DCABES 2014

13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science

24-27 November 2014 | Xian Ning, Hubei, China

Editors in Chief
Craig Douglas and Guo Yucheng

Proceedings

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

DCABES 2014

24-27 November 2014

Xian Ning, Hubei, China

Proceedings

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

DCABES 2014

24-27 November 2014
Xian Ning, Hubei, China

Editors in Chief
Craig Douglas
Guo Yucheng

CPS
Conference Publishing Services

IEEE Computer Society Order Number E5396
ISBN-13: 978-1-4799- 4169-8
BMS Part # CFP1420K-CDR

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: <reprints@computer.org>

Editorial production by Juan E. Guerrero
Cover art production by Mark Bartosik
Printed in the United States of America by Applied Digital Imaging

IEEE Computer Society
***Conference Publishing Services* (CPS)**
http://www.computer.org/cps

2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science

DCABES 2014

Table of Contents

Distributed/Parallel Applications

Cloud/Grid Computing

Distributed/Parallel Algorithms

Computer Networks and System Architectures

E-Business/E-Society

Information Security/Internet of Things (IOT)

Simulation/Image Processing

Preface
DCABES 2014

The DCABES is a community working in the area of Distributed Computing and Applications in Business, Engineering, and Sciences, and is responsible for organizing meetings and symposia related to the field. DCABES intends to bring together researchers and developers in the academic field and industry from around the world to share their research experience and to explore research collaboration in the areas of distributed parallel processing and applications.

The 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES 2014) will be held on November 24~27, 2014 in a very famous resort, the Xian Ning, Hu Bei province, China.

All papers accepted by DCABES 2014 Proceedings have been peer reviewed and carefully modified. Since DCABES 2001, the first DCABES, each DCABES symposium has invited 4 to 5 world famous professors and experts in computer science and technology area to give keynote speeches for the conference. Thirteen years passed and dozens VIPs in distributed parallel processing have attended the DCABES conferences, among them there are Professor Jifeng He, a member of Academia Sinica, Professor Albert Y. Zomaya, the Editor-in-Chief of IEEE Transactions on Computers and Associate Editor-in-Chief of IEEE Transactions on Parallel and Distributed Systems, Professor C.-H. Lai, Editor-in-Chief of JACT (Journal of Algorithms and Computational Technology), Academia Sinica Professor Zhiwei Xu, Editor-in-Chief of Journal of Computer Research and Development, Professor Craig Douglas, distinguished professor in MGNET, et al.

The DCABES series began as a summer short course held at Hong Kong Polytechnic University in 2000 with the support of the British Computer Society - Hong Kong Chapter. The two co-chairs of DCABES, Professors GUO Qingping and LAI Choi-Hong, extended the short course into a series of conferences that continues today and grows yearly.

In recent years, more and more attentions have been put to the distributed parallel computing. We are confident that the distributed parallel computing will play an even greater role in the near future, since distributed computing resources, once properly cooperated together, will achieve a great computing power and get a high ratio of performance/price in parallel computing. In fact the grid computing, cloud computing and the multi-core processor cluster are closely related to and evolved from the distributed parallel computing.

All papers contained in the Proceedings give us a glimpse of what future technology and applications are being studied in the distributed parallel computing area in the world. More papers concerning the distributed parallel algorithms and applications, the intelligent transportation as well as image processing have been selected and included in the DCABES 2014 Proceedings.

We would like to thank all members of the Program Committee, the local organizing committee, and the external reviewers for selecting papers. We would also like to thank the WUT (Wuhan University of Technology, China), CSIR (Center for Studies of Information Resources, Wuhan University, Wuhan, China), the NPCS (National Parallel Computing Society of China), the I2C3 (Institute of Intelligent Computing, Communication and Control, Wuhan) for their supports as local organizers of the conference.

Here thanks are also extended to Mr. Zhang Yuchao and Mr. Zhu Lin of Wuhan University of Technology for their contributions in organizing the DCABES 2014 conference.

Guo Qingping, *Wuhan University of Technology, China*
DCABES Co-chair and I^2C^3 Chair

Conference Organization
DCABES 2014

Conference Chair
Guo Q. P., *Wuhan University of Technology, China*

Conference Co-chairs
Douglas Craig C., *University of Wyoming, USA*
Lai C.-H., *University of Greenwich, United Kingdom*

Program Committee Co-chairs
Craig C. Douglas, *Yale University, USA*
C.-H. Lai, *University of Greenwich, United Kingdom*
Q.P. Guo, *Wuhan University of Technology, Wuhan, China*

Local Organizing Committee Co-chairs
Guo Qingping, *Wuhan University of Technology, Wuhan, China*
Hu Changping, *CSIR, Wuhan University, Wuhan, China*

Local Organizing Committee
Liu Yuhua, *Central China Normal University, Wuhan, China*
Zhang Shesheng, *Wuhan University of Technology, China*
Li Wenjing, *Guangxi Normal University, Nanning, Guangxi, China*
Chen Wei, *Wuhan University of Technology, Wuhan, China*
Xiao Xinping, *Wuhan University of Technology, China*
Zhong L., *Wuhan University of Technology, Wuhan, China*

Secretariat
Guo Yucheng, *Wuhan University of Technology, China*
Zhang Yuchao, *Wuhan University of Technology, China*
Zhu Lin, *Wuhan University of Technology, China*

Steering Committee Co-chairs
Guo Q.P., *Wuhan University of Technology, China*
Lai C.-H., *University of Greenwich, United Kingdom*

Steering Committee
Douglas Craig C., *University of Wyoming, USA*
Tsui Thomas, *Chinese University of Hong Kong, Hong Kong, China*
Xu W., *Jiangnan University, Wuxi, China*

Program Committee
DCABES 2014

Frederic Magoules, *Ecole Centrale Paris, France*
Albert Y. Zomaya, *Chair Professor in Centre for Distributed and High Performance Computing School of Information Technologies, The University of Sydney, Australia*
Professor Hai Jin, *Dean of School of Computer Science and Technology, HUST, Wuhan, China*
Maurício Vieira Kritz, *National Laboratory for Scientific Computation, Petropolis-RJ, Brasil*
W.B. Xu, *Jiangnan University, Wuxi, China*
Xiao-Chuan Cai, *University of Colorado at Boulder, USA*
Jianwen Cao, *Institute of Software Chinese Academy of Sciences, Beijing, China*
Chi XueBing, *Chinese Academy of Sciences, China*
Yakup Paker, *Queen Mary University of London, United Kingdom*
Turgay Altilar, *Istanbul Technical University, Istanbul, Turkey*
Souheil Khaddaj, *Kingston University, United Kingdom*
Lishan Kang, *University of Geosciences, China*
Chen Wei, *Wuhan University of Technology, Wuhan, China*
Xiao Xinping, *Wuhan University of Technology, China*
Wenjing Li, *Guangxi Normal University, Nanning, Guangxi, China*
Shesheng Zhang, *Wuhan University of Technology, China*
Ping Lin, *Professor, University of Dundee, United Kingdom*
Jiachang Sun, *Institute of Software, Academy of Science, China*
Alfred Loo, *Lingnan University, Hong Kong*
Peter Kacsuk, *Hungarian Academy of Sciences, Hungary*
Stefan Vandewalle, *Katholieke Universiteit Leuven, Belgium*
Robert Lovas, *Hungarian Academy of Sciences, Hungary*
Faouzi Alaya Cheikh, *Gjovik University College, Norway*
NIKOS Christakis, *University of Crete, Heraklion, Greece*
Haixin Lin, *Delft University of Technology, Netherlands*
David Keyes, *Columbia University, USA*
Zhihui Du, *Tsinghua University, China*
Meiqing Wang, *Fuzhou University, Fuzhou, China*
Yuhua Liu, *Central China Normal University, Wuhan, China*
Youwei Yuan, *Hangzhou Dianzi University, Hangzhou, China*
V. P. Kutepov, *Moscow Power Engineering Institute, ul., Russia*
Yi Pan, *Georgia State University, USA*
Alan Davies, *University of Hertfordshire, United Kingdom*
Peter Sloot, *University of Amsterdam, Netherlands*
Franck Cappello' CNRS, *Universite Paris-Sud, France*
Simon Cox, *School of Engineering Sciences, United Kingdom*
Xiaojun Tong, *Harbin Institute of Technology at Wei Hai, China*
Liu Dan, *China Criminal Police University, China*
Lamine M. Aouad, *University College Dublin, Ireland*
Thi-Mai-Huong Nguyen, *Ecole Centrale Paris, France*
Haiwu He, *Hohai University, China*
Alan J. Davies, *University of Hertfordshire, United Kingdom*
Ziyue Tang, *Air Force Radar Institute, Wuhan, China*

Implementations and Interpretations of the Talbot-Ogden Infiltration Model

Mookwon Seo
University of Wyoming
Mathematics Department
Laramie, WY 82081-1000, USA
Email: mseo@uwyo.edu

Derrick Cerwinsky and Krishan Gahalaut
King Abdullah University of Science
& Technology (KAUST)
UN 1500 Building 1, Al-Khawarizmi
4th Floor, room 4319-CU01
Thuwal 23955-6900
Kingdom of Saudi Arabia
Email: {derrick.cerwinsky,krishan.gahalaut}@kaust.edu.sa

Craig C. Douglas
University of Wyoming
School of Energy Resources and
Mathematics Department
Laramie, WY 82081-1000, USA
Email: cdougla6@uwyo.edu

***Abstract*—The interaction between surface and subsurface hydrology flow systems is important for water supplies. Accurate, efficient numerical models are needed to estimate the movement of water through unsaturated soil. We investigate a water infiltration model and develop very fast serial and parallel implementations that are suitable for a computer with a graphical processing unit (GPU).**

***Index Terms*—Parallel computing; GPU computing; Hydrology; Water infiltration**

I. Introduction

For more than a century groundwater has been used faster than it has been replenished. The interaction between surface and subsurface hydrology flow systems is important for water supplies that are the keys to the local ecosystem and to economic development. Effective and efficient numerical models are needed for estimating the movement of water through unsaturated soil.

The paper is organized as follows. In Section II, we introduce the van Genuchten, Richards, and Green-Ampt models. In Section III, the Talbot-Odgen model is introduced, which describes subsurface flows in a discretized water content domain. In Section IV, we present a vertically discretized Talbot-Ogden implementation based on matrix and vector models. In Section V, we provide timings from a representative example to show how our implementations compare to each other. In Section VI are some conclusions.

II. Background

In this section, we provide basic mathematical models for water infiltration. In Section II-A, we describe van Genuchten's model for the conductivity and capillary pressure. In Section II-B, we describe Richards' model that is a mass-balanced version of Darcy's law. In Section II-C, we introduce the Green-Ampt model that is widely used in estimating infiltration parameters and states, such as the flux, accumulative water content, and infiltration time.

A. van Genuchten Model

The van Genuchten model is based on Mualem's model [1], [2], which describes the prediction of the hydraulic conductivity function of unsaturated porous media.

Let K_{rel} be the relative hydraulic conductivity, h be the absolute value of the pressure head (i.e., the pressure divided by the fluid specific weight), S_e be the relative water content, θ be the water content, θ_s be the saturated water content, and θ_r be the residual water content. Formally, we define

$$\begin{aligned} S_e &= \frac{\theta-\theta_r}{\theta_s-\theta_r} \text{ and} \\ K_{rel} &= S_e^{1/2}\left[\int_0^{S_e}\frac{1}{h(x)}dx \Big/ \int_0^1 \frac{1}{h(x)}dx\right]^2. \end{aligned} \tag{1}$$

Two equivalent models (Brooks-Corey and van Genuchten [3]) describe the soil-water retention curve. Let α, n, and m be undetermined parameters and h be the absolute value of the pressure head. Then

$$S_e(h) = \left[\frac{1}{1+(\alpha h)^n}\right]^m. \tag{2}$$

From (2),

$$\frac{1}{h(S_e)} = \alpha\left[\frac{S_e^{1/m}}{1-S_e^{1/m}}\right]^{1/n}. \tag{3}$$

Substituting (3) into (1),

$$f(S_e) = \int_0^{S_e}\left[\frac{x^{1/m}}{1-x^{1/m}}\right]^{1/n} dx \text{ and} \tag{4}$$

$$K_{rel}(S_e) = S_e^{1/2}\left[\frac{f(S_e)}{f(1)}\right]^2. \tag{5}$$

Substituting $x = y^m$ into (4),

$$\begin{aligned} f(S_e) &= \int_0^{S_e^{1/m}}\left[\frac{y}{1-y}\right]^{1/n} m y^{m-1}dy \\ &= m\int_0^{S_e^{1/m}} y^{m-1+1/n}(1-y)^{-1/n}dy, \end{aligned} \tag{6}$$

which is a special form of an incomplete beta-function when $k = m-1+n^{-1} \in \mathbb{N}$. If $k=0$, then $m = 1-n^{-1}$ and $0 < m < 1$. Hence,

$$\begin{aligned} f(S_e) &= m\int_0^{S_e^{1/m}}(1-y)^{-1/n}dy \\ &= m\left(-\frac{1}{1-1/n}\right)\left((1-S_e^{1/m})^{1-1/n}-1\right) \\ &= 1-(1-S_e^{1/m})^m. \end{aligned}$$

Since $f(1) = 1$,

$$K_{rel}(S_e) = S_e^{1/2}(1-(1-S_e^{1/m})^m)^2. \tag{7}$$

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.71

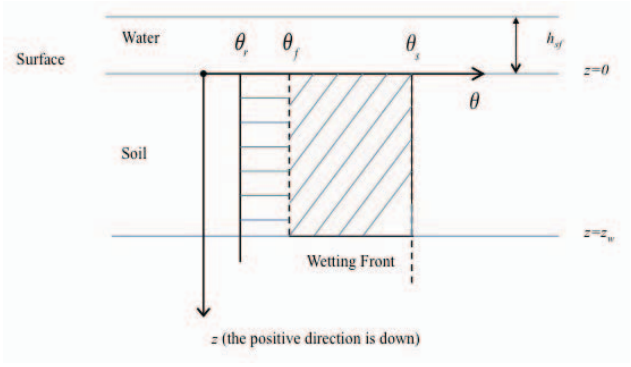


Fig. 1. Green-Ampt model

Let K be the hydraulic conductivity at a particular water content θ, K_{sat} be the hydraulic conductivity at saturation, and ψ be the capillary pressure head. The relative hydraulic conductivity is

$$K_{rel} = \frac{K}{K_{sat}}.$$

From (3), the capillary pressure head is

$$h(S_e) = \mid \psi(S_e) \mid = \alpha^{-1}(S_e^{-1/m} - 1)^{1/n}.$$

The effective capillary pressure head ψ_e [4], [5] is computed using (2), (7), and $m = 1 - n^{-1}$:

$$\begin{aligned}\psi_e &= \int_0^\infty K_{rel}(h)dh = \int_1^0 K_{rel}(S_e)\frac{dh}{dS_e}dS_e \\ &= \int_1^0 S_e^{1/2}(1-(1-S_e^{1/m})^m)^2(\alpha n)^{-1} \cdot \\ &\qquad (S_e^{-1/m}-1)^{1/n-1}(-mS_e^{-1/m-1})dS_e \\ &= \frac{1-m}{\alpha m}\int_0^1 (1-(1-S_e^{1/m})^m)^2 \cdot \\ &\qquad (1-S_e^{-1/m})^{-m}S_e^{-1/m+1/2}dS_e. \end{aligned} \quad (8)$$

We numerically integrate (8) using Gaussian quadrature with 256 Gauss points and fit it by least squares with a rational function:

$$\psi_e \approx \alpha^{-1}\frac{0.046m + 2.07m^2 + 19.5m^3}{1 + 4.7m + 16m^2}.$$

Example 1: For sand samples, $\alpha = 3.6\text{kPa}^{-1}$, $n = 1.56$, $m = 0.36$, $\theta_r = 0.078$, $\theta_s = 0.43$, and $K_{sat} = 2.889 \times 10^{-6}$m/s.

B. Richards Model

Richards provided a general description for the unsaturated subsurface flows based on Darcy's law [6],

$$q = -K\nabla h,$$

where q is the water discharge. Let z be the depth of water, h_z be the pressure head at depth z, and p_z be the water pressure caused by the weight of the water at depth z. For the 1D case, the water discharge with depth z has to be balanced with a change in the soil moisture [7]:

$$\frac{d\theta}{dt} + \frac{dq}{dz} = 0 \text{ or } \frac{d\theta}{dt} = \frac{d}{dz}\left(K\frac{dh_z}{dz}\right).$$

Note that $p_z = z$ since the weight of water is identical to the length of water in the porous media. Hence, $h_z = \psi + z$ and

$$\frac{d\theta}{dt} = \frac{d}{dz}\left(K\frac{d(\psi+z)}{dz}\right) = \frac{d}{dz}\left(K\left(\frac{d\psi}{dz}+1\right)\right).$$

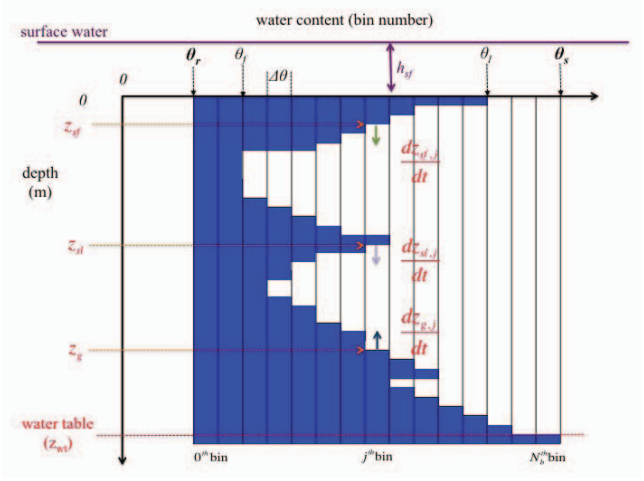


Fig. 2. Talbot-Ogden model

C. Green-Ampt Model

The Green-Ampt equation is derived from Richards equation based on the nonliear form of Darcy's law for a partially saturated flow, as represented in Fig. 1. Let θ_f be the initial water content, ψ_w be the capillary pressure of the wetting front at depth z_w, and h_{sf} be the height of the water above the surface, as represented in Fig. 1.

Let h_0 and h_{zw} be hydraulic pressure heads corresponding to the surface and depth z_w, respectively. At the surface, the water pressure head is same as the height of the water. Define $F(t)$ as the water in the area above the wetting front in Fig. 1:

$$F(t) = z_w(\theta_s - \theta_f).$$

From Darcy's law,

$$\frac{dF(t)}{dt} = K_{sat}\frac{dh}{dz} = K_{sat}\frac{h_0 - h_{zw}}{z_w} = K_{sat}\frac{h_{sf} + z_w + \psi_w}{z_w}.$$

When h_{sf} is sufficiently small, then the vertical infiltration rate is

$$\frac{dF(t)}{dt} = K_{sat}\left(\frac{(\theta_s - \theta_f)\psi_w}{F(t)} + 1\right). \quad (9)$$

III. Talbot-Ogden Model

In this section we describe the Talbot-Ogden model. In Section III-A, we describe the wetting front velocity for the Talbot-Ogden model. In Section III-B, we describe the equation governing the groundwater front. In Section III-C, we show the subsurface slug equation. In Section III-D, we introduce the redistribution of subsurface water.

The Talbot-Ogden model discretizes the water content domain into segments that conduct flows downward vertically in soils [8], as represented in Fig. 2. Let θ be the moisture content variable. Define a bin as a discretized segment from the residual water content θ_r to the saturated water content θ_s. Let $\Delta\theta > 0$ be the width of each bin and θ_j be the midpoint of the j^{th} bin. Define the subscript ℓ as the rightmost bin that has water attached to the surface and f as the leftmost bin that is not fully saturated.

The relative water content S_j for the j^{th} bin is

$$S_j = \frac{\theta_j - \theta_r}{\theta_s - \theta_r}.$$

Let K_j be the conductivity from the 0^{th} bin to the j^{th} bin and ψ_j be the capillary suction in the j^{th} bin. Recall that K_{sat} is the

hydraulic conductivity at saturation and n and m are experimental values depending on the soil type: for $n > 1$ and $m = 1 - n^{-1}$ from the van Genuchten's model and (7),

$$\begin{aligned} K_j &= K_{sat} S_j^{1/2} (1 - (1 - S_j^{1/m})^m)^2, \\ \psi_j &= \alpha^{-1} (S_j^{-1/m} - 1)^{1/n}. \end{aligned}$$

Every subsurface flow is governed by following:

$$\text{Subsurface flow velocity} = \frac{\text{Conductivity factor}}{\text{Water content factor}} \times \frac{\text{Pressure factor}}{\text{Length factor}} \quad (10)$$

and

$$\text{Pressure factor} = \text{Capillary head pressure} + \text{Pressure head}.$$

A. Wetting front velocity

Let $z_{sf,j}$ be a depth of the water attched to the surface in j^{th} bin. For the infiltration rate in the j^{th} bin,

- water content factor $= \theta_\ell - \theta_f$,
- conductivity factor $= K_\ell - K_f$,
- pressure factor $= \psi_j + z_{sf,j} + h_{sf}$,
- length factor $= z_{sf,j}$.

Using (10) the wetting front velocity (i.e., the infiltration rate) is

$$\frac{dz_{sf,j}}{dt} = \frac{K_\ell - K_f}{\theta_\ell - \theta_f} \left(\frac{h_{sf} + \psi_j}{z_{sf,j}} + 1 \right). \quad (11)$$

B. Groundwater front velocity

Let $z_{g,j}$ be the groundwater front depth in j^{th} bin and z_{wt} be be the depth of the water table where pressure head is equal to the atmospheric pressure. For the groundwater front velocity in j^{th} bin,

- water content factor $= \theta_j - \theta_f$,
- conductivity factor $= K_j - K_f$,
- pressure factor $= -\psi_j + z_{wt} - z_{g,j}$,
- length factor $= z_{wt} - z_{g,j}$.

When the groundwater front is above the water table and using (10), the groundwater front velocity is

$$\frac{dz_{g,j}}{dt} = \frac{K_j - K_f}{\theta_j - \theta_f} \left(\frac{-\psi_j}{z_{wt} - z_{g,j}} + 1 \right). \quad (12)$$

Groundwater can join the water table, which causes the latter to change its location in the bin.

C. Slug velocity

When the surface is completely dry, water detaches from the surface and moves down in bins. A hanging water shape appears (see Fig. 2, area pointed to by z_{sl}), which is a *slug*.

For the j^{th} bin, let $z_{sl,j}$ be the bottom depth of a slug and len_j be the slug length. Assume the velocities of the top and bottom of the slug are the same.

- water content factor $= \theta_j - \theta_{j-1}$,
- conductivity factor $= K_j - K_{j-1}$,
- pressure factor $= \psi_j(\text{slug bottom}) - \psi_j(\text{slug top}) + len_j$
  $= len_j$
- length factor $= len_j$.

Using (10) the slug velocity is

$$\frac{dz_{sl,j}}{dt} = \frac{K_j - K_{j-1}}{\theta_j - \theta_{j-1}}. \quad (13)$$

D. Redistribution

For each time step, after subsurface velocities are calculated, water is redistributed based on capillary pressure. Redistribution instantly drags hanging water from right to left since subsurface water movement in right bins is faster than in left bins.

IV. Implementation

There are many ways to implement the Talbot-Ogden model. In Section IV-A, we introduce the classical way to describe the Talbot-Ogden model. In Section IV-B, we describe a linked list based implementation. In Section IV-C, we describe a vertically discretized implementation using binary dense matrix. In Section IV-D, we describe a vertically discretized implementation using just a vector.

A. Classic Talbot-Ogden model

The hydrostatic equilibrium is the condition of the fluid that is not in motion. Groundwater is initially set to the hydrostatic equilibrium,

$$z_{g,j} = z_{wt} - \psi_j.$$

Define $\mathcal{B}_L$ as the leftmost bin that is not completely saturated and $\mathcal{B}_R$ as the rightmost bin that has surface front water:

$$\begin{aligned} \mathcal{B}_L &= \min\{j \in \mathbb{N} | z_{g,j} \neq 0, 0 \leq j < N_b\} \text{ and} \\ \mathcal{B}_R &= \max\{j \in \mathbb{N} | z_{sf,j} \neq 0, 0 \leq j < N_b\}. \end{aligned}$$

Let z_{bot} be the depth of the bottom of the Talbot-Ogden domain and N_b be the number of bins. Before calculating the infiltration we need to consider the amount of water $\mathcal{W}_d$ flowing through the fully saturated bins, which is given by

$$\mathcal{W}_d = K_{\mathcal{B}_L - 1} \Delta t.$$

Let $\mathcal{W}_s$ be the amount of water above the surface and W_r be the amount of water going in or out of the groundwater. Therefore

$$W_s = \begin{cases} \mathcal{W}_s - \mathcal{W}_d, & \text{if } \mathcal{W}_s \geq \mathcal{W}_d, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$W_r = \begin{cases} \mathcal{W}_r + \mathcal{W}_d, & \text{if } \mathcal{W}_s \geq \mathcal{W}_d, \\ \mathcal{W}_r + \mathcal{W}_s, & \text{otherwise.} \end{cases}$$

Define z_{j_0} as the dry depth that is the smallest initial depth of infiltration for the j^{th} bin. From the Green-Ampt equation (9), let $\theta_s = \theta_j$ and $\theta_f = \theta_{j-1}$. Then for the j^{th} bin,

$$\frac{dF(t)}{dt} = K_j \left(\frac{\psi_j \Delta\theta}{F(t)} + 1 \right).$$

Using integration,

$$F = K_j \Delta t + \psi_j \Delta\theta \ln \left(1 + \frac{F}{\psi_j \Delta\theta} \right).$$

Since $F = z\Delta\theta$,

$$z = K_j \frac{\Delta t}{\Delta\theta} + \psi_j \ln \left(1 + \frac{z}{\psi_j} \right). \quad (14)$$

There is a maximum bound of the dry depth that is given by

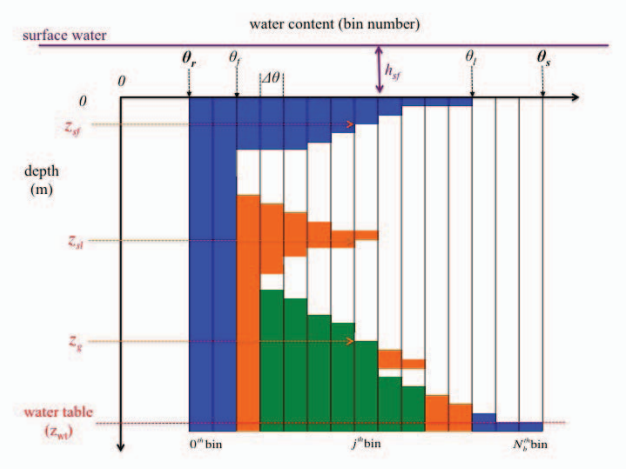
$$z = K_{N_b - 1} \frac{\Delta t}{\theta_f} + \psi_f \ln \left(1 + \frac{z}{\psi_f} \right). \quad (15)$$

Let x and y be the solutions of (14) and (15), respectively, using Newton's method. Therefore

$$z_{j_0} = \min\{x, y\}.$$

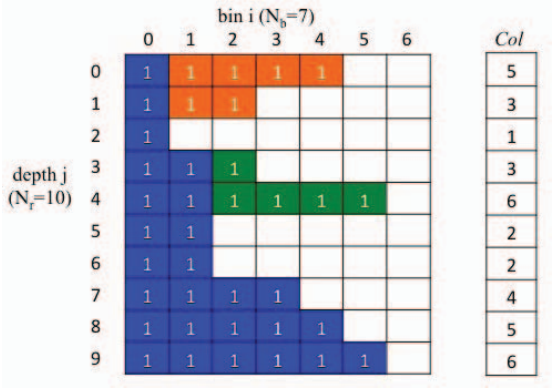
For each bin j, we calculate $z_{sf,j}$ and $z_{g,j}$, respectively, using a fourth order Runge-Kutta method for (11) and (12). We then compute the falling distances for the wetting front $dist_{sf,j}$ and the groundwater front $dist_{g,j}$.

When the surface is completely dry, the water from the surface is detached and governed by (13). Since the distance only depends on the conductivity and water content, but not time, we precompute the



Each bin is its own linked list with the first to third elements of each list colored blue, orange, and green, repspectively.

Fig. 3. Linked lists representation



Blue, orange, and green represent water attached to the groundwater, surface, and detached (a slug), respectively.

Fig. 4. Matrix and Vector models

slug falling distance $dist_{sl,j}$. We only need to recompute it if we change the time interval.

After water moves vertically, redistribution of water from the right bins to the leftmost bins completes the time step.

B. Linked list model

The oldest Talbot-Ogden implementations [8] used a single array to represent the depth of water in each bin in the domain. This method was both fast and memory efficient. The redistribution phase in this implementation unfortunately required sorting the array. Only the maximum depth of water in each bin was approximated, which means it could not model slugs in the domain.

The linked list implementation uses a set of linked lists, each of which represents a single bin in the domain. Fig. 3 shows how Fig. 2 is represented in this format. As the number of slugs changes, elements can be added or removed from the lists. This method provides a memory efficient method for storing the domain.

This implementation is very effective in the infiltration phase, but is very expensive in the redistribution phase. In order to compute the redistribution, a comparison between each element of each linked list and every element of each linked list to its left must be made. If an opening is found in a bin to the left of the current element, the water must be moved. Altering the domain requires altering the elements of the source linked list and the destination linked list. This must be done for every element of the linked list for every bin at every time step. As the number of slugs grows, this becomes very expensive. It is also impractical to make this implementation scale well on a parallel computer.

C. Matrix model

The Matrix model is an implementation of the Talbot-Ogden model that simplifies the redistribution phase by using a different data structure for the domain. This model uses a discretization in the vertical direction as well as in the θ direction. This additional discretization creates a collection of cells that can be represented as a binary array, see Fig. 4. In each cell, a 1 represents the presence of water, and a 0 indicates an empty cell.

The advantage of the matrix representation is the trivialization of the redistribution phase. To redistribute water, cells are moved to the left. This is easily accomplished by summing the row and setting the correct number of matrix elements to 1. The algorithm is embarrassingly parallel since in the infiltration phase each bin is independent of every other bin. In the redistribution phase each row is independent of every other row. This model lends itself well to multicore CPUs and GPU systems.

The major disadvantage to the matrix model is the amount of data that must be moved at each time step. Also, every element of the domain must be touched twice in each time step.

D. Vector model

The Matrix model is a step in the right direction, but the required data movement makes the method prohibitively expensive. The obvious next step is some type of sparse representation. Since in the redistribution phase the water always moves to the left side of the domain, the only required information to construct the matrix is the number of non-zero cells in each row. This idea leads to what we call the Vector model. In this model the entire domain is represented by a single vector, as can be seen on the right in Fig. 4.

This representation of the data requires care in the infiltration step. Since all of the bins are represented as a single number, the movement by each bin is much more complex and more expensive than in the Matrix model. However, the redistribution phase is automatic and is actually free. As in the Matrix model, each row must be examined at each time step.

Some of the parallelism of the Matrix model is lost in the Vector model. By adding the assumption that all water moves like slugs, the parallelism can be recovered and this method can be implemented on a GPU. The final implementation of this method on a GPU has some interesting properties:

1) The run time per time step is almost constant, independent of the water in the domain. Hence, in some cases the serial implementations can skip empty regions of the domain, which makes them faster than GPU implementations that do not skip empty regions.
2) The complexity of the algorithm depends on the conductivity of the soil and the size of the time step. Hence, the larger each of these factors is, the more work that is required by the GPU per time step.

Modifications to the data structures and computations were required to use the algorithm on a GPU. Lookup tables were created

so that each row in the z-domain worked independently of all other rows, thus allowing each thread on a GPU to process a single row. These modifications speeded up the algorithm on a GPU over the existing CPU algorithm.

Unfortunately, the lookup tables were too large to be in GPU shared memory and had to be in GPU global memory. Between the overhead for each kernel launch and the access time to the GPU's global memory, the expected speedup was not achieved. However, the modified data structures and algorithms proved to be extremely effective on the CPU by making more effective use of L1 and L2 memory caching. The serial implementation was provided an unexpected and significant performance advance.

V. Numerical experiments

In this section, we provide timings from representative examples to show how our implementations compare to each other.

The numerical examples were run on a 3.10 GHz quad core Intel Core i7-4930MX CPU with 32 GB of RAM running Windows 7 Ultimate 64-bit. The examples were compiled using Microsoft Visual Studio Professional 2013 Version 4.5.50938 in 64-bit mode. A NVIDIA 780M GPU with 1536 CUDA cores and 3 GB of memory was also used.

The simulations are for a 1 meter deep column of coarse, tightly packed sand using the parameters in Table I. Each example simulates a week of infiltration using two different rainfall patterns that are listed in Table II. The timings and speedups for both examples are in Table III.

TABLE I
Constants used in each numerical example.

Variable	Value	Units
bins	300	
number of rows	100	
conductivity	$2.889e-6$	meters per second
porosity	0.357	unitless fraction
van Genuchten α	3.6	per meter
van Genuchten n	1.56	unitless
residual saturation	0.078	unitless fraction

TABLE II
Rainfall used in each example.

Rainfall	Example 1			Example 2		
	Start	Stop	Rate	Start	Stop	Rate
1	0	0.5	10	0	8.0	4.0
2	12.0	12.1	10	96.0	106.0	2.0
3	26.0	26.1	1.0			
4	37.0	37.1	1.0			
5	49.0	49.2	1.0			
6	62.0	62.1	1.0			
7	70.0	70.1	1.0			

Times are in hours from the initial start time. Rates are in cm/hour.

TABLE III
Numerical Timings and Speedups

Timings	Example 1	Example 2
Linked list	74.526	58.164
Matrix	13.820	20.415
Vector	0.089735	0.089465

Times are in seconds.

Speedups	Example 1	Example 2
Matrix	5.39	2.85
Vector	830.51	650.13

Speedups assume the base time is for the Linked list model.

The first example has seven short, hard rainfalls. This example creates multiple slugs that remain for most of the simulation time. The linked list method suffers because it must track and sort linked lists for each bin and each slug. The vector implementation is not slowed by large numbers of slugs since each row is treated independently.

The second example has two very heavy rainfalls. In this case the linked list implementation performs better than in the first example since there is almost never more than one slug at a time. The vector code runs at almost the same rate as the first example, illustrating that the run time is almost independent of quantity and frequency of water present in the simulation.

VI. Conclusions

In this paper, we investigated a water infiltration model and developed very fast implementations. In the Matrix model, we developed a method that is embarrassingly parallel and appears suitable for GPUs. In the Vector model, we flattened the Matrix model and reduced its memory requirements by a dimension. As a result, the Vector model runs faster on a single CPU than either the Linked list model on a CPU or the Matrix model on a GPU or CPU. We are investigating a Matrix-Vector-free model that shows promise on a GPU, but is still future work.

Our long term goals are to use the fastest implementation in basin scale model of the upper Colorado River [9]. We will need to run tens of thousands of cases in parallel using a domain decomposition method.

Acknowledgments

We would like to thank Fred Ogden, Robert Steinke, and Wencong Li for helpful discussions about hydrology issues. This research was supported in part by the National Science Foundation grant EPS-1135483 and King Abdullah University of Science & Technology.

References

[1] Y. Mualem, "A new model for predicting the hydraulic conductivity of unsaturated porous media," *Water Resources Research*, vol. 12, no. 3, pp. 513–522, 1973.

[2] P. B. O. Ippisch, H.-J. Vogel, "Validity limits for the van Genuchten-Mualem model and implications for parameter estimation and numerical simulation," *Advances in Water Resources*, vol. 29, no. 12, pp. 1780–1789, 2006.

[3] M. T. van Genuchten, "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils," *Soil Sci. Soc. Am. J.*, vol. 44, pp. 892–898, 1980.

[4] H. P. Meyer, H. J. Morel-Seytoux, P. D. Meyer, M. Nachabe, M. T. V. Genuchten, and R. J. Lenhard, "Parameter equivalence for the Brooks-Corey and van Genuchten soil characteristics: Preserving the effective capillary drive," *Water Resources Research*, vol. 32, 1996.

[5] H. J. Morel-Seytoux and J. Khanji, "Derivation of an equation of infiltration," *Water Resources Research.*, vol. 10, no. 4, pp. 795–800, 1974.

[6] H. Darcy, *Les Fontaines Publiques de la Ville de Dijon*. Paris: Dalmont, 1856.

[7] K. Beven, *Rainfall-Runoff Modelling*, 2nd ed. New York City: Wiley-Blackwell, 2012.

[8] C. A. Talbot and F. L. Ogden, "A method for computing infiltration and redistribution in a discretized moisture content domain," *Water Resources Research*, vol. 44, pp. 1–14, 2008.

[9] Various, "CI-WATER: A Utah-Wyoming cyberinfrastructure water modeling collaboration," http://www.ci-water.org, since 2011, last visited 7/25/2014.

ZDLC for the Early Stages of the Software Development Life Cycle

Y.K. Makoondlall[(a)], S. Khaddaj[(a)], B. Makoond[(a,b)]
(a) School of Computing and Information Systems, Kingston University, Kingston Upon Thames, KT1 2EE, UK
(b) Cognizant Technology Solutions, 1 Kingdom Street, Paddington Central, London, W26BD
yashvir_1@hotmail.com, s.khaddaj@kingston.ac.uk, bippin.makoond@cognizant.com

***Abstract*— The cost of fixing of a software defect in the later phases of the Software Development Life Cycle (SDLC) is significantly more than fixing a defect in the earlier phases of the SDLC, especially for distributed systems. The Zero Deviation Life Cycle (ZDLC) has been engineered to ensure that there is minimum deviation from the requirements and there is as little defect injection as possible between the phases of the SDLC. So far the toolset developed for ZDLC do not include a comprehensive tool to automate the translation of Natural language requirements into design models for communication, process and data requirements. The aim of this paper is to establish the premise for such a tool, which could be added to the ZDLC suites of tools.**

Keywords- Zero Deviation Life Cycle (ZDLC), Automation of design artefacts.

I. INTRODUCTION

The task of overseeing an IT (transformation) project or software projects to completion is a complicated one, which is full of pitfalls. Over the years there has been a series of failed software projects [1], [2] and the reasons why a software project fails vary hugely from project to project [3]. Whilst it is true that software failures or rather software malfunction is likely to be around for as long software will be, all the key stakeholders involved in a project need to ensure that the project is completed with as little defects as possible. Apart from software quality it is important to deliver the projects on schedule and on budget. Some of the cancelled projects were simply called off as they exceeded their initial budget and were behind schedule.

In order to ensure that the delivered software do not contain defects and are not error prone, most companies use post implementation Testing and Quality Assurance models [4]. In a research carried out by Adeel et al.[5], the authors have handed out a few questionnaires to IT professionals in the industry with the aim to gather information on their defect removal strategy and then investigate a series of Defect Prevention (DP) techniques. They noticed that System Testing was the most widely used technique (100%), followed by Integration testing (88%) and Unit testing (75%).

In a typical waterfall methodology life cycle, testing is the last phase before a product is delivered. However defects can be introduced during the earlier phases and even in the Testing phase itself, as shown in Figure 1. Even if other traditional approaches like prototyping or Rapid Application Development (RAD as an iterative method) or agile approaches like Scrum or Extreme Programming are used to manage an IT project, the requirements and design phases have to be completed before development and testing can start. As a results, requirement defects and design defects are carried over to the development phase and the testing phase which is very costly.

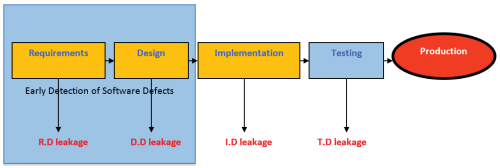


Figure 1. Defect Leakage in the Waterfall Methodology.

If formal testing (Unit Testing, Integration Testing and System Testing) is the primary means of detecting defects, it means that defects will have to be fixed as a rework and a lot of regression testing will be needed to make sure that the software deliverables are working as expected. This widely used approach address the defects after the solution has been designed and coded, but the design defects and requirement defects are not necessarily addressed. The design used may not meet the requirements and other client needs (like scalability). The requirements used to design the solution may not meet all the client's exigencies as the requirements gathered may be incomplete, ambiguous and exclude tacit requirements.

A. Background information and challenges

The cost of fixing a defect in a production environment is much higher than in the initial phases of the Software Development Life Cycle (SDLC). In 2001, Boehm and Basili [6] claimed that the cost of fixing a software defect in a production environment can be as high as a 100 times the cost of fixing the same defect in the requirements phase. In 2009, researchers at the IBM Systems Science Institute state that the ratio is more likely to be 200 to 1 [7], as shown in Figure 2.

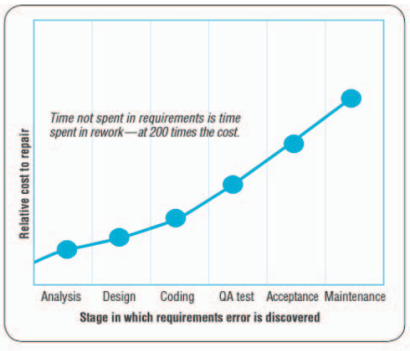


Figure 2. Relative Costs to Fix Software Defects.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.5

It is therefore worthwhile investigating techniques which may help to reduce defects from the earlier phases of the Software Development Lifecycle (SDLC). Software testing does not remove all the defects and therefore other techniques also have to be used to eliminate defects. The situation is bad enough for monolithic systems, but it may be even worse for distributed systems, which are very often developed and implemented across different teams which span across different geographic locations.

In recent years, demand for service oriented distributed systems has been increasing rapidly with new technologies emerging to support the growth and diversity of users' requirements. Both Cloud Computing and Service Oriented Architecture (SOA) are so far the leading trends which businesses and companies are most likely going to adopt [8], [9], [10]. Thus the current and future software architectures are more likely going to be distributed by nature, involving the interaction between diverse systems and platforms, resulting in systems which are more complex than monolithic systems, and require new defect detection mechanisms.

Companies and software departments need to test and validate the pieces of software prior to deployment. Failure to do so early enough in the software development lifecycle (SDLC) can prove very costly. Appropriate verification and validation techniques need to be put in place so that there are as little defects as possible leaking into the latter phases of the SDLC. As more businesses and governments move towards distributed software systems such as SOA and Cloud Computing in order to maximize the capability of web based solution, it is necessary to devise strategies, methodologies and validation techniques to reduce the number of defects.

The defects found in distributed systems may occur in a service or out of the interaction amongst many services. The defects encountered in distributed systems are thus more difficult to fix than the defects encountered in monolithic systems [11]. As the modules in a distributed system start to interact with each other, there are often scenarios or states that have not been modelled upfront that start to emerge.

Thus the defects from those emergent behaviors are harder to fix as they require more time and additional effort for rework analysis. To avoid the additional costs linked to software defects, it is worthwhile investigating techniques which ensure the pieces of software delivered are less defective. As outlined earlier, it can be far less expensive to eliminate the defects in the early phases of the SDLC, especially for distributed systems. The Zero Deviation Life Cycle (ZDLC) has been devised specifically to ensure minimum deviation from the requirements and to reduce defect leakages between the different phases of the SDLC.

II. ZERO DEVIATION LIFE CYCLE (ZDLC)

The Zero Deviation Life Cycle (ZDLC) is an approach to transform any Software Development Life Cycle (SDLC) into a Software Value Chain, which means that at any point in time ZDLC enforces value creation for the industries [12]. ZDLC sustains Value Creation by embedding capabilities such as Automation, Formal Methods, Advanced Simulation and Artificial Intelligence all packaged into an intuitive and simple tool set that employs the concept of gamification and advanced usability techniques to increase ease of use and adoption [13]. ZDLC is supported by a number of tools and software products, designed to practice the core principles of Quality Engineering throughout the Software Life Cycle.

Time, accuracy and quality within the SDLC can be improved, but we need to rethink the use and the dynamics of classical Software Engineering tools. New techniques are required and ZDLC is a framework that provides an approach to merge classical tools with scientific techniques so as to augment accuracy, productivity and quality of delivery whilst reducing effort. This led to the evolution of Software Development Life Cycle to the Software Value Chain [13].

In summary, the aim of Zero Deviation Life Cycle Framework (ZDLC) is to remove ambiguity that is the cause of defect injection and to increase accuracy that is the cause of defect leakage between the phases of the SDLC. Each component that forms the ZDLC can be used independently but when used together they radically improve the process of reducing defect injection and leakage at requirements, architecture/design, build and test.

The ZDLC is a Quality Engineering Platform that comprises the following components [13]:

1. HoQ – House of Quality for requirements elicitation.
2. RMS – Requirements Modeling System to sketch use cases and user stories for the final part of requirements elicitation.
3. TRiZ – The Theory of Inventive Problem Solving used to focus innovation against the hard problems identified in the roof of HoQ-e.
4. TiA – Testable Integration Architecture used to normalize requirements (use cases and user stories from RMS-e) into TiA-e scenarios and to auto-generate a straw man TiA-e model which can be tested against the original and new requirements.
5. SDP – The Systemic Defect Profiler which is used to do both conformance testing in SIT against the TiA-e model and automate root cause analysis of complex defects thus increasing confidence in the quality of the resultant system and reduce the cost of defects.

The tools mentioned above can help to reduce defect leakages between the different phases of the SDLC and also minimize deviation from the requirements. However there are no tools which automate the process of translating the requirements into software models, especially for distributed systems. The aim of this paper is to propose a framework which automates the translation of natural language requirements into design artefacts for distributed systems. The next section exposes the existing tools translating requirements into software artefacts.

III. SOFTWARE REQUIREMENT: TEXTUAL ANALYSIS TECHNIQUES

A. Overview

There are a number of defects management techniques which textually analyses the requirements with the aim to assist in the design process [14], [15], [16], [17]. They employ Natural Language Processing (NLP) techniques to parse the requirements and generate models, which can be used in the design process. These models are typically UML artefacts, which can be used as the blue prints for software construction.

While NLP has been applied to various software engineering fields, IT practitioners were unsure if NLP could be applied to requirement engineering. Ryan [18] claimed that NLP would not be able to understand the requirements. Furthermore even if NLP could be used to understand the requirements, NLP would not be able to infer the requirements which are not explicitly written down but are understood as part of "common domain knowledge". Therefore NLP would not be applicable to requirement engineering, especially for large systems.

However, further research work has continued, and a number of investigations on the applicability of NLP for requirement engineering have appeared in literature [16], [19], [20]. These include AbstFinder [21], which is a prototype tool that can be used to find abstractions in natural language text to be used in requirements elicitation.

Amongst all the published work, the one accomplished by Kof [22] can be regarded as the first to formally dismiss Ryan's claims. Kof demonstrated that NLP was mature enough to be applied for Requirement Engineering as the aim of NLP is not to understand the requirements, but to extract concepts from the requirements. In the approach proposed by Kof, the system engineer is solicited to validate the parses at various stages and can thus ensure that the requirement documents, containing all the key requirements vital for the software, are written down. The method proposed by Kof uses a series of existing techniques in order to formulate an approach which can be used to parse Natural language requirements so as to ensure that the requirements are correctly described, as unambiguously as possible.

The integrated approach aims to establish an application domain ontology which is validated by the user as it is being constructed, during the Requirement Engineering phase. It makes use of a series of techniques and the weaknesses of the individual techniques are compensated by the strengths of the other techniques used. The resulting document is a validated application domain ontology and a corrected textual specification, free from terminology inconsistencies.

The work accomplished by Kof demonstrates that Natural Language processing is mature enough to be applied to Requirement Engineering and also demonstrates that a series of existing techniques can be applied in an integrated approach to parse requirements, written in Natural Languages. NLP can assist in the Requirement Engineering phase, with the help of manual intervention required. However the approach does not assist in the design process and there are other approaches which automate or partially automate the design process. A number of these frameworks are discussed in the following section.

B. Frameworks to model requirements using NLP techniques

1) The Circe framework

Ambriola and Gervasi propose the Circe framework [16], which uses NLP techniques and a series of validation techniques to assist the translation of requirements into (semi) formal models. The user can use the system through a command line interface, a mail interface or a web interface. The tool helps in the elicitation, selection, and validation of the software requirements. It can be used to extract information from Natural Language requirements to build semi-formal and evaluate the consistency of these models.

The system will accept a set of requirements as input, but should also be provided with a glossary of terms. The glossary of terms is defined following some syntactic conventions and can be a set of words describing and classifying all the domain specific and system specific terms of the requirements.

The heart of the solution, which actually parses the requirements and transforms them into a series of (semi) formal models is the Cico engine. The Cico engine uses a custom algorithm and a series of other parameters and functions to parse the Natural Language. It uses MAS (Model Action Substitution) Rules to replace and map terms such as pronouns, which makes use of a fuzzy algorithm and hence enhancing the recognition power of the tool. Every matching is associated with a score of similarity with a model and a weightage system is also used to determine the mapping. It also uses a set of predefined glossaries including a number of relations taken from common sense, whose semantics and correspondence with the domain's underlying model is assumed to be intuitive. A repository accepting glossary terms and requirements is also used to feed the data to Cico when solicited.

The output of the Cico engine is a series of abstract requirements, which are then fed to View module(s) which will produce a graphical representation of that data, which are mostly semi-formal models. The views normally use the abstract requirements but can also access the source requirements, the glossaries or the output from other modules (like the repository) to produce its output.

The Circe framework integrates natural language processing, modelling and validation aspects in an aim to assist in requirement elicitation and validation.

2) The RACE framework.

Mohd Ibrahim and Rodina Ahmad propose the RACE framework [14] which uses NLP techniques to parse the requirements and also defines a set of rules which are then used to start deriving class diagrams from those requirements. The RACE framework is automating part of the design process.

The RACE framework uses a blend of NLP techniques and domain ontology techniques to extract class diagrams from requirements written in Natural Languages (English and Malay). The user is presented with a class diagram

which he can modify, view and organize concepts and relationships through the RACE user interface.

3) The UMGAR Framework

Deeptimahanti et al. [15] propose the UML Model Generator from analysis of Requirements (UMGAR) framework which aims to provide semi-automated support for developing both static and dynamic UML models from Natural Language requirements.

There are two main components in the UMGAR framework and they are:

1. Normalizing requirements component (NLP Tool Layer)
2. Model Generator Component

The Normalizing requirements component (NLP Tool Layer) makes use of a series of tools to normalize the requirements by reconstructing the syntax of the individual sentences. The Model Generator component uses the normalized requirements to produce a Use case model, an Analysis class model or a design class model.

The UMGAR framework is able to generate a use case model, an analysis class diagram and a design class model. These UML diagrams are generated as an XML Metadata Interchange (XMI) [23] file and therefore can be imported by any UML tool which has an XMI Import feature. The UMGAR framework still requires human intervention for the elimination of irrelevant classes and identification of aggregation/composition relationship among objects, and is as such presented as a semi-automated tool to assist the translation of requirements and generate UML based analysis and design models.

4) Other researches

Other researches include the work carried out by Zhou et al. [17] who also propose a framework which uses NLP techniques and domain ontology building techniques to extract class diagrams from Natural Language requirements. It is based on the fact that the core classes are always semantically connected to each other by one to one, one to many, or many to many relationships in the domain. It finds candidate classes using NLP techniques and domain ontology is then used to refine the results.

C. Analysis of existing frameworks.

Natural language requirements generally describes four main categories of requirements which pertain to data (or describing flow of data), the process flow, the communication and non-functional requirements. Non-functional requirements are not modelled but the three other categories of requirements are. In order to define all the features and characteristics of a piece of software, it is important to model the possible behaviors and interactions of the system, especially for a distributed system. Usually data requirements are modelled by an Entity Relationship Diagram (ERD) or a class diagram. The requirements describing a process flow are usually modelled by a Data Flow Diagram or illustrated with a flowchart. The communication sequences within a system are usually modelled using a Sequence diagram and the non-functional requirements are not modelled.

The following table summarizes how the existing approaches cater for the above mentioned categories of requirements:

Feature	Kof' approach	RACE	Circe	UMGAR	Auto-generation of Class Diagram ... (Zhou et al.)
Normalise Requirements	N	N	N	Y	N
Data models	N	Y	Y	Y	Y
Process models	N	N	Y	N	N
Communication models	N	N	Y	N	N
XMI Support	N	N	N	Y	N
Evolving system (learning system)	N	N	N	N	N

Table 1 : Comparative analysis of the existing approaches.

Whilst Kof's approach is able to eliminate inconsistencies and reduce ambiguity in natural language requirements, it does not automate the design process. All of the remaining approaches or frameworks go one step further and model the requirements. The RACE framework, UMGAR and the approach proposed by Zhou et al. only produce data models. The Circe framework is able to model all three types of requirements but the communication model is a communication diagram and not a sequence diagram. Furthermore the Circe framework was developed in 1997, at a time when service oriented architecture had not been used.

It is also noticed that UMGAR is the only one which normalizes the requirements. Normalization of the requirements prior to the extraction object oriented concepts makes the framework more robust and thus more easily applicable to large and complex systems. UMGAR is also the only framework to output the design artefacts as XML files, compliant with XMI standards. This means that an output from UMGAR can be viewed in other tools which also have an XMI import feature. The user is not restrained to viewing and manipulating the design artefacts through the UMGAR user interface alone. For the other frameworks, the user has to view and manipulate the outputs through the framework's own user interface.

All of the existing attempts are either directly aiming to extract object oriented concepts or produce UML diagrams. This is not a flaw but none of the existing approaches have been engineered specifically for service oriented distributed systems. Moreover, all of the existing frameworks are stagnant to a certain extent. By that it is understood that the algorithm and parsing capacity of the framework does not evolve over time. They will produce the same results when presented with the same set of requirements. In some case the user could modify the results, but the initial parse will be the same. For example the user can manually eliminate redundant classes in the UMGAR framework, but the words identified as classes will always be same for a given set of requirements. It is worthwhile having a dynamic system which learns over time through user interaction and grows more accurate.

This research aims to produce a framework which will model all the three categories of requirements, coupled with a learning system which will allow the parses to grow more accurate over time. In so doing, the approach will provide a comprehensive tool to model the natural language requirements, better adapted for modelling distributed systems. The outputs from the tool will be three sets of design artefacts, drawn from one set of requirements which

will then be used to elaborate the architecture of the software. The resulting enterprise architecture and the technical architecture are more likely going to be aligned with the requirements, thus reducing requirement defects.

IV. PROPOSED FRAMEWORK

A. Traditional approach

In a traditional approach, the main steps to design and implement a large and complex software solution would include the requirements phase, the design phase, the implementation phase, the testing phase, the deployment phase and the support phase. It may seem mostly describing the waterfall methodology but even for other traditional approaches or more modern agile approaches, the requirements will have to be ready before the Architectural Design can be started. Figure 3 illustrates the key stages in this traditional approach.

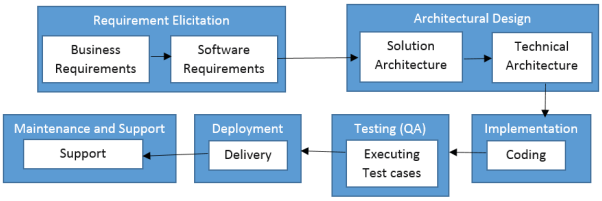


Figure 3. Traditional Approach for developing complex software solutions.

During the requirements phase, the client expresses the business needs which are collated as the business requirements. From these business requirements the software requirements are derived, describing what the software needs to accomplish in order to deliver on the business requirements. Afterwards the architectural design is elaborated. The first step is designing the solutions architecture, which can be considered as the architecture of the solution as a whole. For example, a distributed system may include several pieces of software which communicate which each other through web services or a newly built application may need to retrieve data from or feed data to existing legacy systems. The technical architecture defines the low-level architecture of the program. For example the technical architect may opt for the MVC (Model View Controller) architecture at an application level. Usually the programming language is selected at this stage. During the implementation phase, the solution is coded and then sent to the Quality Assurance (QA) team to be tested. When the solution is given the green light from the QA team, the solution is deployed and any issues are then handled by the maintenance and support team.

B. Proposed framework

The aim of this research is to propose a framework which helps the interpretation of the requirements and automation of the design process so as to reduce the number of defects transpiring from the requirement phase into the later stages of the SDLC. Figure 4 illustrates how the proposed framework alters the SDLC and there are five additional stages involved. Firstly, the Natural Language parser extracts key concepts from the requirements and secondly these concepts are categorized into four main categories. Thirdly, these categorized requirements are used to generate UML models which are presented to the user through the user interface. Fourthly, the user is given the possibility to modify the UML models and these modifications are captured by the learning system. Finally, the user validates the UML design and the framework then creates the finalized models.

The software requirements are fed to the NLP (Natural Language Processing) parser of the framework. From the requirements written in a Natural Language, the parser will identify sentences which pertain to four categories of requirements namely:

- Requirements pertaining to flow of data. (Red)
- Requirements pertaining to a process or process flow. (Blue)
- Requirements pertaining to communication sequences. (Yellow)
- Non-functional requirements. (Green)

The next step is the semantic categorization of the requirements. The requirements pertaining to the flow of data or data attributes are usually modelled with an Entity Relationship Diagram (ERD) or a class diagram. The requirements describing how a process should work are usually modelled with a Data Flow Diagram or illustrated with a flowchart. The communication sequences within a system are usually modelled using a Sequence diagram. The non-functional requirements are not modelled, but can contain information describing certain quality attributes of the software.

The framework will group all the requirements belonging to the same category together (data, process, communication, or non-functional requirements). Those classified requirements are used to generate the models. The framework will analyse the categories individually. An algorithm will loop through all the requirements pertaining to data and then identify the relevant entities or classes. From these classes, the framework will identify the attributes of a class. All this information will then be used to generate the Entity Relationship diagram or the class diagram to describe the flow of data for the system.

Similarly the framework will also parse the process requirements to extract the process flow and produce a Data Flow diagram or a flow chart. The framework will also extract the key attributes pertaining to communication and produce a sequence diagram. The non-functional requirements will not be directly processed by the framework, but may be used at a later stage to extract the test cases.

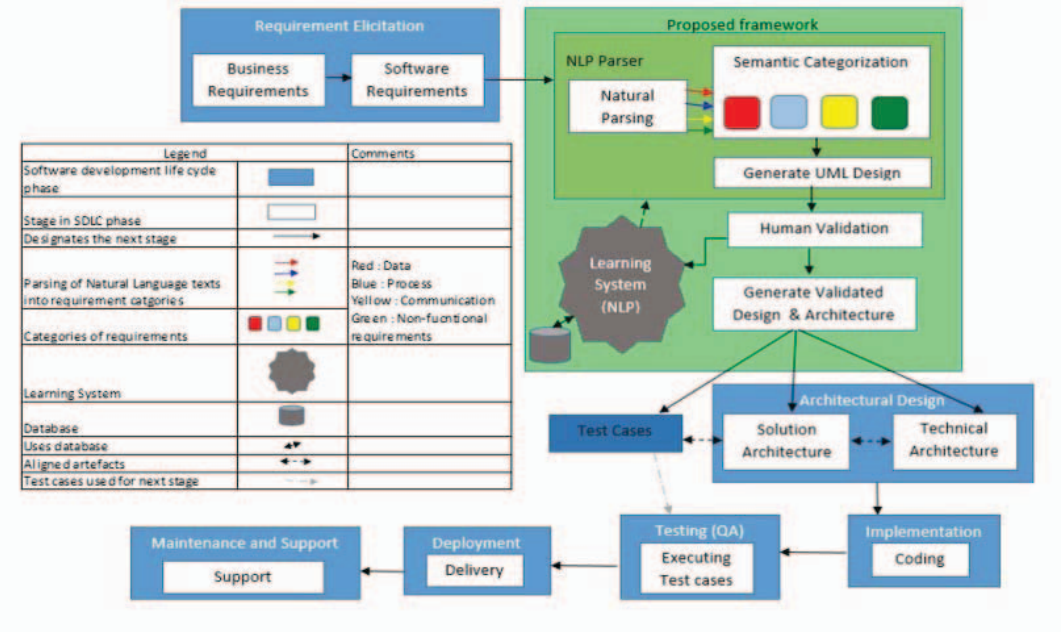


Figure 4. Proposed Approach for designing distributed software solution

All the artefacts produced by the framework are then presented to the user through a User Interface where the models can be viewed in turn. The user will also have the possibility to edit, change and modify the generated models. All the changes made by the user will be captured by the framework's learning system. The learning system is connected to a database, which will be used to store the changes made by the user and other parameters. The NLP parser will then use the data stored by the learning system so that future parses are more accurate. At this stage, a neural network is the most likely learning system which could be used.

When the user is satisfied with the models, the framework will then generate the models to be used for the design of the solution. The solutions architect will still have to validate the design artefacts outputted by the framework. The process has been partially automated and the technical architecture and the solution architecture are more likely going to be aligned, as they are modelled from the same requirement set. The models produced by the framework and the non-functional requirements can also be used to derive the test cases for the system. The test cases will also have to be validated by the QA (Quality Assurance) team but the process has been accelerated for them.

C. Advantages of the new framework

The framework has three advantages and they are listed below:

1. The framework includes a learning system as a reinforcement model. The learning system will ensure that the framework learns from the modifications made by the user and evolves over time. The parsing capacity of the framework is hence enhanced. So far, there are no existing framework which makes use of a learning system, resulting in an evolving algorithm.
2. The requirements are categorized in four broad categories. (Data, process, communication and non-functional requirements). The framework will translate the natural language requirements for three categories of the functional requirements into design models. The proposed framework provides a comprehensive tool to model data requirements, process requirements and communication requirements for distributed systems.
3. The risk of encountering a defect arising from an inadequate architecture is reduced since the framework has automated the generation of the design models. Therefore the solution architecture and the technical architecture are less likely to diverge from the requirements.

V. CONCLUSION AND FUTURE WORK

There are a few existing frameworks which automate the translation of Natural Language requirements into design artefacts. However none of the existing frameworks provide a comprehensive approach to model data, communication and process requirements for distributed systems. Therefore a framework was proposed to this end, including a learning system in order to ensure that the parsing capacity of the tool evolves over time. The proposed framework can be used in conjunction with the toolset available for ZDLC.

The future work will now involve the research and development of the proposed framework. Firstly the detailed design will be elaborated. Then different algorithms to parse the Natural Language requirements will be implemented and evaluated. The learning system will also be coded, tested and evaluated. The user interface and the way the user interacts with the system will have to be crafted. Once the initial

framework is in place, advance coding, testing and evaluation will take place. The overall process will be documented so that the accuracy of the final tool can be evaluated.

REFERENCES

[1] Charette, Robert N. "Why software fails." IEEE spectrum 42, no. 9 (2005): 36.

[2] Zhivich, Michael, and Robert K. Cunningham. "The Real Cost of Software Errors." IEEE Security & Privacy Magazine 7.2 (2009): pp 87–90. © 2012 IEEE

[3] "A study in project failure", http://www.bcs.org/content/conwebdoc/19584, Accessed 24th April, 2014

[4] Mäntylä, Mika V. and Juha Itkonen. "How are software defects found? The role of implicit defect detection, individual responsibility, documents, and knowledge." Information and Software Technology (2014).

[5] Adeel, Kashif, Ahmad Shams and Akhtar Sohaib. "Defect prevention techniques and its usage in requirements gathering-industry practices." Engineering Sciences and Technology, 2005. SCONEST 2005. Student Conference on. IEEE, 2005. pp 1 – 5.

[6] Boehm, Barry and Victor R. Basili, "Software Defect Reduction Top 10 List," Computer, vol. 34, no. 1, Jan. 2001, pp 135–137

[7] "Reducing rework through effective requirements management." (2009), http://public.dhe.ibm.com/common/ssi/ecm/en/raw14192usen/RAW14192USEN.PDF, Accessed 19th June, 2014

[8] Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Volume 25, Issue 6, June 2009, pp 599-616, ISSN 0167-739X

[9] Weinhardt, Christof, Wirt Arun Anandasivam, Benjamin Blau, Nikolay Borissov, Thomas Meinl, Wirt Wibke Michalk, and Jochen Stößer, (2009). "Cloud computing–a classification, business models, and research directions". Business & Information Systems Engineering, 1(5), pp 391-399.

[10] Rings, Thomas, Geoff Caryer, Julian Gallop, Jens Grabowski, Tatiana Kovacikova, Stephan Schulz, Ian Stokes Rees. "Grid and Cloud Computing : Opportunities for Integration with Next generation Network". Number 3, s.l J Grid Computing, 2009, Vols. Volume 7, ISSN 1570-7873 (Print) 1572-9814 (Online)

[11] Khaddaj, Souheil, and Gerard Horgan. "The evaluation of software quality factors in very large information systems." Electronic Journal of Information Systems Evaluation 7.1 (2004): pp 43-48.

[12] http://ovum.com/2013/02/04/zero-deviation-lifecycle-gives-requirements-engineering-and-software-modeling-a-refresh/, accessed July 2014.

[13] B. Makoond, ZDLC Overview, http://0deviation.com/zdlc-platform/overview/, accessed July 2014.

[14] Mohd, Ibrahim and Rodina Ahmad. "Class diagram extraction from textual requirements using Natural language processing (NLP) techniques." Computer Research and Development, 2010 Second International Conference on Computer Research and Development. IEEE, 2010. pp 200-204.

[15] Deeptimahanti, Deva Kumar and Ratna Sanyal. "Semi-automatic generation of UML models from natural language requirements." Proceedings of the 4th India Software Engineering Conference. ACM, 2011. pp 165-174.

[16] Ambriola, V., & Gervasi, V. (1997, November). "Processing natural language requirements". In Automated Software Engineering, 1997. Proceedings., 12th IEEE International Conference (pp. 36-45). IEEE.

[17] Zhou, Xiaohua, and Nan Zhou. "Auto-generation of Class Diagram from Free-text Functional Specifications and Domain Ontology." (2008).

[18] Ryan, K. "The role of natural language in requirements engineering". Proceedings of the IEEE Int. Symposium on Requirements Engineering. San Diego, CA, pp. 240-242., (1993).

[19] Osborne, Miles, and C. K. MacNish. "Processing natural language software requirement specifications." *In Requirements Engineering*, 1996., Proceedings of the Second International Conference on, pp. 229-236. IEEE, 1996.

[20] Harmain, Harmain M., and R. Gaizauskas. "CM-Builder: an automated NL-based CASE tool." *In Automated Software Engineering*, 2000. Proceedings ASE 2000. The Fifteenth IEEE International Conference on, pp. 45-53. IEEE, 2000.

[21] Goldin, L., Berry, D.M.: "AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation". *Automated Software Eng*. 4 (1997) pp.375–412

[22] Kof, Leonid, "Natural language processing: mature enough for requirements documents analysis?" *Natural Language Processing and Information Systems*. Springer Berlin Heidelberg, 2005. pp 91-102.

[23] "OMG XML Metadata Interchange". Object Management Group, MOF 2.0/XMI Mapping, v.2.4.1, http://www.omg.org/spec/XMI/2.4.1/PDF/, Accessed 23rd April, 2014

Reducing Communication Overhead in the High Performance Conjugate Gradient Benchmark on Tianhe-2

Fangfang Liu*, Chao Yang*†, Yiqun Liu *‡, Xianyi Zhang*‡ and Yutong Lu§
*Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
†State Key Laboratory of Computer Science, Chinese Academy of Sciences, Beijing 100190, China
‡University of Chinese Academy of Sciences, Beijing 100049, China
§Dept. of Computer Science & Technology, National University of Defense Technology, Changsha, Hunan 410073, China
{fangfang,yangchao,yiqun10,xianyi}@iscas.ac.cn, ytlu@nudt.edu.cn

***Abstract*—The High Performance Conjugate Gradient (HPCG) benchmark, proposed recently in 2013, has drawn increasingly more attention from both academia and industry. Unlike the High Performance Linpack (HPL) benchmark, which has a very high computation-to-communication ratio, HPCG contains both neighboring and global communication that may severely degrade the parallel performance. To reduce the communication overhead of neighboring communications, we overlap halo updates with halo-independent computations. To hide the cost of the global reductions in vector dot-products, we make use of two reformulated CG algorithms, namely the Gropp's asynchronous CG and the pipelined CG. Some further optimizations are done to decrease the extra overhead introduced in the reformulated CG algorithms. We show by experiments on the world's largest heterogeneous system – Tianhe-2 that the optimized HPCG code scales to 256 nodes (49,920 cores) with a nearly ideal weak scalability of over 90% and an aggregate performance of 10.51Tflops.**

***Keywords*-HPCG; communication-computation overlap; pipelined CG; asynchronous CG; Tianhe-2**

I. Introduction

As the de-facto standard to rank supercomputing systems on the TOP500 List, the High Performance Linpack (HPL) benchmark has been serving the HPC community for over 20 years. However, the gap between HPL and real applications is becoming increasingly larger. It is therefore of great importance, from both academic and industrial points of view, to pay a closer attention to more synthetic benchmarks that have a tighter bound to real applications.

Proposed in 2013, the High Performance Conjugate Gradient (HPCG) benchmark [1], [2] aims to better correlate computation and data access patterns found in many applications nowadays. Unlike HPL that has a very high computation-to-communication ratio, HPCG contains both neighboring and global communication that may severely degrade the parallel performance. It is a demanding task to optimize HPCG on a high computing throughput, yet low data-moving bandwidth system often found on today's TOP500 List.

The challenges in HPCG are twofold. One is how to fully exploit the local computational capacity of each available processing units within a single computing node. The other is how to reduce the communication cost among different computing nodes. In our previous works, we have carried out research on improving the intra-node performance of HPCG on Tianhe-2 by using the Intel Xeon CPUs [3] and by using both the CPUs and the Intel Xeon Phi coprocessors [4]. In this work, we focus on tackling the challenges of the communication overhead in HPCG from two different angles.

1) We first identify that all neighboring communications occur only in sparse matrix kernels for halo updates. By dividing each subdomain into an outer part and several inner parts, we are able to separate halo-independent computations in the inner parts from the halo-dependent ones in the out part. In this way, the cost neighboring communications can be reduced by overlapping with inner part computations.
2) The other type of communication in HPCG is the global reductions required by vector dot-products. Two reformulated CG algorithms, namely the Gropp's asynchronous CG and the pipelined CG, are employed to replace the original one in HPCG. The two variants of CG enjoy a same advantage that the global reduction can be done asynchronously with other sparse matrix kernels. Further optimizations are done to decrease the extra overhead introduced in the reformulated CG algorithms.

To examine the effectiveness of the proposed algorithms and optimization techniques, we carry out experiments on the world's largest heterogeneous system, Tianhe-2, By hiding the neighboring communication, a performance increase of as large as 10% is achieved on 256 computing nodes. Compared to the original CG and the Gropp's asynchronous CG, the pipelined CG successfully hide the cost of global communication thus further increases the overall performance. In summary, the optimized code scales to 256 computing nodes on Tianhe-2 with a nearly ideal weak-scaling efficiency of 90.4% and a aggregated performance of 10.51 Tflops.

II. Basic HPCG Algorithms

HPCG is a benchmark program that solves a sparse linear system arising from the finite difference solution of the three-dimensional Poisson equation $-\Delta u = f$ with homogeneous Dirichlet boundary conditions. The computational domain is a three-dimensional cube covered with a semistructured mesh with equally distributed mesh points along the x, y and z directions, respectively. Due to

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.6

the regularity of the mesh, a second-order accurate finite different scheme of 27-point stencil can be constructed.

The sparse linear system solved in HPCG is $Ax = b$, where A is a sparse matrix obtained from the 27-point stencil and b is the right-hand-side vector generated from a constant exact solution. By using a three-dimensional domain decomposition strategy, the computational domain is divided into subdomains for distributed parallel computing. To achieve high performance, it is allowed to use any data formats of both sparse matrices and dense vectors in HPCG, but it is not allowed to take advantage of the specific structure of the problem to avoid the indirect memory access patterns in all sparse matrix kernels. We use the latest version of HPCG, v2.4, in our research.

In HPCG, the linear system $Ax = b$ is solved by using a preconditioned Conjugate Gradient (CG) algorithm as described in Algorithm 1, where M^{-1} is the multigrid preconditioner to be discussed shortly. In the main loop of Algorithm 1, the major cost consists of one sparse matrix vector multiplication (SpMV) in line 7, one multigrid preconditioner (MG) in line 3, three vector dot-products in line 4, 7 and 10, and three vector updates (WAXPBY) in line 6, 8 and 9.

Algorithm 1 CG for $Ax = b$

Input: A, b, x_0, it_{max}, ε

1: $r_0 \leftarrow b - Ax_0$
2: **for** $i = 0, 1, ..., it_{max}$ **do**
3: $\quad z_i \leftarrow M^{-1} r_i$
4: $\quad s_i \leftarrow (r_i, z_i)$
5: $\quad$ **if** $(i = 0)$ $\quad p_i \leftarrow z_i$
6: $\quad$ **else** $\quad p_i \leftarrow z_i + (s_i / s_{i-1}) p_{i-1}$
7: $\quad \alpha_i \leftarrow s_i / (p_i, Ap_i)$
8: $\quad x_{i+1} \leftarrow x_i + \alpha_i p_i$
9: $\quad r_{i+1} \leftarrow r_i - \alpha_i Ap_i$
10: $\quad$ **if** $(||r_{i+1}||_2 / ||r_0||_2 \leq \varepsilon)$ break;
11: **end for**

Output: x_{i+1}

The preconditioner M^{-1} employed in HPCG is based on a V-cycle geometric multigrid method, as shown in Algorithm 2. In the algorithm, a same method, denoted as SymGS, serves as the coarse-level solver, and the pre- and post-smoother as well. In SymGS(x,r), one symmetric Gauss–Seidel step is applied concurrently on each subdomain, with right-hand-side r and initial guess x. On each subdomain, SymGS(x,r) performs the following operation

$$x[i] = \left(r[i] - \sum_{j \neq i} A[i][j] * x[j] \right) / A[i][i]$$

in a given order, and then doing the same calculation in the reverse order. Here $x[i]$, $r[i]$ denote the i-th component of vector x and r, and $A[i][j]$ denotes the (i, j)-th entry of matrix A.

Algorithm 2 Geometric multigrid V-cycle: $x^h = \text{MG}(r^h)$

Input: r^2

if on the coarse level **then**
$\quad x^h = \text{SymGS}(0, r^h)$ ▷ Coarse-level solver
else
$\quad x^h = \text{SymGS}(0, r^h)$ ▷ Pre-smoother
$\quad r^{2h} = I_h^{2h}(r^h - A^h x^h)$ ▷ Restriction
$\quad x^{2h} = \text{MG}(r^{2h})$
$\quad x^h = x^h + I_{2h}^h x^{2h}$ ▷ Prolongation
$\quad x^h = \text{SymGS}(x^h, r^h)$ ▷ Post-smoother
end if

Output: x^h

III. Proposed Optimizations of Communication

In this section, we provide details on optimizing the communication cost in HPCG. Our work is done based on our recent implementation of a heterogeneous CPU-MIC algorithm to fully exploit the computational capacity of different processing units in each node of Tianhe-2. In the heterogeneous algorithm, each subdomain is divided with an adjustable inner-outer partitioning strategy, with each inner part assigned to an MIC device and the only outer part to the CPU. Consequently, the inner tasks are isolated from each other to avoid inter-node MPI communication. The details of the heterogeneous algorithm can be found in [4].

A. Optimizing neighboring communication

Under the three-dimensional domain decomposition, the computational domain is divided into subdomains with each one assigned an MPI process. Correspondingly, the sparse matrix A and input vector x are both divided by rows, i.e., each processor owns a partial row of A and x. Therefore, before doing computations in SpMV and SymGS, halo data updated from neighboring subdomains are required to be transfered by employing MPI neighboring communication. Because both SpMV and SymGS are very basic kernels that are repeatedly used in HPCG, reducing the cost of neighboring communication in them is of great importance.

Before doing optimization of neighboring communication, we notice that the halo exchange can be safely removed in all SymGS pre-smoothers and the SymGS solver on the coarsest level, due to the reason that the initial value of the input vector in these kernels is zero. After removing the unnecessary neighboring communications, the remaining kernels with halo exchange are SpMV operations and SymGS post-smoothers. In the reference implementation of HPCG, the halo information of each subdomain is predetermined and stored, and the halo exchange is done by posting an `MPI_Irecv` first to establish a nonblocking receive, then an `MPI_Send` to send the halo data to each neighbors one by one, and finally an `MPI_Wait` to complete the point-to-point communication. Only after all send's and receive's are done, the computation is started.

In order to overlap the neighboring communication with computation, we divide each subdomain into an outer part in which computation can only be done after receiving

halo data, and an inner part that is halo-independent. The outer part is processed by CPU, and the inner part can be processed by either CPU or other computing resources or both. In this way, when doing the halo exchange, the computations in the inner part can be started without waiting any halo data. This can be done by starting the computation of the inner part immediately after posting `MPI_Send`. To further improve the communication performance, `MPI_Send` is replaced by its nonblocking variant `MPI_Isend`.

B. Optimizing global communication

As is well known, the time of global collective communication increases tremendously as more MPI processes are used. In the standard CG algorithm, the global communication occurs when calculating the two vector dot-products at line 4 and 7 and the norm of the residual vector at line 10. The computing of a vector 2-norm can be viewed as a vector dot-product with the two input vectors being the same. Therefore, there are three vector dot-products in each iteration of the CG algorithm, with each one requiring a global communication. In addition, the inputs of the three vector dot-products are only available right before the dot-products, and the results of them are immediately required right after. The observation made above indicates that there is little opportunity to overlap the global communication with computation due to the reason that it is impossible to start the vector dot-products earlier or end them later than shown in Algorithm 1.

Recent years, people have made some progresses in reformulating the CG algorithm in a mathematically equivalent way in order to increase the chance of communication-computation overlap. Among the works, we pay special attention to the asynchronous CG proposed by W. Gropp [5], and the pipelined CG proposed by P. Ghysels and W. Vanroose [6]. The two algorithms are listed in Algorithm 3 and Algorithm 4, respectively.

Algorithm 3 Gropp's asynchronous CG for $Ax = b$

Input: A, b, x_0, it_{max}, ε

1: $r_0 \leftarrow b - Ax_0$, $u_0 \leftarrow M^{-1}r_0$, $p_0 \leftarrow u_0$, $s_0 \leftarrow Ap_0$, $\gamma_0 \leftarrow (r_0, u_0)$
2: **for** $i = 0, 1, ..., it_{max}$ **do**
3: $\quad\delta \leftarrow (p_i, s_i)$
4: $\quad q_i \leftarrow M^{-1}s_i$
5: $\quad$**if** $(||r_i||_2/||r_0||_2 \leq \varepsilon)$ break
6: $\quad\alpha_i \leftarrow \gamma_i/\delta$
7: $\quad x_{i+1} \leftarrow x_i + \alpha_i p_i$
8: $\quad r_{i+1} \leftarrow r_i - \alpha_i s_i$
9: $\quad u_{i+1} \leftarrow u_i - \alpha_i q_i$
10: $\quad\gamma_{i+1} \leftarrow (r_{i+1}, u_{i+1})$
11: $\quad w_{i+1} \leftarrow Au_{i+1}$
12: $\quad\beta_{i+1} \leftarrow \gamma_{i+1}/\gamma_i$
13: $\quad p_{i+1} \leftarrow u_{i+1} + \beta_{i+1}p_i$
14: $\quad s_{i+1} \leftarrow w_{i+1} + \beta_{i+1}s_i$
15: **end for**

Output: x_i

Algorithm 4 Pipelined CG for $Ax = b$

Input: A, b, x_0, it_{max}, ε

1: $r_0 \leftarrow b - Ax_0$, $u_0 \leftarrow M^{-1}r_0$, $w_0 \leftarrow Au_0$
2: **for** $i = 0, 1, ..., it_{max}$ **do**
3: $\quad\gamma_i \leftarrow (r_i, u_i)$
4: $\quad\delta_i \leftarrow (w_i, u_i)$
5: $\quad m_i \leftarrow M^{-1}w_i$
6: $\quad n_i \leftarrow Am_i$
7: $\quad$**if** $(||r_i||_2/||r_0||_2 \leq \varepsilon)$ break
8: $\quad$**if** $(i = 0)$ $\beta_i \leftarrow 0$, $\quad\alpha_i \leftarrow \gamma_i/\delta$
9: $\quad$**else** $\quad\beta_i \leftarrow \gamma_i/\gamma_{i-1}$, $\alpha_i \leftarrow \gamma_i/(\delta - \beta_i\gamma_i/\alpha_{i-1})$
10: $\quad s_i \leftarrow w_i + \beta_i s_{i-1}$, $r_{i+1} \leftarrow r_i - \alpha_i s_i$
11: $\quad p_i \leftarrow u_i + \beta_i p_{i-1}$, $x_{i+1} \leftarrow x_i + \alpha_i p_i$
12: $\quad q_i \leftarrow m_i + \beta_i q_{i-1}$, $u_{i+1} \leftarrow u_i - \alpha_i q_i$
13: $\quad z_i \leftarrow n_i + \beta_i z_{i-1}$, $w_{i+1} \leftarrow w_i - \alpha_i z_i$
14: **end for**

Output: x_i

As shown in the two algorithms, there are still three vector dot-products in each iteration of both the asynchronous CG and the pipelined CG. Thanks to the rearranged order of calculations, the strong data dependency is removed so that all global communications in the two algorithms can be overlapped with some computations. For example, in Algorithm 3, the global reductions in the two vector dot-products at line 3 and 5 can be merged to a single one and the calculations of the two vector dot-products can be started as early as at the beginning of line 5 and the results are only needed at line 6. In this way, these two global reductions can be overlapped with the MG preconditioner at line 4. Similarly, the global reduction at line 10 in Algorithm 3 can be overlapped with the SpMV at line 11. In total, there left two global reductions in the asynchronous CG, with one overlapped with MG and another with SpMV. In an analogous way, the global reductions in the three vector dot-products in the pipelined CG can be merged into one and overlapped with MG and SpMV at line 5-6 in Algorithm 4.

In implementation, the communication-computation overlap in the two CG variants can be done by issuing a nonblocking reduction `MPI_Iallreduce` at the earliest possible place to start the global communication, then proceed the computation such as MG or SpMV or both, and finally collect the reduction result by using `MPI_Wait`. To improve the response time of communication, we insert `MPI_Test` right after issuing the nonblocking reduction.

Compared to the asynchronous CG, the pipelined CG enjoys two potential advantages. One is that the total number of global reductions per pipelined CG iteration is reduced to one instead of two in the asynchronous CG. The other is that the global reduction in the pipelined CG is overlapped with the successive application of MG and SpMV, which ensures a larger space to hide the cost of global communication. Instead, in the asynchronous CG, the two global reductions are respectively overlapped with MG and SpMV. The latter advantage of the pipelined CG algorithm is of great value when thousands of MPI pro-

cesses are used and the cost of SpMV is not comparable to a single global reduction.

There are some extra cost in the asynchronous and the pipelined CG. Compared to the standard CG, the asynchronous CG requires to store two more vectors and to perform two more WAXPBY's, and the pipelined CG requires to store five more vectors and to perform five more WAXPBY's. Fortunately, the extra cost of WAXPBY's in the pipelined CG can be reduced by fusing the input and output of related kernels. For example, each two WAXPBY's in a same line at line 10-13 can be fused together. In this way, the cost of memory traverse in theses kernels are greatly reduced.

Similar optimizations can be done when computing the local results of the three dot-products in line 3, 4 and 7 in the pipelined CG. Usually, the three dot-products require to read six vectors, but here in the pipelined CG only three vectors are needed. By fusing the local computation in the three dot-products, it is expected that the memory footprint is reduced by half.

We remark here that although in some cases the number of iterations of the two reformulated CG algorithms may be slightly larger than that of the standard CG due to numerical instability, no changes of the number of iterations are observed in HPCG.

C. Putting all together

The workflow of the two CG variants, with optimizations on both neighboring communication in SpMV (the optimizations on neighboring communication in SymGS are not shown) and global communication in the vector dot-products are shown in Figure 1. Synchronization points to wait the results of nonblocking communications are also marked in the figures. It is clearly seen from the figures that the two algorithms both are able to hide most cost in the neighboring and global communications. The tests done in the next section will confirm it.

IV. Performance and Analysis

Experiments are conducted on the Tianhe-2 supercomputer. Each node of Tianhe-2 is comprised of two 12-core Intel Xeon E5-2692 CPU processors and three 57-core Intel Xeon Phi 31S1P MIC coprocessors. All computing nodes are connected with a customized network named TH Express-2. The bi-directional bandwidth of TH Express-2 can achieve 20GB/s in theory, and offloaded collective operations are supported.

The implementation is based on our hybrid implementation of HPCG v2.4, in which each computing node of Tianhe-2 is assigned with an MPI process. Within each node, all CPU cores and the three Intel Xeon Phi devices are utilized to process a subdomain. In the tests, we fix the sub-domain size to be $NZ * NY * NX$=$128 * 456 * 400$, of which the inner block size are $120 * 448 * 128$.

A. Effects of kernel fusion

In the pipelined CG, each two of the eight WAXPBY's at line 10-13 can be fused together. Similarly, the local computation in the three vector dot-products at line 3,4 and 7 can be fused to avoid some unnecessary cost of memory traverse. We test the effects of these two kernel fusion techniques on a single node of Tianhe-2. The results are shown in Figure 2. The performance improvement is 32% in WAXPBY, and 75% in dot-product. Overall, this single technique increases the total performance by 2.5%.

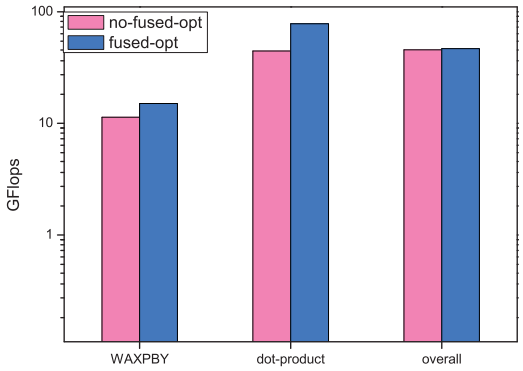


Figure 2. The performance improvement by kernel fusion.

B. Effects of optimizing neighboring communication

We then perform tests to examine the effects of optimizing neighboring communication. The tests are based on the optimized hybrid code of the pipelined CG with kernel fusions. Figure 3 shows the results, from which we observe that as the number of nodes increases, more performance improvement from optimizing neighboring communication is obtained. Compared to the case without optimizing neighboring communication, the speedup of the optimized code is as high as 1.1 when using 256 nodes.

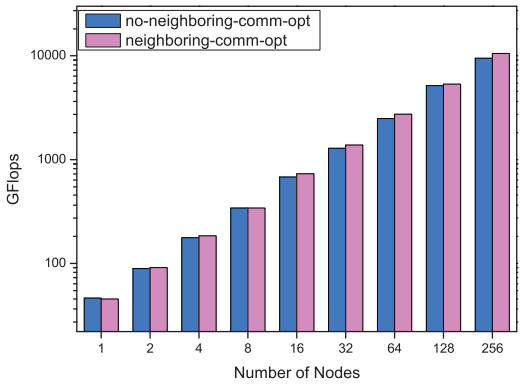
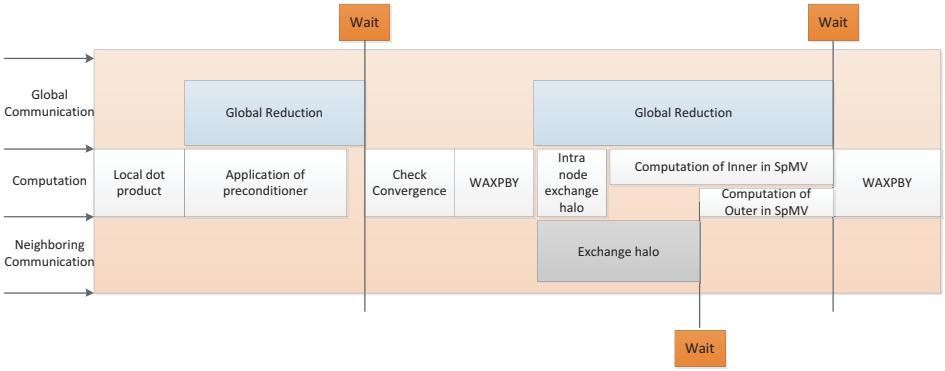


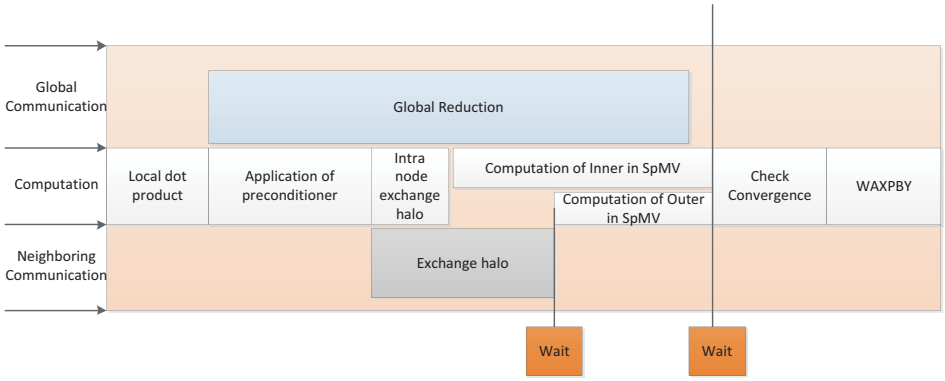
Figure 3. The performance improvement by optimizing neighbor communication.

C. Effects of optimizing global communication

To investigate the effects of optimizing global communication, we count the time of global reductions in the original CG algorithm and its two variants, and show the results in Figure 4. Note that as long as there are computation being done, the time of the nonblocking communication is not counted. In this way, we are able to examine if the communication-computation overlap is successful. From the results we clearly see that both the



(a) Gropp's asynchronous CG.



(b) Pipelined CG.

Figure 1. The workflow of the two reformulated CG algorithms. The optimizations done on global communication in vector dot-products and on neighboring communication in SpMV are shown in the figures.

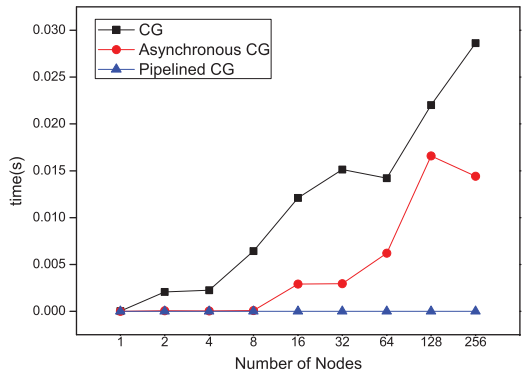


Figure 4. The overhead of global communication per CG iteration.

CG variants reduce the global communication time as compared to the original CG. Between the two algorithms, the pipelined CG is obviously the winner due to the complete hiding of all the cost of global communication.

D. Overall performance

Figure 5 shows the final overall performance of the reference HPCG implementation, the hybrid implementation of the original CG algorithm, the hybrid implementation of Gropp's asynchronous CG, and the hybrid implementation of pipelined CG. For the purpose of performance comparison, all optimizations mentioned in the paper are applied in latter two methods, but not in the hybrid implementation of the original CG. Compared to the reference HPCG implementation, the three hybrid versions all achieve great performance boost. When 256 nodes are used, with the help of communication optimizations, the asynchronous CG outperforms the original CG by 10.6%, and the pipelined CG provides another 4.1% performance increase. For the pipelined CG, when increasing the number of nodes from 1 to 256, a nearly ideal weak scalability of 90.4% is sustained, with an aggregate performance of 10.51 Tflops.

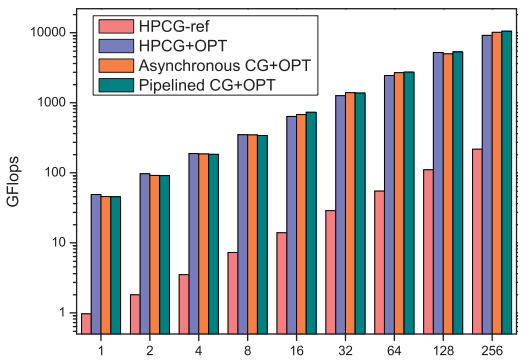


Figure 5. The overall performance improvement.

V. Concluding Remarks

This paper focuses on optimizing the communication performance of HPCG on Tianhe-2. Based on a previously developed hybrid implementation, we first optimize the neighboring communication by overlapping halo exchange with halo-independent computations. Then we make use of two reformulated CG algorithms, namely the Gropp's asynchronous CG and the pipelined CG, to merge the global communication and hide it with other computations. Some optimizations are also done to further reduce the extra costs of the two CG variants. Experiments on Tianhe-2 with up to 256 hybrid nodes show that the optimized code scales to 256 nodes (49,920 cores) with a nearly ideal weak scalability of over 90% and an aggregate performance of 10.51Tflops. We plan to conduct a more thorough research on optimizations of communication in HPCG and carry out a larger scale test on Tianhe-2 with thousands of computing nodes in the future.

Acknowledgment

We thank NUDT and NSCC-Guangzhou for providing us early access to the Tianhe-2 supercomputer. The work was partially supported by NSF China (grants 61170075 and 91130023) and 973 Program of China (grant 2011CB309701).

References

[1] J. Dongarra and M. A. Heroux, "Toward a new metric for ranking high performance computing systems," Sandia National Laboratories, Sandia Report SAND2013-4744, 2013.

[2] J. Dongarra and P. Luszczek, "HPCG technical specification," Sandia National Laboratories, Sandia Report SAND2013-8752, 2013.

[3] X. Zhang, C. Yang, F. Liu, Y. Liu, and Y. Lu, "Optimizing and scaling HPCG on Tianhe-2: early experience," in *Proc. 14th Int'l Conf. on Algorithms and Architectures for Parallel Processing (ICA3PP'14)*. Springer, 2014, to appear.

[4] Y. Liu, X. Zhang, C. Yang, F. Liu, and Y. Lu, "Accelerating HPCG on Tianhe-2: a hybrid CPU-MIC algorithm," in *Proc. 20th IEEE Int'l Conf. on Parallel and Distributed Systems (ICPADS'14)*. IEEE, 2014, under review.

[5] W. Gropp, "Update on libraries for Blue Waters."

[6] P. Ghysels and W. Vanroose, "Hiding global synchronization latency in the preconditioned Conjugate Gradient algorithm," *Parallel Computing*, 2013, in press.

Iterative Krylov Methods for Acoustic Problems on Graphics Processing Unit

Abal-Kassim Cheik Ahamed, Frédéric Magoulès
CUDA Research Center &
Applied Mathematics and Systems Laboratory
Ecole Centrale Paris, France
Email: frederic.magoules@hotmail.com

***Abstract*—This paper deals with linear algebra operations on Graphics Processing Unit (GPU) with complex number arithmetic using double precision. An analysis of their uses within iterative Krylov methods is presented to solve acoustic problems. Numerical experiments performed on a set of acoustic matrices arising from the modelisation of acoustic phenomena inside a car compartment are collected, and outline the performance, robustness and effectiveness of our algorithms, with a speed-up up to 28x for dot product, 9.8x for sparse matrix-vector product and solvers.**

***Keywords*—Linear algebra; Iterative Krylov methods; CSR matrix; GPU; CUDA; Acoustic; Helmholtz equation; Parallel computing;**

I. Introduction

Linear algebra analysis has always been extremely useful when solving partial differential equations arising from many domains such as physics and biology models. Even though Graphics Processing Units (GPUs) were first designed for graphic applications, they also represent a high potential for scientific computing and its applications to both physics and engineering. General-Purpose GPUs allow the developers to harness the high computational power of graphics cards to accelerate general-purpose scientific and engineering computing. The peak performance of CPUs and GPUs is significanlty different, due to the inherently different architectures between these processors. In this work we focus on Compute Unified Device Architecture (CUDA) [45], proposed by NVIDIA in 2006, an appropriate and suitable language for NVIDIA graphics card. CUDA has offered a new vision in high performance computing. In this paper, we analyse double precision complex number arithmetics algorithms of *Alinea* [10], [11], our own research group library, which proposes linear algebra operations and iterative Krylov on both CPU and GPU clusters for real and complex number arithmetics in single and double precision.

The acoustic problem is steered in the frequency domain by the Helmholtz equation with suitable boundary conditions. The matrix of the linear system arising from the finite element discretization of the acoustic problem has a very huge size on high frequency regime. Several discretization techniques like infinite element [1]–[3] or stabilized finite element [21] allows to reduce the size of the matrix. The problem to solve comes from the discretization of the Helmholtz equation in a bounded domain Ω, with outside boundary $\Gamma = \partial\Omega$. The Helmholtz equation is formulated as: $-\nabla^2 u - k^2 u = g$, where $k = \frac{2\pi F}{c}$ is the wavenumber of the frequency $F \in \mathbb{R}$ and $c \in \mathbb{R}$ is the velocity of the medium, which is different in space. In this work, we consider Dirichlet boundary conditions along a part of Γ. Numerical experiments done on a set of acoustic finite element matrices are exhibited and show the performance, robustness and accuracy of linear algebra operations and their uses within iterative Krylov methods for solving acoustic problem modeled by Helmholtz equation.

The plan of this paper is the following. Section II presents the industrial test cases involved to analyze our algorithms. Section III presents numerical results of linear algebra operations required to carry out iterative Krylov methods such as addition of vectors, scale of vectors, sparse matrix-vector multiplication (SpMV), etc. Section IV gives numerical tests on iterative Krylov methods, and Section V gives conclusion.

II. Application: automotive acoustic

This part of the paper gives the main features of the finite element meshes used associated with the acoustic problems arising from the automotive industry [34], namely car compartments: Audi (Audi3D) and Twingo (Twingo3D). The car compartment problem is representative of acoustics cavity. Fig. 1 illustrates respectively the Audi3D and Twingo3D mesh for a given mesh size (h).

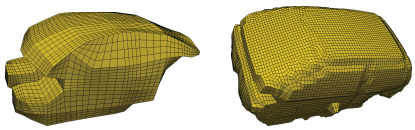


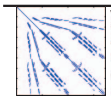
Fig. 1: Audi (Audi3D) and Twingo (Twingo3D)

The matrices used to analyze and evaluate our algorithms are obtained from the finite element discretization of the acoustic problem, governed by the Helmholtz equation. The matrices are sparse large size, *i.e.*, most values are zero. In this way, Compressed Sparse Row (CSR) [6], is considered to store these matrices. Table I reports the matrices associated with the meshes of the car compartments. These features and characteristics are given in the third column. The sparse structure pattern and an histogramm of the distribution of nonzero values are respectively given in the first and second column.

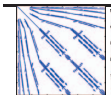
The numerical experiments have been carried out on a workstation based on an Intel Core i7 920 2.67Ghz, which

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.7

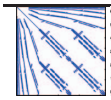
TABLE I: Sketches of Audi, Twingo FE matrices



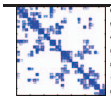
Audi3D-1
h = size = 1,727 density = 0.550
bandwidth = 1,436 nonzero = 16,393
nonzero/h = 9.492 max row = 27
nonzero/h stddev = 10.205
3D acoustic FE matrix. Audi car (mesh size = 0.133425, length wave = 3.5).



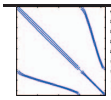
Audi3D-2
h = size = 11,637 density = 0.139
bandwidth = 11,237 nonzero = 188,455
nonzero/h = 16.194 max row = 27
nonzero/h stddev = 11.223
3D acoustic FE matrix. Audi car (mesh size = 0.066604, length wave = 3.5).



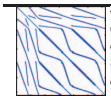
Audi3D-3
h = size = 85,001 density = 0.025
bandwidth = 84,474 nonzero = 1,781,707
nonzero/h = 20.961 max row = 27
nonzero/h stddev = 9.832
3D acoustic FE matrix. Audi car (mesh size = 0.033289, length wave = 3.5).



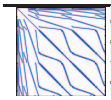
Audi3D-4
h = size = 648,849 density = 0.004
bandwidth = 520,461 nonzero = 15,444,211
nonzero/h = 23.802 max row = 27
nonzero/h stddev = 7.720
3D acoustic FE matrix. Audi car (mesh size = 0.016643, length wave = 3.5).



Twingo3D-0
h = size = 8,439 density = 0.202
bandwidth = 6,268 nonzero = 143,889
nonzero/h = 17.050 max row = 27
nonzero/h stddev = 11.047
3D acoustic FE matrix. Twingo car (mesh size = 0.077866, length wave = 9.5).



Twingo3D-1
h = size = 62,357 density = 0.035
bandwidth = 53,935 nonzero = 1,351,521
nonzero/h = 21.674 max row = 33
nonzero/h stddev = 9.364
3D acoustic FE matrix. Twingo car (mesh size = 0.038791, length wave = 9.5).



Twingo3D-2
h = size = 479,169 density = 0.005
bandwidth = 470,625 nonzero = 11,616,477
nonzero/h = 24.243 max row = 39
nonzero/h stddev = 7.233
3D acoustic FE matrix. Twingo car (mesh size = 0.019379, length wave = 9.5).

has 4 physical cores and 4 logical cores, 12GB RAM, and two NVIDIA graphics card: a Tesla K20c (device #0) with 4799GB memory and GeForce GTX 570 with 1279MB memory (device #1). The cards are double precision compatible. In the following Tesla K20c and GTX 570 will be denoted respectively *gpu#0* and *gpu#1*. For the sake of accuracy, we perform each operation 100 times, and the time indicated corresponds to the average time.

III. Linear Algebra Operations

This section introduces linear algebra algorithms such as assign of a vector, scale of vectors, element wise product, addition of vectors, dot product and sparse matrix-vector products. CUDA was originally dedicated for integer arithmetics and then for real numbers arithmetics, with a decreasing of performance of computations. Since, a complex number is a set of two real numbers composed of real and imaginary part, implementation is feasible by designing a structure of two real numbers. CUDA library includes a structure called `cuComplex`, but for performance reasons, we specify our own complex class template structure `complex<T>` that offers all the operations given by the standard `std::complex`. As a result, in order to get the most benefits of GPU architecture, the elementary linear operation kernel requires to be reimplemented [5], [6], [22]. In reference [10], an analysis carried out on real number artihmetics with double precision with a suitable implementation of the CUDA kernel presents excellent speed-up for linear algebra operations and iterative Krylov methods [11]. The finite element discretization of the Helmholtz equation for acoustic problems conducts to complex number arithmetics matrices. In this paper, we give an extension of this analysis with acoustic problem. We develop efficient iterative Krylov methods for solving linear systems with complex number arithmetics. As proved in [12] for real number arithmetics, our template code gives effective results compared to Cusp [7], CUBLAS [43], CUSPARSE [44]. But performance for complex number arithmetics with double precision remains a defiance, and dynamic auto-tuning of the GPU grid should be considered considered [10].

The complex double precision running times in milliseconds (ms) of the *assign operation* are collected in Table II, with h the size of the vector.

TABLE II: Assign of vector (ZASSIGN)

h	cpu	cpu	gpu#0	gpu#0	gpu#1	gpu#1	ratio#0	ratio#1
	time (ms)	*Gflops*	*time (ms)*	*Gflops*	*time (ms)*	*Gflops*	*cpu/#0*	*cpu/#1*
648,849	1.10	0.59	0.16	4.06	0.20	3.27	**6.88**	**5.54**
2,000,000	4.00	0.50	0.41	4.92	0.46	4.36	**9.84**	**8.72**
9,000,000	18.33	0.49	1.79	5.04	1.69	5.31	**10.27**	**10.82**
14,000,000	27.50	0.51	2.63	5.32	2.86	4.90	**10.45**	**9.63**

In the following, all kernels compute the global index of each thread as follows:

```
unsigned int x = blockIdx.x * blockDim.x + threadIdx.x;
unsigned int y = threadIdx.y + blockIdx.y * blockDim.y;
int pitch = blockDim.x * gridDim.x;
int idx = x + y * pitch;
```

The scale *scale operation* kernel is described as follows

```
__global__ void Scal( stdmrg::complex<double> alpha,
                      const stdmrg::complex<double>* d_x, int size) {
  if ( idx < size ) d_x[idx] = alpha * d_x[idx];
}
```

In Table III, we collect the execution times of the scale *scale operation*.

TABLE III: Scale of vectors (ZSCAL)

h	cpu	cpu	gpu#0	gpu#0	gpu#1	gpu#1	ratio#0	ratio#1
	time (ms)	*Gflops*	*time (ms)*	*Gflops*	*time (ms)*	*Gflops*	*cpu/#0*	*cpu/#1*
648,849	5.56	0.70	0.20	19.47	0.21	18.34	**27.78**	**26.17**
2,000,000	15.71	0.76	0.46	26.04	0.53	22.56	**34.10**	**29.54**
9,000,000	80.00	0.68	1.92	28.08	2.33	23.22	**41.60**	**34.40**
14,000,000	120.00	0.70	2.94	28.56	3.57	23.52	**40.80**	**33.60**

Double-precision complex Alpha X Plus Y (Zaxpy), i.e., $y[i] = \alpha \times x[i] + y[i]$, is a level one (vector) operation between two complex number arithmetics vectors in the Basic Linear Algebra Subprograms (BLAS) package. The simple CUDA kernel of Zaxpy is implemented as follows:

```
__global__ void Daxpy(stdmrg::complex<double> alpha,
                      const stdmrg::complex<double>* d_x,
                      stdmrg::complex<double>* d_y, int size) {
  if ( idx < size ) d_y[idx] = alpha * d_x[idx] + d_y[idx];
}
```

In Table IV, we present the complex number arithmetics with double precision execution times in milliseconds (ms) of *Zaxpy* operation.

TABLE IV: Addition of vectors (ZAXPY)

h	cpu	cpu	gpu#0	gpu#0	gpu#1	gpu#1	ratio#0	ratio#1
	time (ms)	*Gflops*	*time (ms)*	*Gflops*	*time (ms)*	*Gflops*	*cpu/#0*	*cpu/#1*
648,849	5.56	0.93	0.26	20.04	0.27	19.52	**21.44**	**20.89**
2,000,000	16.67	0.96	0.69	23.20	0.81	19.68	**24.17**	**20.50**
9,000,000	75.00	0.96	3.03	23.76	3.33	21.60	**24.75**	**22.50**
14,000,000	120.00	0.93	4.76	23.52	5.26	21.28	**25.20**	**22.80**

The *element wise product* or *element by element product*, i.e., $y[i] = x[i] \times y[i]$. The CUDA kernel, is described simply as:

```
__global__ void EWProduct( stdmrg::complex<double> alpha,
  const stdmrg::complex<double>* d_x,stdmrg::complex<double>* d_y, int size)
        {
  int idx = x + y * pitch;
  if ( idx < size ) d_y[idx] = d_x[idx] * d_y[idx];
}
```

Table V exhibits the double precision execution times of the *element by element product* operation.

TABLE V: Element wise product (ZAXMY)

h	cpu	cpu	gpu#0	gpu#0	gpu#1	gpu#1	ratio#0	ratio#1
	time (ms)	*Gflops*	*time (ms)*	*Gflops*	*time (ms)*	*Gflops*	*cpu/#0*	*cpu/#1*
648,849	8.33	0.47	0.28	13.66	0.29	13.55	**29.25**	**29.00**
2,000,000	25.00	0.48	0.72	16.56	0.85	14.16	**34.50**	**29.50**
9,000,000	120.00	0.45	3.03	17.82	3.33	16.20	**39.60**	**36.00**
14,000,000	180.00	0.47	4.76	17.64	5.00	16.80	**37.80**	**36.00**

Dot product operation can be very costly for large size vectors. Instead of performing a simple loop with simultaneous sums to compute the dot product, which is not very effective on GPUs, we perform it into two distinct tasks. The first is the element wise product of vectors and the second consists in summing all the results obtained at the first step. The reduction done at the second step associates each element of the input data with a thread, and at the end the partial sum of the n^{th} first elements is stored in the first thread of the current block. The final dot product result is then computed as the sum of all the partial sums of the different blocks. The double precision execution times of the *dot product* on both CPU and GPU are exposed in Table VI and Fig. 2. Table VII gives the numerical results

TABLE VI: Dot product (ZDOT)

h	cpu	cpu	gpu#0	gpu#0	gpu#1	gpu#1	ratio#0	ratio#1
	time (ms)	*Gflops*	*time (ms)*	*Gflops*	*time (ms)*	*Gflops*	*cpu/#0*	*cpu/#1*
648,849	5.56	0.93	0.33	15.83	0.33	15.94	**16.94**	**17.06**
2,000,000	16.67	0.96	0.83	19.20	0.76	20.96	**20.00**	**21.83**
9,000,000	80.00	0.90	3.23	22.32	3.23	22.32	**24.80**	**24.80**
14,000,000	130.00	0.86	4.76	23.52	4.55	24.64	**27.30**	**28.60**

of the norm operation.

As shown in Table VI and Table VII, GPUs clearly show better results than CPU with complex number arithmetics in

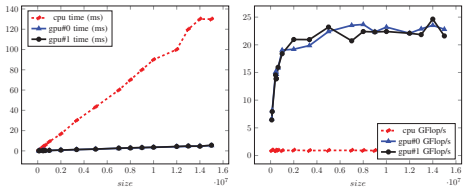


Fig. 2: ZDOT [left: time in ms, right: GFlops]

TABLE VII: NormL2 (ZNORM)

h	cpu	cpu	gpu#0	gpu#0	gpu#1	gpu#1	ratio#0	ratio#1
	time (ms)	*Gflops*	*time (ms)*	*Gflops*	*time (ms)*	*Gflops*	*cpu/#0*	*cpu/#1*
648,849	11.11	0.29	0.31	10.54	0.26	12.65	**36.11**	**43.33**
2,000,000	33.33	0.30	0.73	13.70	0.57	17.60	**45.67**	**58.67**
9,000,000	150.00	0.30	3.13	14.40	2.27	19.80	**48.00**	**66.00**
14,000,000	230.00	0.30	5.00	14.00	3.70	18.90	**46.00**	**62.10**

double precision. Much more than the dot product, the SpMV is probably the most time consuming operation in sparse matrix computation. This is required on all Krylov iterative methods. As proved in [10], the performance of SpMV strongly depends on the properties of the matrix, particularly on the distribution of nonzero values. The following results are obtained with advanced auto-tuned techniques to organize threads on the CUDA grid. References [16], [23], [42], [51], [52] clearly showed the effectiveness of SpMV on GPU compared to CPU for real number arithmetics. The running time and the number of floating operations per second for SpMV with complex number arithmetics with double precision are reported in Table VIII.

TABLE VIII: SpMV CSR

problem	cpu	cpu	gpu#0	gpu#0	gpu#1	gpu#1	ratio#0	ratio#1
	time (ms)	*Gflops*	*time (ms)*	*Gflops*	*time (ms)*	*Gflops*	*cpu/#0*	*cpu/#1*
Audi3D-0	0.01	0.61	0.07	0.12	0.06	0.13	**0.19**	**0.21**
Audi3D-1	0.20	0.67	0.11	1.23	0.12	1.07	**1.84**	**1.60**
Audi3D-2	2.22	0.68	0.37	4.03	0.42	3.56	**5.93**	**5.24**
Audi3D-3	20.00	0.71	2.22	6.41	3.03	4.70	**9.00**	**6.60**
Audi3D-4	180.00	0.69	18.33	6.74	24.00	5.15	**9.82**	**7.50**
Twingo3D-0	1.67	0.69	0.28	4.06	0.33	3.45	**5.88**	**5.00**
Twingo3D-1	15.71	0.69	1.79	6.05	2.44	4.43	**8.80**	**6.44**
Twingo3D-2	140.00	0.66	14.29	6.51	16.67	5.58	**9.80**	**8.40**

IV. Iterative Krylov methods

After the analysis of linear algebra operations for complex number arithmetics with double precision, we now evaluate and analyze their uses within iterative Krylov methods [4], [24], [42], [52]. We have thus implemented a preconditionned bi-conjugate gradient stabilized method (P-Bi-CGSTAB), a preconditionned P-BiCGSTAB parametered (l) and a preconditionned transpose-free quasi-minimal residual method (P-tfQMR) [48], with optimized CUDA and dynamic auto-tuning on GPU. The data transfer between CPU and GPU consists of an important part of optimization [9] for optimal performance on GPGPU. In our Krylov methods codes, we take care to send once all required input data from CPU to GPU

before beginning the iterations. Even so, at each computed dot product or norm, there is one copy back from GPU to CPU. Both CPU and GPU codes are strictly the same, but all linear algebra operations such as Zdot, Znorm, Zaxpy, or SpMV are performed on device (GPU) for the GPU version. The presented iterative Krylov methods are performed with a residual tolerance threshold of 1×10^{-9}, an initial guess of zero and 1000 maximum number of iterations. The numerical experiments presented in the following give an analysis of Krylov methods on CPU and GPU, with the same code, for complex number arithmetics with double precision. The CPU and GPU execution times and corresponding speed-up of Audi3D and Twingo3D are collected in Table IX and Table X. The results corroborate the effectiveness of GPU

TABLE IX: Speed-up of Audi3D

problem	#iter	CPU time (s)	GPU time (s)	speed-up
P-BiCGSTAB				
Audi3D-1	21	0.01	0.030	**0.33**
Audi3D-2	53	0.24	0.106	**2.26**
Audi3D-3	94	4.01	0.703	**5.71**
Audi3D-4	183	85.70	9.209	**9.31**
P-BiCGSTAB(8)				
Audi3D-1	6	0.03	0.110	**0.27**
Audi3D-2	12	0.52	0.286	**1.82**
Audi3D-3	31	12.47	2.162	**5.77**
Audi3D-4	70	266.26	30.100	**8.85**
P-TFQMR				
Audi3D-1	24	0.02	0.040	**0.50**
Audi3D-2	52	0.27	0.113	**2.40**
Audi3D-3	99	4.71	0.755	**6.24**
Audi3D-4	214	102.17	10.786	**9.47**

TABLE X: Speed-up of Twingo3D

problem	#iter	CPU time (s)	GPU time (s)	speed-up
P-BiCGSTAB				
Twingo3D-0	563	1.85	1.008	**1.84**
Twingo3D-1	1000	29.45	5.730	**5.14**
Twingo3D-2	1000	295.66	37.670	**7.85**
P-BiCGSTAB(8)				
Twingo3D-0	1000	31.2	20.970	**1.49**
Twingo3D-1	1000	273.81	54.630	**5.01**
Twingo3D-2	1000	2559.67	324.500	**7.89**
P-TFQMR				
Twingo3D-0	366	1.34	0.626	**2.14**
Twingo3D-1	954	30.4	5.438	**5.59**
Twingo3D-2	1000	318.93	38.090	**8.37**

compared to CPU for solving sparse linear systems. The speed-up grows when the size of the problems increase for all tests, i.e., for a finer mesh GPU is more effective compared to CPU. For a finer mesh the assembled matrix turns into non appropriate size for memory of most of GPUs. To overcome this problem, one way consists in using domain decomposition method [17], [36], [46], [49], [50] based on iterative methods with interface conditions defined on the interface between the subdomains [29]. The Schwarz method [8], [25]–[27] is suitable for solving large size problem. To accelerate the convergence, many references [15], [20], [28], [30], [31] show the importance of these interface conditions. In order to implement this perspective for acoustic problems continuous optimized interface conditions between the subdomains must be implemented as in [19], [32], [33], [35]. Alternative discrete optimization techniques as introduced in [18], [37]–[41], [47] allow a fast and robust convergence of the Schwarz algorithm too. In [13], [14], the authors describe how domain decomposition method is effectively implemented on GPU and proved the robustness of Schwarz methods on a cluster of GPUs. The extension to the complex number arithmetics double precision, of the iterative Krylov methods, to solve the local subproblems defined in each subdomains, leads to similar speed-up. For the Audi car compartment, a speed-up up to 9.2x is obtained for eight subdomains.

V. Conclusion

In this paper we give an analysis of linear algebra operations together with their uses within iterative Krylov methods for solving acoustic problems on Graphics Processing Unit (GPU) with complex number arithmetics with double precision. Numerical tests have been carried out on two different system of accelerated generations of NVIDIA graphics card: GTX570 and Tesla K20c. A set of industrial matrices coming from the finite element discretization of acoustic problems modeled by the Helmholtz equation inside a car compartment are used to demonstrate the interest of using GPU device to perform linear algebra operations, and outline the robustness, performance and effectiveness of the proposed implementation.

References

[1] J.-C. Autrique and F. Magoulès. Numerical analysis of a coupled finite-infinite element method for exterior Helmholtz problems. *Journal of Computational Acoustics*, 14(1):21–43, 2006.

[2] J.-C. Autrique and F. Magoulès. Studies of an infinite element method for acoustical radiation. *Applied Mathematical Modelling*, 30(7):641–655, 2006.

[3] J.-C. Autrique and F. Magoulès. Analysis of a conjugated infinite element method for acoustic scattering. *Computers and Structures*, 85(9):518–525, 2007.

[4] J. M. Bahi, R. Couturier, and L. Z. Khodja. Parallel gmres implementation for solving sparse linear systems on gpu clusters. In *Proceedings of the 19th High Performance Computing Symposia*, pages 12–19, San Diego, CA, USA, 2011. Society for Computer Simulation International.

[5] N. Bell and M. Garland. Efficient sparse matrix-vector multiplication on CUDA. Nvidia Technical Report NVR-2008-004, Nvidia Corporation, 2008.

[6] N. Bell and M. Garland. Implementing sparse matrix-vector multiplication on throughput-oriented processors. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (SC'09)*, pages 1–11, New York, NY, USA, 2009. ACM.

[7] N. Bell and M. Garland. Library cusp website, 2010. Available on line at: http://cusplibrary.github.io/ (accessed on August 7, 2014).

[8] X.-C. Cai, M. A. Casarin, F. W. E. Jr, and O. B. Widlund. Overlapping schwarz algorithms for solving helmholtz's equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, page 391399. Amer. Math. Soc., Providence, RI, 1998.

[9] A. F. Camargos, V. C. Silva, J.-M. Guichon, and G. Meunier. Iterative solution on gpu of linear systems arising from the a-v edge-fea of time-harmonic electromagnetic phenomena. In *Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on*, pages 365–371, Feb 2014.

[10] A.-K. Cheik Ahamed and F. Magoulès. Fast sparse matrix-vector multiplication on graphics processing unit for finite element analysis. In *IEEE 14th International Conference on High Performance Computing and Communication (HPCC)*, pages 1307–1314. IEEE Computer Society, 2012.

[11] A.-K. Cheik Ahamed and F. Magoulès. Iterative methods for sparse linear systems on graphics processing unit. In *IEEE 14th International Conference on High Performance Computing and Communication (HPCC)*, pages 836–842. IEEE Computer Society, june 2012.

[12] A.-K. Cheik Ahamed and F. Magoulès. Iterative Krylov methods for gravity problems on graphics processing unit. In *IEEE 12th International Symposium on Distributed Computing and Applications to Business, Engineering Science (DCABES)*, pages 16–20. IEEE Computer Society, 2013.

[13] A.-K. Cheik Ahamed and F. Magoulès. Schwarz method with two-sided transmission conditions for the gravity equations on graphics processing unit. In *IEEE 12th International Symposium on Distributed Computing and Applications to Business, Engineering Science (DCABES)*, pages 105–109. IEEE Computer Society, 2013.

[14] A.-K. Cheik Ahamed and F. Magoulès. A stochastic-based optimized Schwarz method for the gravimetry equations on GPU clusters. In *Domain Decomposition Methods in Science and Engineering XXI*. Springer, 2014.

[15] P. Chevalier and F. Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.

[16] M. M. Dehnavi, D. M. Fernandez, and D. Giannacopoulos. Finite-element sparse matrix vector multiplication on graphic processing units. *IEEE*, 2010.

[17] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6):1205–1227, 1991.

[18] M. Gander, L. Halpern, F. Magoulès, and F.-X. Roux. Analysis of patch substructuring methods. *International Journal of Applied Mathematics and Computer Science*, 17(3):395–402, 2007.

[19] M. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 24(1):38–60, 2002.

[20] M. J. Gander, L. Halpern, and F. Nataf. Optimized Schwarz methods. In T. Chan, T. Kako, H. Kawarada, and O. Pironneau, editors, *Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan*, pages 15–28, Bergen, 2001. Domain Decomposition Press.

[21] I. Harari and F. Magoulès. Numerical investigations of stabilized finite element computations for acoustics. *Wave Motion*, 39(4):339–349, 2004.

[22] H. Knibbe, C. W. Oosterlee, and C. Vuik. Gpu implementation of a helmholtz krylov solver preconditioned by a shifted laplace multigrid method. *J. Computational Applied Mathematics*, 236(3):281–293, 2011.

[23] M. Kreutzer, G. Hager, G. Wellein, H. Fehske, A. Basermann, and A. R. Bishop. Sparse matrix-vector multiplication on gpgpu clusters: A new storage format and a scalable implementation. *CoRR*, abs/1112.5588, 2011.

[24] R. Li and Y. Saad. GPU-accelerated preconditioned iterative linear solvers, 2010.

[25] P.-L. Lions. On the Schwarz alternating method. I. In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42, Philadelphia, PA, 1988. SIAM.

[26] P.-L. Lions. On the Schwarz alternating method. II. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods*, pages 47–70, Philadelphia, PA, 1989. SIAM.

[27] P.-L. Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.

[28] Y. Maday and F. Magoulès. Non-overlapping additive Schwarz methods tuned to highly heterogeneous media. *Comptes Rendus à l'Académie des Sciences*, 341(11):701–705, 2005.

[29] Y. Maday and F. Magoulès. Absorbing interface conditions for domain decomposition methods: a general presentation. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3880–3900, 2006.

[30] Y. Maday and F. Magoulès. Improved ad hoc interface conditions for Schwarz solution procedure tuned to highly heterogeneous media. *Applied Mathematical Modelling*, 30(8):731–743, 2006.

[31] Y. Maday and F. Magoulès. Optimized Schwarz methods without overlap for highly heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1541–1553, 2007.

[32] F. Magoulès, P. Iványi, and B. Topping. Convergence analysis of Schwarz methods without overlap for the Helmholtz equation. *Computers and Structures*, 82(22):1835–1847, 2004.

[33] F. Magoulès, P. Iványi, and B. Topping. Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 193(45–47):4797–4818, 2004.

[34] F. Magoulès, K. Meerbergen, and J.-P. Coyette. Application of a domain decomposition method with Lagrange multipliers to acoustic problems arising from the automotive industry. *Journal of Computational Acoustics*, 8(3):503–521, 2000.

[35] F. Magoulès and R. Putanowicz. Optimal convergence of non-overlapping Schwarz methods for the Helmholtz equation. *Journal of Computational Acoustics*, 13(3):525–545, 2005.

[36] F. Magoulès and F.-X. Roux. Lagrangian formulation of domain decomposition methods: a unified theory. *Applied Mathematical Modelling*, 30(7):593–615, 2006.

[37] F. Magoulès, F.-X. Roux, and S. Salmon. Optimal discrete transmission conditions for a non-overlapping domain decomposition method for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 25(5):1497–1515, 2004.

[38] F. Magoulès, F.-X. Roux, and L. Series. Algebraic way to derive absorbing boundary conditions for the Helmholtz equation. *Journal of Computational Acoustics*, 13(3):433–454, 2005.

[39] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3742–3759, 2006.

[40] F. Magoulès, F.-X. Roux, and L. Series. Algebraic Dirichlet-to-Neumann mapping for linear elasticity problems with extreme contrasts in the coefficients. *Applied Mathematical Modelling*, 30(8):702–713, 2006.

[41] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approach to absorbing boundary conditions for the Helmholtz equation. *International Journal of Computer Mathematics*, 84(2):231–240, 2007.

[42] K. K. Matam and K. Kothapalli. Accelerating sparse matrix vector multiplication in iterative methods using GPU. In G. R. Gao and Y.-C. Tseng, editors, *ICPP*, pages 612–621. IEEE, 2011.

[43] Nvidia Corporation. Nvidia - library cublas. Available on line at: http://www.nvidia.com/object/cuda_home_new.html (accessed on August 7, 2014).

[44] Nvidia Corporation. *CUDA Toolkit 4.0, CUSPARSE Library*, 2011. Available on line at: http://developer.nvidia.com/cuda-toolkit-40 (accessed on August 7, 2014).

[45] Nvidia Corporation. *CUDA Toolkit Reference MANUAL*, 4.0 edition, 2011. Available on line at: http://developer.nvidia.com/cuda-toolkit-40 (accessed on August 7, 2014).

[46] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, Oxford, UK, 1999.

[47] F.-X. Roux, F. Magoulès, L. Series, and Y. Boubendir. Approximation of optimal interface boundary conditions for two-Lagrange multiplier FETI method. In R. K. et al, editor, *Proceedings of the 15th Int. Conf. on Domain Decomposition Methods, Berlin, Germany, Jul.21-15, 2003*, Lecture Notes in Computational Science and Engineering (LNCSE). Springer-Verlag, Haidelberg, 2005.

[48] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[49] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, UK, 1996.

[50] A. Toselli and O. Widlund. Domain decomposition methods. *Computational Mathematics*, 34, 2004.

[51] A. H. E. Zein and A. P. Rendell. From sparse matrix to optimal gpu cuda sparse matrix vector product implementation. In *CCGRID*, pages 808–813. IEEE, 2010.

[52] A. H. E. Zein and A. P. Rendell. Generating optimal CUDA sparse matrix-vector product implementations for evolving GPU hardware. *Concurrency and Computation: Practice and Experience*, 24(1):3–13, 2012.

Performance Issues and Query Optimization in Big Multidimensional Data

Jay Kiruthika
Kingston University
jay.kiruthika@gmail.com

Dr.Souheil Khaddaj
Kingston University
s.khaddaj@kingston.ac.uk

***Abstract*— SQL queries for big-data and multidimensional data need to be carefully coded to get maximum benefit of optimization and to improve the performance of the system and the applications involved. This is especially true for online applications where there is a race against time in populating the results set. Valuable time and cost can be saved by optimizing query statements during their design process especially when they are required to perform complex process involving relationships with the other tables. The relationships with the tables vary from complex joins to sorting order which has direct impact on system performance. Transactions SQL (TSQL), stored procedures, triggers etc. can be efficiently coded to optimize a query. Queries involved in 3D databases depend on user interaction have a human element added in the optimization process. This paper addresses few of the issues using a number of potential applications.**

Keywords- SQL Query Optimization; Query Performance Optimization; Optimization issues for multidimensional & BIGData.

I. INTRODUCTION

Accessing data in any type of database, involves a query or multiple queries to be formulated, executed and its results reported to the user. The software component that is responsible for the execution of a query is the query processor, which is commonly known as the query optimizer. Its main purpose is to convert the query into low-level language and execute it. Therefore, query optimization plays a major role in querying databases and in fact it has been the point of interest in various areas of research [11][17][18][19]. The query parser checks the validity of the query which is then followed by the query optimizer that examines the queries to estimate the cheapest one and feeds the code generator or the interpreter to transform the results from the optimizer to call the query processor. Finally the query processor executes the query.

Query can range from a single retrieve or select statement to complex inner joins, triggers, stored procedures and "coalesce" statements. The more complex the database is designed, the more response time it takes to execute it depending on the speed and capacity of the system. Especially if they are used via a GUI, the response time taken to execute a single query is larger than when they are executed using a query management console like the SQL server management console for SQL server databases. The GUI front ends, which use widgets and UI gadgets, add to the response time in a web application. The longer the response time it takes for a page to load in a web application, the shorter the number of users visiting the site will be. The threshold or the tolerance response time a user can wait for a web page to load is 2 minutes. If there are multiple queries with an increased complex quotient there is a high possibility of the application running slow, compromising the performance of the application and that of the system.

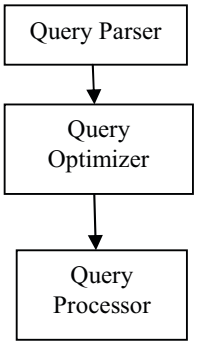


Figure 1. Query Optimization Process

This paper explores few of the issues the companies have to face when designing storage, especially involving big data and 3D databases. It presents case study involving 2 databases one a big data and the other 3D database and analyses the results. It concludes with a recommendation to improve performance by optimizing queries when such databases are involved.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.8

II. QUERIES IN BIG-DATA AND 3D DATABASES

In case of big data and multi- dimensional data such as 3D, it is essential to optimize SQL query as the response time or the cost is longer [14] than medium or small sized applications [9]. If the databases are hidden under secure levels, this adds to the response time. Most of such big and multi-dimensional databases are indexed [12] and contain views to optimize the SQL query performance [10].

TSQLs which process transactions for secure server like bank card validations depending on server side validations tend to incur additional cost to perform such queries[20][21]. Huge companies such as eBay, Amazon etc. perform such transactions in seconds and the number of users accessing the site for such transactions, run into millions. In order to manage such large number of users, care should be taken when optimizing queries. Systems with high speed processors and storage space can perform these transactions faster depending on the hardware configuration.

A cost is incurred for every query executed. There are cost models that are used to estimate the cost of the execution plans. Mathematical formulas are used to estimate the weightage of every inner joins, group by, order by, index, coalesce etc. and is used to estimate the size distribution depending on the query tree [7]. Optimization should be done from the top tree level to the sub tree level down to the child tree where the required table or entity is positioned.

Supporting live interaction with sensors and camera functionality to translate operational space of a user to 3D entity in computer is a complex process. Recent 'wearable' interactive devices like Google glass reflect back the live feed and are able to identify the user's geo location and are able to supply information. The performance of such systems needs to be constantly upgraded and modified to be able to cope with the complex processes.

3D databases are used in a variety of applications in enterprise, science and engineering, for example for storing molecular modelling techniques, which can access large data. The chemical information is stored in such a way that searching is easy by optimizing the query using various statistical and modeling methodologies [1]. 3D structural databases have become a key element in genome sequencing [3], geo spatial database [4], protein clusters database, bio informatics [2] etc. It is also used in gesture-based communication, medical systems and in interactive games based systems. Further to the query optimizer, the gesture based communication system has to identify the gesture corresponding to a task, which adds to the cost to execute such queries. In interactive games applications, especially with multiple players, the complexity of the query increases and the costing is affected depending on the type of query executed.

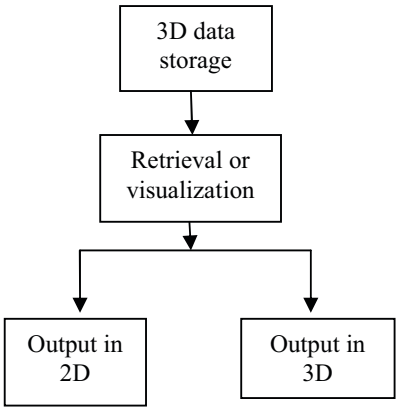


Figure 2. 3D database storage and retrieval

3D database query optimization differs from big data in the sense that the retrieval algorithm retrieves contents either as 2D or 3D. It is a difficult task to optimize the order of the execution this is especially true in object-relational databases. Especially in GIS, CAD maintaining curved faces, shapes and its storage are still a challenge and the execution time differs according to the methodology adopted. When used to store geo spatial information, it uses 3D geometric operations to store them as Object Store classes [5][6] hence the impossibility to implement optimization to such databases arise. The topological data such as the vertical data and the shapes are still in need of an optimized topological model. Visualization of such data involves whole scenes from the result of 3D queries. Few of the algorithms use adjacent objects as the base of related objects to enhance the retrieval in location based information databases. The performance of such databases again depends on the system hardware configuration and speed of the processors.

III. APPLICATIONS

A. Cloud Based Big Data

The applications involved in this paper are from two sources. One is from a Big-data and the other from a 3D database. The response time and the depth of the SQL query are recorded and analyzed. The depth of the SQL query is determined using the hierarchical tree structure. For example the top-level tree will be level 1, the child 2, the sub child 3 etc. as shown in Figure 3. The leaf as represented in the hierarchical tree structure [15] indicates that there is no sub-child under it i.e., the tree branch stops here. There can be many levels like a parent, child, grandchild, direct descendent etc. The 'ORDER By', 'GROUP BY' clauses in the SQL query has various level of tree branches[8] in them if the query involves more than one table structure and will override the hierarchical structure.

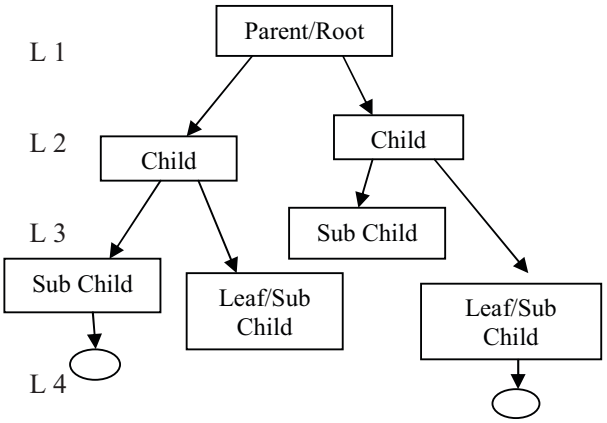


Figure 3. Hierarchical query tree

The SQL depth and the execution time for various levels are recorded in Table 1. This application uses additional UI gadgets, which is added into the time of execution.

Thus, for 50,000-query requests, which are not large in many online applications, the cost is quite significant in terms of performance of the system.

The database involved in this experiment is an RDBMS Oracle database with 248 relational entity tables storing location based information. There are views designed in this database to facilitate easy retrieval of data as the results sets need to be displayed via a web front end. The number of users visiting the site varies and an approximate number of users was around 50,000 users a day. The work load was distributed evenly as it was a cloud based environment and a virtual tool was used to load balance across the servers.

SQL depth	Time of Execution in seconds	Cost per Query in seconds
14	80	40,000
12	60	30,000
5	30	15,000
3	25	12,500

Table 1. Cost for Big-Data

As in many projects, any additional gadget is driven by other requirements for example usability. Thus in this work we are only checking how the query can be optimized which can reduce the execution time for the whole process.

Figure 4 shows the SQL depth level against the execution time in seconds.

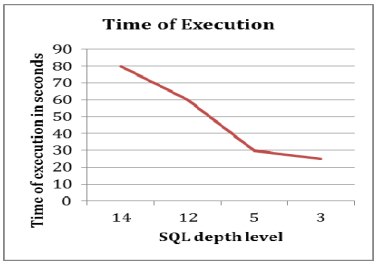


Figure 4. SQL depth level for Big Data application.

B. 3D database

The 3D database involved in this experiment uses a simple interface and 2 level queries to retrieve data from the database. It relies on the gesture of the user to interpret the actions required [13]. There were 3 basic actions, using fingers to select the column (s) and moving the fingers from left to right to navigate the faces of the 3D database and the palm gesture to close the cube. The gesture can be simplified to rotation, selection and clicking which is translated to corresponding actions. The application was developed using Kinect. This is a research based application mainly used for testing purposes aimed solely to record the response execution time of the query based on the gender and the pre-knowledge of SQL. The database used for this experiment was based on Microsoft windows Azure SQL database.

Figure 5. Example of 3D database model [16]

The number of users for this 3D database was from a test sample and the time of execution reflects the response time it takes the users to invoke the required action using gestures.

This application based on gesture analysis; therefore the performance is only related to the query optimization of the data. There was a disparity between the users based on the gender and user's knowledge of SQL. The users who took the shortest time to execute a task were the ones with a pre-knowledge of SQL. The user's difficulty level is measured using a questionnaire. The number of query request was about 100. Table 2 is the recorded SQL depth level and the time of execution for this gesture based application.

SQL depth	Time of Execution in seconds	Cost per Query in seconds
3	230	23000
1	200	20000
2	110	11000

Table 2. Cost for 3D data

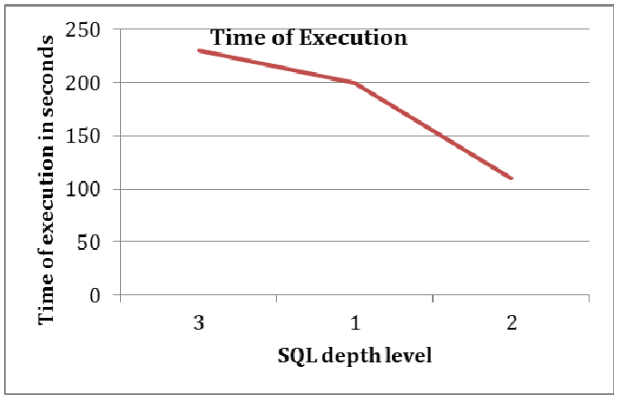


Figure 6. SQL depth level for 3D application.

The SQL depth level against the time of execution in seconds in plotted in Figure 6.

The results discussed in this section includes the execution time of the query as well as the gadgets/gesture into account.

IV. RECOMMENDATIONS

There are various ways to improve the performance using optimization of queries in databases involving 3D and Big data. It is recommended to take into account few issues listed below when designing such databases.

Big-Data

- Use indexing wherever necessary to store and retrieve databases easily
- Use views as much as possible to optimize the SQL queries
- Simpler database designs to facilitate faster searches and access
- Tree structure should be optimized to the sub tree level or the level which the query is based on
- Use of Stored procedures and triggers to access the data readily
- TSQL as a batch processing so as to save cost of transactions and processing

3D Data

- Type of storage, object based or flat 2D known data and 1 D of unknown data to facilitate easier searches
- Optimize the query based on the type of storage, complexity of the sql query such as finding the neighbour or related objects
- Retrival of the data using an algorithm to suit the purpose of the business in order to access and retrieve data faster and in an optimized fashion

V. CONCLUSION AND FUTURE WORK

As companies gravitate to object based storage when handling large amount of data, an efficient optimization technique to handle suitable storage process and retrieval process is sought after. Using

techniques suited for the purpose and the type of data is crucial as it differs when optimizing the query when handling such complex data. This is true especially in GIS, storing chemical structure and genome sequencing.

A faster response from SQL queries to reduce the costing is always sought after. Experimenting on a large 3D database involving complex structures will produce more results to work on. Further work can be carried out by isolating the execution time of the gadgets (client cost of the program) from the sql query execution to improve the performance of the whole system.

Gesture based communication and gamification of many applications has led to increase in 3D storage. Efficient ways to query such databases will continue to evolve. Especially, when used in 3D databases and storage care should be taken when costing queries. As hardware are versatile to handle huge data, a need to handle such storage of data will be a paramont of importance in coming years.

REFERENCES

[1] Julia Weber, Janosch Achenbach, Daniel Moser, and Ewgenij Proschak, 2013. VAMMPIRE: A Matched Molecular Pairs Database for Structure-Based Drug Design and Optimization. Journal of Medicinal Chemistry 2013 56 (12), 5203-5207

[2] Pickett, B. E., Sadat, E. L., Zhang, Y., Noronha, J. M., Squires, R. B., Hunt, V., ... & Scheuermann, R. H. 2012. ViPR: an open bioinformatics database and analysis resource for virology research. *Nucleic acids research*, *40*(D1), D593-D598.

[3] Ågren, J. 2014. The materials genome and CALPHAD. *Chinese Science Bulletin*, *59*(15), 1635-1640.

[4] Lewis, P., Mc Elhinney, C. P., & McCarthy, T. 2012. LiDAR data management pipeline; from spatial database population to web-application visualization. In *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications* (p. 16). ACM.

[5] Rogers, R., Isherwood, B., McDONALD, M. M., Pannese, D. P., & Pinkney, D. 2014. "System and method for optimizing protection levels when replicating data in an object storage system". *U.S. Patent Application 14/189,431*.

[6] Gao, T., Gao, Z., Li, J., Sun, Z., & Shen, M. 2011. The perceptual root of object-based storage: An interactive model of perception and visual working memory. *Journal of Experimental Psychology: Human Perception and Performance*, *37*(6), 1803.

[7] He, X., Luo, L., & Petitclerc, M. 2012. "Query optimization".*U.S. Patent Application 13/612,611*.

[8] Mandal, P., & Seputis, E. A. 2014. "Query optimization with awareness of limited resource usage.".*U.S. Patent No. 8,712,972*. Washington, DC: U.S. Patent and Trademark Office.

[9] Özcan, F., Tatbul, N., Abadi, D. J., Kornacker, M., Mohan, C., Ramasamy, K., & Wiener, J. 2014. Are we experiencing a big data bubble?. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (pp. 1407-1408). ACM.

[10] Chaudhuri, S., Dayal, U., & Narasayya, V. 2011. An overview of business intelligence technology. *Communications of the ACM*, *54*(8), 88-98.

[11] Gonçalves, F. A., Guimarães, F. G., & Souza, M. J. 2014. Query join ordering optimization with evolutionary multi-agent systems. *Expert Systems with Applications*.

[12] Day, P. R., Egan, R. L., & Mittelstadt, R. A. 2014. "Database query optimization using index carryover to subset an index." *U.S. Patent No. 8,745,033*. Washington, DC: U.S. Patent and Trademark Office.

[13] Nickel, K. and Stiefelhagen, R. 2007. Visual recognition of pointing gestures for human-robot interaction. Image and Vision Computing 25, v12, p1875–1884.

[14] Bausch, D., Petrov, I., & Buchmann, A.2012. Making cost-based query optimization asymmetry-aware. In *Proceedings of the Eighth International Workshop on Data Management on New Hardware* (pp. 24-32). ACM.

[15] Celko, J., 2012. Joe Celko's Trees and hierarchies in SQL for smarties. Elsevier. ISBN: 9780123877338.

[16] Herrera-Acuña, Raul A., Argyriou, Vasileios and Velastin, Sergio A. 2013. Graphical interfaces for development exploiting the third dimension using Kinect. *In: 9th International Conference on Intelligent Environments - IE'13;* 18-19 Jul 2013, Athens, Greece

[17] Wu, S., Li, F., Mehrotra, S., & Ooi, B. C. 2011, Query optimization for massively parallel data processing. In *Proceedings of the 2nd ACM Symposium on Cloud Computing* (p. 12). ACM.

[18] Lan, W. 2013. Based on The Variety Constraint Model of Remote Education Database Query Optimization Algorithm [J]. *Bulletin of Science and Technology*, *1*(29), 155-160.

[19] Farnan, N. L., Lee, A. J., Chrysanthis, P. K., & Yu, T. 2013. Enabling intensional access control via preference-aware query optimization. In *Proceedings of the 18th ACM symposium on Access control models and technologies* (pp. 189-192). ACM.

[20] Ramachandra, K., & Guravannavar, R. 2014. Database-Aware Program Optimization via Static Analysis. *IEEE Data Eng. Bull.*, *37*(1), 60-69.

[21] Cao, W., & Shasha, D. 2013. AppSleuth: a tool for database tuning at the application level. In *Proceedings of the 16th International Conference on Extending Database Technology* (pp. 589-600). ACM.

PLDSRC: A Multi-threaded Compressor/Decompressor for Massive DNA Sequencing Data

Ke Zhan*†, Chao Yang*‡, Changyou Zhang*, Jingjing Zheng*, Ting Wang*,
*Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
†University of Chinese Academy of Sciences, Beijing 100049, China
‡State Key Laboratory of Computer Science, Chinese Academy of Sciences, Beijing 100190, China
{zhanke10,yangchao,changyou,wangting}@iscas.ac.cn, sdzjjing@163.com

***Abstract*—To face the rapid growth of DNA sequencing data, it is of great importance to study high efficiency compression techniques to reduce the cost of storing the massive amount of sequencing data. In this paper, we propose a parallel DNA data compressor/decompressor, PLDSRC, based on the famous serial DSRC software. We first analyze the compression and decompression algorithm in DSRC and identity three basic operations, namely read, work, and write. Then a single pipeline parallel algorithm is proposed to accelerate the compression/decompression procedure. To further exploit today's popular multi-core, multi-socket systems based on the non-uniform memory access (NUMA) architecture, we extend the single pipeline approach to the multi-pipeline case. Experiments on two different platforms are done and show that PLDSRC in both single and multiple pipeline forms is able to speed up DNA sequencing data compression/decompression greatly, while maintaining the same compressing ratio. Examples indicate that the maximum speedup of PLDSRC on compressing and decompressing is respectively around 24.71x and 22.00x, as compared to the serial DSRC software.**

***Keywords*-DNA sequencing compression; DSRC; PLDSRC; Multi-pipeline; NUMA**

I. Introduction

With the rapid development of next-generation sequencing technology, biological experiments generate massive data for DNA sequence reads [12]. Although the cost of magnetic disk has steadily declined over time, it has been left far behind the explosive growth of sequencing data. There is an urgent need to save storage space by employing data compression techniques [4] to exploit information redundancy.

Several storage formats are available to store DNA sequencing data, such as FASTA[11], SAM/BAM [2] [13]. Among them, FASTQ [3] is one of the most widely utilized formats. Originally proposed by the Wellcome Trust Sanger Institute, the FASTQ format is used to bundle a FASTA sequence and its quality data. It has become the *de facto* standard for storing the output of high throughput platforms such as the Illumina Genome Analyzer [10], the SOLiD [15] and Roche 454 [14].

A FASTQ file usually contains four lines for each DNA sequence. An example is show as follows.

```
@ERR000025.19809 BGI-FC30ALEAAXX_8_1_194:393/1
AGCAGCTTAACACAACACACACCAGTCTCAGCCACCATCTCCACA
+
I%%#$%%$)%$$&$$&)%$+$%($$"(#%$"$)$'&$"$"'%%%(
```

In the above example, each record consists four data streams, corresponding to the four lines. The first line, starting with "@", is the identifier of the record. The second line is the raw sequence letters comprised of the four bases "A", "T", "C", and "G". The third line is the identifier indicating the end of the second line, and also the beginning of the fourth line. The fourth line is the quality score, with every value corresponding to a base character in the second line.

Substantially large storage space is usually needed to store data in the FASTQ format. There are several software for compressing/decompressing FASTQ format data. Examples include G-SQZ [17], Quip [6], and DSRC [16]. Among them, G-SQZ employs a classic Huffman algorithm [5] for the compression. Both Quip and DSRC use different methods to process different lines of each DNA record, the methods Quip utilizes are based on more advanced statical models. Overall, DSRC enjoys both comparable compression ratio to Quip and the highest compression speed among the three. Therefore, we select DSRC as the start point in our research.

The compression or decompression of DNA sequencing data may be time consuming. For example, on a typical x86 machine, compressing a FASTQ data of 1PB usually takes more than 9712 hours [16]. Therefore it is necessary to speed up the process of data compression or decompression by taking advantage of the computing capacity of modern parallel computers that may be equipped with mutli-core multi-socket CPUs. The purpose of this research is to parallelize DSRC software by employing multi-threading techniques.

In this work we first identify the serial compressing or decompressing algorithm in DSRC to the combination of three basic operations, namely read, work, and write. Based upon this, a multi-threaded algorithm is presented for DSRC on multi-core platforms. In the algorithm, a read thread and a write thread processes the input and the output files respectively. At the same time there are several work threads in charge of compressing or decompressing the data. Input and output lists are used as buffer between different threads. The algorithm is then extended to a multi-pipeline approach to further take better advantage of the non-uniform memory access (NUMA) nature of multi-sock platforms. Experiments indicate that the parallel algorithm is able to accelerate the process of DNA sequencing compression or decompression by over

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.9

24.71x or 22.00x times, as compared to the serial DSRC software.

II. Algorithms and Implementations

A. Serial DSRC algorithm

DSRC organizes data in superblocks. Each superblock contains 512 data blocks. And each data block is comprised of 32 sequencing records. In DSRC, each superblock is compressed independently. The multiple blocks in a same superblock share the same statistical model in the compressing algorithm. When decompressing, DSRC first locates and computes the addresses of all blocks in each superblock, then decompress the blocks one by one by reading the shared statistical information.

Algorithm 1 Serial compression in DSRC.

```
Input: file.fastq
Output: file.dsrc
 1: clear(buffer)
 2: sb_id=0                                   ▷ superblock id
 3: while !EOF(file.fastq) do
 4:     data ← read(file.fastq, sb_id++)
 5:     while buffer ⇐ compress(data) do
 6:         if size(buffer)==1MB then
 7:             file.dsrc ⇐ write(buffer)
 8:             clear(buffer)
 9:         end if
10:     end while
11: end while
```

The serial compression algorithm in DSRC is shown in Algorithm 1, in which we use "←" to represent the assignment operation and "⇐" to represent the appending operation. In Algorithm 1, the input FASTQ data is processed by a reading operation (line 4) first. After reading one superblock, it begins to compress the FASTQ format data into DSRC format data (line 5), with the output appended to a buffer. During the compressing process, the program checks if the size of the buffer is equal to 1MB (line 6). If true, the compressed data is appended to the output file and the buffer is cleared (line 7-8). If not, the algorithm continues to compress the input data. When one superblock is processed completely, the next one will be processed until reaching the end of the input file.

During the compressing process in line 5, the algorithm appends the compressed result to the buffer in a byte-wise manner and constantly check if the buffer is full. Until every time the buffer is full, the algorithm writes the buffer to the output file and clears the buffer. Therefore, it is possible that when the buffer is full the compression of a superblock is not yet finished. The compressing process and the writing process are intersected with each other in the algorithm.

The decompressing procedure in DSRC is similar to the compressing algorithm.

B. Parallel algorithms

In order to parallelize the serial algorithm, we first separate the compressing and the writing operations by changing the rule of buffer usage. Instead of writing the buffer to the output file every time the buffer is full, we change the writing frequency to the unit of a superblock; i.e., the compressed result in the buffer is appended to the output file and the buffer is cleared up every after compressing a superblock. Then, we may define three types of operations: (1) `read` represents for reading data from the input file; (2) `work` represents for compressing or decompressing data; (3) `write` represents for writing data to the output file. The three operations can be done independently by different threads. Based upon this, a multi-threaded algorithm can be designed by utilizing the idea of pipeline [1] [7].

Shown in Algorithm 2 is the single pipeline multi-threaded algorithm. In the algorithm, there are one thread

Algorithm 2 Parallel compression in PLDSRC.

```
Read thread:
Input: file.fastq
Output: i_list[0:w_num-1]
 1: w_id ← 0;
 2: while !EOF(file.fastq) do
 3:     data ← read(file.fastq, sb_id)
 4:     i_list[w_id++] ← push(data, sb_id++)
 5:     if (w_id ≥ w_num) w_id ← w_id-w_num end if
 6: end while
Work threads:
Input: i_list[0:w_num-1]
Output: o_list[0:w_num-1]
 7: for w_id = 0:w_num-1 do in parallel
 8:     content ← pop(i_list[w_id])
 9:     buffer ⇐ compress(content.data)
10:     o_list[w_id] ← push(buffer, content.sb_id)
11: end for
Write thread:
Input: o_list[0:w_num-1]
Output: file.dsrc
12: for sb_id = 0:sb_num-1 do
13:     for w_id = 0:w_num-1 do
14:         if (o_list[w_id].sb_id == sb_id) break end if
15:         file.dsrc ⇐ write(pop(o_list[w_id]))
16:     end for
17: end for
```

for `read`, one thread for `write`, and several (here, w_num) threads for `work`. Between the three types of threads, input and output lists (i.e., i_list and o_list) are used to ensure the independence of threads. The number of input lists is equal to the number of `work` threads, so is the number of output lists. The length of the input/output list is set to a fixed number. When popping from or pushing into the list (as done with `pop` and `push`), thread lock is set to avoid multi-thread I/O locking issues.

The single pipeline algorithm can be extended to the multiple pipelines case, which is more suitable for multi-socket computers. In the multi-pipeline algorithm, each pipeline processes one section of the input file. The

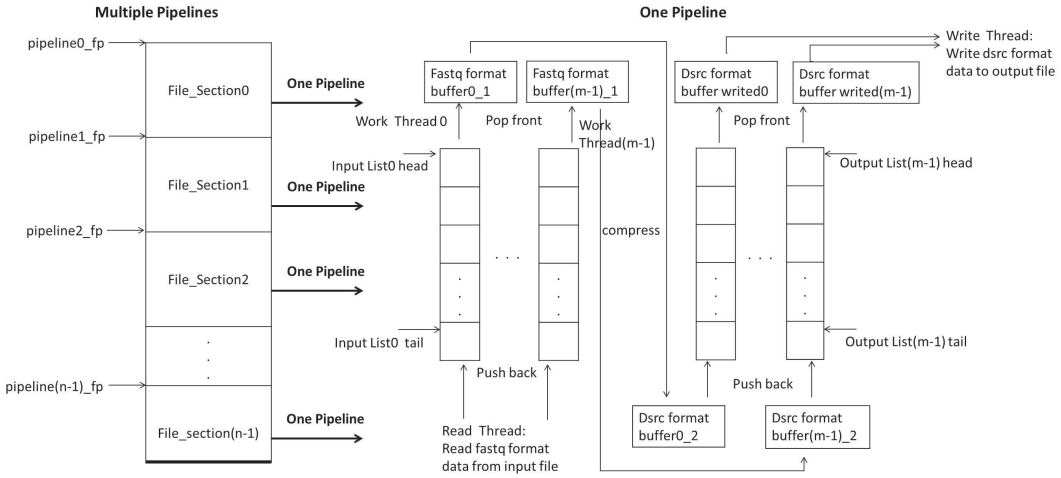


Figure 1. A demonstration of the multi-pipeline compression algorithm in PLDSRC. There are n pipelines, each of which consists of one `read` thread, one `write` thread, and multiple `work` threads. The multiple pipelines are working independently.

multiple pipelines are independent with each other. As shown in the Figure 1, a multi-pipeline compression model is demonstrated. In the figure, the number of pipelines is n, the input file are divided into n sections. File_Section0 represents the section 0 of the input file, File_Section1 represents the section 1 of the input file and so on. Every file section is pointed by a file pointer; for example, pipeline0_fp represents the file pointer that points to File_Section0. There is no relevance between different records, multiple pipelines are used to read data simultaneously. Inside each pipeline, there is one `read` thread, one `write` thread, and multiple `work` threads.

The ideas of the single and multiple pipeline decompression algorithms are similar to the compression versions. And we omit the details for brevity.

To implement the single and multiple pipeline parallel DSRC algorithms for compression and decompression, we employ the POSIX threads (Pthreads, [8]) programing API. Pthreads is flexible to apply on any shared memory CPU platform for establishing and managing threads. In Pthreads, a set of data types, functions, and constants are defined for ease of utilization.

The input list serve as an interface between `read` and `work` threads. There are one `read` thread and several `work` threads inside each pipeline. Every `work` thread and `read` thread share a same input list. Pthreads mutex (mutual exclusion) is used to prevent racing conditions: the `read` thread pushes the data block back to the input list, and at the same time the corresponding `work` thread pops the data block from the input list. Similarly Pthreads mutex is also applied on the output list to avoid data racing between `work` and `write` threads.

On a multi-socket platform with NUMA property, thread affinity may be an important to achieve high performance. For the single-pipeline parallel compression or decompression algorithm, it is expected that the `work`, `read` and `write` threads are binded to cores in a same socket. In this way, the threads in a same socket share the same cache resources, resulting a substantially significant performance boost. However, standard Pthreads APIs do not offer explicit thread binding mechanism. Therefore, we use a non-standard subroutine, pthread_setaffinity_np to bind threads to proper physical cores. For the multi-pipeline case, we alway try to bind all threads in each pipeline to CPU cores that belong to a same socket.

III. Performance Results and Analysis

A. Experimental environment

Two platforms are employed for the performance tests. The first one, referred to as "Machine I" in the sequel, is an Intel Xeon E5-2650 dual-socket×8-core platform with with 32GB local memory. The second one, referred to as "Machine II" in the sequel, is an Intel Xeon fat-node equipped with eight 8-core X7550 2.0GHz CPUs and 512GB local memory. Although the CPU frequencies of the two machines are the same, the second platform has four times more sockets and 16 times more memory capacity. In addition, the second platform supports hyper-threading automatically, which means that a maximum of 128 threads may be used to fully exploit the computing power of the whole system. Two FASTQ format DNA sequencing data files are used for the tests in our work, namely SRR013.fastq and SRR741.fastq.

B. Results and analysis

We first investigate the compressing ratio of PLDSRC compared to DSRC. Except the two test files listed previously, we have also tried many others. Test results show that the compression ratio of PLDSRC is exactly the same as DSRC. Because DSRC compresses each superblock independently, and we divide the superblocks in the same way DSRC does, there is in fact no loss of compression ratio when the parallel PLDSRC algorithm is employed.

We then examine the performance of the single pipeline PLDSRC algorithm. Machine I is used in the test, with the number of `work` threads varying from 1 to 14.

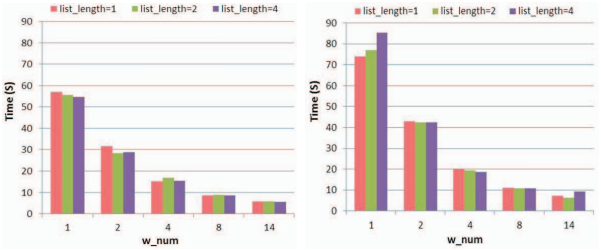


Figure 2. Time in seconds for compressing (left panel) and decompressing (right panel) file SRR013.fastq on Machine I, with respect to different numbers of `work` threads and different lengths of the input/output lists.

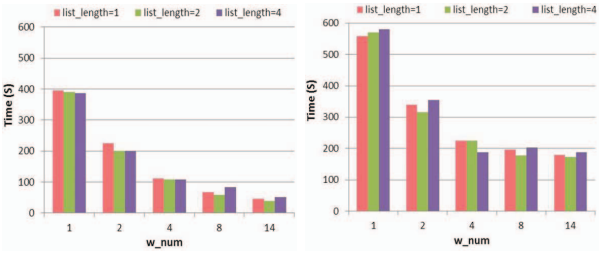


Figure 3. Time in seconds for compressing (left panel) and decompressing (right panel) file SRR741.fastq on Machine I, with respect to different numbers of `work` threads and different lengths of the input/output lists.

Three different lengths of the input/output list are utilized. The performance results for compressing and decompressing SRR013.fastq are shown in Figure 2, and that for SRR741.fastq are presented in Figure 3. It is evident from both figures that as the number of `work` threads increases, the compressing/decompressing time for both files decreases correspondingly. Considering the large amount of I/O throughput, the overall parallel performance of the single pipeline algorithm is satisfactory.

From Figure 2 and 3, there shows no obvious effect to improve the compression/decompression performance by changing the length of the input/output list. We analyze that the above result indicates that processing a data block by a `read` or `write` thread is much less costly than processing the same data block by a `work` thread; i.e., the computation done by the `work` thread is the dominant factor in the DSRC algorithm. To verify our analysis, we examine the measured time consumed by different types of threads in both the compressing and decompressing algorithms; the results are shown in Table I. It is evident

Table I
TIME COSTS PER DATA BLOCK FOR DIFFERENT TYPES OF THREADS.

	Comp.	Time (us)	Decomp.	Time (us)
Machine I	Read	582	Read	285
	Work	122484	Work	148511
	Write	968	Write	3366
Machine II	Read	1881	Read	489
	Work	128337	Work	140254
	Write	2473	Write	4328

to see that the cost processing data by the `work` thread is magnitudes larger than others. Based on the above facts, we fix the input/output list length to be 2 in all the other tests in this work. The reason of using 2 instead of 1 is to avoid potential I/O conflicts.

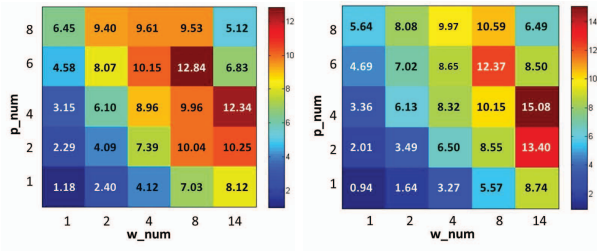


Figure 4. Speedup of compressing (left panel) and decompressing (right panel) file SRR013.fastq on Machine II, with respect to different numbers of pipelines and different numbers of `work` threads per pipeline.

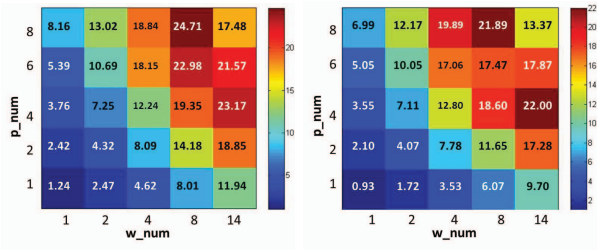


Figure 5. Speedup of compressing (left panel) and decompressing (right panel) file SRR741.fastq on Machine II, with respect to different numbers of pipelines and different numbers of `work` threads per pipeline.

Finally we carry out tests on the performance of compressing/decompressing files on Machine II, where much more threads can be utilized. Both data files are used in the tests. We examine the parallel speedup as both the total number of pipelines (p_num) and the number of `work` threads (w_num) per pipeline change. Figure 4 shows the speedup of compressing and decompressing file SRR013.fastq on Machine II. From the figure we observe that when the total number of pipelines is fixed, in most cases the speedup increases as more `work` threads are used. For compression, the highest speedup is obtained when p_num=6 and w_num=8. While for decompression, p_num=4 and w_num=16 leads to the highest speedup. Results on the speedup of compressing and decompressing file SRR741.fastq on the same machine is presented in Figure 5, which shows similar results to file SRR013.fastq. For a given number of pipelines, in most cases the speedup increases as we use more `work` threads. For compression, we obtain the highest speedup when p_num=8 and w_num=8. While for decompression, the highest speedup is achieved when p_num=4 and w_num=14. No matter in which case, the multi-pipeline approach is superior to the single pipeline case, and the optimal speedup is obtained in the case that the total number of threads is around the total number of physical core (64, for Machine II).

We end this section by presenting a comparative result on the highest speed of both DSRC and PLDSRC on processing the two files on the two machines. The result is shown in Table II. From the table we conclude that PLDSRC is able to accelerate the serial DSRC algorithm by taking better advantage of today's multi-core, multi-socket systems.

Table II
PERFORMANCE RESULTS

	Compression/Decompression	Machine I	Machine II
DSRC	SRR013: Compression	59.40	96.53
	SRR741: Compression	576.90	697.14
	SRR013: Decompression	93.98	90.46
	SRR741: Decompression	530.30	559.13
PLDSRC	SRR013: Compression	5.59	7.52
	SRR741: Compression	38.60	28.21
	SRR013: Decompression	6.30	6.00
	SRR741: Decompression	150.49	25.42

IV. CONCLUSION

Today's DNA sequencing technology generates massive amount of DNA data. There is an urgent need to compress these data in a fast and efficient way. In this paper, we select DSRC, which is one of the most famous DNA sequencing data compressor/decompressor as the base of our research. We make an attempt to speed up DSRC by using multi-thread parallelism and propose a parallel compressor/decompressor, PLDSRC. We first analyze the compressing and decompressing algorithm in DSRC and identity three basic operations, namely read, work, and write. Based upon this, both single and multiple pipeline parallel algorithms are proposed. In the single pipeline approach, a read thread and a write thread processes the input and the output files respectively. At the same time there are several work threads in charge of compressing or decompressing the data. Input and output lists are used as buffer between different threads. The single pipeline approach is then extended to multi-pipeline case by dividing the input file into sections and letting each pipeline working on each section simultaneously. The multi-pipeline approach is more suitable for today's popular multi-core, multi-sock platforms based on the non-uniform memory access (NUMA) architecture. We show by experiments on two different platforms that PLDSRC in both single and multiple pipeline forms is able to accelerate DNA sequencing data compression/decompression on multi-core, multi-socket systems, while maintaining the same compressing ratio. Examples indicate that the maximum speedup of PLDSRC on compressing and decompressing is respectively around 24.71x and 22.00x, as compared to the serial DSRC software.

To further exploit today's emerging multi- and many-core processors, we plan to extend our research to general purpose graphic processing units (GPGPU) and the Intel Xeon Phi coprocessor based on the many integrated cores (MIC) architecture.

ACKNOWLEDGMENT

The work was supported in part by NSF China (grants 61170075 and 91130023) and 973 Program of China (grant 2011CB309701).

REFERENCES

[1] A. Navarro, R. Asenjo, S. Tabik, and C. Cascaval, "Analytical modeling of pipeline parallelism", in *Proc. 18th Int'l Conf. on Parallel Architectures and Compilation Techniques (PACT'09)*. IEEE Computer Society, 2009, pp. 281–290.

[2] The SAM/BAM Format Specification Working Group, "Sequence alignment/map format specification". 2011.

[3] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants". *Nucleic Acids Res*. 38:6, 2010, pp. 1767–1771.

[4] D. Salomon, Data Compression: the Complete Reference. 3rd edition. Springer-Verlag, New York, 2004.

[5] D. A. Huffman, "A method for the construction of minimum-redundancy codes", *Proceedings of the IRE*, 40:9, 1952, pp. 1098-1101.

[6] D. C. Jones, W. L. Ruzzo, X. Peng, and M. G. Katze, "Compression of next-generation sequencing reads aided by highly efficient de novo assembly". *Nucleic Acids Res*. 40:22, 2012, p. e171.

[7] D. Sanchez, D. Lo, R. M. Yoo, J. Sugerman, and C. Kozyrakis, "Dynamic fine-grain scheduling of pipeline parallelism", *Proc. 22nd Int'l Conf. on Parallel Architectures and Compilation Techniques (PACT'11)*. IEEE Computer Society, 2011, pp. 22–32.

[8] D. R. Butenhof, Programming with POSIX Threads. Addison-Wesley Educational Publishers Inc. 1997.

[9] http://trace.ddbj.nig.ac.jp/DRASearch/centerList

[10] M. Kircher, U. Stenzel, and J. Kelso, "Improved base calling for the Illumina Genome Analyzer using machine learning strategies", *Genome Biology*, 10:8, 2009, p. R83.

[11] NCBI Learning Center: `http://www.ncbi.nlm.nih.gov/staff/tao/tools/tool_lettercode.html`, Single Letter Codes for Nucleotides. National Center for Biotechnology Information. Retrieved 2012-03-15.

[12] Y. Kodama, M. Shumway, R. Leinonen, and International Nucleotide Sequence Database Collaboration, "The sequence read archive: explosive growth of sequencing data". *Nucleic Acids Res*. 40 (Database issue), 2012, pp. D54-6.

[13] H. Li, B. Handsaker, A. Wysoker, T. Fennell, et. al, "The Sequence Alignment/Map (SAM) Format and SAMtools". *Bioinformatics*, 25:16, 2009, pp. 2078–9.

[14] M. Margulies, M. Egholm, W. E. Altman, S. Attiya, et. al, "Genome sequencing in microfabricated high-density picolitre reactors". *Nature*, 437:7057, 2005, pp. 376-380.

[15] V. Pandey, R. C. Nutter, and E. Prediger, "Applied Biosystems SOLiD System: Ligation-Based Sequencing". *Next Generation Genome Sequencing*, Wiley-VCH Verlag GmbH & Co. KGaA, 2008, pp. 29–42.

[16] S. Deorowicz, and S. Grabowski, "Compression of DNA sequence reads in FASTQ format". *Bioinformatics*. 27:6, 2011, pp. 860–2.

[17] W. Tembe, J. Lowey, and E. Suh, "G-SQZ: compact encoding of genomic sequence and quality data". *Bioinformatics*. 26:17, 2010, pp. 2192–4.

Optimized data I/O strategy of the algorithm of parallel digital terrain analysis

Yan Li
Nanjing Normal University
Nanjing, China
467094983@qq.com

Wanfeng Dou
Nanjing Normal University
The information security technology and engineer research center of Jiangsu province
Nanjing, China
douwanfeng@njnu.edu.cn

Kun Yang
Nanjing Normal University
Nanjing, China
757389791@qq.com

Shoushuai Miao
Nanjing Normal University
Nanjing, China
871641021@qq.com

***Abstract*—In order to improve the analysis performance of the algorithm of parallel digital terrain analysis, a series of parallel tasks should be proceed by the appropriate partition and scheduling strategy, but the algorithm of data I/O strategy can also be optimized. Based on the conventional data I/O model, the DEM data I/O strategy of the parallel algorithm for digital terrain can be optimized by the parallel pipeline method. Then it can further enhance the efficiency of the parallel algorithm of digital terrain analysis. Take the maximum slope algorithm for example, it was conducted many experiments in the small-scale cluster environment. The results show that the parallel algorithm based on the optimized data I/O strategy can improve the analysis efficiency of the previously conventional algorithm.**

Keywords- the parallel pipeline; the asynchronous data I/O;the parallel computation; the digital terrain analysis;

I. INTRODUCTION

With the rapid development of geography observation and acquisition technology, the spatial data can be obtained more and more efficient, and the data quantity is also fleetly increasing [1]. So the digital terrain analysis technology will be faced with a problem which is that the calculation of spatial data and geographic problems is more complicated and diverse than before. This current situation will become a challenge to some existing digital terrain analysis algorithms of single PC and technology. Then the contradiction between the utilization of computing source and the required analysis efficiency of huge amounts data tends to deepen. Many researchers have used the high performance computing to resolve the mismatch problem between the load of calculation and the computing power of the digital terrain analysis. Then they put forward the concept of the parallel digital terrain analysis [2].

At present, many optimized methods of the existing algorithm of parallel digital terrain analysis mainly concentrated on the division method of DEM data and the scheduling method of many executive tasks in the algorithm when they are executed in the cluster environment. In the cluster environment, because of CPU's competitive relation between access and storage and the limitation of the synchronization lock, simply increasing the number of processor and memory capacity is not necessarily obtain the corresponding advance of the speed-up ratio. So in the field of the parallel computation, the parallel pipeline technology is widely applied, such as Burton Smith has designed the Tera's level shared memory super computer[3] to achieve a special parallel pipeline algorithm, then makes all processors can access all data of the storage system at the same time. In the executing procedure of the parallel digital terrain analysis algorithm, due to the limit of the intensive algorithm`s high access and the complexity of the algorithm, based on the parallel I/O [4,5,6] of assembly line technology, it can lessen the effect, which is due to the congestion of network and the disk I/O of computers, to the performance of parallel algorithms. Therefore, in order to solve the problem of data reading and writing latency in some existing parallel digital terrain analysis algorithms, we can make some schedule and optimization to the parallel data I/O. Through the new data I/O method based on the parallel pipeline method, the data reading and writing can be effectively organized and controlled.

In this paper, by analyzing the input/output and the communication efficiency between each node of the cluster environment, as well as by combining with I/O characteristics in the parallel digital terrain analysis algorithm, we can put forward an optimized data I/O strategy to the data I/O process in the algorithm of parallel digital terrain analysis in cluster environment. Then the strategy realizes the optimized management of the reading, sending, recycling and writing process in the algorithm of parallel digital terrain analysis in the cluster environment.

II. THE OPTIMIZED DEM DATA I/O STRATEGY IN A CLUSTER ENVIRONMENT

At present many DEM data reading and writing methods of the algorithm of parallel digital terrain analysis in a cluster environment mainly include two kinds. One kind is to set up the master node to read all DEM data, then the DEM data are distributed, by some data partitioning and task scheduling methods, to a certain number of child nodes to process and the child node returns the results to the master node. Then the master node sort and output the results. Another kind is that each node in cluster can read and write DEM data from the file server to process through the Network File System[7,8,9](NFS) . In this paper, we introduce the technique of the parallel pipeline to the first kind. Through the improvement of the first method we can improve the

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.10

data reading and writing efficiency of the parallel digital analysis algorithm, and then raise the speed-up ratio of the algorithm.

In this paper, the basic principle of the parallel pipeline is to set the time-consuming operations, like data reading, data writing, data communication and data synchronization lock, in the local data of independent operation. When performing these operations, the CPU needs to wait for and the CPU judges whether the later program block have strong dependency with the current obstructed program block. If there is no strong dependency between them, CPU can tend to perform the later program in advance. DEM data reading and writing process of the algorithm of parallel digital terrain analysis can realize "process while reading" after this introduction of the parallel pipeline technology. As shown in figure 1 a), the main node needs to read all of DEM data. Then it distributes data blocks which are divided by the data partitioning strategy to the sub-process. So when the main node is reading, it can not process some other operations and the sub-process must wait to the data block. But in figure 1 b) we can see that the main node can start to read and write data without the need for blocking that all of the DEM data is finished reading while it is processing some operations, like distributing data blocks and so on. Comparing a) with b), we can see that the time consumption of b) is less than a) due to eliminate the backup between the reading operation and the distributing operation of the master node.

As shown in figure 1, we assume that the amount of DEM data is G, the number of processes is n, the reading speed of the node in the cluster is V_R, the writing speed is V_W, the processing speed is V_P and the communication speed is V_C. In the workstation cluster, the master node is used for reading, writing and distributing DEM data, the time spent on reading all of the DEM data is $\frac{G}{V_R}$ in the master node and it's time to write results is $\frac{G}{V_W}$. By the number of process, the amount of data which is distributed to process by the master node is $\left\lfloor \frac{G}{n} \right\rfloor$. So the communication time between master node and each sub-process is $\frac{G}{nV_C}$, as while the time spent on returning the results to the master node by sub-process is $\frac{G}{nV_C}$. Assume that the calculation load of every node is equal, so the computation time of each sub-process is $\frac{G}{nV_P}$.

According to the previous process execution sequence diagram which is not improved, the time consumption of the entire process can be calculated as follows:

$$T_1 = \frac{G}{V_R} + (1+2+3+\ldots+n-1)\frac{G}{nV_C} + \frac{2G}{nV_C} + \frac{G}{nV_P} + \frac{G}{V_W} \quad (1)$$

$$= \frac{G}{V_R} + \frac{(n-1)G}{2V_C} + \frac{2G}{nV_C} + \frac{G}{nV_P} + \frac{G}{V_W}$$

$$= G\left(\frac{1}{V_R} + \frac{(n-1)}{2V_C} + \frac{2}{nV_C} + \frac{1}{nV_P} + \frac{1}{V_W}\right)$$

In order to achieve DEM data "process while reading" strategy of the master node, the master node need to use the thread -level parallel technology. So "process while reading" DEM data I/O algorithm can be seen like that:

Step1: The master node start the same number of threads with the number of data partition blocks

Step2: The master node limit that the reading operation of data block can only be performed by a single thread at the same time. So the number n thread can read the number n data block after the number n-1 thread read the number n-1 data block.

Step3: If the number n thread has read the number n data block, it can send the data to the corresponding sub-process. Then it needs to wait for the result returned from the sub-process.

Step4: The sub-process receives the data block and processes it.

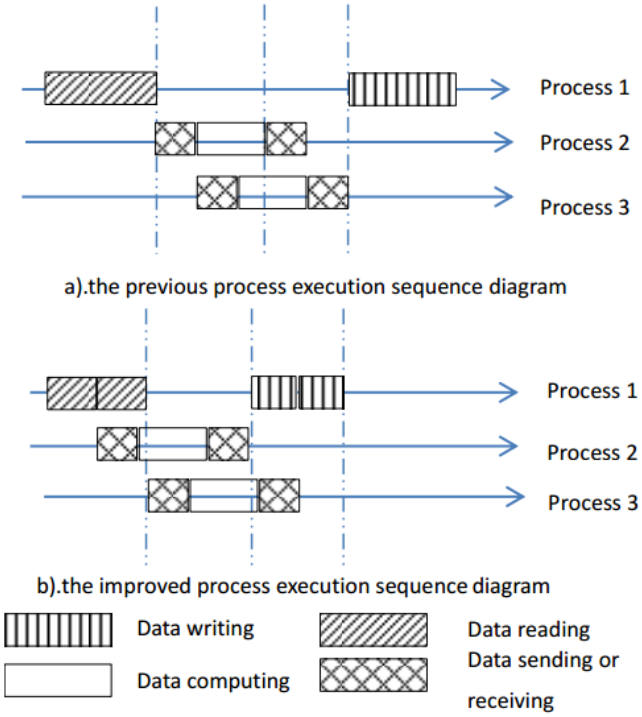


Figure 1 conventional I/O and asynchronous I/O

Step5: The sub-process returns processing results to the corresponding thread.

Step6: Because the thread of the master node is waiting for the result returned from the sub-process, it can receive the result and write it to the DEM file in time. the master node output the DEM file.

Because both threads and child nodes are in receiving or sending state, this avoids the receiving or sending jam. So each thread and child node forms to the parallel pipeline. This strategy improves the efficiency of data transmission and enhances the speed-up ratio of the parallel digital terrain analysis algorithm. The specific process is shown in figure 2.Compared to the previous (the conventional data I/O) process execution sequence diagram which is not improved in figure 1a), from the improved (process while reading) process execution

sequence diagram in figure 1b) the time consumption of entire process execution can be calculated for:

$$T_2 = \frac{G}{nV_R} + \frac{2G}{nV_C} + \frac{G}{nV_P} + n \cdot \frac{G}{nV_W}$$
$$= G(\frac{1}{nV_R} + \frac{2}{nV_C} + \frac{1}{nV_P} + \frac{1}{V_W}) \quad (2)$$

Thus the time which can be saved can be calculated like:

$$\Delta T = T_1 - T_2 = (n-1)G \cdot (\frac{1}{nV_R} + \frac{1}{2V_C}) \quad (3)$$

By the formula (1), (2) and (3) can be seen, ΔT is greater than 0 , so the strategy using the parallel pipelining mode to realize asynchronous I/O can improve the computation efficiency of parallel computing and save the time of the data reading and the results writing. In particular when the number of process is large (n increases), data reading and writing time can be almost implied in the computation time. The limitation of data I/O time consumption on the calculation efficiency is gradually weakening in the parallel processing function. Meanwhile due to the efficiency will be greatly reduced if there are many random, small or high frequency read operations in the common mechanical disk, while designing the asynchronous I/O, we should especially consider on this restrictive factor.

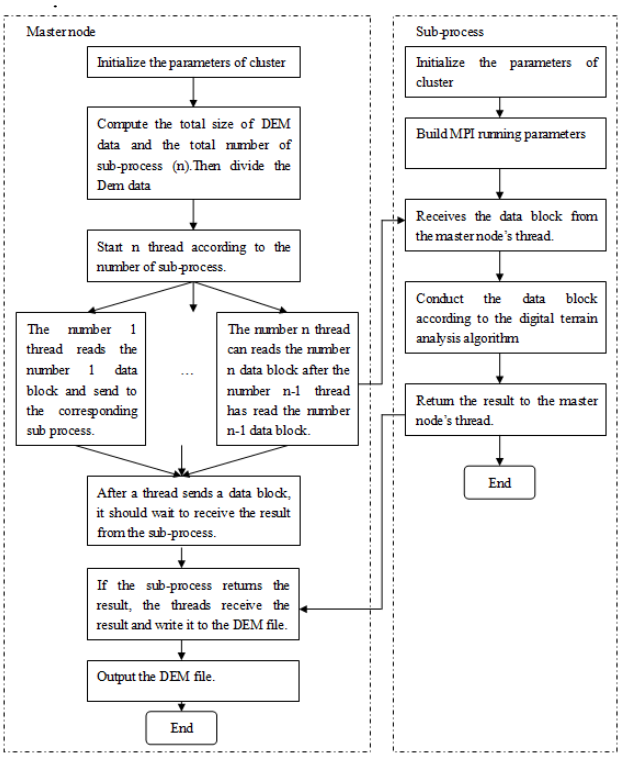


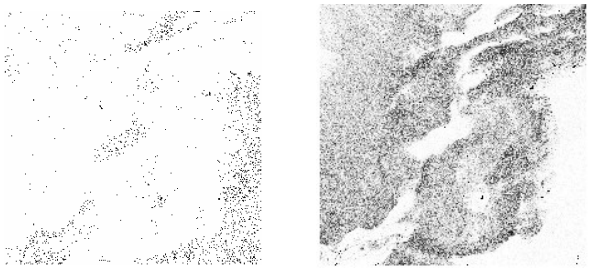
Figure 2 the flow chart of the improved DEM data I/O process

III. EXPERIMENT AND ANALYSIS

In many parallel digital terrain algorithms, they often lack the parallel processing of the data reading/writing and data sending/recycling, this is because that the data reading/writing and sending/recycling process involved in the complex hardware environment and it is very difficult to do strictly synchronous processing on the highly real-time requirements parallel computing environment. In order to adapt to the hardware condition of the cluster environment, in the small cluster environment this experiment design an asynchronous I/O. When the master node reads or writes the DEM data, it can open a new thread to perform the reading and writing operation.

A. The experimental data and computing environment

The Cluster environment is 16 computers in a Cluster structure and use the gigabit Ethernet to connect each node. The processor of each computer is Intel (R) Xeon (R) CPU E5645 @ 2.40 GHz. Its memory is 24 GB DDR3. The operating system of Cluster is centos 5. The software environment of the experiment is GDAL 1.6.1, openmpi 1.5.4 and GCC 4.4.7. The size of experimental data is the 6001 horizontal size , 6001 vertical size DEM data, as shown in figure 3 a). The experiment uses the Maximum Slope algorithm to compute the slope of this DEM data. Using the above data, the experiment introduces the optimized data I/O strategy to the parallel Maximum Slope algorithm. Data blocks will be processed by the asynchronous I/O processing strategy which are divided based on the actual algorithm of the data partitioning strategy. Its experimental results are shown in figure 3 b).



a).original data b).result data

Figure 3 the experimental data and calculated results

B. The analysis of the data I/O algorithm performance

First of all, we do not introduce the improved data I/O algorithm into the parallel digital terrain analysis algorithm. After that we only design a "distribute and reading" algorithm, and test whether the optimized data I/O strategy can accelerate the data reading and writing speed in the cluster environment. Then we can analyze whether the improved data reading and writing strategy can be used in the subsequent algorithm of parallel digital terrain analysis and realize "process while reading" operation. Table 1 is the experiment processing time of "distribute and reading" algorithm in different number of processes while not being involved in the parallel digital terrain analysis algorithm.

Table 1 THE TIME OF "DISTRIBUTE AND READING" EXPERIMENT

The number of processes	Conventional I/O time (s)	" distribute and reading "time(s)
5	0.7308	0.6453
7	0.8090	0.6456
9	0.8145	0.6538
11	0.7755	0.6725
13	0.8294	0.6818

From table 1, it can be summarized that the time consumption of the "distribute and reading" algorithm is shorter than the consumption of conventional data I/O strategy. This will provide a basis principle for the subsequent application, such as the "process and reading" data I/O strategy in the parallel digital terrain analysis algorithm. Although with the increase of the number of processes the time consumption is increasing both the conventional I/O algorithm and the "distribute and reading" algorithm, this is because the time consumption of experiment in table 1 is mainly on the communication consumption between different nodes.

Then we use the "process while reading" algorithm to change data I/O mode of the parallel Maximum Slope algorithm, the time consumption of the experiment can be shown in figure 4. According to the chart, when the number of processes increases to a certain number, the main factor, restricting promoting the parallel speed-up ratio, is the data I/O time consumption and the time consumption of communication between different nodes. In the experiment by optimizing data I/O strategy, we can reduce the CPU waiting time in the entire parallel process and improve the parallel efficiency. So in the figure 4 the time consumption of the Maximum Slope algorithm improved by the "process while reading" algorithm is shorter than the conventional Maximum Slope algorithm.

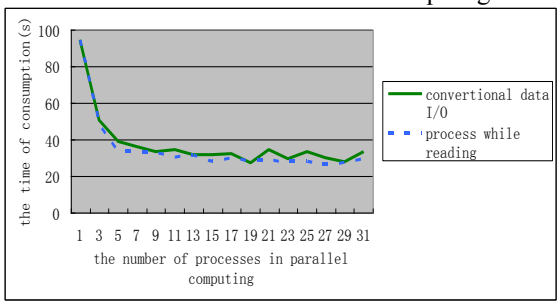


Figure 4 The calculation time of the parallel algorithm

As well as, the speed-up ratio comparison is shown in figure 5. So the "process while reading" algorithm in view of the pipeline technology put by this article can effectively improve the parallel speed-up ratio. Through the "process while reading" of data can help to get a better speed-up result.

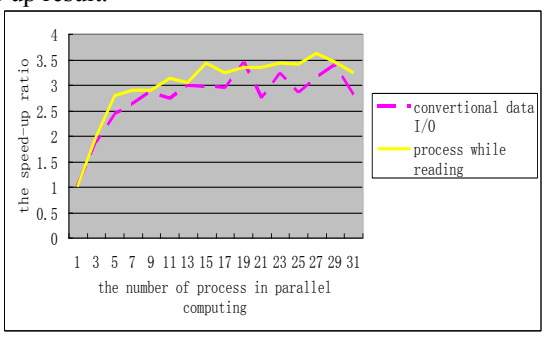


Figure 5 the speed-up ratio of parallel algorithm

The bottleneck of parallel time consumption is the reading, sending and writing time consumption under the influence of some low speed devices such as disk and network. When the number of processes increases to a certain number, the time consumption of parallel computing does not continue to fall, but presents a balance state. At the same time, the difference in height between two curves shows that time consumption due to CPU waiting. This suggests that the optimization of data I/O strategy has very important significance in parallel digital terrain analysis. So the "process while reading" data I/O strategy which is put forward in this paper is necessary.

IV. CONCLUSION

The conventional parallel data reading and writing strategy is optimized in digital terrain analysis algorithm to further enhance the operational efficiency of the parallel digital terrain analysis algorithm. This paper has used the "process while reading" algorithm to change data I/O mode of the parallel Maximum Slope algorithm. Then we try to improve the algorithm to verify this optimization data I/O strategy. The result of the experiment show that the "process while reading" optimized data I/O strategy can further enhance the operational efficiency of the parallel digital terrain analysis algorithm. In this article the next step of work is experimented on the different size of DEM data and the different digital terrain analysis algorithm to further study the application of "process while reading" optimized data I/O strategy in the parallel digital terrain analysis algorithm.

V. ACKNOWLEDGE

This work has been substantially supported by the National Natural Science Foundation of China (NO. 41171298).

REFERENCES

[1] Lixin Wu, Yizhou Yang etc. The based parallel algorithms research facing the new hardware structure of a new generation of GIS [J]. Geography and geo-information science, 2013 (29) : 1-8.

[2] Jing Zhao . The research based on the parallel granularity model of digital terrain analysis and fault-tolerant scheduling [D]. The master's thesis of nanjing normal university, 2012.

[3] Alverson R, Callahan D, et al. The Tera's computer system [J]. ACM SIGARCH Computer Architecture News,1990,18(3b):1-6.

[4] Yanying Wang, Yan Ma etc. The optimized parallel Mosaic algorithm based on dynamic grouping strategy and multi-thread parallel IO [J]. Journal of remote sensing information. 2012 (2) : 3-8.

[5] Haoyu Peng , Zhefan Jin etc. The application of the double parallel lines based on PC cluster machine [J]. Journal of Computer-Aided Design & Computer Graphics. 2006 (18) : 1581-1586.

[6] Wei Pan, Liaoyuan Chen etc. The research of MPI + OpenMP hybrid programming model based on SMP cluster [J]. Journal of Application Research of Computer. 2009 (12) : 4592-4594.

[7] Martin R, Culler d. NFS Sensitivity to High Performance Networks [C]. Proc. Of SIGMETRICS '99. 1999-05.

[8] Lever C, Honeyman p. Linux NFS Client Write Performance [C]. Proceedings of the Usenix Technical Conference on FREENIX Track, Monterey. The 2001-06.

[9] Liqiang Cao, Hongbing Luo. Cluster environment that influences the bandwidth of the NFS file system test and analysis [J]. Journal of Computer engineering, 2007 (19) : 125-127.

A Task Assignment Method For Phi Structure

Yunchun Li
Beijing Key laboratory of network technology
Computer Science Department, Beihang University
Beijing, China
lych@buaa.edu.cn

Tianyu Zhang
JSI, Computer Science Department
Beihang University
Beijing, China
zhangtianyugt@foxmail.com

***Abstract*—Xeon Phi is a high performance co-processor launched by Intel in 2012. Though Phi is specifically designed for Exascale super computer, the task assignment for Phi is yet to be studied. Based on the special needs of task assignment for Phi, this paper presents an algorithm evolved from graph bisection algorithm: a graph is formed based on the memory dependence of tasks, by traversal the graph with an assuming cut point, iteratively finds out the groups of tasks with least dependence on tasks outside group, this algorithm can provide a task assignment solution between CPU and Phi aiming at memory optimization. Experiment reveals that this algorithm can significantly reduce the total memory usage of the job, also increase the efficiency of the execution. By reducing the memory usage, this algorithm can eliminates the memory bottleneck of Phi and expand the using range of Phi.**

Task Assignment; Phi; Memory Usage; CG Algorithm

I. INTRODUCTION

Xeon Phi is a x86 structure co-processor brought out by Intel, oriented towards HPC domain. By launching this, Intel tries to lead the industry into Exascale area[1]. Phi co-processor consists of around 60 (differs from products) x86 cores, each core supports four hardware process with 32 512bits width vector processing unit. Phi comes from P54C structure, which is a simple, short pipeline, sequential execution structure[1]. The simplicity of Phi causes great efficiency lose once the pipeline pauses, hence single core sequential performance is not as good, but lower both the complexity of the structure and power consumption. This makes Phi core a good build block for the MIC product. Phi can dramatically improve the computational capability via massive parallelization. All cores on Phi share one 8GB main memory. Each core has two private cache, 64kb L1 cache among which 32kb instrument cache and 32kb data cache; 512kb L2 cache. Cores communicate through the core ring interface, which connects core, memory controller and PCI-E interface.

There're three program models for Phi, those are: CPU center model, CPU initiates main function, Phi do the auxiliary calculation; MIC center model, Phi can do all the computation or use CPU as secondary processor; Symmetric model, both CPU and Phi cores are treated equally, such as MPI. Among above models, CPU center model is the most convenient and commonly used. Using Phi only is the easiest model to porting existing programs to Phi, but can't fully utilize all computation power; symmetric model can be optimized and modeled as traditional cluster programs.

As discussed above, task assignment for Phi has its own characteristics. First, tasks only need to be divided into two portions, one on CPU, the other on Phi, the total computation cost is smaller; second, due to the lack of memory of Phi, all tasks on Phi can only use less than 8GB memory. The memory consumption must be taken into consideration while assigning tasks, otherwise the tasks on Phi may exceed the memory capacity; finally, the granularity of task is remarkably finer than existing general purposed task assignment. Hence, with the total calculation amount gone down, limitation increased, general purposed task assignment algorithm can't suit Phi perfectly.

While designing task assignment algorithm for Phi, the following matters should be considered: the memory consumption of task portion, the transfer overhead, etc. Most existing algorithm fails to take memory or other resource limitation into consideration, in some circumstances the tasks can not be run due to resource limit.

This paper is organized as follow: section two introduces related works, section three introduces our novel approach of task assignment aiming at reduce memory usage, section four validates the effects of this algorithm by porting and optimizing CG algorithm, last section summarizes this paper.

II. RELATED WORKS

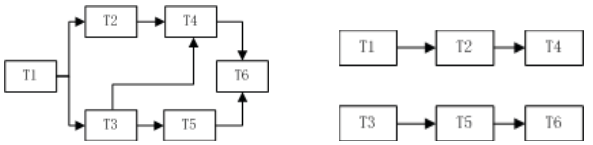
The commonly used definition of task assignment are as follow:

- A set of N tasks, , is the ith task;
- A set of M processors, , is the ith processor;
- An matrix C[m,m], C[i,j] is the delay of passing data from to ;
- An matrix Et[m,m], Et[i,j] is the estimated time consumption when is running on ;
- A task dependence graph uses a DAG to represent the task relationship.

The aim of the algorithm is to find a strategy of task assignment, which assigns each task to each processor, decides the order of tasks execution, trying to minimize the job time consumption under the condition of task dependence graph. This is considered an NPC problem, can't be solved in polynomial time, a full traversal of the entire solution space is needed to get an optimal solution. The model is shown in Figure 1(a).

The solution for task assignment is shown in Figure 1. (b). To get this solution, existing research on task assignment are: reference [8] uses iterative graph bisection algorithm, initiates two sets using two most separated nodes. Separates other nodes based on the distance between node

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.11



(a)task dependence graph (b)task assignment graph

Figure 1. Task dependence and assignment graph.

and set. Reference [9] uses greedy algorithm, first marks the node with the smallest degree, using it as a center of a set. After that iterates on the rest of the graph, each time add the node nearest to center node into the center set, when a size of a set achieved the preset threshold, a set of tasks for a processor is produced. Reference [10][11][12] uses K-L algorithm, in a load balanced bisection graph, calculates the benefit of moving a node, and applies the most benefiting move until the benefit becomes negative. Reference [13] uses A* algorithm to find the best task algorithm. Reference [7] parallels heuristic algorithm to speed up the calculation. There are newer researches using random algorithms, reference [3][4] uses harmony search, reference [5][6] uses genetic algorithms.

We can conclude from above, the researches generally divide into two categories: one is based on classical graph algorithms, specifically designed for the purpose of task assignment; the other is all sorts of random or parallel algorithms aiming to deal with the NPC characteristic of task assignment. All focus on reducing the total time consumption, and bypassing the complexity of the task assignment algorithm itself.

III. MEMORY USAGE BASED TASK ASSIGNMENT

This paper bring out an algorithm called MUBTA(Memory Usage Based Task Assignment) which mainly focuses on the memory usage rather than the time usage. MUBTA uses a similar yet different description of the problem. This paper ignores the information related with time to reduce the complexity of the algorithm. In this paper, a node represents a data in the program, the dependency of data will be represented by arcs. Task assignment can be seen as a bisection of the graph described above. The formal definition is as follow:

- Memory dependence graph is a weighted DAG G, G={n, e, nw, ew}.
- n_i is the ith node in the graph, each represents a data during calculation, which is a portion of memory used.
- nw_i is the weight of n_i, representing the cost storing and transferring data , this paper uses the size of data in memory.
- e_{ij} is an edge from n_i to n_j, representing the calculation of n_i needs n_j, meaning there is a formula like $n_i = f(...,n_j,...)$ during calculation.
- ew_{ij} is the weight of edge e_{ij}, which is when the calculation n_i and n_j is on different device, the transfer cost of n_j, here $ew_{ij} = nw_j$.

Using this model, the memory dependency graph from experiment in section IV. is shown as Figure 2.

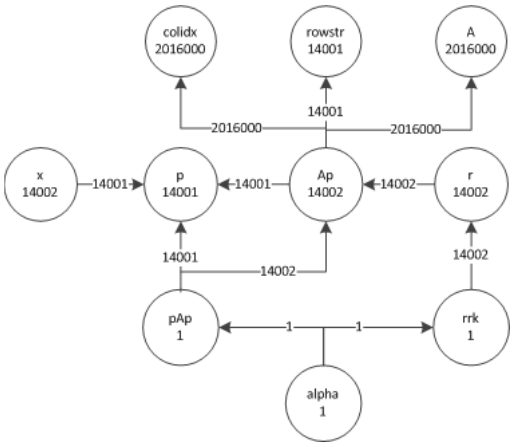


Figure 2. Memory dependence graph in following experiment.

This algorithm does not pay much attention to the order of tasks, dependency is not about time. Also the result of this algorithm won't be able to give users suggestions on tasks' order, it can only provide two portions of tasks for CPU and Phi. Balancing between time and memory consumption.

The task assignment is essentially bisecting a graph with the least total weight of edges between the two subgraphs. This is similar to the algorithm of finding cut points of a undirected connected graph, which is a node in a graph, once is deleted the graph will be no longer connected. But the graph in task assignment is a DAG, the algorithm has to be modified to fit this.

This algorithm is an iterative algorithm, each iteration selects a node as a centroid, and finds the task group with least transfer cost with the other group using it, the group found can be seen as a new task, and shrink into a node, then one iteration is done.

One iteration is as follow: in graph G, initialize a set A = {} using an assumed centroid, traverse on the edge set of G, if all the head nodes of the edges ends with , add into A, A = A {}, after one iteration A is a disconnected subgraph when a is the cut point. Iterates until the A does not change or exceed the memory limitation. This algorithm is similar to the greedy algorithm in [9], but the aim is to make the coupling inside a group higher than tasks cross groups. This can provide a good group to be a candidate for a shrink point.

The total weight of nodes inside A is the memory usage, the total weight of edges across groups is the transfer cost. The pseudo code of algorithm described above is as follow:

```
def find_task_group(G, n):
    A = Queue()
    B = Set()
    push(n)
    memory = 0
    while(A.not_empty()):
        a = A.pop()
        memory += a.weight()
        if(memory exceeds the memory size of Phi):
            memory -= a.weight()
            break;
        G -= a
        B.add(a)
```

```
        Remove all the edge with a as head
        Find out nodes with out-degree equals 0
        Add them to A
    c = new node(weight = memory, origin = B)
    G += c
    for edge in G:
        if(the head node of edge is in B):
            edge.tail = c
```

Run this algorithm with each node, we can get a group of nodes, the cost and benefit with this node as cut point. This group is an optimal task group based on the knowledge of memory usage. After parallelism analysis, a good task group can be obtained. If task imbalance occurs, these task groups can be shrink into a new node, after this a new bigger task group can be obtained by running the algorithm on the new graph to exploit the capability of Phi.

IV. OPTIMIZE CG ALGORITHM USING MUBTA

Conjugate gradient is a iterative method, designed to calculate the numerical solution of linear equations, it works especially well for sparse matrix. CG is used to calculate the x in Ax=b, A must be symmetric and positive-definite, theoretically the calculation can be done in n(n is the rank of the matrix) steps. CG is considered stable fast and parameter-free, is broadly used in electronic, machine learning, geography and graphics etc.

CG is one of five NPB core programs as the representative of random memory access program model. After simply porting CG to Phi, our experiment reveals that the random memory access behavior and the low computational cost per byte makes CG not so efficient when using MIC center model, the bottleneck is Phi's lack of memory bandwidth.

This paper uses the implementation in NPB as a base program to optimize. Before optimization, CG is ported to C, and some minor logical modification is done, such as using parallel memcpy, asynchronized transfer and nocopy transfer, etc.

This paper uses MUBTA to optimize CG algorithm by do task assignment between CPU and Phi. Formulas in CG are as follows: Ap = A*p, pAp = p*Ap, alpha = rrk1/pAp, x = x + alpha * p, r = r - alpha * Ap, rrk = r * r, beta = rrk / rrk1, p = r + beta * p. Based on these, Ap depends on A and p, in this way this paper builds up the memory model and do the task assignment. After a full review, the memory dependence model is shown in Figure 2. After two iterations and load balancing analysis, the calculation of Ap and pAp is put on CPU, achieving balanced performance. Colidx, rowstr and A don't need to be transferred to Phi, the transfer cost is the size of Ap and p.

The experiment is carried out on a single workstation with one Phi co-processor. CPU is Xeon E5-2609 at 2.4GHz, 32GB main memory, 64bit CentOS 6.4. Co-processor is Xeon Phi coprocessor 5110P, MPSS is version 2.1.6720-13. Experiments are divided into two groups: NPB version and optimized version, each group consists of three scales: A, B and C, each runs in four parallelism: 60, 120, 180 and 240, or 1 to 4 threads each core. Each experiment runs five times, the average time is used as the final result.

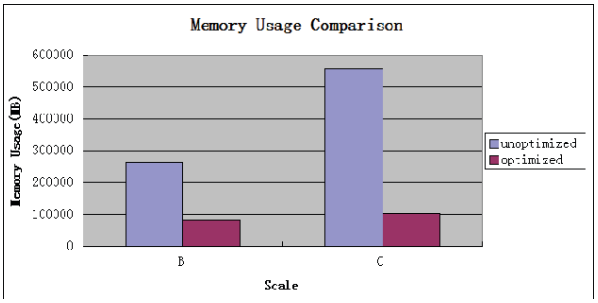


Figure 3. Memory usage comparison

The memory usage comparison diagram is shown in Figure 3. . After the application of task assignment, memory usage in class C reduces 81.8%, class B reduces 68.2%. With some portion of data is only needed on CPU, the transfer overhead is also significantly reduced. Meanwhile, the total computation time is also massively improved. The time comparison diagram is shown in

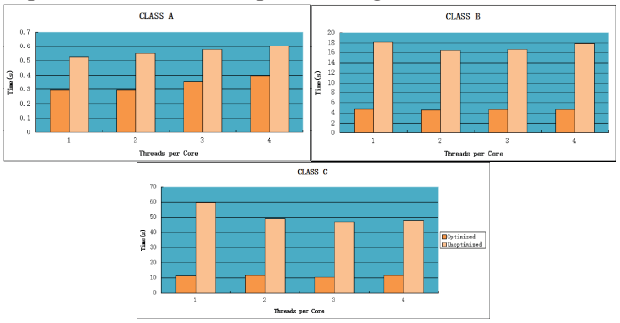


Figure 4. Computation time comparison

With task assignment computation time is better than MIC center model. In best case, optimized time is 19% of original time, worst case is 65%.

V. SUMMARY AND OUTLOOK

This paper analyzes the characteristics of task assignment between CPU and Phi, brings out a novel task assignment algorithm, which is based on classical algorithms yet simpler and still efficient. This algorithm focuses on the memory usage in task execution, the granularity of task is also finer, is more suitable for Phi.

The algorithm in this paper is still naive in several ways: first, there is hardly one real number to evaluate the task assignment result, the task assignment can't be done automatically; meanwhile, this algorithm has not taken time into consideration. This will be investigated afterwards.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 61202425) and National Hi-tech R&D program of China (863 program)(Grant No. 2012AA01A302).

REFERENCES

[1] Endong Wang and Qing Zhang, High Performance Programming Guidance On MIC[M]. Beijing: China WaterPower Press, 2012.(in Chinese)

[2] Huang B. The key power of Tianhe 2, introduction to Intel Xeon Phi architecture[J]. Micro Computer, 2013(22):131-137. (in Chinese)

[3] Salman A, Ahmad I, Hanaa A L R, et al. Solving the task assignment problem using Harmony Search algorithm[J]. Evolving Systems, 2013, 4(3): 153-169.

[4] Zou D, Gao L, Li S, et al. A novel global harmony search algorithm for task assignment problem[J]. Journal of Systems and Software, 2010, 83(10): 1678-1688.

[5] Omara F A, Arafa M M. Genetic algorithms for task scheduling problem[J]. Journal of Parallel and Distributed Computing, 2010, 70(1): 13-22.

[6] Sathappan O L, Chitra P, Venkatesh P, et al. Modified genetic algorithm for multiobjective task scheduling on heterogeneous computing system[J]. International Journal of Information Technology, Communications and Convergence, 2011, 1(2): 146-158.

[7] Mohan R, Gopalan N P, Prasanth S H D, et al. Parallel heuristic graph matching algorithm for task assignment problem in distributed computing systems[C]//Computer & Information Science (ICCIS), 2012 International Conference on. IEEE, 2012, 2: 575-579.T.N.

[8] Bui, S.Chaudhuri, F.T.Leighton, M.Sipser, Graph Bisection Algorithms with good average case behavior, Combinatorica, 7 (1987), pp.171-191.

[9] R.D.Williams, Performance of Dynamic load balancing algorithms for unstructured mesh calculations, Concurrency: Practice and Experience, 3 (1991), pp. 457-481.

[10] B.W.Kernighan and S.Lin, An efficient Heuristic procedure for partitioning graphs, Bell Systems Tech, J. 49(1970), pp 291-308.

[11] C.Farhat, A simple and efficient automatic FEM domain decomposer, Computers and Structures, 28(1988), pp.579-602.

[12] C.M Fiduccia and R.M.Mattheyses, A liner-time heuristic for improve network partitions, ACM IEEE Nineteenth Design Automation Conference Proceedings, vol.1982,ch.126, pp 175-181, 1982.

[13] Chien-chung Shen and Wen-Hsiang Tsai, A Graph Matching Approach to Optimal task assignment in Distributed computing systems using a Minimax Criterion, IEEE Transactions on Computers, vol. C-34,No.3, March 1985.

[14] Cramer T, Schmidl D, Klemm M, et al. OpenMP Programming on Intel R Xeon Phi TM Coprocessors: An Early Performance Comparison[J]. 2012.

Parallel Computer Technology Study on Hydrodynamic and Sediment Transport Mathematical Model in Estuaries Based on MPI

Cheng Wenlong[1,2], Shi Yingbiao[1,2], Wu Xiuguang[1,2], Li Zhiyong[1,2], Wang Rongsheng[1]
1. Zhejiang Institute of Hydraulics and Estuary, Hangzhou, 310020, China;
2. Key Laboratory of Estuarine and Coastal of Zhejiang Province, Hangzhou, 310016, China
Email: chengwl@zjwater.gov.cn

***Abstract*—Serial estuarine hydrodynamic and sediment transport model is parallelized by MPI method based on domain decomposition techniques, the parallel model is applied to a hydrodynamic example of tidal current and sediment transport in the estuary and offshore area of Zhejiang in China. The benchmark result shows that MPI method has good parallel effect, whose speedup is accessible to 41 by 80 processors. Simulations of the annual sediment transport by tidal currents in local region have been reduced to less than one day of compute time.**

Keywords-hydrodynamic and sediment transport model; domain decomposition; Parallel computer technology; MPI; Metis; estuaries

I. INTRODUCTION

The estuarine hydrodynamic and sediment transport model is a general term for the mathematical models applied to simulate the processes of shallow water flow, sediment transport and the resulting seabed deformation in the estuary and offshore area, thus it is of great significance to compute the model by numerical solution. In the modern simulation of local shallow water flow, there is a pressing demand for accurate computational methods, and inevitably the grid scale is narrowed to a few meters from a few kilometers, which results in the geometry multiple growth of calculation amount. Furthermore, the CPU manufacturers gradually shift their research focus from simply raising single CPU frequency to the integration of multiple computing cores within single CPU. At present, though considerable progress has been achieved in the computer technology, the hydrodynamic and sediment transport mathematical modeling is still facing an awkward situation of insufficient computing capacity. Meanwhile, for the other numerical calculation field such as fluid dynamics, combined with the new computer technology, parallel computing through multiple CPU has become increasingly common to solve the problem of insufficient computing power. In recent years, much more achievements have been obtained in the shallow water dynamics and the parallelization of hydrodynamic and sediment transport model at home and abroad[1-5], while parallelization method often is relatively single and closely associated with the model algorithm, which means the lack of commonality and difficulties in performance.

In this paper, based on MPI (Message Passing Interface) and domain decomposition techniques, it is possible to make the serial estuarine hydrodynamic and sediment transport model parallelization with a little change to the original serial code and no change to the framework and algorithm of model. In addition, the integration of subroutine for parallel modules makes it much more portable to be inserted into other modules. Then a parallel model of tidal current and sediment transport in the estuary and offshore area of Zhejiang is established to simulate the annual seabed deformation and analyze the parallel performance.

The paper is organized as follows. The hydrodynamic and sediment transport model is described in Section II, Section III which form the basis of this paper deal with domain decomposition techniques and message passing interface (MPI) protocols. An application and performance measurements are presented in Section IV and the conclusions in Section V.

II. MODEL DESCRIPTION

The serial hydrodynamic and sediment transport mathematic model in estuaries (called SW2D) is a computational fluid dynamics with application to the study of geophysical flows and seabed deformation in estuarine and coastal regions[6]. SW2D computes the water surface elevations, current and sediment concentration for two-dimensional free surface flows, by solving the Reynolds form of the Navier Stokes equations for turbulent flows.

The conservation form of the control equations include the continuity equation and momentum conservation equation, and the control equations of divergence form are expressed as follows:

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=H \tag{1}$$

Where:

$$U=\{h,hu,hv,hs\}$$

$$F=\left\{hu,hu^2+\frac{1}{2}gh^2-2h\nu_t\frac{\partial u}{\partial x},huv-h\nu_t\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right),hus-h\varepsilon_x\frac{\partial s}{\partial x}\right\}$$

$$G=\left\{hv,huv-h\nu_t\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right),hv^2+\frac{1}{2}gh^2-2h\nu_t\frac{\partial v}{\partial y},hvs-h\varepsilon_y\frac{\partial s}{\partial y}\right\}$$

$$H=\{H_1,H_2,H_3,H_4\}$$

Here:

$$H_1=\frac{E-D}{\rho_s^{'}}$$

$$H_2=-gh\left(\frac{\partial z_0}{\partial x}+\frac{u\sqrt{u^2+v^2}}{C_z^2h}\right)-\frac{\Delta\rho}{\rho_m}\frac{gh^2}{2\rho_s}\frac{\partial s}{\partial x}-$$

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.12

$$\frac{\rho_0-\rho}{\rho_m}\frac{(E-D)u}{\rho_s}+fhv+\frac{W_x}{\rho_m}$$

$$H_3=-gh\left(\frac{\partial z_0}{\partial y}+\frac{v\sqrt{u^2+v^2}}{C_z^2h}\right)-\frac{\Delta\rho}{\rho_m}\frac{gh^2}{2\rho_s}\frac{\partial s}{\partial y}-$$

$$\frac{\rho_0-\rho}{\rho_m}\frac{(E-D)v}{\rho_s}-fhu+\frac{W_y}{\rho_m}$$

$$H_4=E-D$$

Where, h=water depth; u, v = depth averaged velocities in the x- and y-directions, respectively; Z_0=riverbed elevation; Cz=Chezy coefficient expressed by Manning formula; n=Manning coefficient; s=depth-averaged sediment concentration; E,D= sediment entrainment and deposition fluxes across the bottom boundary of flow, representing the sediment exchange between the water column and bed; ρ_m=density of water-sediment mixture; ρ_s=density of sediment.

In order to accurately fit the actual estuarine irregular shore boundary, the irregular triangle element is selected for the numerical control volume, and the calculated variables are arranged in the triangle centre. The governing equations are solved by an explicit finite-volume method with good conservation. Control interface flux is calculated based on the scheme of the approximate Riemann solver. In order to achieve higher precision, a stable second-order spatial precision numerical scheme is obtained through the grid variables linear reconstruction with variable slope limit. Limited by paper's length, the numerical calculation method could be seen in the literature[6].

III. PARALLEL IMPLEMENTATION DETAILS

The serial SW2D code is parallelized by MPI method based on domain decomposition techniques. The global mesh is decomposed to as many subdomains as the number of processors in the communicator. The interprocessor communication is explicitly defined using Message Passing Interface (MPI) calls[7]. The resulting implementation is highly portable across all the parallel machines and no modifications need to be done in codes.

Given a serial finite volume code, its parallel framework can be broadly written as follows.

1) Domain Decomposition: Divide the global computational domain into subdomains.

2) Domain Mapping: Assign each subdomain as the local domain of a processor to set up an SW2D integration.

3) Interprocessor Data Exchange: Each processor communicates with its adjacent processors to ensure correctness of the boundary fluxes, before marching to the next time step.

B. Domain Decomposition

For high efficient parallel computing, load balancing and minimum interprocessor data communication are crucial, which ask to assign elements to partitions under the following rules:

1) Each partition will contain roughly the same number of elements;

2) The total length of the boundary between partitions is to be minimized.

The first rule pertains to the concept of load balancing, and with the second rule in the domain decomposition, the communication volume is minimized. It is unrealistic and not necessary to decompose the domain manually, many open-source free software, such as Chao and METIS[8], can be performed domain decomposition, which is performed using the METIS in this work. METIS uses multilevel graph partitioning techniques that first partition a coarser graph. After the coarse partitioning is complete, the graph is uncoarsened to obtain the partition for the full graph. This technique leads to better quality partitions and higher computational efficiency. The partitioning occurs before the calculation and requires a negligible uptime relative to the wole execution period.

C. Domain Mapping

After having partitioned the grid using the METIS, each processor has been assigned a subdomain in which to integrate SW2D. In the whole process, every grid in each subdomain must establish a one-to-one connection with the global grid, and vice versa.

At the conclusion of this domain mapping, each processor has the correct initial and boundary conditions with which to drive a full integration of its subdomain.

In addition, the variables output need be performed by domain mapping to global flow field information.

D. Interprocessor Data Exchange

In a domain decomposition method, data must be exchanged to preserve the correctness of the variables on and near subdomain boundaries. Data exchange is based on a mapping procedure where interior nodes along the interprocessor boundaries in some processor are mapped to the corresponding boundary nodes of the exchange partner and vice versa. Interprocessor communication of data is made using standard MPI (Message Passing Interface) non-blocking sends and receives.

IV. ANALYSIS OF THE PARALLEL PROGRAM PERFORMANCE AND APPLICATION

A. Experiment Environment

The hardware environment of experiment is based on a PowerEdge M1000e enclosure with two 2.27-GHz Intel(R) Xeon(R) E5520 processors and 24GB memory in each blade. Each Intel(R) Xeon(R) E5520 includes four processor cores. The blades use high-performance 20Gb/s Infiniband card and switch to exchange data. In this work, the high performance cluster contains 10 computing nodes. Thus, there are 80 processor cores in total.

All experiments are executed in Red Hat Enterprise Linux Server release 5.2 with the 2.6.18 Linux kernel. In addition, MPICH2-1.2.1p1 and METIS are installed.

B. The Parallel Program Application

The performance of the parallelized SW2D(called PARSW2D) model has been evaluated on the PowerEdge M1000e cluster. The benchmark case was a real case, which was run to simulate the tidal current and sediment

transport for the period of one year in the estuary and offshore area of Zhejiang in China. The computational mesh as shown in Fig. 1 contains 126351 triangular elements and 68962 nodes, the number of control volumes is equal to the number of triangular elements.

TABLE I. PERFORMANCE DATA FOR CUT DECOMPOSITION

Number of cores	Uptime （secs/1000 steps）	Total uptime （Days）	Speed up	Acceleration efficiency
1	162.51	39.54	1.0	1.00
2	82.91	20.18	2.0	0.98
3	56.43	13.73	2.9	0.96
4	42.77	10.41	3.8	0.95
5	34.58	8.41	4.7	0.94
6	29.44	7.16	5.5	0.92
7	25.80	6.28	6.3	0.90
8	23.08	5.62	7.0	0.88
16	12.24	2.98	13.3	0.83
24	8.57	2.09	19.0	0.79
32	6.86	1.67	23.7	0.74
40	5.97	1.45	27.2	0.68
48	5.29	1.29	30.7	0.64
56	4.84	1.18	33.6	0.60
64	4.30	1.05	37.8	0.59
72	4.10	1.00	39.6	0.55
80	3.98	0.97	40.8	0.51

As previously mentioned, the performance test was carried out across a variety of processor cores from 1 to 80 cores. Local computational domains when split across 80 cores is shown in Fig. 2. Each color represents the subdomain assigned to a given processor. The depth-average velocity and suspended sediment concentration in Hangzhou bay in china are shown in Fig. 3. We have compared the of serial and parallel solutions and the results are consistent with the single-precision range. The simulation was carried out to a total time period of one year. Fig. 3 plots the observed speedup values to the theoretical ones. Table 1 compares the execution time every 1000 steps and a total time period of one year for the PARSW2D Zhejiang current and sediment transport model benchmark on a variety of cores. The second column indicates the total execution time every 1000 steps . The total uptime is shown in column three, which is 39.54 days when the benchmark case was run to serial code. The observed timings indicate that the speedup is 7.0 for 8 processors, it is not linear because the discharge or water level boundary element need extra CPU times in some computational node. For increased number of processors, the time spent on communication will increase, and the computational work of every node will reduce, a slight drop in speedup is evident. At last the measured speedup is accessible to 41 on 80 cores. The parallel model greatly improve the computing speed, simulations of the annual sediment transport by tidal currents in the estuarine and coastal zone of Zhejiang have been reduced to less than one day of compute time.

V. CONCLUSIONS

In this work, a good two-dimensional hydrodynamic code, SW2D was ported onto parallel platform without changing the structure and algorithm of the serial code. The parallel code is based on domain decomposition principles and uses MPI for all interprocessor communication. The critical data exchange between processors is programmed using non-blocking sends and receives. All message passing is coded in the MPI standard interface which is portable to a large variety of parallel machines. The parallelized SW2D(PARSW2D)code can be utilized for a study of flow and sediment movement on decadal timescales in the estuarine and coastal zone of Zhejiang.

ACKNOWLEDGMENT

The research reported in this paper was supported by the National Natural Science Foundation of China (Grant No. 40806037) , the Ministry of water resources Nonprofit Research program of China (Grant No. 201101056) and the Fostering Talents and Innovation team building Project of Zhejiang province, China (Grant No. 2012F20031). Finally the authors would like to thank Master Wang binyu from TongJi University for her valuable suggestions.

REFERENCES

[1] Yu Xin, Yang Ming,Wang Min, et al, “MPI based concurrent calculation and study runoff and sediment mathematical model of the Yellow River,” Yellow River, vol. 27, Feb. 2005, pp. 49-53.

[2] Prasada Rao, “A parallel RMA2 model for simulating large-scale free surface flows,” Environmental Modelling & Software, vol. 20, Jan. 2005, pp. 47-53.

[3] Wang Jian-jun and ZHANG Ming-jin, “Parallel computer technology study on 2D numerical model of flow and sediment in rivers,” Journal of Waterway and Harbor, vol. 30, Mar. 2009, pp. 222-225.

[4] Ali Khosronejad, Seokkoo Kang, Iman Borazjani, et al, “Curvilinear immersed boundary method for simulating coupled flow and bed morphodynamic interactions due to sediment transport phenomena,” Advances in Water Resources, vol. 34, Jul. 2011, pp. 829-843, doi:10.1016/j.advwatres.2011.02.017.

[5] Chao Yang and Xiao-Chuan Cai, “A parallel well-balanced finite volume method for shallow water equations with topography on the cubed-sphere,” Journal of Computational and Applied Mathematics, vol. 235, 2011, pp. 5357- 5366.

[6] Shi Ying-biao, Pan Cun-hong and Cheng Wen-long, “Numerical simulation of sediment transport under the action of Strong hydro-dynamic,” Non-uniform non-equilibrium sediment transport of the symposium, Tianjin, 2008.

[7] Du Zhi-hui. High-performance calculation parallel programming techniques-- MPI parallel programming[M] . Beijing: Tsinghua University Press, 2001.

[8] G. Karypis and V. Kumar. METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices version 4.0. University of Minnesota, Department of Comp. Sci. and Eng., Army HPC Research Center, Minneapolis, 1998.

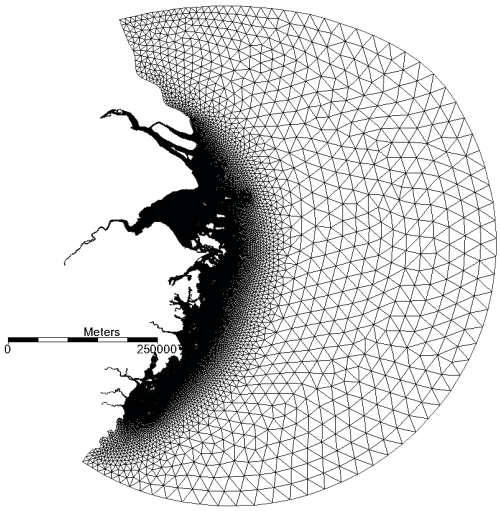


Figure 1. Computational Mesh

Figure 2. 80-way partitioning of Study Region using METIS

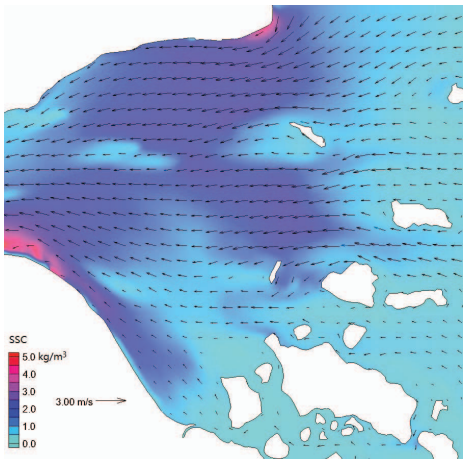


Figure 3. PARSW2D-computed depth-average velocity and sediment concentration in Hangzhou bay in china

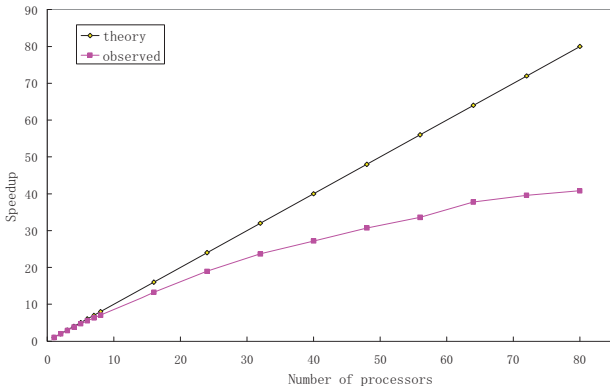


Figure 4. Relation betwwen parallel speedup and number of processor

Energy Consumption Analysis on Graphics Processing Units

Abal-Kassim Cheik Ahamed, Frédéric Magoulès
CUDA Research Center & Applied Mathematics and Systems Laboratory, Ecole Centrale Paris, France
Email: frederic.magoules@hotmail.com

***Abstract*—In this paper, we propose to investigate the compromise of the relative gains between the computation time and the energy consumption on Graphics Processing Unit (GPU). The energy consumed by GPU can not be neglected anymore, even if code acceleration is the main target. We aim to propose a prediction model of both time computation and energy consumption of linear algebra operations. Numerical experiments on a worstation equiped with two GPUs GTX275, have been performed on a set of matrices arising from the finite element discretization of the gravity equation with real data issued from the Chicxulub crater. The results exhibit performance, robustness and efficiency of Alinea library, our research group library, compared to Cusp library in terms of speed-up and energy consumption for double precision number arithmetics.**

***Keywords*-Green computing; Energy consumption; GPU computing; GPGPU; CUDA; Linear algebra; Iterative Krylov methods; CSR format; Cusp**

I. Introduction

Since the past decade, parallel computing is largely used to solve complex problems in a cost-effective way, so that the work-load can be shared between different processors. *Compute Unified Device Architecture* (CUDA) [21], was proposed by NVIDIA in 2006, in order to exploit the computational power of GPU architecture, which provides a high level GPGPU-based programming language (General-Purpose computing on GPUs). The aim in this paper consists in analysing the energy consumption of a GPU with a compromise with the execution time. Actually, most of the researches mainly focus on reducing the execution time without taking care of the behaviour of the energy consumed by the GPU. Over the past years, the community of HPC has been concerned by green computing, as reflected in the Green 500 toplist [10], which gives the top list of green supercomputers. In this paper, we design an original experimental protocol that allows to measure with a strong accuracy the energy consumed by a GPU. To evaluate and validate our protocol we use our research group linear algebra library, *Alinea* [5] [6] and compare it to Cusp [4]. *Alinea* is a linear algebra library extended to CUDA, which also includes some advanced iterative methods for sparse matrices. The optimized CUDA kernel used in this experiments are those detailed by the authors in references [7] [8] [9]. A specific feature of GPU compared to the CPU is the type of memory used. CUDA devices have four main types of memory: *global, local, constant and shared*. The properties of these memories are recalled by the authors in [8]. A hierarchy on the relative speeds to access memories is: register memory $<$ shared memory $<$ constant memory $\lll$ global memory.

The rest of the paper is organized as follows. In Section II, we describe in detail the proposed experimental protocol, which allows measurement of the energy consumed by a GPU. In Section III, the obtained numerical results are presented on a benchmark with a set of different gravimetry problems. Finally, Section IV concludes this paper.

II. Experimental protocol

We propose an experimental protocol that enables to measure the energy consumed during GPU computations. To evaluate the protocol we compare different algorithms both in terms of speed-up and energy consumption. The ideal case consits in maximizing the speed-up and minimizing the energy consumed. Most GPUs do not include sensor capable to provide the characteristics of power consumption. Reference [1] gives an analysis of power and performance of GPU-Accelerated systems, and reference [20] proposes a statistical model of energy consumption of GPUs based on hardware performance counters. First, we have to connect *an amperometric clamp*, on power supply wire of the GPU. An amperometric clamp determines an intensity of the magnetic field generated from the flowing current. The output physical quantity is an electric tension that is measured by an oscilloscope. We cannot plug directly a device on the PCI Express slot, so we have to use an extension cable, a PCI Express riser, to plug the clamp on it, as shown in Fig. 2(a). We also have to measure the current circulating through the cables that directly link the power supply unit and the GPU. We easily compute the original intensity using the linearity of the relation between the tension (voltage) and the intensity. Fig. 1 presents the experimental protocol we have designed. The riser consists of two types of power wires, one whose voltage is $+12V$ and another one whose voltage is $+3.3V$. Object (1) in Fig. 2(a) shows the $+12V$ wires. These two types of wires are seperated in order to first plug the clamp around the wires, as shown in Fig. 2(b). We use two clamps where the first is plugged on the $+3.3V$ wires and the second is plugged on the $+12V$ wires. We have two measures of the power circulating through the riser, using the following formula: $P_{riser} = 3.3 * I_{3.3V} + 12 * I_{12V}$. The intensity and the voltage given by the clamp are constant.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.13

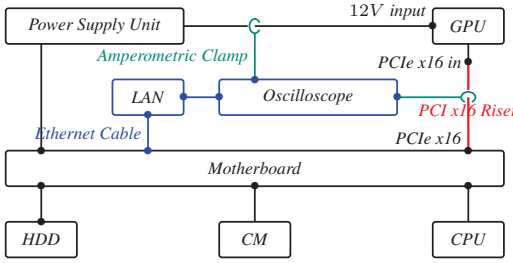
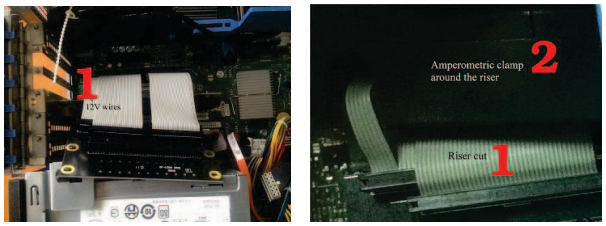


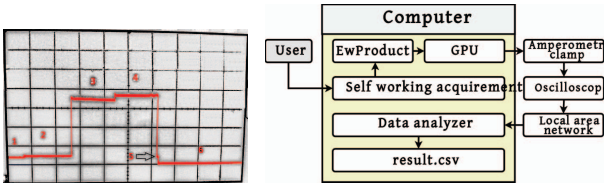
Figure 1: Design of the protocol for experimentation



(a) Riser uncut (b) Riser cut

Figure 2: Protocol for experimentation

In Fig. 2(b), object (1) correponds to the riser cut and object (2) to the amperometric clamp around the riser. The foreground clamp is plugged into the power supply unit whereas the background one is plugged into the riser. To retrieve the measure we use a numerical oscilloscope, which is connected to the local area network using an ethernet cable. The computer we used has two NVIDIA GTX 275 GPUs. The experiments have been performed on the first GPU and the second one being mostly used for the video output. The measures retreived are noisy due to a disturbance caused by the other components of the machine, making the results hard to analyze. To remove the noise of the signal, we use *weighted averaging filter or Gaussian filter* with a standard deviation σ corresponding to the distribution of a Gaussian function. This filter consists in averaging each point of the signal with a weight. To evaluate our results and validate our experimental protocol we studied qualitatively the response of the GPU for the elementary operations such as memory allocation, addition of vectors Daxpy ($y = \alpha \times x + y$), which is a level one (vector) operation between two vectors in the Basic Linear Algebra Subprograms (BLAS) and product of two vectors (element wise product), and communication (memory copy) between the GPU and the CPU. We have designed a program that executes an operation to evaluate and to change the size of the vectors. From the numerical oscilloscope we follow the execution steps of the program, *i.e.*, CPU instruction and kernel launch. The execution steps are the following: (i) n, the size of the vectors, is given, (ii) the random vectors are generated on the CPU (RAM memory), (iii) the



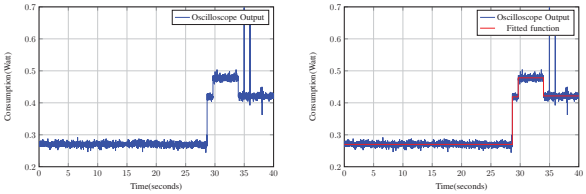
(a) Screen of the oscilloscope (b) Program execution

Figure 3: Screen of the oscilloscope and program execution

parameters: n_axpy, number of Saxpy and n_prod, number of products to be performed, number of threads per block, block load factor, are entered, (iv) the global GPU memory are allocated, (v) the parameters: n_copy, number of copies to be performed, which allows to pause the execution and look at the consumption of the memory allocation process, are entered, (vi) n_copy of the vector are done into the GPU memory, (vii) the addition kernel is executed n_axpy times and then the total elapsed time is saved, (viii) the product kernel is executed $nprod$ times and then the total elapsed time is saved, (ix) the results of all these computations are copied back to the CPU. Fig. 3(a) clearly shows these six different phases, which corresponds to the six phases of the program: (1) the memory is allocated but nothing has been copied inside, (2) the program is copying data from CPU to GPU memory, (3) the program is performing the sum, (4) the program is performing the element wise product, (5) the program is copying data back, deallocating the memory and finally (6) the GPU is back to stand-by. Let us remark that after the last step and a little waiting time that depends on the other GPU usages, the GPU will enter in an energy saving mode and the consumption will decrease until it reachs its default regime. The first results show that there are significant changes in the consumption during the different phases of the program execution. As a result, we are able to follow the execution of a program on the oscilloscope. It then validates our protocol to achieve our goal, *i.e.*, carrying out a benchmark of elementary operations. In the following section we look into the realization of this benchmark. The signal results from the oscilloscope have been collected via LAN connection. The data of the ouput results are saved into "*Comma-Seperated Values*" file format, which gives two columns that respectively correspond to the times in seconds and the measured tensions in volts.

III. Numerical results

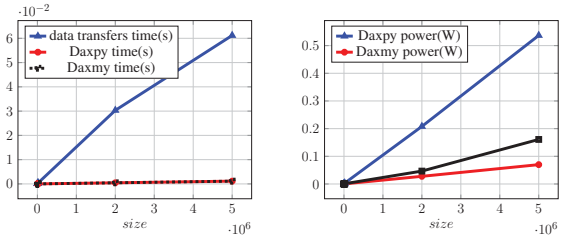
In this part, we collect the numerical results to examine and corroborate the efficiency of our experimental protocol. The workstation used for the experiments consists of an Intel Core i7 920 2.67GHz with eight cores, 5.8 GB RAM and two NVIDIA GTX275 GPU, double precision and CUDA 4.0 compatible, equipped with 895MB memory. This configuration is adequate for performing all linear algebra



(a) Energy consumption for the different phases of the execution

(b) Signal after Gaussian filter

Figure 4: Function approximating the curve



(a) Time in seconds (s)

(b) Power in Watt (W)

Figure 5: Double precision: data transfers, Daxpy, Daxmy

operations. In order to better understand the behaviour of the execution time and the energy consumption of the program, we start to evaluate the correlation between the execution time and the energy consumption of elementary operations executed on a single GPU by changing the size of the vector. The proposed benchmark consists in performing the same operation among allocation, copy, sum and product several times, *i.e.*, with the same size of vectors, at least 50 times corresponds to 50 measures. Fig. 4(a) shows an example of a measure of the energy consumption for the different phases of the execution obtained from the oscilloscope. The figure represents the electrical power in Watt over the time in second. The corresponding energy is computed with the formula: Energy = Power $\times$ Time. As seen in Fig. 4(a) the intensity reachs two positions: a position when the GPU is not working and a position when the GPU is working. The experimental results show that the GPU consumes energy even when it is not working. We can see in Fig. 4(a) four noisy phases. Each phase is associated respectively with *sleep*, *sum*, *product*, or *copy* operation. To compute the execution time and the energy consumed by the GPU for each state, we implement a program that can find each state accurately. We approximate the energy consumed during a phase by a constant. Indeed, our measures hint that the theoretical curve follows a simple constant function defined on a subset of $\mathbb{R}$. As a result the algorithm has to determine the moment when the pertinent leaps happen, which corresponds to a transition between two different phases and then to compute a constant value that will best approximate the function between jumps. The idea behind the considered automate algorithm consists in setting the value of the approximated function between two jumps as the mean value of the measured curve during the phases. The pertinent leaps are identified by iteratively computing the mean of the first values of the curve. We conclude that the jump is relevant if the ten next values are too far from the examined mean value. Taking into account the variations of the standard deviation (σ) gives a more accurate approximation. Fig. 4(b) provides an example of an approximated function computed by the algorithm.

In this paper, Compressed-Sparse Row (CSR) format available in both Alinea and Cusp libraries, is considered. In Fig. 5, we report respectively the execution time in seconds and the power consumption in Watt (W) for double precision data transferring from CPU to GPU, the addition of vectors (Daxpy) and the element-wise (Daxmy) operations. According to the results, the linear prediction of the GPU execution time in second and the electrical power in Watt of the double precision data transfers between CPU and GPU are respectively formulated as: $0.00007 + 10^{-8}x$ and $0.00002 + 10^{-7}x$, where x is the size of the problem, and where the coefficient of determination 0.9922 and 0.9998 respectively. The coefficient of determination, shows how data are close to the fitted regression model.

The linear prediction of the GPU execution time and electrical power for Daxpy are expressed as follows, where x is the size of the problem: Time (s): $4.905\times10^{-6}+2.272\times10^{-10}x$ (0.999) and Power (W): $2.682 \times 10^{-4} + 1.389 \times 10^{-8}x$ (0.999). The corresponding coefficients of determination are given in brackets. Thereafter, we will use the same convention as previously, *i.e.*, the coefficients of determination will be in brackets.

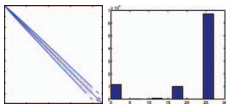
The linear prediction of the GPU execution time and electrical power for double precision element wise product are expressed as follows: Time (s): $-6.668 \times 10^{-6} + 2.164 \times 10^{-10}x$ (0.9857) and Power (W): $-1.417 \times 10^{-4} + 3.131 \times 10^{-8}x$ (0.9856).

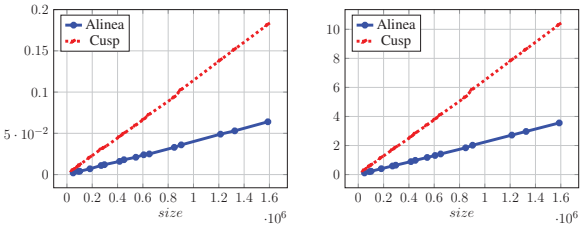
Besides the execution time, the other phases illustrated in Fig. 4(b) satisfy the following conditions: P_{idle}, power (instantaneous) at rest: *47.4 W*, P_{active}, power when the card is activated (memory allocated but not yet used): *58.43 W*, P_{break}, power during a break in the calculation: *60.77 W*, P_{end} , power at the end of the program, before the process is completed: *59.14 W*. The energy consumed is given in the following formula (1):

$$\begin{aligned} Energy \quad = \quad & P_{idle} \times t_1 + P_{active} \times t_2 + P_{break} \times t_3 + P_{end} \times t_4 \\ & + P_{CPU \leftrightarrow GPU}(n) \times T_{CPU \leftrightarrow GPU} * n \\ & + P_{compute}(n) \times T_{compute} * n \end{aligned} \quad (1)$$

where t_1, the time during the program runs without using the graphics card, t_2, the time during the memory is allocated without being used, t_3, the time during the GPU is not used in the middle of the calculations, *e.g.*, computation is done on the CPU and t_4, the time during the main program continues to run and the GPU is not used.

Table I: Patterns of the matrices

	**gravi 49,248**
	h = size = 49,248 density = 0.044
	bandwidth = 3,779 nonzero = 1,067,632
	nonzero/h = 21.678 max row = 27
	nonzero/h stddev = 9.425

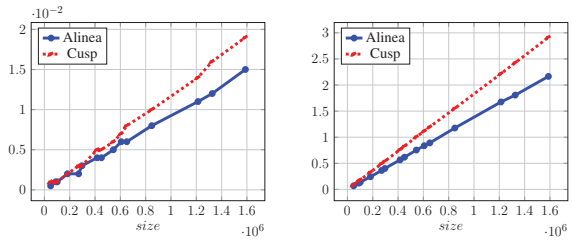


(a) Time in seconds (s) (b) Energy in Joule (J)

Figure 6: Double precision dot product

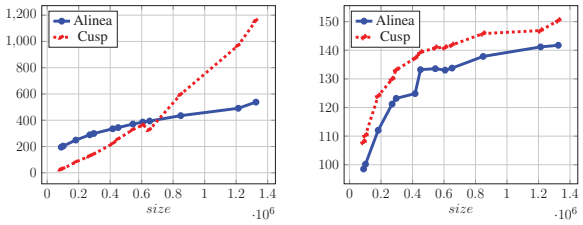
In order to evaluate and consolidate our experimental protocol, we consider a real engineering problem namely the solution of the gravity equation in the Chicxulub crater in the Yucatan Peninsula in Mexico. The discretized domain consists of a parallelepiped 200 km $\times$ 200 km $\times$ 10 km. The associated matrices vary from $49,248$ to $1,325,848$ rows and the numbers of nonzeros values vary from $2,010,320$ to $33,321,792$. They have the same struture pattern given in the first column of TABLE I. Fig. 6 gives the kernel time in seconds and the energy consumed in Joule (J) by the GPU for both Alinea and Cusp double precision dot product. The linear prediction of the GPU execution time and energy consumption for double precision dot product are formulated as follows, where x is the size of the problem: Alinea-Time(s): $-2.216 \times 10^{-4} + 4.002 \times 10^{-8}x$ (0.9996), Cusp-Time(s): $-6.325\times10^{-4}+1.142\times10^{-7}x$ (0.9996), Alinea-Energy(J): $-1.897 \times 10^{-2} + 2.241 \times 10^{-6}x$ (0.9996), Cusp-Energy(J): $-5.477 \times 10^{-2} + 6.514 \times 10^{-6}x$ (0.9996),

Fig. 7 represents the kernel time in seconds given by the oscilloscope and the energy consumed by the GPU in Joule (J) for both Alinea and Cusp double precision sparse matrix-



(a) Time in seconds (s) (b) Energy in Joule (J)

Figure 7: Double precision CSR SpMV



(a) Time in seconds (s) (b) Energy in Joule (J)

Figure 8: Double precision CSR Conjugate Gradient

vector multiplication for CSR format, the most time consuming operation [2], [3]. The linear prediction of the GPU execution time and energy consumption for double precision SpMV are expressed as follows: Alinea-Time(s): $-3.552\times10^{-5}+9.286\times10^{-9}x$ (0.9996), Cusp-Time(s): $1.165\times10^{-5}+1.177\times10^{-8}x$ (1.000), Alinea-Energy(J): $-1.223\times10^{-3}+1.371\times10^{-6}x$ (0.9999), Cusp-Energy(J): $-6.667\times10^{-4}+1.834\times10^{-6}x$ (1.000) As we can see in Fig. 6 and Fig. 7, the dot product and the SpMV exhibit better results for Alinea compared to Cusp in terms of kernel execution time and energy. In Fig. 8 we report the time in seconds of the GPU part of the Conjugate Gradient (CG) solver and the energy consumed by the GPU in Watt seconds for both Alinea and Cusp double precision for CSR format. The CG benchmark fix the residual tolerance threshold $\epsilon = 10^{-6}$. The results clearly confirm the effectiveness and the robustness of Alinea. As we can also seen in Fig. 8 that the energy consumed increases when the size of the problem increases. The prediction of the GPU execution time and energy consumption for double precision solver are expressed as follows: Alinea-Time(s): $-6.637\times10^{-1}+6.138\times10^{-6}x$ (0.9739), Cusp-Time(s): $-7.440\times10^{-1}+7.493\times10^{-6}x$ (0.9838), Alinea-Energy(J): $-1.179\times10^{2}+8.826\times10^{-4}x$ (0.9704), Cusp-Energy(J): $-7.264+5.494\times10^{-4}x$ (0.9986). In the previous analysis no preconditioner have been used for the CG, but it is fundamental to mention that CG can be used very efficiently on GPU with some preconditioning based on domain decomposition methods [11], [15]. For this purpose, interface conditions between the subdomains are tuned with a continous approach like in the FETI method [12] or like in the Schwarz method [13], [14]. These interface conditions can also be tuned with a patch substructuring approach as introduced in [16]–[19]. The associated problem, with the Lagrange multipliers, condensed on the interface between the subdomains, is then solved with an interative algorithm on the CPU. At each iteration, each subproblem defined in each subdomain is solved with the CG method on a different GPU. The proposed protocol is naturaly applied on each GPU to estimate the energy consumption, upon the size of the submatrix defined in each subdomain. The results we have obtained for multi-GPU confirm the efficiency of our protocol for Schwarz methods.

When the computation time increases the energy consumption becomes important. The GPU is effective for large size problems in terms of computation time, but consumes much more energy than the CPU.

IV. Conclusion

In this paper, we have presented an original experimental protocol for measuring the energy consumption of GPU during computation. We have focused our analysis on investigating the compromise between the relative gains in computation time and power consumption of GPU. We proposed some mathematical models to predict the execution time and energy consumption according to the properties of the problem. The effectiveness and robustness of our experimental protocol are evaluated and validated by numerical experiments performed on a machine with 2 GPUs NVIDIA GTX275. The presented results, conducted on a real engineering problem, highlight the robustness and performance of Alinea, our research group library, compared to Cusp and demonstrate the efficiency of our proposed experimental protocol to measure the energy consumption of GPU.

References

[1] Y. Abe, H. Sasaki, M. Peres, K. Inoue, K. Murakami, and S. Kato. Power and performance analysis of gpu-accelerated systems. In *Presented as part of the 2012 Workshop on Power-Aware Computing and Systems*, Berkeley, CA, 2012. USENIX.

[2] N. Bell and M. Garland. Efficient sparse matrix-vector multiplication on CUDA. Nvidia Technical Report NVR-2008-004, Nvidia Corporation, 2008.

[3] N. Bell and M. Garland. Implementing sparse matrix-vector multiplication on throughput-oriented processors. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis (SC'09)*, pages 1–11, New York, NY, USA, 2009. ACM.

[4] N. Bell and M. Garland. Cusp: Generic parallel algorithms for sparse matrix and graph computations, 2012. Available on line at: http://cusp-library.googlecode.com (accessed on September 29, 2012).

[5] A.-K. Cheik Ahamed and F. Magoulès. Fast sparse matrix-vector multiplication on graphics processing unit for finite element analysis. In *IEEE 14th International Conference on High Performance Computing and Communication (HPCC)*, pages 1307–1314. IEEE Computer Society, 2012.

[6] A.-K. Cheik Ahamed and F. Magoulès. Iterative methods for sparse linear systems on graphics processing unit. In *IEEE 14th International Conference on High Performance Computing and Communication (HPCC)*, pages 836–842. IEEE Computer Society, june 2012.

[7] A.-K. Cheik Ahamed and F. Magoulès. Iterative Krylov methods for gravity problems on graphics processing unit. In *IEEE 12th International Symposium on Distributed Computing and Applications to Business, Engineering Science (DCABES)*, pages 16–20. IEEE Computer Society, 2013.

[8] A.-K. Cheik Ahamed and F. Magoulès. Schwarz method with two-sided transmission conditions for the gravity equations on graphics processing unit. In *IEEE 12th International Symposium on Distributed Computing and Applications to Business, Engineering Science (DCABES)*, pages 105–109. IEEE Computer Society, 2013.

[9] A.-K. Cheik Ahamed and F. Magoulès. A stochastic-based optimized Schwarz method for the gravimetry equations on GPU clusters. In *Domain Decomposition Methods in Science and Engineering XXI*. Springer, 2014.

[10] Green500, 2013. Available on line at: http://www.green500.org (accessed on August 7, 2014).

[11] Y. Maday and F. Magoulès. Absorbing interface conditions for domain decomposition methods: a general presentation. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3880–3900, 2006.

[12] Y. Maday and F. Magoulès. Optimal convergence properties of the FETI domain decomposition method. *International Journal for Numerical Methods in Fluids*, 55(1):1–14, 2007.

[13] F. Magoulès, P. Iványi, and B. Topping. Convergence analysis of Schwarz methods without overlap for the Helmholtz equation. *Computers and Structures*, 82(22):1835–1847, 2004.

[14] F. Magoulès, P. Iványi, and B. Topping. Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 193(45–47):4797–4818, 2004.

[15] F. Magoulès and F.-X. Roux. Lagrangian formulation of domain decomposition methods: a unified theory. *Applied Mathematical Modelling*, 30(7):593–615, 2006.

[16] F. Magoulès, F.-X. Roux, and L. Series. Algebraic way to derive absorbing boundary conditions for the Helmholtz equation. *Journal of Computational Acoustics*, 13(3):433–454, 2005.

[17] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3742–3759, 2006.

[18] F. Magoulès, F.-X. Roux, and L. Series. Algebraic Dirichlet-to-Neumann mapping for linear elasticity problems with extreme contrasts in the coefficients. *Applied Mathematical Modelling*, 30(8):702–713, 2006.

[19] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approach to absorbing boundary conditions for the Helmholtz equation. *International Journal of Computer Mathematics*, 84(2):231–240, 2007.

[20] H. Nagasaka, N. Maruyama, A. Nukada, T. Endo, and S. Matsuoka. Statistical power modeling of gpu kernels using performance counters. In *Green Computing Conference, 2010 International*, pages 115–122, Aug 2010.

[21] Nvidia Corporation. *CUDA Toolkit Reference Manual*, 4.0 edition. Available on line at: http://developer.nvidia.com/cuda-toolkit-40 (accessed on September 29, 2012).

A Parallel Grey Theoretic Model of Inland Water Transport Management

Chao Sun[1], She-sheng Zhang[2]

[1] Zhixing College of Hubei University, Wuhan, P. R. China
[2] Wuhan University of Technology, Wuhan, P. R. China.
10368260@qq.com

Abstract: **Development of inland water transport management is the major macroeconomic research of the transport. Using grey theory method, GM (1,1) management parallel calculation model is established. The model analysis the growth rate of highway, waterway and railway, forecasts the trend of the carrying capacity of water transportation, inland waterways mileage and high-grade waterways mileage in China.**

Keywords: **GM (1, 1) model; grey management; inland river shipping.**

I. Introduction

Compared to highway and railway, inland waterway transport can make use of water resources, so that its construction cost is low.

There are long inland waterways in Russia, China, Brazil, and United States. Density of inland waterway freight (freight turnover completed each kilometers waterway) in American and western European countries ,especially in United States and Germany, are higher than other countries. The United States has developed a channel network whose route is the Mississippi River. Its main and main tributaries has realized the highly channelizing according to need, 2.74 m deep, 9700 km long in uniform standard, accounting for approximately 50% of the total mileage, connected to the north and the Great Lakes, east along the Saint Lawrence Seaway to the Atlantic ,whose estuary connects with gulf coast canal. Rhine in Western Europe [2] Originates in Switzerland, by way of France, Germany, flows into North Sea in the Netherlands, whose tributaries are highly channelizing or governance, connected with EIbe river and Weser. After construction in the decades period, the Rhine - the United States -the Danube canal engineering has been completed, and the Rhine and Danube is connected, from east out the Black Sea. This channel network navigation 1350 tons self-propelled barge, channel network extends more than 20,000 kilometers. In China [3], there are more than 1500 rivers and 900 lakes whose drainage area is over 1000 square kilometers, in which navigable for ships of over 500 tons takes up less than 10% of the total mileage of waterways, most of which is navigable for ships of less than 100 tons. China inland waterways are mainly distributed in the Yangtze river and the Hearl river, Huai river and Heilongjiang river, in which the navigable mileages of Yangtze river and Pearl river and the Huai river account for 82.3% of national total navigable mileage. The Yangtze river system with the best general conditions, 7 million kilometers mileage, accounts for 70% of the navigation mileage, Yangtze river route has 3638.5 km navigable mileage.

The paper is financially supported by China national natural science foundation (No.51139005),

II. Status of waterway transportation

This paper has studied various transportation modes for the carriage of goods, and general cargo transportation ways consist of highway transportation, railway transportation and waterway transportation [5]. According to the statistical data from Ministry of Transport of the People's Republic of China and Ministry of Railways of the People's Republic of China (1999 ~ 2008), The main modes of cargo transportation are increasing with economic development of China. Especially waterway transportation development speed (due to the influence of the world financial crisis, growth rate reduced in 2008) is growing rapidly. Annual growth rate of various transportation ways are shown in **Fig 1.**

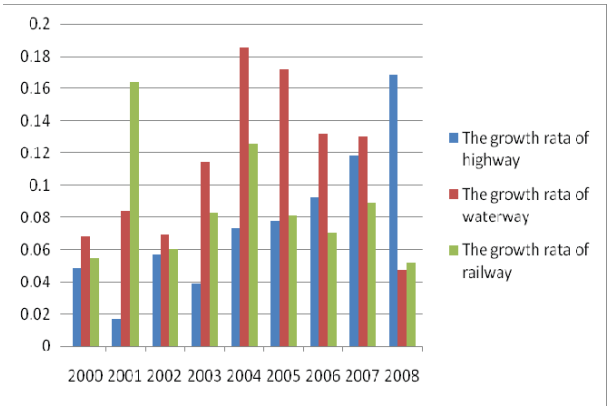


Fig.1. The growth of vigorous transportation

Based on analysis of figure 1, the annual growth rate

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.14

of waterway transportation is higher than railway transportation and highway transportation except 2001 and 2008, and it is more than 16% in 2004 and 2005.The average growth rate from 2000 to 2008 is 11.15%, so waterway transportation is increasing year by year.

III. The grey prediction of waterway based on GM(1,1)

A. The statistical data of waterway transportation from 1999 to 2008

According to the data from Ministry of Communications [4], the water transport cargo volume (1999~2008) may be obtained and used in the paper.

B. Data processing and Establishing GM (1, 1) model

Definition: The water transport cargo volume of year n is $x^{(0)}(n)$. $x^{(1)}$ is the ordinal accumulation of $x^{(0)}(n)$,and $z^{(1)}$ is the mean of sequence of $x^{(1)}$.

$$P=\begin{bmatrix} a \\ b \end{bmatrix}, Y=\begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \cdots \\ x^{(0)}(n) \end{bmatrix}, B=\begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \cdots & 1 \\ -z^{(1)}(n) & 1 \end{bmatrix}$$

Establishing GM (1, 1) model:

The time response function:

$$x^{(1)}(k+1)=(x^{(0)}(1)-\frac{b}{a})e^{-ak}+\frac{b}{a}$$

$$k=1,2,...,n$$

The value restoring function:

$$\hat{x}^{(0)}(k+1)=\hat{x}^{(1)}(k+1)-\hat{x}^{(1)}(k)=(1-e^{a})(x^{(0)}(1)-\frac{b}{a})e^{-ak}$$

$$k=1,2,...,n$$

Least squares estimate parameter vector P:

$$P=(B^{'}B)^{-1}B^{'}Y$$

The paper uses the data from 1999 to 2006 to establish the gray prediction model, and uses MATLAB to solve it.

$$P=\begin{bmatrix} -0.1282 \\ 9.1631\times10^{8} \end{bmatrix}$$

So the prediction function expression is:

$$\hat{x}^{(0)}(k+1)=(1-e^{-0.1282})(x^{(0)}(1)-\frac{9.1631\times10^{8}}{-0.1282})e^{0.1282k}$$

$$k=1,2,...,n$$

C. Substituting for the variable to predict

According to the model and original data, the water transportation cargo volume from 2007 to 2019 can be predicted, and shown in Fig.2.

In 2007, the actual amount of water transport is 2811990000 (ton), and the predicted value with GM(1,1) model is 2782497680 (ton). The relative error is only 1%, or the accuracy is higher than 99%. So the GM(1,1) model can be used to predict water transport cargo. The trend of water transport cargo (1999~2019) is shown in Fig.2.

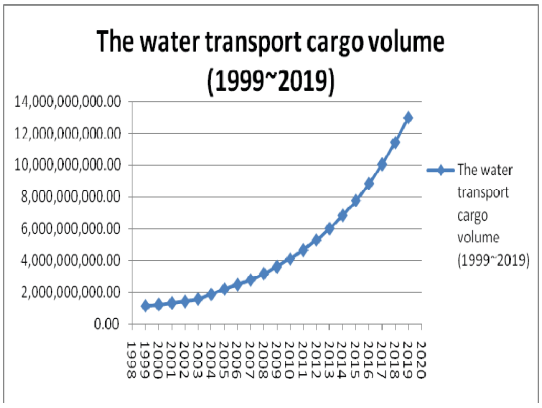


Fig.2. The water transport cargo volume (1999~2019)

IV. The prediction of inland waterways mileage and high-grade waterways mileage

Similarly, the paper establishes GM (1,1) model to predict inland waterways mileage and high-grade

waterways mileage. The inland waterways mileage and high-grade waterways mileage from 1999 to 2019 is shown in Fig.3, and Fig.4.

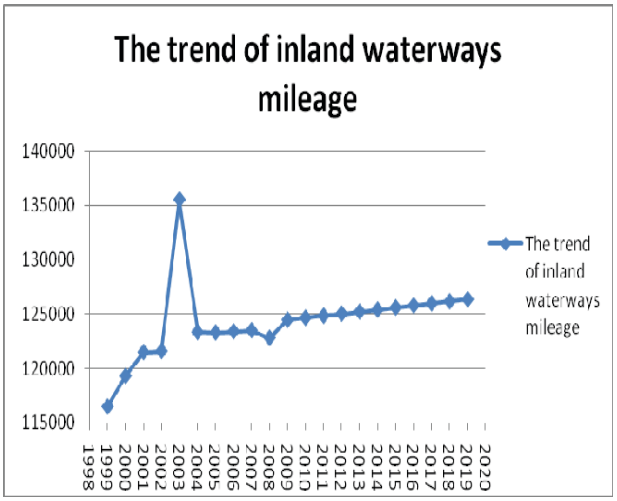


Fig.3. The trend of inland waterways mileage(1999~2019)

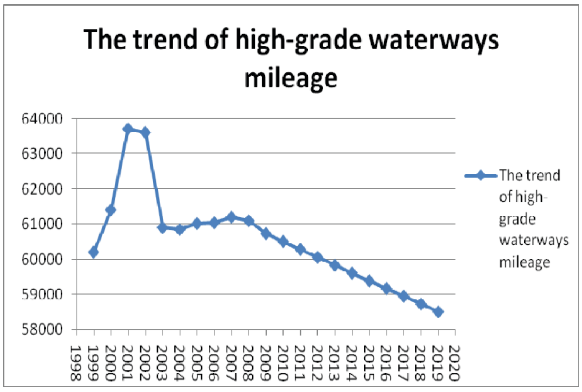


Fig.4. The trend of high-grade waterways mileage (1999~2019)

V. Data analysis and Strategy by using parallel computer

In the case data is hug, the parallel computer is used to calculation. The N=16 processors is chosen in this paper. After calculation, the results are shown in the figures. It's obvious that the water transport cargo volume will increase dramatically in the few years (from **Fig.2.**). According to the present economic situation, the inland river shipping cargo volume accounted for a proportion of the water transport cargo volume will not reduce, so in the next few years the inland river shipping cargo volume will increase substantially. But the inland waterways mileage in the next few years increases slowly (from **Fig.3.**). Especially, the construction of high-grade waterways mileage will decrease (from **Fig.4.**). Apparently this can not adapt to the growing interrelatedness inland water transport.

For inadaptability between inland river shipping transportation and channel development, the government should pay more attention to the construction of inland water transport. With its special advantages inland water transport will play a very important role in future logistics transport. In China, inland water transport has its natural advantages, but at present insufficient of channel development especially high-grade waterways severely is restricting the development of inland water transport. Due to construction of highway and railway in the future is gradually saturated, government should mainly take the development of inland water transport into consideration in the next economic plan and do our best to realize various transportation ways optimal allocation and maximum possible use.

VI. Conclusion

The parallel computer gray theoretic GM (1,1) management model is discussed in this paper for the inland water transport management. This model analysis the growth rate of highway, waterway and railway, forecasts the trend of the carrying capacity of water transportation, inland waterways mileage and high-grade waterways mileage in China.

Reference

[1] Yang Chengqing, American inland navigation development experience and enlightenment[J], China port &waterway, 2008(7),33-35;

[2] Li Cheng, Zhou Zuofu, Matters needing attention for inland development of china from the view of measures takes taken by german government for supporting inland navigation development[J], Port and waterway engineering, 2004(6), 17-19.

[3] Wu Peng, Development opportunities and Challenges

of inland river shipping transportation in China [J], Port &waterway engineering, 2010(2),11-15.

[4] Ministry of Transport of the People's Republic of China, 《1998-2008 inland water transport mileage TAB from Ministry of Transport of the People's Republic of China》 http://www.moc.gov.cn/zhuzhan/tongjigongbao/hangyenianjian/

[5] Ministry of Railways of the People's Republic of China, 《1998-2008 statistical yearbook from Ministry of Railways of the People's Republic of China》

. http://www.china-mor.gov.cn/zwgk/zwgk_tjxx.html

Design of Fuzzy Control System for Tank Liquid Level Based on WinCC and Matlab

Zhu Jian-Jun
College of Information & Control Engineering, Jilin Institute of Chemical Technology
Jilin City,China
Zjj099@163.com

***Abstract* — In the industrial control area, WinCC is widely applied in industrial production process of real-time monitoring control and has become one of the important configuration software. But it is still unable to realize complex control algorithm with shortage of programming language. While Matlab can deal with large amounts of data with high efficiency with its toolbox and Simulink simulation environment, complex control algorithm can be realized through the research on the model and simulation of the control system. Thus, in this paper, they are combined organically under the background of level control of the tank, and a real-time fuzzy control scheme based on OPC communication technology is put forward. Finally intelligent control of the tank level is realized and its effectiveness is verified as well.**

Keywords- OPC; WinCC; Matlab; Simulation

I. INTRODUCTION

As a popular controlled object in process control system, liquid level control system finds its typical application in industrial manufacturing. Liquid level controlled plants are characterized by nonlinearity, long time delay and great inertia[1][2]. Conventional control methods, therefore, cannot fulfill process control. The application of traditional PID controller shows unsatisfactory result and its system needs more regulation time. Compared with conventional control approaches, fuzzy control of nonlinear or time-delayed objects performs better than PID and it is more adaptable in the process of liquid level control because mathematical modeling is not necessarily established and control quality is less affected by features of controlled plants and parameter changes. Focusing on tank systems, the present paper adopts Matlab to conduct background control strategy calculations and data processing. Simulation of configuration is realized by linking and setting up parameters in WinCC, which not only reduces dependence on hardware but also runs better real-time monitoring of controlled objects. The research has a promising application prospect in experimental teaching[3].

II. STRUCTURE OF LIQUID LEVEL CONTROL SYSTEM

Liquid levels of overhead water tank, middle water tank and lower water tanks are objects controlled by tank level system, which aims to control liquid level within the session of a given level on a smaller scale. This paper mainly studies the lower tank, the structure of whose control system is shown in Fig 1.

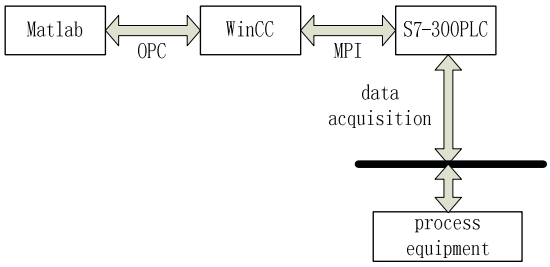


Figure 1. Control system structure

III. DESIGN OF FUZZY CONTROLLER

A. *Structure of Fuzzy Controller*

Based on features of liquid level process control and its requirements, fuzzy controller adopts a two-dimensional structure shown in Fig 2. Fuzzy controller is composed of input fuzzification, fuzzy inference and defuzzification. E and Ec in Fig 2 are fuzzy variables of deviation e and deviation change rate ec. u* is incremental output which is then defuzzified. Ke, Kec is respectively fuzzy quantitative factors E and EC and Ku is scale factor u*.

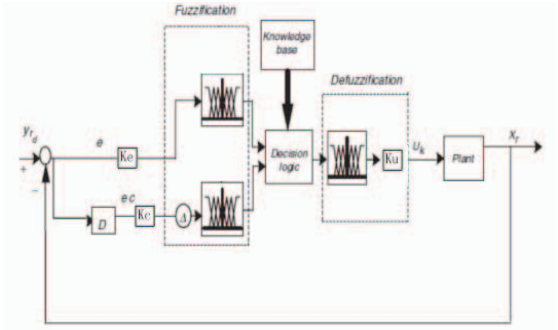


Figure 2. Fuzzy control block diagram of the level control system

B. *Fuzzy Domain*

The scope of the input and output variables are determined based on experiment. The range of deviation of the E level is [-15, +15], range deviation change rate Ec for [-0.1, +0.1]. u* is the fuzzy controller output which regulates valve, whose range is 0%～100%.

C. *Variable Settings*

Setting of variables level and rate include the value range and distribution of fuzzy subsets and the type of membership function. In order to simplify the computation of the fuzzy controller, five equal span triangular membership functions are employed for controller input variables and output variable. They are NB,NS,ZE,ZO,PS,PB. The membership functions of these fuzzy variables are shown in Fig 3.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.15

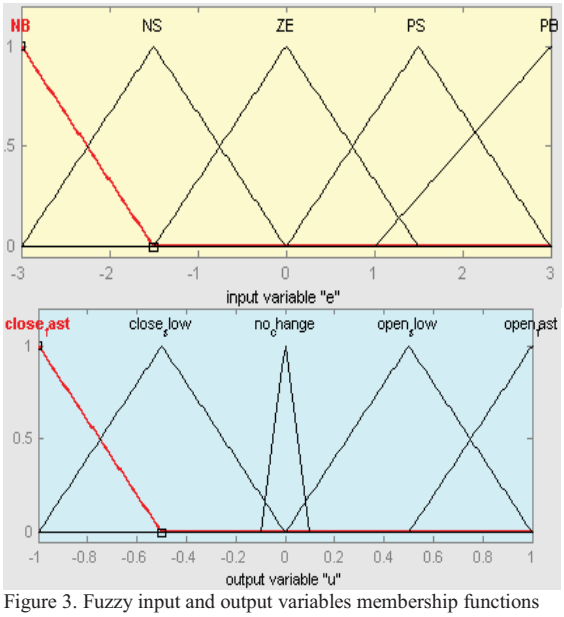


Figure 3. Fuzzy input and output variables membership functions

D. *Fuzzy Control Rules Table*

The major components of a fuzzy controller are a set of linguistic fuzzy control rules and an inference engine to interpret these rules[4][5].

These fuzzy rules offer a transformation between the linguistic control knowledge of an expert and the automatic control strategies of an activator. Every fuzzy control rule is composed of an antecedent and a consequent. A general form of the rules can be expressed as:

Ri: IF X is A1 and Y is A2, THEN U is C1

While Ri is the ith rule, X and Y are the states of the controlled system inputs respectively and U is the control output. A1,A2 and C1 are the corresponding fuzzy subsets of the input and output universe of discourse respectively.

According to this theory, 25 fuzzy rules are established. Those fuzzy rules are listed in Table I.

TABLE I. FUZZY RULES TABLE FOR THE LIQUID LEVEL TYPE

E	Ec				
	NB	NS	ZE	PS	PB
NB	close_fast	close_fast	close_slow	close_slow	no_change
NS	close_fast	close_slow	close_slow	no_change	open_slow
ZE	close_slow	close_slow	no_change	open_slow	open_slow
PS	close_slow	no_change	open_slow	open_slow	open_fast
PB	no_change	open_slow	open_slow	open_fast	open_fast

IV. SYSTEM SOFTWARE DESIGN

A. *Supervisory configuration Structure of Fuzzy Controller*

The present experiment adopted the industrial configuration software of Siemens Company, and completed a monitoring platform for the basic functions, such as real-time curve, curve of history, historical data, the realization of animation, alarming module, reports module, printing module, all of which are shown in Fig 4.

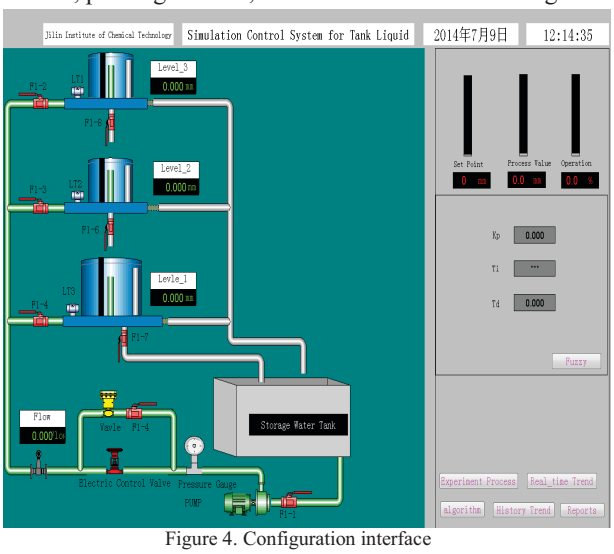


Figure 4. Configuration interface

B. *Establishment of System Model*

The Simulink is used to write a conventional PID and fuzzy PID two control algorithms. The models of two control algorithms are shown in Figure 5. The implementation scheme is realized based on realization of real-time communication between Simulink and WinCC by the OPC technology[6]. A real-time control algorithm in simulink is established to control tank level by direct calling “OPC Read /Write” function in this paper. A clear and visual comparison of the advantages and disadvantages of different control strategies are made by the trend of different control algorithm[7].

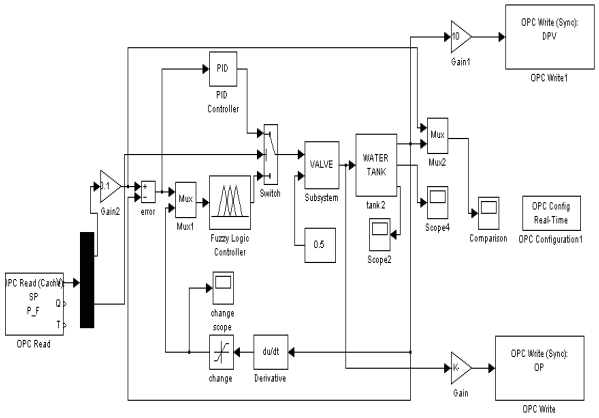


Figure 5. The simulation model of the system

V. EXPERIMENTAL DATA ANALYSIS

At this point, real-time fuzzy control system for level of tank has been designed based on Matlab and WinCC. WinCC and Matlab are put into operation, the curve of the actual level tracking the desired level is gained through WinCC and Simulink simulation, as shown in Figure 6.

In the same conditions, the experiment results show that the PID reaction speed is slow, but the dynamic quality is better and can eliminate the static error. Fuzzy

control has no overshoot. Besides, both rising speed and stable speed are the fastest in spite of the static error.

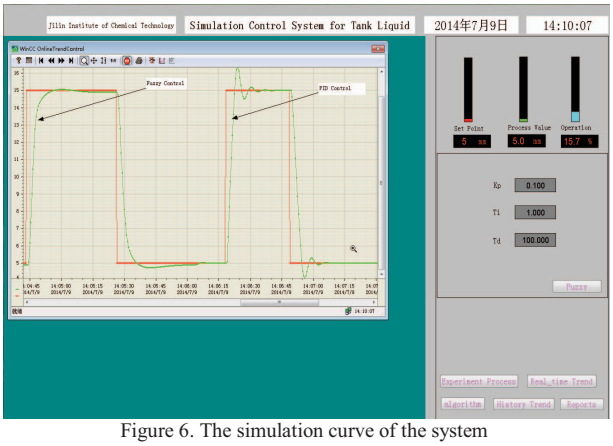


Figure 6. The simulation curve of the system

The water level controlled by the fuzzy controller can response fast and reaches the stable state quickly in comparison with that controlled by the conventional PID controller, so it helps to improve the performance of the system.

CONCLUSION

In the system, the algorithm analysis and calculation are carried out in Matlab. The real time monitoring, operation, animation and data display are operated in configuration environment with Matlab and WinCC data exchanged by OPC technology. This transformation technology can effectively combine advantages of configuration real time monitoring conveniently and reflect the system control algorithm analysis and calculation. At the same time, this technology can also effectively eliminate the reliance of the configuration software on hardware support. The experimental transformation exerts a better demonstration effect.

REFERENCES

[1] SHA Quan. The Communication Between KingView and Matlab.Journal of Shang hai Institute of Technology, Vol 4, pp.286-289, DEC. 2006 (In Chinese).

[2] WANG Wei-quan, MA Yang, LEI Yan-hua, et al. Coupled-tank Liquid-level Fuzzy Control System Based on KingView and Matlab, Industrial Control Computer, Vol 27, pp 23-24, Mar. 2014 (In Chinese)

[3] FENG Jiang-tao. DDE Conmunication Design of KingView and Matlab.Journal of Electric Power, Vol 21, pp.291-293, Mar. 2006 (In Chinese).

[4] HU Yun. The Data Exchange Technology between Matlab and Configuration Software WinCC.Journal of East China Jiaotong University, Vol 25, pp.43-46, Aug. 2008 (In Chinese).

[5] YANG Xiao-wu, LI Jin-song, LI Gan-rong, et al. The Design of Boiler Level Fuzzy Control System Based on MATLAB. Chemical Engineering & Equipment, pp. 11-14, Jan. 2014 (In Chinese)

[6] MA Yang, LEI Yan-hua, BAO Yan. The Design of Intelligent Water Level Experiment Platform based on OPC, Journal of Shenyang Institute of Engineering(Natural Science), Vol 10, pp. 163-166, Apr. 2014 (In Chinese)

[7] LI Er-chao,Liu Wei-rong,LI Wei. An easy real-time online simulation method base on Wincc and Matlab.Experimental Technology and Management,Vol 25, pp.69-71, Mar. 2008 (In Chinese).

Research on Chinese Polar Knowledge Repository and Its Infrastructure

Wenfang Cheng[1,2], Jie Zhang[1,2], Xia Zhang[1], Jiangang Zhu[1], Rui Yang[1,2], Hao Wu[1,2]

[1]Polar Research Institute of China, Shanghai, 200136, China
[2] Chinese National Arctic and Antarctic Data Center, Shanghai, 200136, China
chengwenfang@pric.gov.cn, zhangjie@pric.gov.cn, zhangxia@pric.gov.cn, zhujiangang@pric.gov.cn, yangrui@pric.gov.cn, wuhao@pric.gov.cn

***Abstract*— The knowledge repository of China polar expedition is an open and authoritative online information repository. It publishes information entries relating to Chinese Arctic and Antarctic expeditions' activities. To better make use of the polar resource, this paper designs a basic structure on knowledge query, and presents an automatic query system for data sharing. The system gives the function module and its function description. This article states investigations in related with data source, data process and network structure, and proposes in the end the extension and development of the knowledge repository.**

Keywords- Polar Region; Entry; Knowledge Repository; Data Sharing

I. INTRODUCTION

China polar expedition has accomplished 30 Antarctic voyages and 5 Arctic voyages, and realized fully free data sharing collected on expeditions. However, some basic information and data about polar expeditions are stored respectively in the Chinese Arctic and Antarctic Administration, Polar Research Institute of China, Wuhan University and several participating units in polar expeditions, which causes the shortage, missing and mismatch of the data, bringing to some extent negative effect and losses to the state archives and on-the-spot investigation. For example, information of expedition team per voyage involves the starting off and returning date & time, itinerary, projects, personnel and their positions & roles. Without a comprehensive information repository, the authenticity of the data, on one hand, cannot be verified with overwhelming false information in cyber age and, on the other hand, history data can be found nowhere and therefore fails in guiding future itinerary planning and the decision making in projects [1]. In view of history and current status, this article deals with a China polar expedition knowledge repository (CPEKDB) which is developed by dedicated people and maintained collectively based on the ideas of Wikipedia.

II. MODELING TO AUTOMATIC QUERY SYSTEM FOR DATA SHARING

According to the characteristics of knowledge query and the neural network to information processing, this paper designs a basic structure to knowledge query, proposes a system model of the basic structure and is mapping the system model to the neural network, thus establishes a knowledge query model based on neural network, and discusses the composing of the model. The design of the system model consists of two main functional modules, i.e., data source and data processing, keywords extraction and decision support operation. The main modules consist of the submodules, respectively, as shown in Figure 1. Each module and their function are described in the following section.

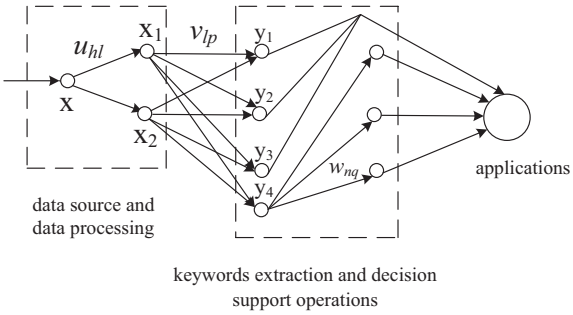


Figure 1. System model of knowledge query

In Figure 1, x denotes the first main module, and x_1, x_2 is two submodules of x. The second main module is the combination of the submodules y_1 .. y_4 denote four subfunctions, respectively. Moreover, the submodule y_4 consists of other three submodules. u_{hl} , v_{lp} and w_{nq} are the weight. The nine functional modules are discussed respectively as follows.

III. DATA SOURCE AND PROCESSING

In the data source and processing module layer, there are three modules that are a data source module and two data processing modules. By adjusting the weights from the data source to the data processing, we can obtain the varying data sources in a small period of time, and input these different component sources to the data processing modules.

As an open knowledge repository in polar expedition, the creation of knowledge in CPEKDB is achieved through allowing creation of new entries by anybody and the edition of existing entries under the policy of editing guide and centralized verification. Country-wide knowledge producers enhance their production efficiency and get the CPEKDB entries updated and developed in a dynamic way through knowledge sharing.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.16

A. Data source

Considering the particularity of the field of polar knowledge, polar expedition organizations and their staffs mainly maintain knowledge in CPEKDB. In the nearly 30 years' history of China polar expeditions, 447 organizations and 3,200 people have participated in the course of China polar expeditions. The development and promotion of CPEKDB would undoubtedly be affected if those organizations and individuals created all the entries. Hence, in the starting period, the majority of the data in the repository are provided and initialized by the Chinese National Arctic and Antarctic Data Center, which means the data source of entries are mainly acquired through some major institutions and referential information in polar expeditions. In choosing the organizations for data collection, such organizations are picked out under the guideline of "centralized and authoritative data" as Polar Expedition Office of the State Oceanic Administration (in charge of the organization and collaboration of polar expedition team set-up), Polar Research Institute of China (the executor of polar expedition), Wuhan University (partial information development unit in polar expedition), Scientific Committee of Antarctic Research (the highest international academic and authoritative unit in Antarctic science, in charge of the draft, launch, progression and collaboration of international Antarctic research plan).

To support the creation of entries, apart from manual collection, sorting and entry, CPEKDB accomplishes the systematic programmed import of knowledge products through Beautiful Soup web crawler [Figure 2]. Beautiful Soup is a Python library designed for quick turnaround projects like screen-scraping. It creates a parse tree for parsed pages that can be used to extract data from HTML, so this library is useful for web scraping — extracting data from websites [2]. Three features make it powerful:

1） Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need.

2） Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8.

3） Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing people to try out different parsing strategies or trade speed for flexibility.

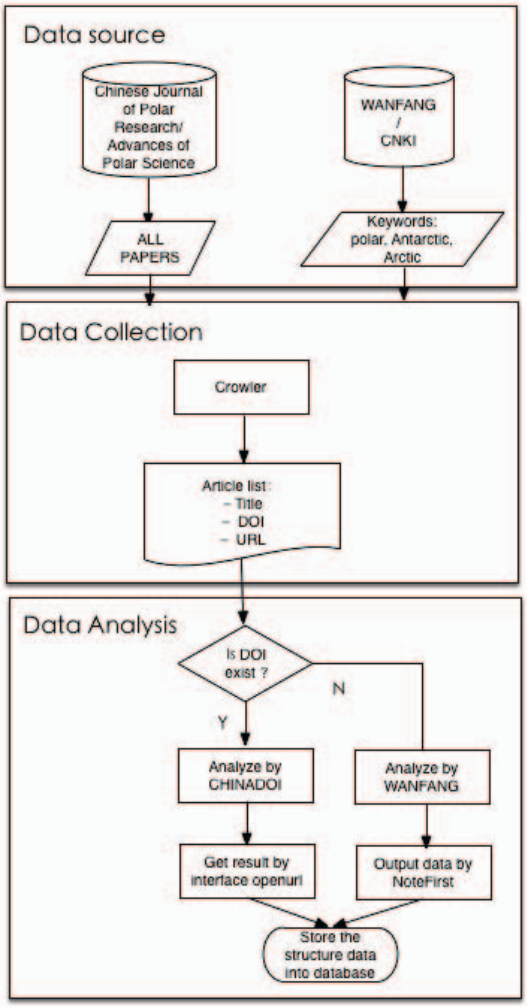


Figure 2. Flow chart of data collection

B. Data processing

Although CPEKDB is similar to Wikipedia and Baidu encyclopedia which allow free creation and edition of entries, CPEKDB is, in a sense, an authoritative and comprehensive knowledge repository which provides reference to scientists, polar information contributors and investigators and therefore it is equipped with rigid data verification process. The verification process is separated in two segments: one is to have all the data verified by system data specialists before its release; and the other is to record and monitor the status of data renewal via version number [figure 3].

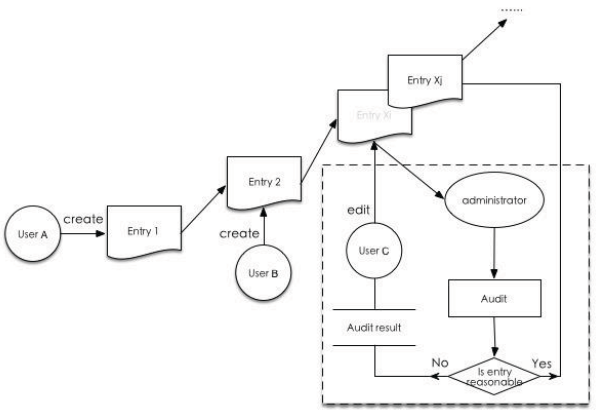


Figure 3. CPEKDB entry release process

As is shown in the dot-line box in figure 3, following the completion of a creation or edition of an entry by any user, the administrator will be prompted to do verification. As a result, the entry can be released directly if it passes the verification, otherwise feedback will be sent back to its user for correction until it can be released. When an entry is released, its user will get detailed information such as its URL, etc.

IV. KEYWORDS EXTRACTION AND DECISION SUPPORT OPERATION

In the keywords extraction and decision support operation module layer, the 4 modules are the infrastructure of system, carry out the most important function for data query and have the expert knowledge. Based on the similarity of keywords and expert knowledge, the important characteristics of data are extracted to be from the data processing layer, and the adjustment to the input weight v_{lp} is: if the input similarity is smaller, we reduce the value of v_{lp}, otherwise increase the value of v_{lp}, where $l = 1,2$, $p = 1,2,3,4$ are the number of modules in the data processing layer and the keywords extraction and decision support operation layer, respectively.

A. Infrastructure of system

If each type of entry in CPEKDB is regarded as a node and super links among entries as directed edges, then CPEKDB can be described as a directed network in which we can learn the internal interconnected characteristics of its knowledge repository through studying the character of the network.

CPEKDB is a directed network from inner point of view and all columns are interconnected. [Formula 1]

Expedition = [basic information, personnels] (1)
Personnel = [basic information, exploration information, institutions, papers, projects, data, samples]
Paper = [basic information, personnel, projects]
Project = [basic information, personnel]
Institution = [basic information, URL, personnel]
Term = [basic information]
Gazetteer = [basic information].
Observation system (exploration stations, exploration vessels, loading platforms and sensors) = [basic information].

B. Category tree

Unlike Wikipedia and Baidu encyclopedia, there are only eight categories in CPEKDB. Based on formula (1), we sort and get the figure of CPEKDB category tree structure [figure 4]. Investigation on degree distribution is the basic content of network study [3]. For any node i, number of links pointing from other nodes is recorded as $K_{i,in}$, which is named in-degree and the number of links pointing from i to other nodes is recorded as $K_{i,out}$, which is named out-degree. Then the degree of node i which is $K_i = K_{i,in} + K_{i,out}$. From the analysis of figure 4, we get the degree distribution status of each node in CPEKDB. The study finds that 1) the distribution trend of K_i is much more affected by $K_{i,out}$. 2) Peak value mainly centralizes among such nodes as personnel, expedition, article and project, which shows that those knowledge plays an important role in the whole Chinese polar expeditions. 3) Through analysis of the relationship among the peak value nodes, we get the rule of WHW (Who, How, What), which can be expressed as the accomplishments (papers) achieved by polar expedition staffs (personnel) through polar expedition activities (expeditions) for certain scientific research tasks (projects). WHW rule shows in hint not only the emphasis government put on scientific activities, but also their great concerns over the significance and value of scientific research, which is known as output. 4) WHW rule is limited within the CPEKDB, only explaining the connectivity and reciprocity among all entries. Due to the lack of interaction and application analysis for polar basic data of scientific research and the polar resource assessment, WHW cannot explain all the scientific value of the whole polar scientific research.

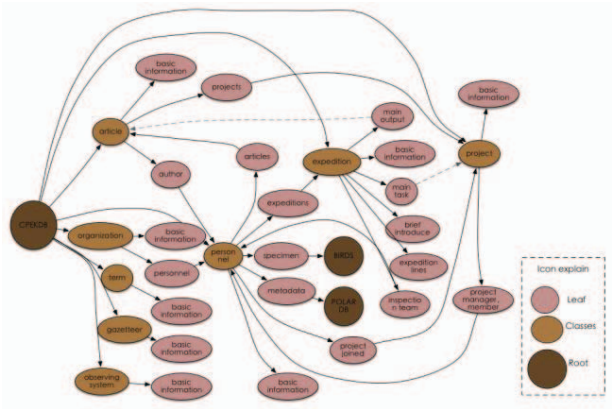


Figure 4. The tree structure of the category

C. Description of four function modules

CPEKDB is divided into four modules by function

characters, they are retrieval module, release module, service module and interface module.

1) Retrieval module

The module is divided into login and retrieval sub-modules.

a) Login sub-module

CPEKDB neither bears any user entity, nor does it store any user information, and all users are from one-stop login on polar expedition information portal (www.polar.gov.cn). The portal adopts OAUTH as access request protocol, which provides a safe, open and simple standard for the request for authorization of the user resources. It allows users to admit third-party applications' access to their private resources stored on certain web site (such as photos, videos, contact lists) without having to provide the user name and password to the third party applications. OAUTH allows the user to provide a token rather than a user name and password to access their data stored at a specific service provider. Each token authorizes a specific web site in a specific period of time (for example, the next 2 hours) to access a particular resource (for example, just may be a set of data). Therefore, OAUTH is safe [4]. There have been many famous OAUTH service providers so far such as GOOGLE, FACEBOOK, DROPBOX, Tencent QQ, Sina Weibo and Douban. Considering a humanized design for login, polar expedition information portal cooperates with Tencent Company to allow users to access CPEKDB resources via their Tencent QQ accounts.

b) Retrieval sub-module

CPEKDB adopts three data retrieval methods which are compound queries, Lucene full-text retrieval and unified retrieval. Compound query, a professional query, fits for structured data, while Lucene full-text retrieval fits for unstructured data [5] such as documents, XML files, Html files, etc. and unified query meets the need for parallel and the integrated export of the query result.

In using compound query, sub-modules extract filter conditions according to user-interested models and the filtered conditions are arranged in reverse order in time and quantity. In handling the query results, three optimized methods are designed:

i) Create partition table for the scattered storage of big data in order to shorten the data query time and improve query efficiency. The establishment of the partition table is based on models in which users are interested.

ii) Set up caching mechanism for the query result in order to avoid I/O bottleneck.

iii) Make simple the sorting of the query results through sorting in two benchmark parameters as 'latest edition time' and 'highest visitation.

2) Release module

Release module is designed to realize the process from the point any registered users create and edit entries to the point of their release. Besides, permission distribution and entry verification are very important functions.

Different permissions are set to achieve different capabilities. The management of permission in CPEKDB system adopts the method of access control based on roles . Role-based access control method is recognized as an effective way to solve the access control of unified resource. Its two significant characteristics are:

i) Reducing the complexity of authorization management and lowering maintenance cost;

ii) Flexibility in supporting system security strategy, and high elasticity in the encapsulation of the components and the change of the application.

According to the functions provided by permission management system, two main sub-modules are included as the authorization and access control sub-modules. Their functions are:

Authorization sub-module: accomplishes permission assignment. Authorization module is mainly used for the management of roles, users and user groups including basic information maintenance, maintenance of relationship among roles, users and user groups, as well as granting all types of permissions to roles, users and user groups respectively.

Access control sub-module: realizes the verification of authority. Access control module accomplishes identity check for users logged in, and validates uses' operation permission during system operation.

3) Service module

The pattern of "from the shallower to the deeper " is introduced into the process of design of this module to solve the problems in the usage of CPEKDB. The underlying layer provides general overview of polar knowledge base through system introduction. The middle layer helps guiding the users in the creation, edition and search of entries via detailed helping documents. The top layer is designed for interactive services, offering services based on online customer service of Tencent QQ platform.

4) Interface module

In the keywords extraction and decision support operation module layer, the fourth module consists of three service interfaces. According to the degree of benefit to data opening, the adjustment to the input weight w_{nq} is: if the degree of benefit is smaller, we reduce the value of w_{nq}, otherwise increase the value of w_{nq}, where $n=1$, $q=1,2,3$ are the number of submodules to connect the fourth modules in the keywords extraction and decision support operation layer.

Three service interfaces of CPEKDB are WebService, Web Widget and entry reference. The benefit of their opening is:

- To avoid duplicate entry of polar expedition basic data in each application system and

ensure the consistency of the basic information.

- Each application system can easily quote basic public data with the minimum price of program modification.
- To improve the correlation between the data of each application system and the basic public data.

V. APPLICATION

Since the release of CPEKDB from 2013, although its users and pages amount is far less than Wikipedia and baidu encyclopedia [6] (table 1), it is the only knowledge repository in Chinese polar areas with the highest authority. In addition, its peculiar English version plays a certain role in promoting international cooperation and knowledge sharing and dissemination in polar field.

Through the interface API provided by CPEKDB, polar information platforms, such as POLARDB, BIRDS and *Polar Exploration Online System*, have successfully called data from CPEKDB, which fully guarantees the consistency of the data. For example, in the process of inputting author information during metadata editing process on POLARDB, all the data is called in real time from CPEKDB, and its asynchronous transmission is realized with Ajax technology. The metadata information in POLARDB will be triggered and updated automatically when revision happens to the entries in CPEKDB. BIRDS's news module refers to CPEKDB dictionary application to read the data from BIRDS and as long as these entries are included in the BIRDS classes, reference link with CPEKDB library can be set up.

Table 1 Comparisons among CPEKDB, Wikipedia and Baidu Encyclopedia

Items	Wikipedia	Baidu Encyclopedia	CPEKDB
Pages quantity	≈ 30 million	≈6.27 million	≈45 thousand
Number of users	365 million	3.462 million	≈3 thousand
Release time	2001	2006	2013
Language	287 types	Chinese, English	Chinese, English
Commercial sites	No	Yes	Yes
Censorship	No	Yes	Yes
Field	All	All	Antarctic & Arctic

VI. CONCLUSION

In order to increase the openness of the polar expedition data, CPEKDB also can be expanded in categories and be added with more historical exploration documents, archives, exploration reports, achievement reports, etc., integrating into an aggregation of polar exploration library and polar research knowledge base so as to provide users with more comprehensive, objective and timely information service.

ACKNOWLEDGMENT

This research is funded by projects: basic condition platform of Ministry of Science and Technology – Data Sharing Infrastructure of Earth System Science (2005DKA32300), China polar science strategy research fund project (20120106), and the State Oceanic Administration polar science key lab open research fund (KP201110).

REFERENCES

[1] Zhang xia, Cheng Shaohua, Zhu Jiangang. The research and construction of the Chinese polar science database system|. The organization of data resource and research of relationship between metadata and dataset[J]. Chinese journal of polar research.2002, 14(1): 62-72.

[2] Beautiful Soup website http://www.crummy.com/software/BeautifulSoup/

[3] Fei Zhao , Tao Zhou , Liang Zhang , et al.: Journal of university of electronic science and technology of China , 2010, 39(3): 322.

[4] Hammer E. OAuth 2.0 and the Road to Hell [J]. 2012.

[5] Wikipedia: http://zh.wikipedia.org/wiki/%E6%AD%A3%E5%88%99%E8%A1%A8%E8%BE%BE%E5%BC%8F

[6] Wenfang Cheng , Jie Zhang , Mingyi Xia , et al.: *System design and implementation of a resource-sharing platform for polar samples* [J]. Chinese journal of polar research, 2013, 25(2): 185-196.

A load balancing scheme for distributed key-value caching system in cloud environment

Tao Wang, Xin Lv*, Fang Yang, Wenhuan Zhou, Rongzhi Qi
College of Computer and Information
Hohai University, HHU
Nanjing, China
e-mail:yuecjn@163.com
*Corresponding author: lvxin.gs@163.com

HuaiZhi Su
College of Water Conservancy and Hydropower Engineering
Hohai University, HHU
Nanjing, China

Abstract—Distributed key-value caching system has been deployed in many kinds of clouds. The effect of load balancing between each node is crucial in key-value caching system in clouds. Invalidation is a feature of the data in the key-value caching system, and making some cache invalid is efficient instead of adjusting the load location when load balancing. It is worth studying that how to utilize the feature reasonable to reach maximum load balancing. Aiming at this point, a new cache-invalidation-scope model in key-value caching system is proposed. Combined with greedy algorithm, the scheme provides a better load balancing algorithm for different load cases (CLB). CLB algorithm utilizes entropy and the scope of invalid cache invalid as the evaluation basis of load balancing effect. Compared with existing algorithm, CLB load balancing algorithm lifts up the performance of key-value caching system more than one times.

Keywords-Cloud Computing, key-value system, Load balancing, Cache Invalidation

I. INTRODUCTION

Cloud computing is a calculation and business mode providing users with elastic scalable resources by virtualizing them [1-2]. Cloud computing provides all kinds of resources and information on the Internet. The flexibility is the key requirement in these applications of cloud computing. The distribution of cache data should be adjusted according to the dynamic changes of network and application in key-value caching system and the performance of the system is affected by the load on the cache and bandwidth of the system. How to adjust the cache data dynamically to minimize the scope of invalid cache invalid with minimal cost and make the smallest uniform of cache load is crucial in the whole application system.

The static load distribution is used in many present cache systems, such as Memcache and Voldemort which use consistent hash distributed algorithm [3-4]. These algorithms have resolved the problem of redistributing key values, however, they are unable to adjust the distribution of key values dynamically. In case of hot issues or uneven bandwidth, static load distribution can't meet the requirement of dynamic. So dynamical hash algorithm is proposed to satisfy this requirement in the cache system. In the new algorithm, hash values will be adapted for the appropriate node and the cache will be reconstructed which lead to some invalidation of cache instead of migrating load. It is worth studying that how to narrow the scope of invalid cache and reduce the time of cache reconstruction. In this paper, a cache-invalidation-scope model based on the improved consistent hash algorithm is put forward.

II. RELATED WORK

In the research of elastic distributed key-value system, much achievement has been made. Ala etc al. discussed the impact of data migration on performance in key-value database, and then they proposed an elastic load balancing scheme based on multiple linear regression model and a model to compute the time of transferring load [5]. Qin etc al. advanced a kind of elastic feedback evaluation model of Voldemort and did a thorough research to evaluation model of elastic distributed key-value system [6]. Jose etc al. proposed a scheme based on a variety of mixed data communication for Memcache distributed key-value system which can ensure the reliability of data communication in case of large-scale nodes in the distributed environment [7]. Chiu etc al. proposed an elasticity key-value system based on extended hash algorithm without compromising elasticity [8].

As above mentioned, so many solutions has been proposed to satisfy the requirement of elasticity, however, there are still some shortcomings such as lacking of an feedback controller that can intelligently acquire and release resources. In this paper, the scheme for solving the problem of load balance between nodes is designed.

III. PRELIMINARIES

This paper intends to build an elastic key-value caching system in cloud environment. The key point to the system is elasticity and cloud-based. Figure 1 shows the basic control framework of our system. Control structure is the key which is responsible for the elastic demand of the system. CloudStack is used to satisfy the elastic and both database and cache nodes are virtual machine of CloudStack [9]. The tools adopted in each application of this system is selected in consideration of open source and usability (list in table 1). Our framework consists of the following six major components.

CloudStack: An open source cloud computing platform to provide virtual nodes, including the cache nodes and database nodes.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.17

Controller: It is a component that records the service status of the whole system (CPU utilization, load size, bandwidth utilization, etc.). The controller decides to get data from which cache node according to the request. And the controller executes load balancing in a section of interval according to the load of each node.

YCSB: An open source framework which is called benchmark tool and allows creating various load scenarios [10].

Sensor: A component that detects database load conditions and operation parameters (CPU utilization, load size, bandwidth utilization, etc.).

Memcache: A distributed key caching system which stores cache.

Mysql: A distributed relational database which stores applications data.

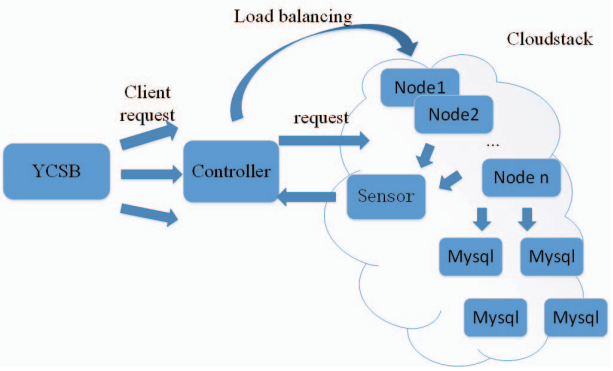


Figure 1. System Framework

IV. MODEL AND ALGORITHM

A. Consistent Hash

Firstly, hash value is calculated for and each node ranging from 0 to 2^{32}, and the algorithm assigns them to the correspond one. Finally, the algorithm searches the nearest server in clockwise direction, then stores the data on the server. If the hash value beyond 2^{32}, then the value will be stored on the first server. Consistency hash algorithm is static load balancing algorithm and the range of hash value to each node is fixed. Adding node is the only way to share the load when the load is imbalance or the load of one node is too heavy.

B. Cache Invalidation and Cache Reconstruction

The cache ranging from A to B will be invalid if the hash value of a node reduced from B to A. The controller will get data from the node when there comes the request for the data which is invalid. Then the node sends the request to database and save the value in the node which is called cache reconstruction. Cache reconstruction is costly and how to make the range of invalid cache minimum is particularly important.

C. Improved Consistent Hash Algorithm

Data is distributed from the first node in consistent hash algorithm, actually each node is the equal in a cloud environment. As shown in figure 2, a cyclic group of nodes is organized from the first virtual machine to the last of the virtual machine, and hash value can be assigned from any virtual machine. Table 2 shows that the controller keeps a table of hash value of each node and the hash value of each key can be changed in the system. The best load balancing can be achieved through set an appropriate beginning position.

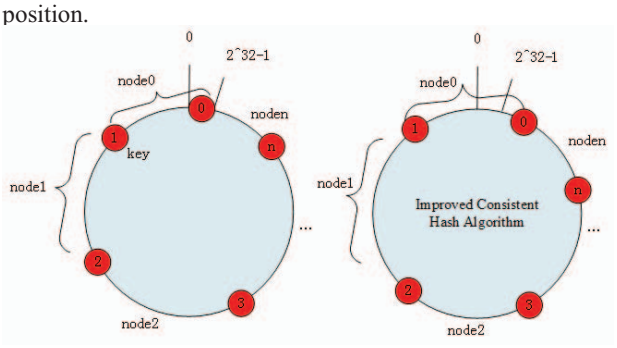


Figure 2. consistent hash and improved consistent hash

TABLE I. HASH VALUE OF EACH NODE

Node	*Hash Value*
0	*key0——key1*
1	*Key1——key2*
2	*Key2——key3*
.....	.....
n	*keyn——key0*

D. Load Entropy and Cache-invalidation-scope Model

Suppose there are n nodes and the load of each node is $l_i(\mathrm{i}=1,2.\ldots\ldots\mathrm{n})$. p_i represents the ratio of load for each node. The load entropy for the whole system is according to the Shannon's information theory [11].

$$p_i = l_i / \sum_{i=1}^{n} l_i \tag{1}$$

$$H(\mathrm{P}) = -\sum_{i=1}^{n} p_i \bullet \log p_i \tag{2}$$

The purpose of load balancing is to calculate a group of solutions of l_i and to make the entropy of the system largest.

Suppose there are n nodes and the load of each node is $l_i(\mathrm{i}=1,2.\ldots\ldots\mathrm{n})$ at beginning. After calculation the load of each node is $t_i(\mathrm{i}=1,2.\ldots\ldots\mathrm{n})$. Suppose the beginning index of each node is $a_i(\mathrm{i}=1,2.\ldots\ldots\mathrm{n})$ and after calculation beginning index of each node is $b_i(\mathrm{i}=1,2.\ldots\ldots\mathrm{n})$. Figure 3 shows the load balance of system before and after load balancing if hash distribution of nodes is the consistent hash.

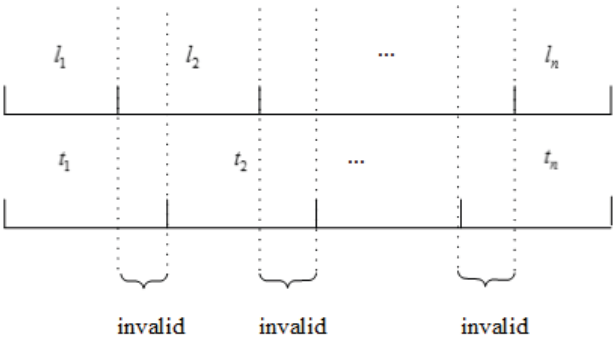


Figure 3. the length of the load for every node before and after load balancing

To node 1, cache data is valid. To node 2, the scope occupied by node 1 is invalid. Obviously the scope of each node occupied by other node is invalid. But the starting point of the node 0 can't be changed. Figure 4 shows load balance of system before and after load balancing if hash distribution of nodes is the improved consistent hash. Notice that the starting position of first node is changeable.

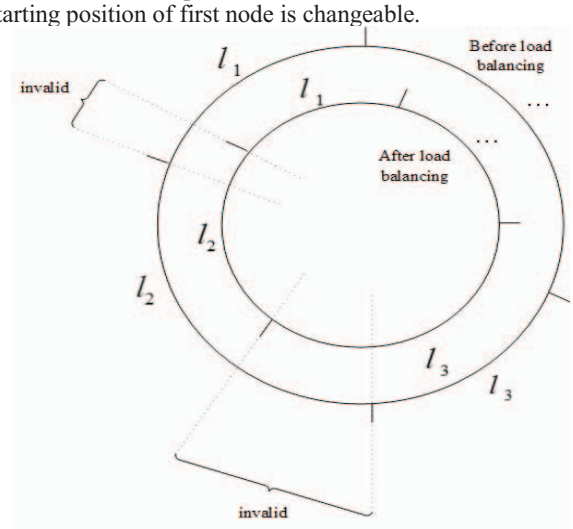


Figure 4. the distribution of hash value for every node before and after load balancing

Equation (3) shows how to calculate the scope of invalid cache according to geometrical knowledge. $Len()$ represents the length of the overlap between two lines.

$$L=\sum_{i=0}^{n}(l_i-Len((a_i,a_i+l_i),(b_i,b_i+t_i))) \tag{3}$$

$$Len()=\begin{cases}0, a_i>b_i+t_i \text{ or } b_i>a_i+l_i\\ l_i, b_i+t_i>a_i+l_i \text{ and } b_i<a_i\\ t_i, a_i+l_i>b_i+t_i \text{ and } a_i<b_i\\ b_i+t_i-a_i, a_i<b_i+t_i<a_i+l_i \text{ and } b_i<a_i\\ a_i+l_i-b_i, b_i<a_i+l_i<b_i+t_i \text{ and } b_i>a_i\end{cases}. \tag{4}$$

Let x be the starting position of node 0 after load balancing. Equation (5) shows how to calculate b_i.

$$b_i=x+\sum_{j=0}^{i-1}t_j\,(0<i\le n) \tag{5}$$

Equation (6) can be obtained by equation (5) and equation (3).

$$L=\sum_{i=0}^{n}(l_i-Len((a_i,a_i+l_i),(x+\sum_{j=0}^{i-1}t_j,x+\sum_{j=0}^{i}t_j)))(0<i\le n) \tag{6}$$

So L is a piecewise linear function about x in (6). To get the minimum scope of invalid cache, x should be solved to make L minimum. x can not be achieved by traversal because the scope of it is very big $(0\sim 2^{32}-1)$. Then a method combined with the characteristics of linear function and greedy algorithm is presented to solve x.

The function about the invalid cache of node 0 can be divided into 5 segments. As the same as node 0, the invalid cache of the other node is also divided into 5 segments. At last, L is a function with 5^n segments and every function is liner about x which cannot be solved if n is big. According to the characteristics of linear function, the minimum value of each section function is the minimum value of two endpoints. Each endpoint can be solved by a_i, b_i, t_i and the total number of endpoints is $5*n$. Let $m_i(0<=i<5n)$ be the coordinate of endpoints. Equation (7) shows the minimum invalid scope.

$$L_{\min}=\min_{k=0}^{5n-1}(\sum_{i=0}^{n}(l_i-Len)(0<i\le n) \tag{7}$$

$$Len=Len((a_i,a_i+l_i),(m_k+\sum_{j=0}^{i-1}t_j,m_k+\sum_{j=0}^{i}t_j)). \tag{8}$$

E. Load Balancing Algorithm

Let $l_i(i=1,2.....n)$ be the load of each node. The starting position of node 0 is a. The goal of load balancing is to get a set of load $t_i(i=1,2.....n)$ and the starting position of node 0 x which can make the scope of invalid cache minimum. This is a multi-objective optimization problem which is a NP problem. An algorithm with greedy algorithm is presented to solve this problem below.

First step: Let threshold ε be the percentage of scope which is invalid. Scope of cache $(0\sim 2^{32}-1)$ is divided into s segments and the length of every segment is $2^{32}/s$.

Second step: The biggest load of l_i is decreased by $2^{32}/s$ and the smallest load of l_i is increased by $2^{32}/s$.

Third step: go back to second step if $L_{\min}/2^{32}$ less than ε, otherwise the iteration is over.

The core of the algorithm is to make the biggest load share some load to the smallest load. According to the theory

of entropy, the entropy of P will increase after the second step and the cost of adjustment is smallest which meets the requirement of load balancing.

V. EXPERIMENT

A. Setup

Table 3 shows the hardware and software of the system. Twenty virtual nodes are be created in CloudStack platform.

TABLE II. HARDWARE AND SOFTWARE

hardware	***Processor***	***4***
	CPU Cores	***4***
	Model name	***I7***
	Memory	***4*8G***
	Frequence	***3.2GHZ***
software	***Physical system***	***Ubuntu10.04***
	Virtual system	***Ubuntu10.04***
	Key-valu system	***Mencached1.4.17***
	Database	***Mysql***

Table 4 shows the setup in YCSB in our system.

TABLE III. YCSB SETUP

Parameters	***Values***
Number of records inserted in warm-up	***100000***
Write Percentage (%)	***10%***
Read (%)	***90%***
Showing Result Interval (Sec)	***60(s)***
Throughput (Ops/Sec)	***4000(ops/s)***

B. First Experiment

In first experiment CLB algorithm is compared with consistent hash algorithm and CARP algorithm to verify the validation of the load balancing. Figure 5 shows the result of the first experiment.

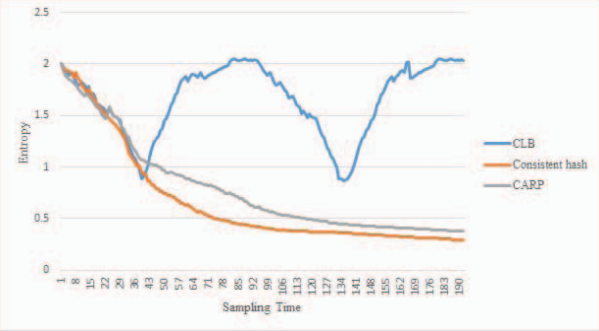


Figure 5. The entropy of system in different algorithm

Figure 5 illustrates the comparison of load entropy between different algorithms. Compared with other two algorithms, the effect of CLS algorithm is better in the same load test. The load entropy of consistent hash algorithm and CARP algorithm decreases over time without the effect of elastic load balancing. Load will be balanced when the load entropy of system is reduced to a certain point (threshold). Initially, the system is running under the condition of unbalanced load and the load entropy decrease over time. Then the load entropy is increased after the first load balancing. Finally, the system balances load in a section of interval. CLS load balancing algorithm makes entropy of the system in fluctuated state. The average entropy of CLS load balancing algorithm is twice of the average entropy of consistent hash algorithm and CARP algorithm.

In order to verify the validity of the CLB algorithm, Figure 6 shows the time needed for each load balancing and time intervals between load balancing.

Figure 6. The time for load balancing and the time interval between load balancing

The time for load balancing is long and the time interval between load balancing is short in the beginning of the program. As the program runs, the time for load balancing gradually become shorter and time interval between load balancing become longer. Although the CLB algorithm will make the system loss some performance at the beginning of the program, dynamic load balancing losses reduced gradually with the running of the program. It can be concluded that the CLB algorithm.

C. Second Experiment

In second experiment three thresholds are tested to verify the effect of load balancing. Figure 6 shows the result of the second experiment.

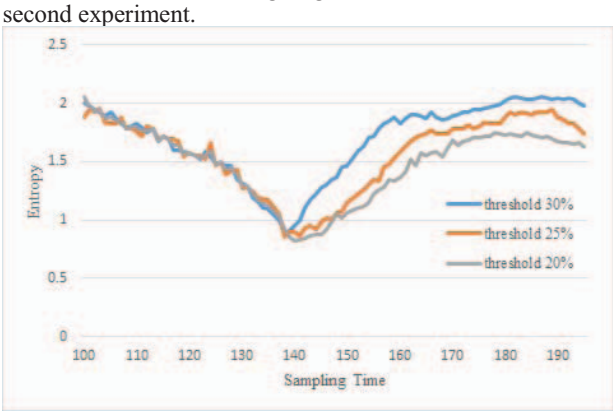


Figure 7. Load balancing of different threshold

Figure 7 shows the load entropy of system in different threshold. Along with the augment of the threshold ε, the tolerable of invalid scope turn bigger, meanwhile, the load

entropy become smaller after load balancing, however the execution time of load balancing is larger than usual. So threshold can be used to adjust the effect of load balancing.

VI. CONCLUSION

In this paper, a new scheme of load balancing for key-value cache system in cloud environment is proposed in consideration the effect of load balancing and the scope of invalid cache. Cache-invalidation-scope model is established to improve the effect of load balancing. The percentage of invalid cache and load entropy are utilized to quantify the effect of load balancing. The most contribution of this paper is to improve the existing consistent hash algorithm and make it suitable for load balancing, besides, a cache-invalidation-scope model is proposed providing a favorable load balancing scheme. The results of the experiment show that the proposed algorithm lifts up the performance of key-value caching system more than one times.

In further research, we are planning to consider of adding-deleting node dedicating to improve the effect of load balancing.

VII. ACKNOWLEDGEMENT

This paper is supported by National Natural Science Foundation of China: “Research on Trusted Technologies for The Terminals in The Distributed Network Environment” (Grant No. 60903018), “Research on the Security Technologies for Cloud Computing Platform” (Grant No. 61272543), and “The National Twelfth Five-Year Key Technology Research and Development Program of the Ministry of Science and Technology of China” (Grant No. 2013BAB06B04), “Key Technology Project of China Huaneng Group” (Grant No.HNKJ13-H17-04).

REFERENCES

[1] Chen K, Zheng W M. Cloud computing: system instances and current research[J]. Journal of Software, 2009, 20(5): 1337-1348..

[2] L.Wang, R.Ranjan, J. Chen et al. Cloud computing: Methodology, systems and applications[M]. Boca Raton: CRC Press, 2012.

[3] Devine R. Design and implementation of DDH: A distributed dynamic hashing algorithm[M]//Foundations of Data Organization and Algorithms. Springer Berlin Heidelberg, 1993: 101-114.

[4] Karger D, Lehman E, Leighton T, et al. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web[C]//Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. ACM, 1997: 654-663.

[5] Arman A, Al-Shishtawy A, Vlassov V. Elasticity Controller for Cloud-Based Key-Value Stores[C]//Proceedings of the 2012 IEEE 18th International Conference on Parallel and Distributed Systems. IEEE Computer Society, 2012: 268-275.

[6] Qin X, Wang W, Zhang W, et al. Elasticat: A load rebalancing framework for cloud-based key-value stores[C]//High Performance Computing (HiPC), 2012 19th International Conference on. IEEE, 2012: 1-10.

[7] Jose J, Subramoni H, Kandalla K, et al. Scalable memcached design for infiniband clusters using hybrid transports[C]//Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on. IEEE, 2012: 236-243.

[8] Chiu D, Shetty A, Agrawal G. Evaluating and Optimizing Indexing Schemes for a Cloud-based Elastic Key-Value Store[C]//Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on. IEEE, 2011: 362-371.

[9] Baun C, Kunze M, Nimis J, et al. Open source cloud stack[M]//Cloud Computing. Springer Berlin Heidelberg, 2011: 49-62.

[10] Cooper B F, Silberstein A, Tam E, et al. Benchmarking cloud serving systems with YCSB[C]//Proceedings of the 1st ACM symposium on Cloud computing. ACM, 2010: 143-154.

[11] Cover T M, Thomas J A. Elements of information theory[M]. John Wiley & Sons, 2012.

A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing

Chun-Yan LIU
Department of Information Engineering
Wuhan university of technology HuaXia college
Wuhan, China
liuchunyan210@126.com

Cheng-Ming ZOU,Pei WU
School of Computer Science and Technology
Wuhan University of Technology
Wuhan, China
zoucm@hotmail.com

***Abstract*—An efficient approach to task scheduling algorithm remains a long-standing challenge in cloud computing. In spite of the various scheduling algorithms proposed for cloud environment, those are mostly improvements based on one algorithm.And it's easy to overlook limitations of the algorithm itself. Aiming at characteristics of task scheduling in cloud environment, this paper proposes a task scheduling algorithm based on genetic-ant colony algorithm. We take the advantage of strong positive feedback of ant colony optimization (ACO) on convergence rate of the algorithm into account.But the choice of the initial pheromone has a crucial impact on the convergence rate. The algorithm makes use of the global search ability of genetic algorithm to solve the optimal solution quickly, and then converts it into the initial pheromone of ACO. The simulation experiments show that under the same conditions, this algorithm overweighs genetic algorithm and ACO, even has efficiency advantage in large-scale environments. It is an efficient task scheduling algorithm in the cloud computing environment.**

Keywords-cloud computing; task scheduling; genetic algorithm; ant colony optimization

I. INTRODUCTION

Cloud computing is an emerging technology where information technology resources are provisioned to users in a set of a unified computing resources on a pay per use basis[1].Cloud computing using virtual technology parts the huge computing task into a number of small tasks through the network, which are next allocated into the huge system consisting of multiple servers by some allocation methods, then returning the results to the user after computing[2-5]. Thus, key points and difficulties of cloud computing are how to reasonably carry out the task scheduling and resource allocation. It is an important topic that designs an excellent performance scheduling algorithm to improve quality of service in cloud environment.

Task scheduling in cloud computing has attracted great attentions. Many researchers have proposed different scheduling algorithms which run under the cloud computing environment. However, most task scheduling algorithms that have been proposed are based on an improved algorithm. Here, we review the most relevant research works done in the literature for scheduling algorithm.A bandwidth-aware algorithm is proposed for dicisible task scheduling in could computing environment ,which is on the basis of the optimized allocation scheme[6]. the Jianfeng LI,Jian PENG design a task scheduling algorithm based on double fitness genetic algorithm. It improves the efficiency of cloud computing, through setting two optimization goals. One is total task completion time and the other one is average task completion time[7]. Liangliang FENG, Tao Zhang, Zhen-hong Jia, Xiao-yan Xia, Xi-zhong Qin propose a task scheduling algorithm based on improved particle swarms, considering the total task completion time and the total cost to complete tasks[8].The pricing and peak aware scheduling algorithm for cloud computing is proposed in 2012,which demonstrated feasibility of interactions between distrebutors and one of their heavy use customers in a smart grid environment[9].Xia-yu Hua, Jun Zheng, Wen-xin Hu introduce a cloud computing resource allocation method based on ACO, which is fully taking inherent properties of computing resources and the node's load into account, in order to predict the execution speed[10]. Jing-zhao ZHANG,Tao JIANG propose an improved adaptive genetic algorithm, to a certain extent, solve the traditional genetic algorithm "premature" issue, and accelerate the convergence rate[11]. Zong-bin ZHU,Zhong-jun DU propose improvements GA cloud computing task scheduling algorithm, considering two elements, the time and cost of the task scheduling[12]. Jian-ping LUO,Xia LI,Min-rong CHEN address the problem of resource scheduling based on shuffled frog leaping algorithm, combining with the tasks and resources, and propose two types of network coding structures on the basis of which make merits of individual choice according to the value of Qos[13]. Ming-hai XU,Yuan ZI propose a network selection based on ant colony optimization, where the feedback mechanism is introduced into the network selection algorithm to choose the right path using pheromone concentration[14]. What's more,there has an improved differential evolution algorithm based on the proposed cost and time models on cloud computing environment ,and this algoruthm can optimize task scheduling and resource allocation[15].Analyzing the above task scheduling algorithms, we can find that it is easy to overlook the inherent limitations of algorithm itself, as optimization ability of genetic algorithm at late stage is poor and prone to premature degradation, where colony algorithm's searching is inefficient. Combining the

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.18

advantages and disadvantages of various intelligent algorithms and taking global search capability of genetic algorithm and high accuracy of ACO into account, the paper proposes a genetic-ant colony scheduling algorithm (GA-ACO algorithm) integrating the global search capability of genetic algorithm and high accuracy of ACO, which shows its good performance through simulation experiments.

II. PROBLEM OF TASK SCHEDULING IN CLOUD COMPUTING

In cloud computing, task scheduling policy directly affects the efficiency of the user's tasks and the efficient usage of resources under the cloud environment. Hence, how to achieve optimal allocation of user's tasks is the key issue of task scheduling in cloud computing. The process of task scheduling under the cloud environment is as follows.Firstly, tasks and resources will be mapped according to current task and resource information in accordance with certain strategy. Then follow the mapping between the resources allocated to the implementation of the task to ensure the efficiency of the task and the quality of service requirements of users. Finally the summary of the results is excuted to the submitting user.

The current cloud computing environment is mostly built according to MapReduce programming model, which is an efficient task scheduling model especially for the generation and processing of large data sets[16-20]. The specific implementation process was shown in Fig.1.

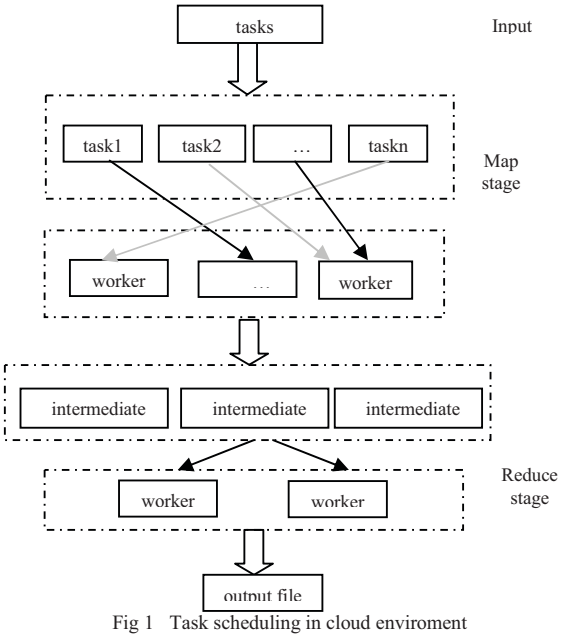


Fig 1 Task scheduling in cloud enviroment

Fig.1 illustrates the proposed model for task scheduling under cloud environment which consists of two stages, namely, the map and the reduction. The core idea of MapReduce is to divide a parallel processing task execution stage into Map and Reduce stages. In the Map stage user's task is divided into smaller sub-M tasks by MapReduce function, which allocates them to multiple workers, and then there will output an intermediate file . In the reduce stage ,results will be outputted after the treatment of the map pooled analysis of the pre-result

III. TASK SCHEDULING ALGORITHM BASED ON GENETIC-ANT COLONY ALGORITHM IN CLOUD COMPUTING

This study aims at task scheduling problem in the cloud computing.In order to get the best result of task scheduling and takes less time,an integrate of the genetic algorithm and ACO effectively can be made r,eferring the MapReduce model.Accordingly, there proposes a GA-ACO algorithm.

A. Design Ideas

The main idea of the GA-ACO algorithm are as follws. In the early stage of task scheduling ,it takes advantage of genetic algorithm's global search ability, and forms chromosome by indirect encoding.Then choose reciprocal of task completion time as the fitness function.After selection, crossover and mutation, generate the optimal solution and convert this solution into ACO's initial pheromone, and form optimal solution of task scheduling through the feature of positive feedback and efficiency.

B. Rules of Genetic Algorithm

1) Chromosome Encoding and Decoding

To solve the problem of the task scheduling under cloud environment, we should encode scheduling scheme into chromosomes where each chromosome represents a particular scheduling scheme. This paper applies an indirect encoding method. Specific operation is as follows: Each task occupying the resource is encoded. The length of the chromosome equals to the total number of sub-tasks. The number of each bit position represents the gene sub-tasks' number and the value of Gene-bit represents the number of the occupied resource.

The amount of sub-tasks is calculated using (1):

$$subtaskNum = \sum_{t=1}^{m} taskNum(t) \qquad (1)$$

Where:

m = number of tasks

t = the order of task

$taskNum(t)$ = the number of sub-tasks assigned to task t

For example, it assumes that there are 3 tasks ,then m=3.And 3 resources means n=3 and 3 tasks are divided into 3,4,2 sub-tasks, meaning that $taskNum(1)$=3, $taskNum(2)$=4, $taskNum(3)$=2,so $subtaskNum$=9.It means that the length of the chromosome is 9. Setting the value scope of the gene to be (1, 3). Applying method of indirect encoding to generate a set of chromosomes: {2, 3, 1, 2, 3, 1, 2, 1, 1}. The first sub-task is assigned to the second resource, the second sub-task is assigned to the third resource, and so on. Then decode these chromosomes, and obtain the distribution of various resources on tasks, w1:{3,6,8,9} w2:{1,4,7} w3:{2,5}

2) The Objective Function and the Fitness Function

Calculate the execution time to perform tasks for each resource using decoded sequence and ETC matrix. It follows

that the total time to complete the task of resource scheduling,as in:

$$F(x) = \max_{r=1}^{n} \sum_{i=1}^{w} work(r,i) \quad (2)$$

Where:

Work(r, i) = the time spent by the resource *r* performing sub-task *i* which is on this resource.

w= the quantity of the sub-tasks assigned to the resource.

Equation (2) is defined as objective function.

The fitness function is used to evaluate chromosomes' pros and cons. The value of the function is bigger, then the chromosomes' survivability is stronger and the function's solution is better. Since the value of the fitness function is the reciprocal of objective function, and the time is shorter, and the fitness value is bigger, and the probability of being selected is larger.

The fitness function is defined as:

$$f(x) = 1/F(x) \quad (3)$$

3) Genetic Manipulation

Genetic manipulation of genetic algorithm including selection, crossover and mutation. And through these operations it continues to generate new individuals so as to search out the optimal solution.

a. selection

Probability of selection for each individual is calculated based on the value of fitness function. Equation (4) illustrates how the probability of selection is computed:

$$P(i) = \frac{f(i)}{\sum_{j=1}^{SCALE} f(j)} \quad (4)$$

b. crossover

This paper chooses adaptive crossover methods. Larger crossover probability exchange some bit between individuals, so that it can avoid the occurrence of premature. In the latter part of the algorithm, as crossover probability decreases, it is easier to generate new good individual and accelerate the convergence rate.

c. mutation

This paper adopts single point mutation to change some individual bits in groups for smaller probability, like "1"to "0","0"to "1".

In actual operation, it eliminates the new individuals whose value of fitness function is less than the average value after several recursive iterations, and gets the optimal solution of certain groups as a basis for obtaining a pheromone ACO.

C. Exact Solution based on ACO

1) Combining Genetic Algorithm and ACO

Evaluate the chromosome population according that successive five generations' evolutionary rates are small.Then the genetic algorithm can be terminated and enter the ACO.

When genetic algorithm is terminated, sort individuals in the population according to the size of the fitness function values, from which the top 10% of individuals are selected as an optimization solution and converted to initial pheromone. The specific initialization rule is shown in (5):

$$T_i^G(0) = \rho S_n \quad (5)$$

Where:

ρ = self set constant

S_n = as the genetic algorithm optimization solution

Through the operation of genetic algorithm we can obtain the distribution of pheromone. The initial value of resource pheromone is set in (6):

$$T_i(0) = r_i + T_i^G(0) \quad (6)$$

Where:

r_i = processing capacity of the resource

$T_i^G(0)$ =the pheromone value transformed from the optimal solution when the current Genetic Algorithms is terminated.

2) Path Selection

Each ant determines the probability of the next resource according to the information of the current resource.

$$P_{k,}(i,j) = \frac{[T_j(t)]^{\alpha}[\eta_j]^{\beta}}{\sum_{\in U}([T_u(t)]^{\alpha}[\eta_u]^{\beta})} \quad (7)$$

Where:

$T_j(t)$ = the value of the pheromone in resource j at the moment t

η_j = processing capacity of the resource j

α or β = the importance of the pheromone

3) Update Pheromone

By comparing the performance of ACO, the method of global updating the pheromone can improve the convergence efficiency. That is, when an ant successfully completes a resource selection , the pheromone will change. Pheromone update rule is shown in (8):

$$T_j^{new} = \rho T_j^{old} + \Delta T_j \quad (8)$$

Where:

ρ = termination condition of ACO

When the cycle counter N reaches the maximum number of iteration's range, the current value is the optimal scheduling scheme, and then the ACO terminates.

IV. RESULTS AND DISCUSSION

In order to verify the feasibility and effectiveness of GA-ACO algorithm, we need to simulate it from the performance of its task scheduling. Subsequently, experimental results are compared to genetic algorithm and ACO under the same environment.

A. Parameter Settings

For advantages of each searching and solving by genetic algorithm and ACO with repeated experiments, various numbers of scenarios with different parameters values are taken into consideration during simulation. Table 1 summarizes the simulation parameters used in these experiments.

TABLE I. THE PARAMETERS SETTINGS

Algorithm	Parameter	Value
GA	Number of population	100
	Crossover rate	0.6
	Mutation rate	0.1
ACO	Number of ants	100
	α	1
	β	1
	ρ	0.3
	a	1
	b	0.8

B. *Experimental Results and Analysis*

All algorithms are implemented and tested on the platform of clouds IM simulator. In the experiments carried out in this study, Fig.2 depicts the results under the same environment where the number of tasks are 50, and compares the success rate of searching the optimal solution and the number of iterations among GA-ACO algorithm, GA and ACO.

From the Fig.2, it can be noted that success rate of searching the optimal solution reaches up to 98% when the number of iteration is 28 in GA-ACO algorithm experiment, but when the number of iteration is 50, GA algorithm's rate reaches up to 63% and ACO algorithm's rate reaches up to 95%.

From the Fig.2, it can also be concluded that GA-ACO algorithm requires less iterations to find the optimal solution. And it's solving efficiency is better than GA and ACO significantly. This is because the GA-ACO algorithm converts several optimization solution generated by GA into pheromone of ACO, greatly shortening the time to collect pheromone.

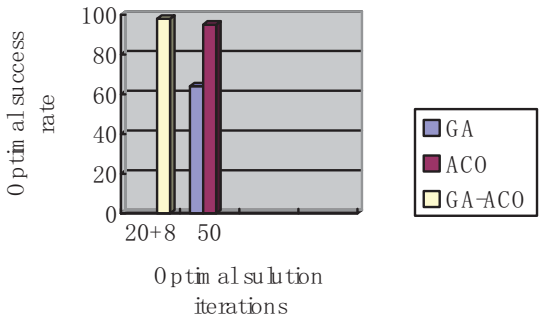


Fig.2 Comparison of optimal success rate

In another experiment, we used CA-ACO algorithm, GA and ACO respectively to test the performance of schedule tasks. It is sampled data once in each task number 50, 100, 200,300,400,500. The simulation effect diagram by the three Algorithms is as follow.

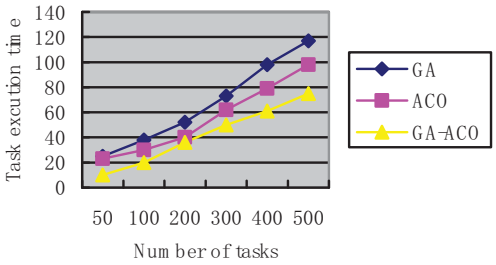


Fig 3 Comparison of task execution time

As we can see from Figure 3, it illustrates the observation of the execution time with the increasing number of tasks for each algorithm. From the figure, it is clear that when the number of tasks is smaller, the resources are more adequate in cloud environment. As for the task execution time ,all these three algorithms relatively cost little.And among the three algorithms, GA-ACO algorithm is slightly better than ACO and GA, though the gap is not obvious. With increasing number of tasks, the increasing trend of the execution time spent by CA-ACO is significantly less than the other two algorithms. What's more, the performance's improvement is obvious. This is main due to that the increasing number of tasks results in a high load for each algorithm which leads to extend the execution time. However, at a larger number of tasks, GA-ACO algorithm makes use of its own advantages, avoiding the defects of GA's local searching and ACO's lacking initial pheromone.

V. CONCLUSIONS

This paper makes some researches on task scheduling under cloud environment, aiming at solving the slow convnergence problem caused by the lack of initial pheromone of ACO .Then there introduces the GA-ACO(the integration of genetic algorithm and ACO), which uses the strong global search capability of GA to get better solution, and then converts it into the initial pheromone of ACO, and finally gets optimal scheduling through positive feedback of ACO. Based on the simulation results, it shows that the integration of GA and ACO is beneficial to be used in cloud computing to solve the task scheduling, as it effectively improves the searching efficiency of algorithm.

ACKNOWLEDGMENT

We would like to thank to the organizers of the committee of DCABES 2014 and the editor of the IEEE CPS who offer us the opportunity and posibility to have our paper published.Also we want to thank to the professors who give us many suggestions to this paper.The paper is supported by "the Fundamental Research Funds for the Central Universities (WUT:2014-Ⅶ-027)

REFERENCES

[1] LeiLa Ismail,Rajeev Barua.Impementation and performance evalution of a distributed conjugate gradient method in a cloud computing environment,Software-practice and Experience,2013,pp.281-304.

[2] Abadi D J. Data management in the cloud:Limitations and opportunities.Bulletion of the IEEEComputer Society Technical Committee on Data Engineering,2009,32(1):3-12.

[3] FOSTER I,ZHAO Y,RAICU I, et al.Cloud computing and grid computing 360-degree compared[C].In Proc of IEEEGrid Computing Environments Workshop,2008:1-10.

[4] Peng Liu. Cloud computing[M].BeiJing: Electronic Industry Press,2007:2-13

[5] Rodrigo N.Calherios,Rajiv Ranjan,Anton Beloglazov.CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,Software Practice and Experience,2011,41(1),pp.23-50.

[6] Chen Liang,James Z.Wang,Rajkumar Buyya.Bandwidth-aware divisible task scheduling for cloud computing,Software-practice and Experience,2014,44:163-174.

[7] Jian-Feng Li,Jian Peng.Task scheduling algorithm based on improved genetic algorithm in cloud computing environment[J].Jounal of Computer Applications. 2011,31(1):184-186.

[8] Liang-liang Feng, Tao Zhang, Zhen-hong Jia, Xiao-yan Xia, Xi-zhong Qin.Task schedule algorithm based on improved particle swarm under cloud computing environment[J].Computer Engineering, 2013,39(5):183-186.

[9] Xuan Li,Jine-Chung Lo.Pricing and peak aware scheduling algorithm for cloud computing,2012 IEEE PES Innovative Smart Gird Technologies,pp.1-7.

[10] Xia-yu Hua, Jun Zheng, Wen-xin Hu. Ant colony optimization algorithm for computing resource allocation based on cloud computing environment[J].Journal of East China Normal University(Natural Science), 2010,2(1):127-134.

[11] Jing-zhao ZHANG,Tao JIANG.Improved adaptive genetic algorithm[J].Computer Engineering and Applications,2010,46(11):53-55.

[12] Zong-bin ZHU,Zhong-jun DU.Improved GA-based task scheduling algorithm in cloud computing[J]. Computer Engineering and Applications,2013,49(5):77-80.

[13] Jian-ping LUO,Xia LI,Min-rong CHEN.Guaranteed QoS resource scheduling scheme based on improved shuffled frog leaping algorithm in cloud environment[J].Computer Engineering and Applications,2012,48(29):67-72.

[14] Ming-hai XU,Yuan ZI.ACO-based network selection algorithm[J].Computer Engineering and Applications,2012,48(5):84-88.

[15] Jinn-Tsong Tsai,Jia-Cen Fang,Jyh-Horng Chou.Optimized task scheduling and resource allocation on cloud computing environment using imprived differential evolution algorithm,Computers & Operations Research,2013,pp.0305-0548.

[16] J.Dean,S.Ghemawat.Mapreduce.Simplified data processing on large clusters,Sixth Symposium on Operating System Design and Implementation,San Francisco,CA,USA,December 2004

[17] Bhandarkar,M.MapReduce programming with apache Hadoop,Parallel&Distributed Processing(IPDPS),2010 IEEE International Symposium on Digital Object Identifier,2010.

[18] J.Dean and S.Ghemawant,"MapReduce:Simplified Data Processing on Large Clusters,"In Communications of the ACM,Vol.51,Issue 1,2008,pp.107-113.

[19] X.H.Zhang,Z.Y.Zhong,S.Z.Feng,B.B.Tu,and J.P.Fan."Improving Data Locality of MapReduce by Scheduling in Homogeneous Computing Environments," In IEEE 9th International Symposium on Parallel and Distributed Processing with Applications,2011,pp.120-126,doi:10.1109/ISPA.2011.14.

[20] Kristi Morton,Magdalena Balazinska, and Dan Grossman,"Paratimer: a progress indicator for mapreduce DAGs,"In SIGMOD'10:Proceedings of the 2010 international conference on Management of data,pp.507-518,New York,NY,USA,2010.ACM.

A Cloud Gaming System Based on NVIDIA GRID GPU

Qingdong Hou, Chu Qiu, Kaihui Mu, Quan Qi, Yongquan Lu
Communication University of China
Beijing, China
e-mail: { houqd2012, chqiu, khmu, qi_quan, yqlu}@cuc.edu.cn

***Abstract*—Like Internet sweeping the world 20 years ago, cloud gaming, today, is gaining in popularity with unimaginable speed. However, many technical issues must be solved through its exploring process. Among all of these problems, bandwidth limitation and support for multi-client concurrent access become the bottleneck for further development of cloud gaming. Cloud gaming is based on real-time audio and video stream, therefore, it will be a great contribution to cloud gaming on research how to encode the stream with minimum time and maximum compression radio. So we are committed to improve the ways of encoding and multi-client concurrent access in the server of cloud gaming. In order to reduce the delay in limited bandwidth and improve multi-client concurrent access efficiency, a concurrent server is built up based on the integration of NVIDIA GRID GPU and the open source cloud gaming platform GamingAnywhere. As the results show, the latency is reduced and the concurrency is improved in this system.**

Keywords-component; cloud gaming; NVIDIA GPU; low latency; high concurrency

I. INTRODUCTION

The so-called cloud gaming in fact is a service. It is designed to separate the traditional game from game machine and PC, subvert the close contact between game development and hardware upgrades. Users do not need to download and install games on their own computers, they only need a simple client to play any games provided by the gaming service. The game runs on the server which is provided by the company of cloud gaming service, the client is only responsible to transmit the player's keyboard, mouse and joystick input to the server, decompose and play the compressed video and audio sent back from the server in real-time. The concept of cloud gaming is almost perfect for players who no longer need the high-end configuration, the tedious installation process, and endless upgrades; for the game developers, they no longer face the problem of piracy; and for game operators, to maintain and upgrade the game on the server side becomes a simple thing [1].

CiiNOW[2], OnLive[3] is a relatively mature commercial operation of cloud gaming company. CiiNOW was the first to lead the trend of cloud gaming solution by using AMD Radeon graphics, which provides a cost-effective, terminal-to-terminal, turnkey solutions including the necessary hardware and software components. When OnLive was introduced at the Game Developer's Conference in 2009, it attracted a significant amount of attention from the mass media and the public. The server is well-known partly because of its high-profile investors and partners. It was released in June 2010 and now offers more than 120 games. OnLive's client is available on Microsoft Windows, Mac OS X, and as a TV set-top box. OnLive has recently expanded its market to mobile devices like Andriod and IOS platforms [4].

However, those cloud-based video games still suffered from the bandwidth-bottleneck. The bandwidth constraints restrict the bit rate of gaming videos, while the jitter and delay affect the quality of experience (QoE) for the players [4]. In order to solve the bandwidth limitations, the conventional method is to set up data centers in multiple locations. When a game client accesses a game, it will automatically connect to the game from the nearest cloud data center in order to get the best gaming experience. Although this method is to some extent alleviate the bandwidth-delay problem, but the cost is too high. Generally cloud gaming operations revenues cannot pay the cost of maintaining so many data centers. Therefore, efficiently encoding and transmitting real-time gaming videos become the most critical issue in cloud-based video gaming system.

NVIDIA introduced the NVIDIA GRID cloud gaming technology in 2008, which makes video games transmitted over the web in the form of streams like any other media. At the same time NVIDIA GRID Toolkit provides a complete development kit to perform efficient image capture and remote processing for the NVIDIA GPU. NVIDIA GRID series make full advantage of GPU processing power to provide GPU accelerated applications and games through the network to the user. NVIDIA GRID is a series of products which includes GPU virtualization, remote processing and session management libraries, it allows multiple users to take advantage of GPU while experiencing graphics-intensive desktop applications and games. The CUDA (Computer Unified Device Architecture) is an architecture that allows the execution of parallel general purpose algorithms in the GPU, acting as a co-processor to the CPU. With CUDA, there is no need to deeply know the graphics pipeline to exploit the GPU resources, since features are provided to facilitate access to resource of the GPU [5].

GRID SDK provides two ways for the GPU rendering content efficiently read-back, namely: NvFBC and NvIFR. NvFBC is ideally suited to desktop capture and remoting. The NVIDIA Framebuffer Capture(NvFBC) API captures and optionally compress the entire Windows desktop or full-screen applications running on the supported Operating Systems. It provides essentially the same output one would see on a monitor connected to the GPU: a full desktop, with application windows, menu bar, composited overlay and hardware cursor. NvFBC has many advantages over existing method of framebuffer capture. It is resilient to Aero DWM enable/disable and changes in resolution, and operates asynchronously to graphics rendering, using dedicated hardware compression and copy engines in the GPU. It delivers frame data to system

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.19

memory faster than any other display output or readback mechanism, with minimal impact on rendering performance. Unlike NvFBC, the out from NvIFR does not include any window manager decoration, composited overlay, cursor or taskbar; it provides solely the pixels rendered into the render target, as soon as their rendering is complete, ahead of any compositing that may be done by the windows manager. In fact, NvIFR does not require that the render target even be visible on the Windows desktop. It is ideally suited to application capture and remoting, where the output of a single application, rather than the entire desktop environment, is captured [6].

GA is an open-source clouding gaming platform. In addition to its openness, it also has high extensibility, portability, and reconfigurability [7]. GA currently supports Windows and Linux, and can be ported to other OS's including OS X and Android. The value of GA, however, is from its openness that researchers, service providers, and gamers may customize GA to meet their needs. This is not possible in other closed and propriety cloud gaming platforms.

The remainder of this paper is organized as follows. The modification of the overall framework of GA with NVIDIA GRID SDK is introduced in Section RELATED WORK. The performance measurement results in the processing delay, bandwidth and concurrent access of GA platform with NVIDIA GPU is presented and the correlation analysis is made in Section PERFORMANCE MENSUREMENT. Finally, the concluding remarks are presented in Section CONCLUSION AND FUTURE WORK.

II. RELATED WORK

NVIDIA GRID series make full advantage of GPU processing power to provide GPU accelerated applications and games through the network to the user. NVIDIA GRID SDK also provides a way to capture and compress the render target called NvIFR, which uses on-chip hardware video encoder to a H264 video stream and copies the H264 encoded elementary bit stream to system memory at host. GA is the first open source cloud gaming system, which realizes game video capture, encoding and transmission on GA server, and data flow receiving, decoding and display on GA client. The purpose of this study is to integrate GA with NVIDIA GPU to re-implement a more efficient cloud gaming system.

In this study we have archived the transformation of GA by using NVIDIA GPU, and modified the overall architecture of GA, the main work is as follows:

1) *Modified the game video streaming capture mode by using NVIDIA GRID SDK.*
2) *Improved the current access of GA, so that the user can request to use different types of games in the same gaming server.*

After modification, the basic architecture diagram of our cloud gaming framework as shown below:

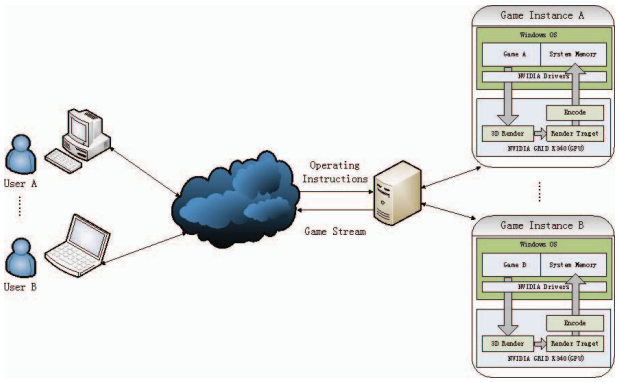


Figure 1: The architecture diagram of our cloud gaming framework

As shown above, the basic process of our cloud gaming framework is that, the server starts and waits for connecting requests from a gaming client, user A requests for game A through the client and sends the game name as parameter to a server, the server reads the specified configuration file according to the game name, then the game will be loaded from the server to the clients. At the same time the server captures game audio and video stream by using NVIDIA GPU and compresses to H.264 format, finally sends the data flow to the client. The client will display the game screen after receiving audio and video stream and performing decompression. All of the gaming operation instructions about user A will be sent to the server, the game running in the server sends back the corresponding response according to the instructions. Then the updated video will be sent to the client again after capturing and compressing by GPU and the client updates the display picture accordingly.

A. *Modify The Capture Mode of GA Server Using NVIDIA GPU SDK*

The server of our cloud gaming framework is installed on 64-bit Windows Server 2012 Datacenter Evaluation, with 128G memory, Intel(R) Xeon(R) E5-2650 2.00GHZ, and the NVIDIA GRID K2 card and its driver installed.

The kernel module of our cloud gaming framework includes : game capture module, image format conversation module, the H.264 data compression module and the communication module, Among which, the capturing of the game screen, the image format conversion and data compression are processed by three separate threads.

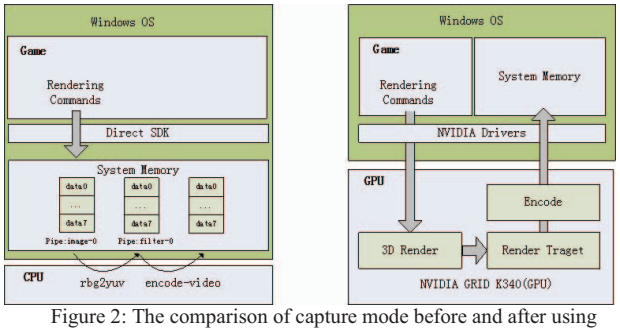


Figure 2: The comparison of capture mode before and after using NVIDIA GPU

As shown in figure 2 the comparison the capture mode of GA and our cloud gaming framework with NVIDIA GPU. The operating process about GA is as follows. First it enters resource initialization and screen capture process after the game window installs the hook [8]. Second in the phase of resource initialization a raw data channel pipe

image-0 is created, which maintains a circular linked list of eight blocks of data. When the data of game screen is captured, the data is stored in the original channel image-0. After that, channel image-0 notifies the image format conversion thread "filter_rgb2yuv" to perform image format conversion. When the thread "filter_rgb2yuv" is in initializing process, a channel pipe filter-0 is also created, which composed by a circular linked list with eight blocks of data. When "filter_rgb2yuv" thread receives a signal, it fetches the data from pipe image-0 circularly, converts them from RGB to YUV and then stores them into pipe filter-0. Finally, "filter_rgb2yuv" thread notifies "encode_video" thread to encode data. The "encode_video" thread fetches the data from pipe filter-0 to perform H.264 compression, and sends compressed data to the clients.

The basic process of our cloud gaming system framework is that, when the game starts, GPU begins to receive the rendering commands from the game screen, then performs 3D rendering and outputs image sequence. Those data will also be stored in the Render Target cache. What the server system needs to do is to read the cache data from GPU back to the continuous memory of system. Here we use the capture mode of NvIFRH264. NvIFRH264 captures and compressed the render target, using on-chip hardware video encoder to a H.264 video stream and copies the H.264 encoded elementary bit stream to system memory on the host.

As Figure 1 shown, the CPU is used to capture the game video, convert image format and compress the data in the architecture of GA. In our cloud gaming framework using NVIDIA GPU, this part of the work is performed by GPU. The most obvious difference is that, the complex computing process is transferred from CPU to GPU when using NVIDIA GPU to realize the capture process.

B. Modification of Concurrent Access

The GA does not support concurrent connection. The server starts a particular game according to specified configuration file, and then waits for the client to connect. When a client is connected, the game screen is captured and compressed by H.264. Then compressed data will be sent to the client using RTP and RTSP. When multiple clients are connected, they will see the same video, which comes from the same port.

The structure of concurrent access modification is shown below:

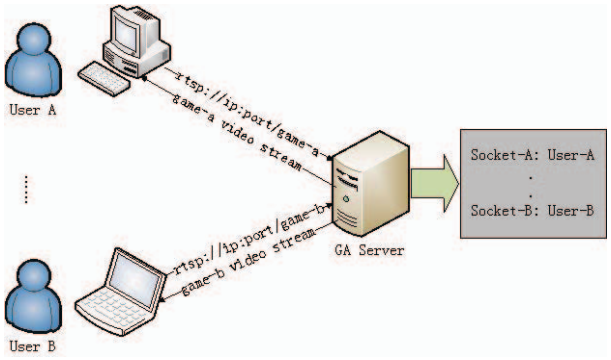


Figure 3: Concurrency transform of our cloud gaming framework

As shown above, we implement the following functions:

1) Support concurrent connections of multiple clients.

First, the server starts to listen on a port. When multiple client connection requests arrived, the server forks separate threads for each client to serve and preserves the socket created for connection with client. This socket will be used in subsequent communication in RTSP. Each incoming client will see the respective game screen due to the using of the separate socket in communication. And there is no interference in data transmission between the individual connections.

2) Support ordering game on demand

The RTSP communication of GA client is implemented by using Live555, performings the RTSP stages one by one. The stages are, in order, OPTION, DESCRIBE, SETUP, PLAY and TEARDOWN. In RTSP communication process, the OPTION phase is used to ask the server which methods are available, then the server returns all of the available methods. The implementation of GA does not use the OPTION stage, which is added in the client of our cloud gaming framework and used to send the game name to the server. Finally the game name will be parsed and the specified game will be loaded in the server.

Figure 4 shows the game video capture of our cloud gaming framework. The figure 4-A is the game screen which is running on the server-side, and the figure 4-B is the game screen received by the client. It has higher resolution and runs more smoothly without any experience of delay.

Figure 4: The game video capture of our cloud gaming framework

III. PERFORMANCE MENSUREMENT

In our experiments, processing delay, bandwidth and concurrency are measured in our cloud gaming framework which integrate with NVIDIA GPU. The measurement of this study is carried out in a local area network.

The response delays of a cloud gaming system consist of three parts: network delay, processing delay and play-out delay. Network delay is the time required to deliver a player's command to the server and the time of sending back a game screen to the client. It is usually referred to as the network round-trip time. Processing delay is the difference between the time the server receives a player's command and the time it responds with a corresponding frame after processing the command. Play-out delay is the difference between the time that a client receives the encoded form of a frame and the time the frame is decoded and presented on the screen [9]. The processing delay, a way of measuring the performance of our solution, will be reduced by using NVIDIA GPU to capture and encode the cloud gaming stream in the server.

The purpose of the concurrent access is to achieve the largest number of games running in one gaming server.

A. *Processing delay*

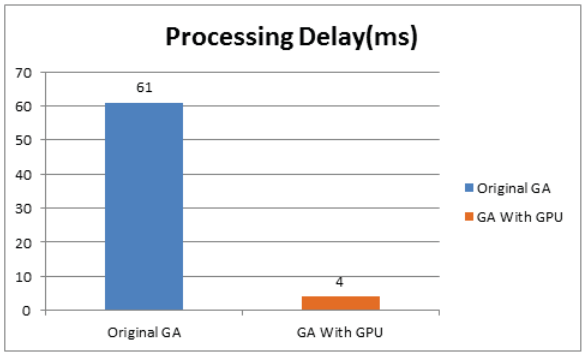


Figure 5: The comparison of processing delay

As shown above, it takes about 61ms to capture the game screen, convert the data format and compress the data, all stages processed by CPU. Using the NVIDIA GPU, data format conversion and data compression H.264 is completed in about 4ms. There are two reasons to cause the processing delay reduced: 1) The GPU completes the image format conversion and H264 compression through hardware acceleration. 2) The system does not need to spend a large cost on create, maintain and destroy the circular linked list.

B. *Bandwidth*

Iperf tool is used to measure the bandwidth between the server and client. Iperf is a network performance testing tool, which can test the maximum TCP and UDP bandwidth performance, Iperf can report bandwidth, delay jitter and packet loss. Because it is in the local area network environment and network transmission is good, the data loss factor need not to be considered. Figure 6-A show the network load generated by GA system and Figure 6-B show the network load generated by our cloud gaming framework. Figure 6-A shows that the bandwidth fluctuates between 5.5 and 6.5MBps without GPU. Figure 6-B shows that the bandwidth is fluctuate up and down in 3.25MBps using the GPU, because of the GPU using hardware compression. Due to its higher compression efficiency, when the same data is input, it will produce the less output, so that the network load will be produced accordingly.

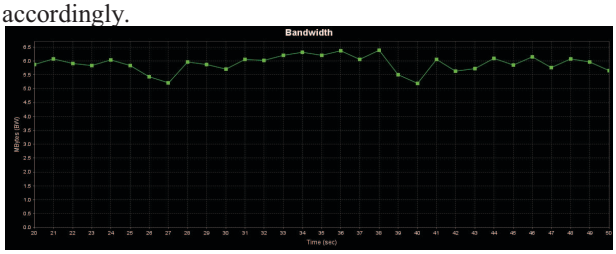


Figure 6-A:The bandwidth of original GA

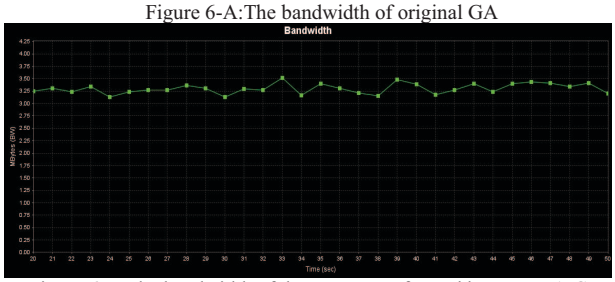


Figure 6-B:The bandwidth of the game transformed by NVIDIA GPU

C. *Concurrent Access*

The GA platform does not support multi-user concurrent requests, and it supports in our cloud gaming framework. The server of our cloud gaming framework is installed on Windows Server 2012 Datacenter Evaluation with 128G memory and NVIDIA GRID K2 graphics, which can support 15 concurrent connection requests of users under better transmission quality.

IV. Conclusion And Future Work

In this study, we proposed a solution to reduce the response latency and improve concurrent access for cloud gaming system integration with NVIDIA GPU. But the study also has some deficiencies that need improvements in further work:

- Optimizing the concurrency to support more client connection;
- Optimizing the capture mechanism to support more types of games;
- Optimizing the data transfer to make the transfer process more stable;

Cloud gaming has broad prospects, although it will take a long period of time for overall ascension of the basic network environment, yet when the wind coms, a single spark can start a prairie fire.

Acknowledgment

The authors acknowledge the financial supports by the National Key Technology Support Program(2012BAH17B03) and the Program Project of CUC(XNG1138, YXJS2012319, YXJS2012206, BY2012230, BE2013054 and JSWHCY-2013-(98)).

References

[1] D. Mishra, M. El Zarki, A. Erbad, H. Cheng-Hsin, N. Venkatasubramanian. “Clouds + Games: A Multifaceted Approach,” IEEE Internet Computing, vol. 18, May. 2014, pp. 20-27, doi:10.1109/MIC.2014.20.

[2] http://www.linkedin.com/company/ciinow.

[3] http://www.onlive.com.

[4] Wei Cai, Leung V.C.M. “Mlutiplayer Cloud Gaming System with Cooperative Video Sharing,” CloudCom, Taiwan, Dec. 2012, pp. 640-645, doi:10.1109/CloudCom.2012.6427515.

[5] F. Tsuda, R. Nakamura. “A technique for collision detection and 3D interaction based on parallel GPU and CPU processing,” SBGAMES, Brazilian, 2011, pp. 36-42, doi: 10.1109/SBGAMES.2011.20.

[6] NVIDIA, GRID SDK 2.1 PROGRAMMING GUIDE, 2013.

[7] C-Y Huang, C-H Hsu, Y-C Chang and K-T Chen. “GamingAnywhere: An Open Cloud Gaming System,” MMSys ACM. Oslo Norway, 2013.

[8] K-T Chen, Y-C Chang, P-H Tseng, C-Y Huang and C-L Lei. “Measuring The Latency of Cloud Gaming Systems,” ACM, New York NY USA, 2011, pp.1269-1272, doi:10.1145/2072298.2071991.

[9] K-T Cheng, Y-C Chang, H-J Hsu, D-Y Chen, C-Y Huang and C-H Hsu. “On the Quality of Service of Cloud Gaming Systems,” IEEE Transactions on Multimedia, vol. 16, Feb. 2014, pp. 480-494.

[10] S. Ryan, L Jiangchuan. “On GPU Pass-Through Performance for Cloud Gaming: Experiments and Analysis,” NetGames Annual Workshop, Canada, 2013, pp. 1-6, doi:10.1109/NetGames.20136820614.

[11] D. Vintache, B. Humbert, D. Brasse. “Iterative Reconstruction for Transmission Tomography on GPU Using Nvidia CUDA,” Tsinghua Science and Technology, vol. 15, Feb. 2010, pp. 11-16.

[12] Z-P Lu, X-M Wen and Yong Sun. “A Game Theory Based Resoure Sharing Scheme in Cloud Computing Environment,” WICT, Trivandrum, 2012, pp.1097-1102, doi:10.1109/WICT.2012.6409239.

Improving Random Read Performance of Glibc

Mei Wang
School of Computer Engineering
Shenzhen Polytechnic
Shenzhen, China
name@xyz.com

Yuanyuan Zhou, Feng Xiao, Qiuming Luo
College of Computer Science and Software Engineering
Shenzhen University
Shenzhen, China
lqm@szu.edu.cn

Abstract—The Cloud data services, specifically, key/value stores and NoSQL database that require a large number of index lookups that fetch small amount of data. Random I/O becomes the critical performance factor. However, compared with sequential read, the efficiency of random read is very low. Our experiment will explain this. File I/O operation is closely associated with the implementation of I/O mechanism both in kernel space and user space. In this paper, we aims at analyzing the standard I/O mechanism, improving the standard I/O mechanism in user space for random read and implement into the glibc. Our experiment test result proves that our improved I/O mechanism will greatly improve the performance of random read.

Keywords-glibc; standard I/O; random read; I/O buffer; kernel file pointer

I. INTRODUCTION

Current leadership-class machines, such as the Tianhe-2 and Titan systems, consist of tens to hundreds of millions of cores [2]. While the computational power of supercomputers increases, the I/O system for these machines is often less powerful. Data access rates to hard disks are not improving as quickly as multicore processor computation rates, and the increasing number of processing elements in the systems can generate an overwhelming volume of I/O requests. I/O has become a critical bottleneck for data-intensive scientific applications on HPC systems and leadership-class machines. And the same time, cloud data services, specifically, key/value stores and NoSQL databases that require a large number of index lookups that fetch small amounts of data. Random I/O becomes the critical performance factor [1].

File I/O operation involves device driver, I/O scheduler layer, generic block layer, I/O forwarding layer, kernel page cache layer, the VFS abstraction layer, and user space buffer layer, etc. There are many I/O optimization research works from each layer [3] [4] [5]. Kazuki Ohta's work [3] presented two optimization techniques at the I/O forwarding layer to further reduce I/O bottlenecks on leadership-class computing systems. [4] presents a set-associative page cache for scalable parallelism of IOPS in multicore systems. [5] proposes an I/O buffer cache (kernel space) mechanism based on the frequency of file usage. Some I/O optimization works base on using the I/O analysis tools to collect a lot of trace information about I/O calls and leave it for programmers to understand [6] [7] [8].

In this paper our main optimization work is at the user space buffer layer. Through experiment and analysis, we found that the cost of switch to the kernel space to change the kernel file pointer is very high, when the read randomly rang is not very large, the cost of changing kernel file pointer is the main reason for the poor performance of random read. Therefore, we have improved the glibc I/O mechanism to minimize the cost of changing the kernel file pointer. We redesigned the fgetc, fseek, flush, fclose function and implement them into glibc. Our test result show that the improved I/O mechanism for glibc can greatly improve the performance of random read when the random range is not too big, for some cases, the performance can almost equival to sequential read.

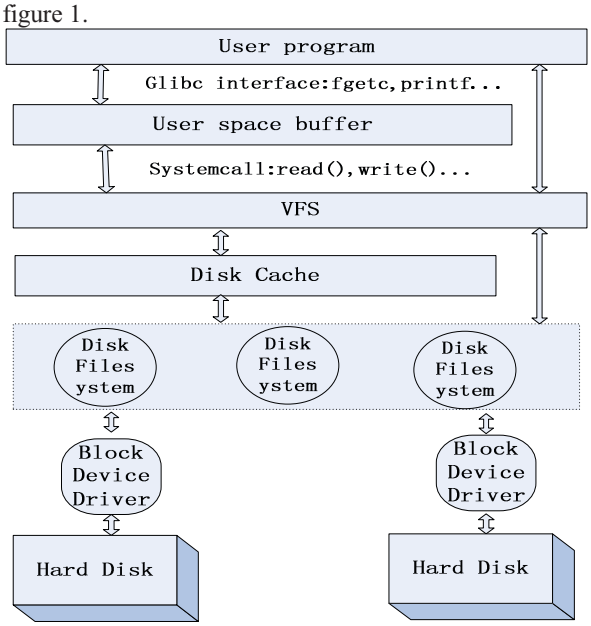
The set of the paper is organized as follows. In section II, we descripts the glibc standard I/O mechanism. Section III focuses on the performance test of random read using the glibc interface and analyzing the cause of the performance degradation. In section IV we illustrate our modification to the glibc I/O mechanism and presents our experiment result about random read with the improved glibc I/O mechanism. Our conclusion and future work is in section V.

Our optimization work for random read mainly based on linux, so in the below, the standard C library represent the glibc, and the standard I/O mechanism refers to the linux I/O mechanism.

II. STANDARD LINUX FILE I/O MECHANISM

A. Accessing a Regular File

Accessing a regular file is a complex activity that involves the VFS abstraction, the handling of block devices, and the use of disk caches and so on, as showed in figure 1.



978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.20

Figure 1. The hierachical of accessing a regular file

The linux kernel uses the page cache as the disk cache to buffer the disk file data in the kernel space to improve system performances by reducing disk accesses as much as possible, all access to regular files made by read, write, and mmap system calls is done through the page cache [11]. For the similarly reason, in order to reduce the cost of switching form user model into the kernel model, the standard I/O library usually set up a buffer area in user space for every opened file of the user program to minimizing the number read or write system calls. Therefore, for the standard file input operation, the file data will get from disk devices to the kernel buffer space, and them copy to the user space buffer, and finally assigned to the application's user space variables. For standard file output, it performs the opposite operation.

B. *User Space Buffer*

The goal of the buffering provided by the standard I/O library in the user space (always be called user space buffer) is to use the minimum number of read and write calls. Also, it tries to do its buffering automatically for each I/O stream, obviating the need for the application to worry about it. Three types of buffering mode provided by the glibc standard I/O library: unbuffered, line buffered and fully buffered.

Unbuffered means the standard I/O library does not buffer the characters. When we write some characters with the standard I/O fputs function to an unbuffered stream, it probably be handled with the write system call. The standard error stream, for example, is normally unbuffered. This is so that any error messages are displayed as quickly as possible, regardless of whether they contain a newline. For Line buffered, the standard I/O library performs I/O when a newline character is encountered on input or output. Line buffering is typically used on a stream when it refers to a terminal: standard input and standard output, for example [12].

In case of fully buffered mode, actual I/O takes place when the standard I/O buffer is filled. Files residing on disk are normally fully buffered by the standard I/O library. Some pointers in the struct of FILE in glibc standard I/O library are used mainly to control the buffer for reading and writing, in which the _IO_buf_base and _IO_buf_end are used to point to the bottom and top of the buffer respectively, _IO_read_base and _IO_read_end points to the start and end of the input buffered area, and _IO_write_base and _IO_write_end points to the start and the end of output area, the last two are the current read pointer and current write (or put) pointer named _IO_read_ptr and _IO_write_ptr.

The buffer used is usually obtained by one of the standard I/O functions calling malloc the first time I/O is performed on a stream, so if the file is opened with O_RDONLY mode, then the first time I/O operation will be reading operation and all the buffer assigned for buffering reading data. Otherwise, if the file is opened with O_WDONLY mode, the buffer will be assigned for buffering putting data, also means the _IO_write_end equals to _IO_buf_end. In the final case, the file is opened with O_RDWR mode, the buffer (default size is 4KB) is shared by reading and writing.

III. RANDOM READ

The so-called sequential read to a file refers to reading data from a file without intermediate seeks, and the random read is a reading pattern of seek-read-seek-read, etc. For example, a file consists of many records, the data an application needs is a part of data of each record, in this case, the application must do random read to cross the width of a record.

A. *Experimental Data of random read Performance*

There are two methods we can use to do random read with the glibc library function [9]. One method we called fseek_read is that call the lseek or fseek function to adjust the kernel file pointer before calling the function about reading each time, and the another method is to call the pread function to make the operation of adjusting the kernel file pointer and reading at one go.

The picture below (figure 2) shows our test results about bandwidth of reading 10MB data from a text file using three different methods. The version of glibc we used is 2.19, which is the latest version. We use the fgetc function (each time read a char) to read data, and use the fseek function to adjust the kernel file pointer. The sequential_read means do sequential read (just calls the fgetc function without fseek or lseek function), while the main code of fseek_read is like this:

```
for(i = 0; i <    numB; i++)
{
        fseek(fp, i, SEEK_SET);
        rch = fgetc(fp);
}
```

Before each time of reading, there is an fseek function to adjust the kernel file pointer.

The pread function is used to read from a file descriptor at a given offset. We need to pass an offset parameter to the function each time we call it. There is only one statement (call the pread function) in the loop of for as well as the sequential_read.

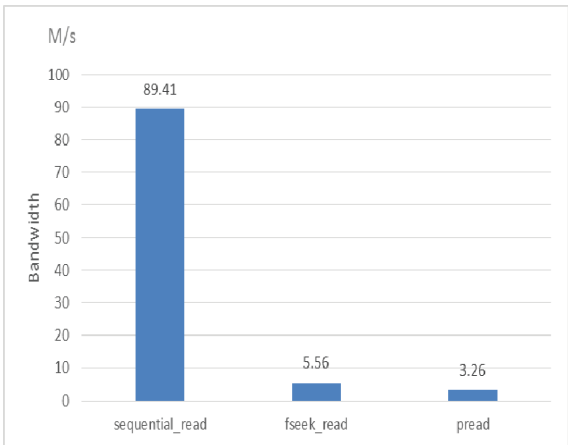


Figure 2. The bandwidth of Random Read and Sequential Read

B. Analysis

As we can see from the figure 2, the bandwidth of sequential_read is much bigger than the random read (both fseek_read and pread method). The bandwidth is almost 16 times that of the fseek_read and is more than 27 times compared with pread. This definitely demonstrates that the cost of switching to kernel space to change the kernel file pointer is very high. However, the bandwidth of pread which adjust the kernel file pointer and read operation at one go is almost half of the fseek_read which do this in two steps. This is because when use the pread function to read data from a file, the glibc provides unbuffered type for it. We can print the value of _IO_read_base and _IO_read_end to check it. It is unlike fseek_read with fully buffered type. For fseek_read, the fseek operation will change the kernel file pointer, and then adjust the _IO_read_ptr to the right point. Therefore, when the kernel file pointer after changed is within the scope of the buffer (the range of file offset all the data in the buffer corresponding to). The next read operation just need get the data from the user space buffer directly rather than use the read system call to get data from kernel space.

In fact, we have do similar test on Windows, the problem of low efficiency about random read is more serious.

IV. IMPROVEMENT TO GLIBC'S I/O MECHANISM

A. New Mechanism and strategy for I/O

According to the analysis in the previous section, we find that the cost of random read mainly in two aspects, one is the cost of switching into kernel to change the kernel file pointer, and the other aspect is about the needed data can't be buffered in the user space buffer, especially the cost of changing kernel file pointer. As a consequence, we improve the efficiency of random read from these two aspects.

The buffer used is usually obtained by one of the standard I/O functions calling malloc the first time I/O is performed on a stream, considering the data the application need can't be buffered. We can set up an appropriate size of buffer according the range the file pointer jumps. For the cost of changing the kernel file pointer, if we think carefully, we will find that the kernel file pointer need be used for reading data from read-only file just when the user space buffer need to be refreshed. At that time the glibc I/O library need the kernel file pointer in the kernel space's file struct to get right data from the kernel space cached page to the user space buffer. That means if the kernel file pointer the nfseek or lseek will change is in the range of file offset that the data in the buffer corresponding to. It is not necessary to do really changing of the kernel file pointer, it just need to change the _IO_read_ptr to the right position so that the next read operation can read the correct data, and then record the last value of the kernel file pointer in user space. Adjusting kernel file pointer according the last value of the kernel file pointer recorded till the user space buffer need to be refreshed.

In general, we can buffer the last position of the file pointer changed by fseek or lseek function with a global variable and put off the change to the kernel file pointer until the user space buffer need to be refreshed. In order to reduce the large number of changes of the kernel file pointer and improve the efficiency of random read.

B. Implementation into glibc

On one hand because most application on linux use the glibc's standard I/O library, changing the glibc is easy to make the system crash, even the system can't be restart , because some system initialization process also use the glibc's standard I/O library to deal with standard input and err, etc. On the other hand, our improving work mainly on random read, changing the source code of glibc I/O library function we must consider the other read or write mode, this will make our work very complex. Therefore, we employ a simple and feasible method: we create a global structure variable (named nfops) and an initial function. The members of the structure include a random_read flag which represents the application will do random read operation if this flag is set, and a last_seek variable which use to record the last value of the kernel file pointer changed by fseek or lseek, and a buffer_size variable which use to set up an appropriate size of buffer according the range the kernel file pointer jumps, and some pointer of function about file reading. The user application can call the initial function to initialize the structure. If the random_read is set, the initial function will initialize the function pointers about file reading to new set of function we designed for random read. Otherwise, this function pointers will be initialized to the old file reading function of the glibc. Therefore, we just add some file reading function about random read into glibc and do not affect the use of the original gibc standard I/O library function.

We have designed a set of file reading functions for random read and rename them as nfseek, nfgetc, nfflush, nfclose, etc. The main code of nfseek as follow:

```
if((nfops.read_base<=offset)&&(offset<
  nfops.read_end ))
        fp->_IO_read_ptr = fp->_IO_read_base +
        (offset - nfops.read_base);
else
        fp->_IO_read_ptr = fp->_IO_read_end;
nfops.last _seek = offset.
```

The offset is the position of kernel file pointer the nfseek function set. If the offset is not beyond the scope of the buffer, this function will just adjust the _IO_read_ptr to the right position and record the last_seek. Otherwise, it will set the IO_read_ptr equal to _IO_read_end to invalidate the data in the buffer, so that the buffer will be refreshed in next read operation.

We can see that the nfseek is unlike fseek, it just adjust the _IO_read_ptr and record the last value of the kernel file pointer. In the nfgetc function, each time user space buffer need to be refreshed, it will call the lseek function to adjust the kernel file pointer according to the value of last _seek, showed as follow:

```
lseek(fd, nfops.last _seek, SEEK_SET);
count=_IO_SYSREAD(fp, fp > _IO_read_base, fp >
 _IO_buf_end-fp>_IO_read_base);
```

C. *Results comparison*

In this part, we show the efficiency of random read using our improved method, and compared it with the other two random read methods introduced in the part A of section III. Because the default size of user space buffer is 4KB in the standard file I/O mechanism of glibc, in the improved random read method (marked with nread) we also initialize the buffer size with 4KB. We use the fgetc function to read a character each time just like the fseek_read method. We designed a program which used to write a specified number of random characters to a file and use this program to create a text file with size of 80GB, we use this file as our random read file in our experiments and read 10MB data in each experiment.

The result showed as figure 3, the vertical axis represents the bandwidth of random read using three different method, the unit is MB/s. The blue curve is the trend of bandwidth using improved random read functions with different random read step (the file offset distance of the ordered two file read). The horizontal axis shows the different random read step from 1 to 8192, the maximum step is 8192 and read 10MB data in each experiment. That is why we have created an 80GB file (8192*10MB).

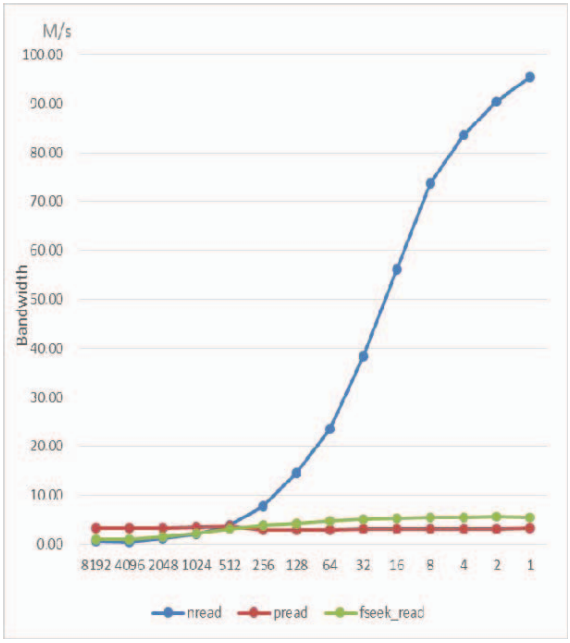


Figure 3. The bandwidth of Random Read with different step

From the figure 3, we clearly see that the bandwidth of nread seems to be growing exponentially with the reduction of step, this is because the smaller the step, the frequency of the buffer need to refresh is lower and also the number of switching into kernel model to change the kernel file pointer is less, when the step is 1, the nread is equivalent to sequential_read, so the bandwidth is almost equal to sequential_read, even a little bigger, this may because the new fgetc function is more simple without too many jumps of function pointer and strict security checks. Due to the reduced number of refreshing buffer with the reduction of step, the growth of the bandwidth of fseek_read grows slowly. However, because the pread with unbuffered type, so the bandwidth of it is basically unchanged.

From another point of view, when the random read step is bigger than the one in four of the total buffer size, it is better to use the pread method to do random read.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced the standard file I/O mechanism briefly and specify the user space buffer, and then we illustrated the problem of low efficiency about random read through the experiments and our analysis. With that we concluded that the cost of switching into kernel model to change the kernel file pointer is very high and that the data can't be buffered also has certain influence to the performance of rand read. In the next section we presented our improvement of the standard file I/O mechanism for random read and introduced how to implement it into glibc. Finally, we showed the high efficiency of our improved random read mechanism when the jumping step of read is not too large.

In fact, there is another kind of file reading and writing mode which causes a lot of buffer refreshing and changes of the kernel file pointer. That is the interval reading and writing with O_RDWR mode. Because the read and write shares a buffer, this will make the read buffering data and the write buffering data be covered by each other, and each time read or write may need to change kernel file pointer to return to their previous operating position. In this case, we can designed a double buffer to isolate the read and write buffer data and record respective kernel file pointer in the user space. That is our future work.

ACKNOWLEDGMENT

The research was jointly supported by the following grants: China 863-2012AA010239, NSF-China-61170076, Foundation of Shenzhen City under the numbers JCYJ20120613161137326, JCYJ2012061310222457, Shenzhen Polytechnic Foundation 601422K20008, and Academician Workstation Construction Projects in Guangdong Province (2012B090500020).

REFERENCES

[1] K. Muthukkaruppan. Storage infrastructure behind facebook messagesUsing HBase at Scale. In High-Performance Transaction Systems, 2011.

[2] " Top500 list," http://www.top500.org/. [Online]. Available: http://www.top500.org/.

[3] K. Ohta, D. Kimpe, J. Cope, K. Iskra, R. Ross, and Y. Ishikawa, "Optimization techniques at the I/O forwarding layer," in IEEE International Conference on Cluster Computing,IEEE press, sept. 2010, pp.312-321, doi: 10.1109/CLUSTER.2010.36.

[4] Da Zheng , Randal Burns , Alexander S. Szalay, "A parallel page cache: IOPS and caching for multicore systems," Proceedings of

the 4th USENIX conference on Hot Topics in Storage and File Systems, USENIX Association Berkeley press, June. 2012

[5] Tatsuya Katakami, Toshihiro Tabata and Hideo Taniguchi,"I/O Buffer Cache Mechanism Based on the Frequency of File Usage," The Third International Conference on Communications and Information Technology, IEEE press, Nov. 2008, pp 76-82, doi: 10.1109/ICCIT.2008.107.

[6] Y. Yin, S. Byna, H. Song, X.-H. Sun, and R. Thakur, "Boosting Application-Specific Parallel I/O Optimization Using IOSIG," in Proceedings of IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, 2012.

[7] S. Seelam, I.-H. Chung, D.-Y. Hong, H.-F. Wen, and H. Yu, "Early experiences in application level I/O tracing on Blue Gene systems," in Proceedings of the IEEE International Parallel and DistributedProcessing Symposium, 2008.

[8] "HPC-5 open source software projects: LANL-Trace," [Online]. Available: http://institute.lanl.gov/data/software/#lanl-trace.

[9] S. Loosemore, R. Stallman, R. McGrath, A. Oram, U. Drepper, "The GNU C library reference manual for glibc 2.19," Free Software Foundation.

[10] Daniel Bovet , Marco Cesati , Andy Oram, "Understanding the Linux Kernel," Second Edition, O'Reilly & Associates, Inc., Sebastopol, CA, 2002

[11] W. Richard Stevens, "Advanced programming in the UNIX environment," Addison-Wesley, 1992

MVEI: An Interference Prediction Model for CPU-intensive Application in Cloud Environment

Xiaoli Sun, Qingbo Wu, Yusong Tan, Fuhui Wu
College of Computer
National University of Defense Technology
Hunan, China
E-mail: mengshang_90@163.com

***Abstract*—Consolidation and overbooking are two important methods to improve the resource utilization in cloud computing. In overbooking cloud environment, the resources occupied by virtual machine are restricted to guarantee performance fairness between multi-tenants. However, the interaction among co-host applications results in performance degradation because of the resource sharing and competition. Previous studies about performance interference mainly focused on traditional virtual environment and didn't take the resource restriction into account. In order to predict the effect of interference in cloud environment, which can give suggestions for resource scheduling in turn, we conduct comprehensive experiments about CPU-intensive application in Cloud. Firstly, the interfering factors of the performance are analyzed, and the relationships between different factors and interference are obtained by using the least squares method. Then, a multivariable exponent interference (MVEI) model is proposed to predict the interference in resource restricted environment. Finally, the experimental results indicates the MVEI model can predict interference more precisely than linear and quadratic model.**

Keywords-Cloud computing; CPU-intensive application; Interference predication; MVEI model.

I. INTRODUCTION

Cloud providers adopt overbooking and consolidation strategies to increase resource utilization and revenues. And they encapsulate applications in the virtual machines (VMs) which is consolidated to the physical machines (PMs) by using the virtualization technology. After consolidation of VMs to smaller number of PMs, some physical machines will be idle so that they can be turned off or set to the low power mode [1-3]. Therefore, the virtual machine consolidation can improve the energy efficiency of data center. Overbooking refers to submitting more resource requirement than the overall resources available from cloud providers [4]. The cloud environment usually employs CPU reservation and CPU utilization restriction strategies to guarantee the CPU, network and disk I/O performance for each application. Hence, the cloud environment is quite different from traditional virtual environment.

Virtualization technology provides strong isolation measures to prevent a VM's failure from influencing others. But some researches show that virtualization technology, which slices the memory, hard drives and assigns them to different virtual machines, can't provide effective performance isolation [5-7]. As some shared resources (e.g. memory bandwidth and cache) can't be sliced, it causes interaction among the co-host applications and performance degradation, which is known as performance interference. Performance interference means that cloud users can't obtain the quality of service (QoS) corresponding to the resources they purchased.

In order to solve the problem of performance interference, many researches have been focused on quantitative analysis and modeling [8-11]. Most of them are about the performance interference of data and I/O intensive application in traditional virtual environment. But few of them are concerned about performance interference in cloud computing environment in which the resources are restricted, and it's different from traditional environment.

In this paper, some experiments are performed in resource restricted environment to analyze the interference of CPU-intensive applications. A NAS Parallel benchmark is selected as CPU-intensive application, which is deployed on physical machine to collect the data of performance degradation. After analyzing the reason of performance degradation and the interfering factors, the multivariable exponent interference (MVEI) model is proposed to predict the performance interference. The parameters of the model are calculated by the Newton iterative. Compared with the linear and quadratic model, the MVEI is more accurate.

The rest of the paper is organized as follows: Section II discusses related work. Section III presents the interference statement and analyzes the performance interference mechanism. Section IV proposes a performance interference prediction model. In Section V, we compare MVEI with linear and quadratic models. Section VI presents the conclusion.

II. RELATED WORK

Since the virtualization technology was employed in cloud computing, the interference among co-host virtual machines has received widespread attention. The purpose of researches about interference is to ensure the QoS of applications. R. Nathuji et al. [5] proposed a MIMO feedback model, and used Q-states to distinguish different level of QoS. Q. Zhu et al. [12] considered the time variances in resource usages, and proposed an interference model to predict the application QoS. Compared with these researches our proposed approach focuses on a specific type of resources, instead of a holistic analysis of interference caused by all the resources.

Some prior researchers analyzed the interference according to the type of resources competed by the co-host VMs, such as cache, I/O. Govindan et al. [13] analyzed the

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.21

performance interference caused by the last level cache contention and proposed a simulated cache, which was leveraged to predict the interference among VMs. Yiduo Mei et al. [14, 15] presented an in-depth performance interference analysis focused on network I/O applications. R. C. Chiang et al. [11] presented three interference prediction models (the weighted average, linear and nonlinear models) for data-intensive application on a pair of VMs, and used machine learning method to determine the parameters. Similarly we analyze the interference among CPU-intensive applications, and establish the interference prediction model. But our work differs in three aspects. Firstly, we deeply analysis the key source of interference about CPU-intensive applications. Secondly, we establish the multivariable exponent interference model (MVEI) according to the analysis. Finally, we study the interference in the overbooking cloud environment which is different from the traditional virtual environment.

III. INTEFERENCE STATEMENT

A. Experimetal Setup

1) Hardware environment

All experiments are conducted on a server host featuring two Intel Xeon series processor E5-2430 CPU, with each features 6 cores running at the frequency of 2.2GHz. As the CPU enables hyper-thread technique, there are 24 cores in total. The last cache is 15M, and the system memory is 48G. The operating system of the host is kylin server 3.2.4, and KVM is used to establish a virtual system. The operating system used in virtual machines is same with the host. Each virtual machine is configured with 4 virtual processers and a 4G memory.

KVM virtualization technology can realize binding between CPU and VM by using cgroup. As VM is just a process in KVM server, we can use cpulimit achieve the control of CPU utilization. In cloud environment, CPU utilization of VM is restricted by cpulimit. And the physical machine reserves some CPU resources to ensure the host running normally.

2) Benchmark

In the experiment, we select the NAS parallel benchmark [16] as CPU-intensive applications. This benchmark consists of eleven procedures, and each has different sizes of test data sets. The data sets are divided into several categories according to the data size. Usually, the NAS parallel benchmark has three different versions: sequential, OpenMP and MPI. Since the single-threaded CPU-intensive applications can't take full advantage of several VCPU in VM and the MPI has many I/O operations, the OpenMP version is chosen. Table I shows the resource usage in isolated environment. For most procedures, the VCPU utilization rate is very high, but they are different in memory and I/O. Both FT and DC contain many I/O operations. MG and FT take up more memory resources. EP and LU need less memory resource, but LU access cache more frequently. Compared with EP, BT and CG occupy more memory but less I/O and cache operations. To analyze CPU-intensive application, the procedures should contain less memory and I/O operations. Therefore, we choose EP and BT as the benchmark.

TABLE I. THE RESOURCE USAGE OF NPB

Test	CPU (%)	Mem (%)
BT	100	18
EP	100	0.2
LU	100	15.7
CG	100	23.3
MG	10	89
FT	11.4	86.2
DC	100	50
IS	100	27.6
SP	100	19.4
UA	100	12.9

We use Sysstat and Oprofile [17] to monitor CPU utilization and other parameters about CPU events. Oprofile is a low-cost system management and global monitoring tool, which samples the processor event. And it can record the number of the cache line failure, hence we adopt the tools to monitor the cache miss rate.

B. Interference analysis

We deploy the NPB applications on the VM, and take the runtime in isolation environment as a baseline. The isolation environment is that there is only one VM running on the server. The interference is defined as follow:

$$I = \frac{t_{inf}}{t_{iso}} \tag{1}$$

where t_{iso} and t_{inf} respectively represent the runtime of the application in isolation and interference environment, and I is the interference of application. Obviously, the interference is greater than 1, and it increases with the degree of interference.

1) Experimental results

We gradually add the virtual machines to the PM and test the performance of applications. The interference increases linearly with the number of virtual machine, as shown in Fig. 1. In overbooking situation, the performance degradation rate doesn't change which indicates the cloud environment can provide effective performance guarantee.

Since the virtual machine in KVM environment is a process in PM, the VM scheduling is process scheduling. In the PM, the CFS (Completely Fair Scheduler) is adopted to schedule the process, so that the VM can be fairly scheduled. Thus, the performance of applications executed in the same PM simultaneously has slightly differences, as shown in Fig. 2. It shows that the KVM virtualization technology can basically achieve CPU performance isolation and fair scheduling.

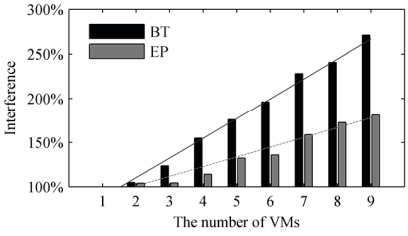


Figure 1. The performance interference in cloud environment

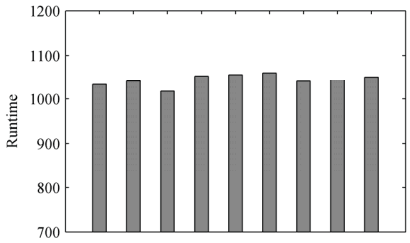


Figure 2. The runtime of applications running on the same PM simultaneously

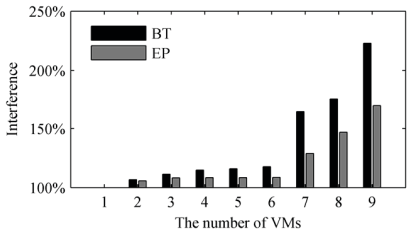


Figure 3. The interference in traditional virtual environment

Fig. 3 shows the interference in the traditional virtual environment. It is easy to see that the interference is almost unchanged in non-overbooking situation (N<6, N is the number of VMs). And when the number of VMs is seven, the interference will increase by 50%. So the interference in traditional virtual environment is totally different from the interference in cloud computing environment shown in Fig. 1.

IV. PERFORMANCE INTERFERENCE MODEL

In this section, we analyze the reason of performance interference for CPU-intensive applications. Based on this analysis, we establish the multivariable exponent interference (MVEI) model to predict the interference.

A. Interference mechanism analysis

The CPU-intensive applications have high processor utilization, perhaps at 100% usage for many seconds or minutes. This kind of application consumes more CPU resources and less I/O resources. So the interference of co-host CPU-intensive applications is caused by sharing and competing CPU, cache and memory resources.

To understand the competitive situation of resources, the variations of the CPU utilization, VCPU utilization and cache miss rate are recorded.

Fig. 4 shows the relationship between the CPU utilization of PM and the number of VMs. It indicates that the CPU utilization of PM increases nonlinearly. In non-overbooking situation, the CPU utilization of PM increases linearly. While in overbooking case (N<6), the CPU utilization of PM exhibits a nonlinear growth, but the increase rate is very small. The reason why the CPU utilization can't reach 100% is that cloud computing provides CPU reservation measures to ensure the PM running normally. In non-overbooking situation, the applications don't share the CPU time slices. In contrary they compete for the CPU slices time in overbooking situation. Therefore, CPU utilization is an important interfering factor.

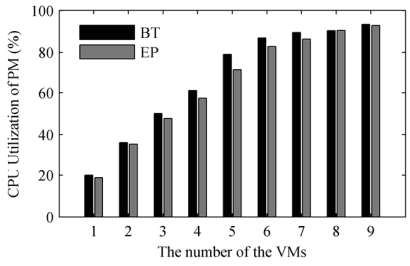


Figure 4. The CPU utilization of PM

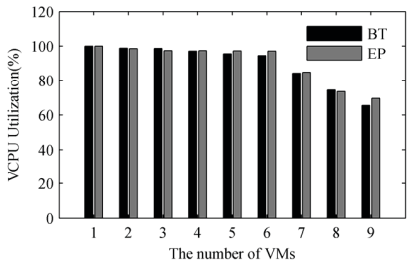


Figure 5. VPU utilization

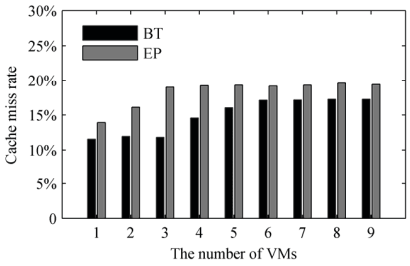


Figure 6. The cache miss

The CPU utilization of virtual machine is the ratio of VCPU scheduled time to the total time. In Fig. 5, VCPU utilization decreases with the number of virtual machine. When multiple CPU-intensive applications are deployed on the same PM simultaneously, the time VCPU scheduled corresponding reduces and the waiting time becomes longer due to the CPU time slice competition. So the performance and the VCPU utilization related closely.

Another important factor of interference is the cache miss rate. Since CPU-intensive application inevitably fetches data from cache and multiple cores share last level cache in SMP hardware structure, the cache competition will be more intense. Fig. 6 shows the relationship between the cache miss and the number of virtual machine. We can see that the cache miss increases with the number of VMs. In overbooking situation, the cache miss is almost unchanged.

In a word, as sharing cache and CPU time slices, the co-host CPU-intensive applications will affect each other. In overbooking situation, the CPU time slices is a more important factor. And in non-overbooking situation, the cache miss is more important.

B. Interference model

In this section, a multivariable exponent model is established according to the former analysis. In overbooking cloud environment, when an application is going to be deployed on the physical computing server, the resource utilization of PM and resource usage of the

application are always known. Our model is based on this condition to predict performance interference.

1) MVEI model

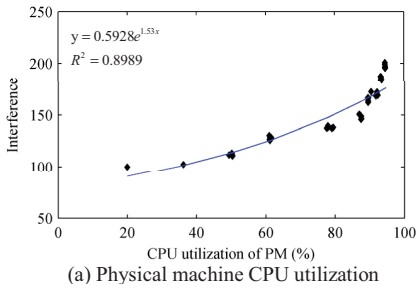
We point out the existence of interference very clearly in last section. When nine VMs are deployed on the PM, the performance of application decreases three times more than this in isolation environment. Therefore, forecasting the CPU-intensive application interference is very necessary.

In order to establish the performance interference prediction mathematical model, we need to choose the model variations. For CPU-intensive applications, the major interference factor is the usage of CPU and cache resources. So we chose the CPU utilization and cache miss rate as variations of the model. In order to characterize the relationship between the interference and resource utilization, the linear regression method and the least squares method are used to confirm whether they are linear or non-linear relationship.

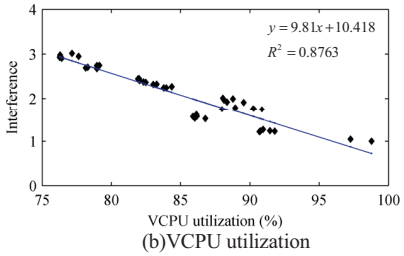
Fig. 7 shows the interference changes with the resource utilization of virtual machines and server. As shown in Fig. 7(a), there is a non-linear relationship between the performance interference and the utilization of PM. Then we can assume there is an exponential relationship between them. In order to verify the assumption, the least squares method is used to obtain the exponential expression. And the correlation coefficient of curve fitting is 0.8989, which indicates the assumption is correct.

Meanwhile, there is a linear relationship between the interference and VCPU utilization in Fig. 7 (b). By using linear regression method, we determine the linear relationship between them. And the correlation coefficient is 0.8763, which indicates that the two parameters have a strong linear correlation.

When an application deploys on a physical machine, the resource usage in isolation environment can be known according to the historical data or a special experiment. At the same time, we can acquire the resources usage of the VM deployed on the server and the resource utilization of server. Through the above analysis, the MVEI model can be expressed as follows:



(a) Physical machine CPU utilization



(b)VCPU utilization

Figure 7. The relationship between the resource utilization and the interference

$$I_i = a + b \cdot R_{cache} + c \cdot N + e^{d + f \cdot U_{host}} + g \cdot \sum_{j=1}^{N-1} U_{vmj} \quad (2)$$

where I_i indicates the interference of virtual machine i; R_{cache} is the cache miss rate of the application i in isolation environment; N is the number of virtual machine; U_{host} donates the current CPU utilization of PM; U_{vmj} is VCPU utilization of the VM j deployed on a physical machine. The rest are coefficients.

2) Newton iterative method

Since MVEI model is a multivariable nonlinear model, the conventional method is very difficult or even impossible to calculate coefficients. We test a large number of data sets and substitute them into the equations to calculate the coefficients. Afterward we get a set of nonlinear equations. These nonlinear equations cannot be calculated by using the formula of its numerical solution. Therefore, we used the Newton iteration method to find the optimal solution.

Newton iterative method is a method for finding successively better approximations to the roots (or zeroes) of a real-valued function. It can be summarized in following steps:

（i）Chose a reasonable initial value;

（ii）Compute the Jacobian matrix;

（iii）Obtain the iteration step according to the Jacobian matrix;

（iv）Judge whether the error meets the precision. If it meets the requirement, finish the program.

（v）If not, judge whether the number of iterations reaches the maximum. If not, turn to the step (ii). If it meets the requirement, finish the program.

Needless to say, the selection of initial value is critical. It is uneasy to convergence if an unreasonable initial value is selected. Therefore, we chose the coefficients calculated by linear regression as the initial value. After multiple iterations, the optimal coefficients can be obtained.

V. EXPERIMENTAL EVALUATION

To verify the accuracy of MVEI model, we compare it with other two models. According to previous researches [11], most CPU-intensive applications interference prediction model is linearly or quadratic. On the basis of these theories, the linear and quadratic interference model of CPU-intensive applications can be donated as follows:

$$linear: I_i = a + b \cdot N + c \cdot U_{host} + d \cdot \sum_{j=1}^{N-1} U_{vmj} \quad (3)$$

$$quadratic: I_i = a + b \cdot N + c \cdot U_{host} + d \cdot U^2_{host} + e \cdot \sum_{j=1}^{N-1} U_{vmj} + f \cdot \sum_{j=1}^{N-1} U^2_{vmj} \quad (4)$$

where I_i indicate the interference of virtual machine i; R_{cache} is the cache miss rate of the application i in isolation environment; N is the number of virtual machine, U_{host} donates the current CPU utilization of PM; U_{vmj} is the VCPU utilization of the VM j deployed on a physical machine. The rest are coefficients. We also use Newton iteration method to calculate coefficients.

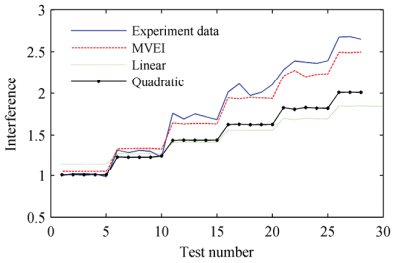


Figure 8. The interference prediction

TABLE II. THE PREDICTION ERROR

	Min	Max	Ava
Linearly	1.16%	30.8%	20.38%
Quadratic	1.24%	25%	14.39%
MVEI	1.07%	7.7%	4.12%

To verify the effectiveness of the model, another procedure (CG) introduced in section III is deployed on the experimental environment to evaluate these models. Three different models are used to predict the interference. Fig. 8 shows the interference prediction of the three models. As can be seen from the figure, the MVEI model can track the performance interference more accurately. Although the quadratic model is more precise than linearly model, it still has a large error. Besides, the linear and quadratic model can't detect the tiny changes of the interference.

Table II shows the maximum, minimum and average prediction error of each model. As shown in Table II, the MVEI model has the stable performance compared with the other two models. Although the minimum error of the three models is very closely, the maximum has a big difference. In general, the MVEI model can predict the interference more accurately.

VI. CONCLUSION

We have investigated the performance fluctuation of CPU-intensive application in cloud computing environment which is different from the traditional virtual environment. As the virtualization technologies can't provide strict performance isolate, the interference exists in co-host applications. Then we analyze the cause of interference and the interfering factor from cache and the CPU usage. To accurately predict the interference, we establish a multivariable exponent interference (MVEI) model according to the analysis. The experimental result indicates that the proposed model has a higher prediction accuracy than the linear and quadratic model in overbooking cloud computing.

ACKNOWLEDGMENT

This work is supported by project (2013AA01A212) from the National 863 Program of China, project (61202121) from the National Natural Science Foundation of China, Science and technology project (2013Y2-00043) in Guangzhou of China and project (20114307120013) from the New Teachers' Fund for Doctor Stations, Ministry of Education of China.

REFERENCES

[1] M. A Vouk, "Cloud computing–issues, research and implementations," CIT. Journal of Computing and Information Technology, vol. 16, Apr. 2008, pp. 235-246.

[2] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," Proc. 2008 conference on Power aware computing and systems. Dec. 2008, pp. 20-40.

[3] A. Berl, E. Gelenbe, M. Di Girolamo, and G. Giuliani, H. De Meer, et. al., "Energy-efficient cloud computing,". The computer journal, vol. 53, Jul. 2010, pp. 1045-1051.

[4] L. Tomás, and J. Tordsson, "Improving cloud infrastructure utilization through overbooking," Proc. 2013 ACM Cloud and Autonomic Computing Conference. ACM Press, Nov. 2013, p.5.

[5] R. Nathuji, A. Kansal, and A. Ghaffarkhah, "Q-clouds: managing performance interference effects for QoS-aware clouds," Proc. 5th European conference on Computer systems, ACM Press, Apr. 2010, pp. 237-250.

[6] Y. Koh, R. Knauerhase, P. Brett, M. Bowman, et.al, "An analysis of performance interference effects in virtual environments," Proc. IEEE Symp. In Performance Analysis of Systems & Software(ISPASS'07), IEEE Press, Apr. 2007, pp. 200–209.

[7] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell, "Modeling virtual machine performance: challenges and approaches," ACM SIGMETRICS Performance Evaluation Review, vol. 37, Dec. 2009, pp. 55-60.

[8] I. S. Moreno, R. Yang, J. Xu, and T. Wo, "Improved energy efficiency in cloud datacenters with interference-aware virtual machine placement," Proc. IEEE Eleventh International Symp. Autonomous Decentralized Systems (ISADS), IEEE Press, Mar. 2013, pp. 1–8.

[9] M. Kambadur, T. Moseley, R. Hank, and M. A. Kim, "Measuring interference between live datacenter applications," Proc. International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE Computer Society Press, Nov. 2012, p. 51.

[10] G. Casale, S. Kraft, and D. Krishnamurthy, "A model of storage i/o performance interference in virtualized systems," Proc. the 31st International Conference on Distributed Computing Systems Workshops(ICDCSW'11), Jun. 2011, pp. 34–39.

[11] R.C. Chiang and H. H. Huang, "Tracon: Interference-aware scheduling for data-intensive applications in virtualized environments," Proc. 2011 International Conference for High Performance Computing, Networking, Storage and Analysis(SC'11), Nov. 2011, pp. 1–12.

[12] Q. Zhu and T. Tung, "A performance interference model for managing consolidated workloads in qos-aware clouds," Proc. IEEE 5th International Conference on Cloud Computing (CLOUD), IEEE Press, Jun. 2012, pp. 170–179.

[13] S. Govindan, J. Liu, A. Kansal, and A. Sivasubramaniam, "Cuanta: quantifying effects of shared on-chip resource interference for consolidated virtual machines," Proc. 2nd ACM Symposium on Cloud Computing(SOCC'11), ACM Press, Oct. 2011, p.22.

[14] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, and C. Pu, "Understanding performance interference of I/O workload in virtualized cloud environments," Proc. 3rd IEEE International Conference on Cloud Computing(CLOUD'10), IEEE Press, Jul. 2010, pp. 51–58.

[15] Y. Mei, L. Liu, X. Pu, S. Sivathanu, and X. Dong, "Performance analysis of network I/O workloads in virtualized data centers,"IEEE Transactions on Service Computing, vol. 14, Jun. 2011, pp.48-63.

[16] http://www.nas.nasa.gov/Software/NPB/

[17] http://oprofile.sourceforge.net/new

MapReduce Model Implementation on MPI Platform

Guo Yucheng

Computer Science Department
School of Computer Science and Technology
Wuhan University of Technology, Wuhan, China
Email: ycheng.g@gmail.com

***Abstract*— With development of Multicore clusters the task-scheduling problem in heterogeneous cluster has become hot point of research. The method to solve this problem in Cloud computing is virtualization, which can make the heterogeneous nodes being isomorphic and then using MapReduce model for task scheduling in isomorphic nodes. But the approach has some shortcomings: virtualization itself will cause the loss of performance; and there are much more disk IOs in the MapReduce model, which can also cause performance degradation. Based on our earlier work which successfully adds fault-tolerance functions in MPI, this paper proposes a MPI based MapReduce approach which implements internodes communication with efficient MPI communication functions to achieve task scheduling on heterogeneous nodes directly by improved work pool and thread pool. By this way the load balancing can be achieved efficiency. The proposed MPI based MapReduce model can efficiently deal with a kind of data intensive as well as computation intensive problems.**

Keywords-MapReduce, MPI, task scheduling, load balancing

I. INTRODUCTION

Multi-core processors and multi-core cluster systems [1] have become popular in recent year. Meanwhile requirements for big data processing are explosive. Generally speaking processing tasks can be classified as three types: the data intensive, the computation intensive and the data as well as computation both intensive problems. Development of cloud computing has proposed a new parallel programming model, the MapReduce model. This model has good universality in dealing with same type data. On the platform of Hadoop, application developers only need to write their own Map function and Reduce function. The Hadoop system can partition the task automatically for parallel execution. But the traditional MapReduce model is usually only applicable for data intensive tasks. It focuses on data processing. There are some shortcomings of solving problems of both data intensive and computing intensive in MapReduce model. The message passing of MapReduce model is realized through the low layer of the distributed file system. The strategy of file system is store-forward, that is loading data firstly and reading out them to forward secondly. It stores all information in disk, and then reads them from the disk while required. When the intermediate data increase and amount of them is big, the model is bound to create a number of useless disk IO operations, that will cause the disk I/O bottleneck definitely, and it is expensive for high performance computing.

MPI (Message Passing Interface) is the most widely used distributed parallel programming tools currently. It has an open source library, and has been supported by a large number of research institutions. MPI has become a parallel scientific computing standard and norms of the representative [2, 3]. MPI is an open source project, has been supported by Argonne national laboratories and numerous universities in United States. It has good scalability, efficiency and portability, since the low layer of MPI is implemented with C language [4].

In scientific parallel computing, MPI with its high efficiency has gained a lot of user recognition. In recent years, for dealing with both data and computation intensive problems, many scholars and experts consider a hybrid model which combines MapReduce model with the MPI [5], implementing the MapReduce programming model on the MPI platform [4] to improve its performance and efficiency, enabling it to support computation-intensive as well as data-intensive distributed processing tasks. However at present researches of the MPI based MapReduce model only stay in proposal stage, some experiment prototypes do not consider the lack of fault tolerance ability in the MPI, but the disaster recovery is not possible without complete fault-tolerant realization, that is the basic requirement in the MapReduce model.

II. KEY TECHNOLOGIES IN TRADITIONAL MAPREDUCE MODEL

The basic character of the MapReduce model is the master-workers structure, by which the nodes are divided into master and workers. Generally speaking there are three key technologies in MapReduce data processing:

1) Data segmentation technology
2) The strategy of task allocation
3) Fault-tolerance and disaster-tolerance technology [6, 7]

The traditional MapReduce model for Worker node fault tolerance is based on redundant backup. The strategy to deal with a Worker node failure is if the Worker has not finished a Reduce work, all works on this Worker must be done again. So all the previous jobs of the worker node are wasted. [8, 9, 10]

III. MPI BASED MAPREDUCE MODEL

A. Three-level fault tolerant technology of MPI

In order to overcome the shortcomings of traditional MapReduce model, this paper proposes a MPI based MapReduce model. However the MPI cannot give a satisfactory support to fault tolerance, which is the key requirement of the MapReduce model, so the earlier work of our team is focused on enhancing the MPI fault tolerant ability. A three-level fault tolerance mechanism has been implemented [8,11]:

1. Task rescheduling. If volume of computing task is not large, then the mission elapsed time is relatively small. In this case when a node fails, all other nodes and the parallel processing task should be terminated immediately,

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.72

and the parallel processing task is put into the end of the task queue. Although the cost in termination of the whole task is relatively large, but it is worth in comparison with reschedule cost. The MPI itself has such function, without modification. From the system design point of view this strategy can be accept for the relatively small tasks.

2. Checkpoint and recovery. If real time requirement of a processing task is not high, the MPI should do task state check at periodic checkpoints. While a node is failure and causes mission failure, the node can be restarted after recovery, and then the interrupted task can be restated from the nearest checkpoint, thereby reducing the loss.

3. Task migration. If a task has a big amount calculation, and requires sensitive real-time characteristic, when a node is failure and other nodes are not affected, then the master node (Master) will do dynamic migration of the unfinished task of the failure node to other nodes for continue processing, thus greatly reducing the adverse effects of MPI task failure.

B. *Monitoring Structure*

In proposed model a monitoring mechanism is required. The monitoring structure is shown in figure 1.

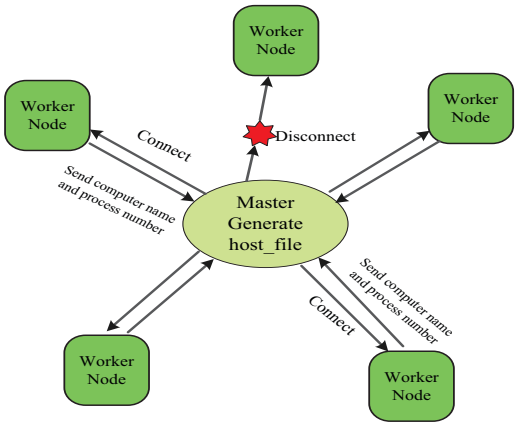


Figure 1 The structure of Monitoring program

C. *The work pool technology and implementation*

Nodes in the cluster are divided into two categories: the Master and the Slaver (worker). Master nodes distribute tasks of Map and Reduce; Worker nodes execute the tasks of Map and Reduce.

The work pool technology has some shortcomings [6]: the main process can only send a task to a secondary process, that is, after the initialization of the task pool, the main process can only response to new task requests by one at a time. Therefore, when the number of the secondary process is big, there may be many secondary processes which apply for tasks at the same time, the main process can only respond to the request of one process, this affects the efficiency of the task distribution. To overcome this shortage a double cache structure is taken in our scheme.

The improvement of the work pool scheduling method is as follow: making each task node has two tasks at the same time, and sending request to the main process soon after the execution of a tasks is completed. The double cache structure can reduce the waiting time of getting a task. In general, the task states of each worker node are: one task is executing, another task is waiting for executing.

D. *The application of the thread pool technology*

Using the thread pool technology has more advantage when the nodes have much more logical processors. First, the runtime cost of thread is much smaller than process. Second, multiple MPI processes on one node are difficult to control, but each node only containing one process is more conducive to control. In the present model, a combine of the improved work pool and thread pool is adopted to distribute the tasks: A task is taken from task queue and divided into first level slice which then assigned to each node. The second division is taken place on each node to produce second level slices, which then are assigned to threads. This method can not only improving efficiency but also can be conducive to load balancing.

IV. THE IMPLEMENTATION OF MPI BASED MAPREDUCE

A. *The system module structure*

The main modules and the relationship of the system are shown in figure 2. There are three major modules:

Job management module: managing the job scheduling, monitoring the implementation of the job, saving the job queue, providing the corresponding fault tolerant ability, saving automatically the last unfinished job, supporting the remote job submitting and result returning.

Monitoring module: managing the available host list. The available host list is the basis data structure of assigning tasks. Monitoring procedures can find the "bad node", and make the "bad node" excluded.

Task management module: the task division, the task assignment, the task execution, getting the result, returning the results.

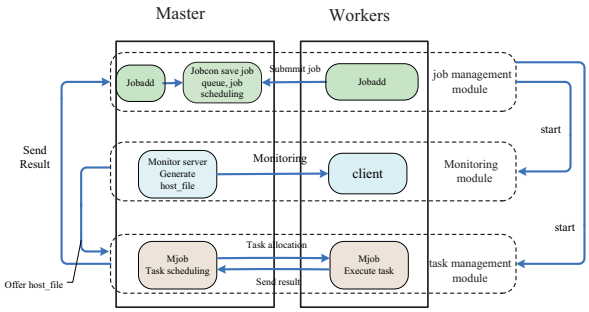


Figure 2 System Block Diagram

B. *The Two Level Slice Data Structure*

The two level slice divisions of processed data are illustrated in figure 3, 4 and 5.

In the figures the contents of each element in the tables are 0, 1, and 2. The number 0 represents that the corresponding division is not executed, 1 means the corresponding division is executed, and 2 means the corresponding division have executed.

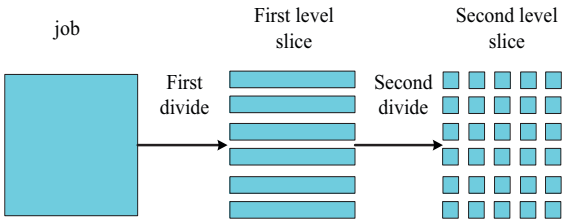


Figure 3 Job division diagram

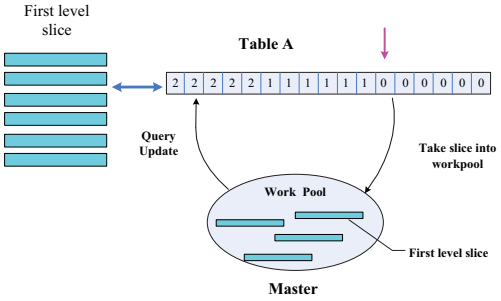


Figure 4 First level slice schedule

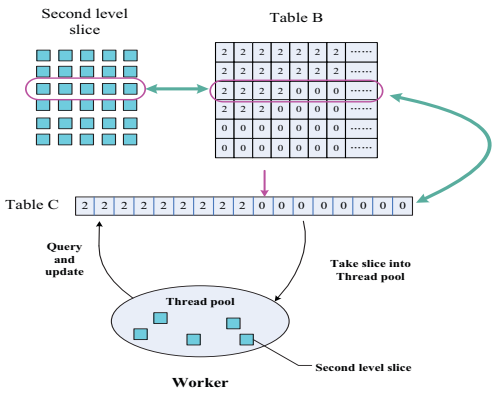


Figure 5 Second level slice schedule

V. THE PERFORMANCE TESTING

The test data are pairs of key/value set, that is the two-tuples set. We take the Hadoop provided "wordcount" data set as the test data set. The test data are divided by row, each row is a Key, and the content of the row is Value. The test case for the system is: finding a batch of numbers that is closest to one number from several documents, the contents of the document are the int type digitals. The Key of the system is the position number of the digital; The Value is digital value of this position, and finally the result of top eight is returned.

The performance comparison of the model and Hadoop is shown in figure 6.

It can be seen from the figure, that the efficiency of the MPI based MapReduce model developed by our laboratory is much higher than the Hadoop MapReduce model. Especially, with bigger size of slice and smaller intermediate data the higher efficiency is achieved. When the sizes of slice are 64MB, 16MB and 4MB, the rates of corresponding execution times (MPI / the Hadoop) are 16.2%, 22.9%, and 23.8% respectively.

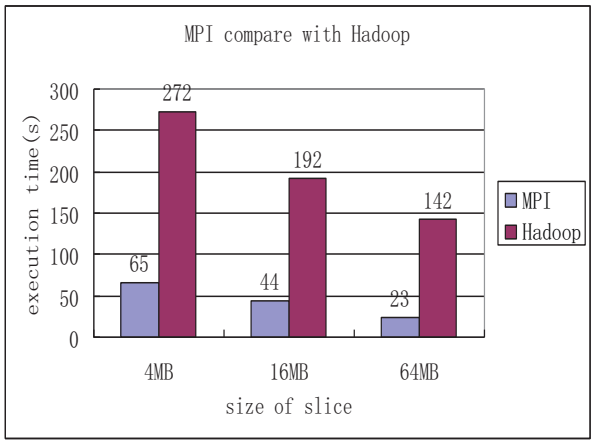


Figure 6 The performance comparison of Hadoop and MPI

The main reasons of the results are: (1) MPI itself has been done some optimization in cluster communication; communication efficiency in MPI is much higher than the direct use of socket communication. (2) Our system adopts a memory mapping strategy to deal with processed file; whole job execution is in memory without frequent disk IO operation; however the Hadoop job execution needs a lot of disk IO: first reading disk file as an input of a Map, writing the input of the Map to disk, then reading out the Map results and taking them as Reduce input, and finally writing the Reduce results to disk. (3) Intermediate results generated in the test cases have relatively small size and big number of multiple disk IOs, and then using a distributed file system to transfer them; however by directly using MPI for communication can achieve much higher efficiency then Hadoop.

VI. CONCLUSION AND FUTURE WORKR

The traditional MapReduce is not efficient. The MPI based MapReduce model, developed in WUHT distributed parallel processing laboratory, has a good Fault-Tolerance ability, so it can deal with data intensive as well as computation intensive problem in MPI Cluster. The experimental results show that the efficiency of MPI based MapReduce model is much higher than the traditional MapReduce model.

At present, the size of first level slice and second level slice of this project is a fixed value, which is a problem we are trying to solve. That is how to adjust the size of the slice automatically so as to improve the versatility of this system. The slice size adjustments should consider cluster size, topology and the size of data which to be processed.

REFERENCES

[1] Nidhi Aggarwal, Parthasarathy Ranganathan, Norman P. Jouppi, James E. Smith. Configurable Isolation: Building High Availability Systems with Commodity Multi-Core Processors, In Proceedings of the 34th Annual International Symposium on Computer Architecture (2007)

[2] Yu-Fan Ho, Sih-wei Chen. A MapReduce Programming Framework Using Message Passing. Computer Symposium (ICS). 2010.11.P 883-888.

[3] Ying Peng, Fang Wang. Cloud Computing Model Based on MPI and OpenMP [J]. Computer Engineering and Technology (ICCET). 2010.4.Vol.7.P 85-87.

[4] Zeng Yan, On MPI-Based Master-Slave Parallel Task Allocation and its Implementation, Journal of Computer Applications and Software, Vol.27 No.6 Jun 2010 (in Chinese)

[5] Torsten Hoefler, Andrew Lumsdaine, Jack Dongarra. Towards Effcient MapReduce Using MPI. Lecture Notes in Computer Science. 2009. P 240-249

[6] M Bhandarkar. MapReduce Programming with Apache Hadoop[J]. 2nd IEEE International Conference on Cloud Computing Technology and Science. 2010, 721-726

[7] Chang F, Dean J, Ghemawat S, et al. BigTable: A distributed storage system for structured data. ACM Trans [J]. on Computer Systems, 2008, 26(2):1−26.

[8] Peng Wu. Research of Key Technology of Fault-tolerant and Disaster Recovery in MPI cluster [D]. Wuhan: Wuhan University of technology, 2012. (In Chinese)

[9] D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

[10] Dongxu Hu. Researcch of Key Technology of MPI-based Multilayered Fault-tolerant high performance cloud platform [D]. Wuhan: Wuhan University of technology, 2013. (In Chinese)

[11] Yucheng Guo, Dongxu Hu, Peng Wu. MPI-based Heterogeneous Cluster Construction Technology[C]. DCABES 2012, Guilin, 2012: 120-124.

Associate Task Scheduling Algorithm Based on Delay-Bound Constraint in Cloud Computing

Yingchi Mao[1,2], Lili Zhu[1], Xi Chen[1], Qing Jie[1]

1 College of Computer and Information
Hohai University
Nanjing 211100, China
e-mail: maoyingchi@gmail.com

2 Huaian Research Institute of Hohai University
Huaian 223001, China

***Abstract*—Task scheduling is one of the most important issues in the cloud computing environments. In the cloud systems, the main goal of the task scheduling algorithms is to balance the workload among the computing nodes and maximize the utilization while meeting the bound of the total execution time. Concerning the delay of the associated tasks scheduling in cloud computing, a hierarchical task model was discussed and the associated task scheduling algorithm based on delay-bound constraint (ATS-DB) was proposed. The associated tasks and the task execution order were represented by one directed acyclic graph (DAG). The proposed hierarchical task model can improve the task execution concurrency. The independent tasks in each layer was grouped into the corresponding task set belonging to the task layer. Through the calculation of the total tasks execution time bound in each task layer, the associated task was dispatched to the resources with the minimum execution time. Extensive experimental results demonstrated that the proposed ATS-DB algorithm can achieve better performance than HEFT algorithm in the terms of the total execution time and resource utilization.**

Keywords- Cloud computing; associated task scheduling; hierarchical task model; delay-bound constraint

I. INTRODUCTION

Cloud computing is emerging as a new paradigm of large-scale distributed computing, which support convenient and on-demand network access to a shared pool of computing resources. Cloud computing can many advantages, including transparency of resources, flexibility, location independence, reliability, and so on [1]. To provide these facilities, the tasks should be scheduled to the appropriate resources in order to achieve maximum performance in minimum time.

Task scheduling is one of most important issues in the cloud computing environments. In the cloud systems, the goal of the scheduling algorithms is to spread the workload among the computing nodes and maximize the utilization while minimizing the total task execution time. The task scheduling process is performed in two stages. In the first stage, the scheduler allocates resources to the cloud as requested by an application. In the second stage, the incoming tasks are assigned to the appropriate virtual nodes in an effort to balance loads within the virtual nodes [2]. To minimize the total execution time of all incoming tasks, the scheduling algorithm should reduce the amount of transferred data and ensure the load balance [3].

At present, the existing task scheduling algorithms have been proposed to meet the different goals considering the different constraints. For example, First Come First Served algorithm (FCFS), Round Robin algorithm (RR), Min-Min algorithm and Max-Min algorithm can achieve good scheduling performance for non-associated tasks. Concerning the delay of the associated tasks scheduling in cloud computing, a hierarchical task model was discussed and the associated task scheduling algorithm based on delay-bound constraint (ATS-DB) was proposed.

The contribution of this paper are as follows:

1) At present, almost research on task scheduling in cloud computing focus on the single and independent task scheduling in order to reduce the complexity of scheduling problem. In this paper, we proposed an associated task model for the associated task scheduling considering the real application requirements in cloud computing.

2) Based on the dependency of associated tasks, a directed acyclic graph (DAG) was presented to describe the execution order for the associated tasks. Also, we proposed the hierarchical task graph to decompose the associated tasks, which can improve the tasks execution concurrency and reduce the execution time.

3) In order to execute all of the associated tasks in the specific delay-bound, we proposed the concept of tasks processing capacity and the corresponding calculation method, and further established the mapping between the task processing capacity and execution time.

The rest of the paper is organized as follows: Section 2 addresses the existing task scheduling algorithms. In section 3, the associated tasks scheduling model is defined. The details of proposed associated tasks scheduling algorithm is discussed in Section 4. Extensive experiments results are presented in Section 5. Section 6 concludes the paper and discussed some future work.

II. RELATED WORK

Task scheduling algorithms can be categorized in two main groups in cloud computing: Batch mode scheduling algorithms, and online mode algorithms. For the first group, tasks are queued and gathered into a set when arriving in the cloud. The scheduling algorithm will execute after a fixed period of time. The typical examples of batch mode scheduling algorithms are First Come First Served algorithm (FCFS), Round Robin algorithm (RR), Min-Min algorithm and Max-Min algorithm, Genetic Algorithm (GA)[8]. These scheduling algorithms can achieve good performance for a group of independent tasks. A group of independent tasks simplified the tasks model for the real applications in the cloud. In fact, the real application, there exist a large of number of associated tasks. These tasks have dependency and the execution order. If one of the associated tasks has delayed beyond the execution time bound, the execution of its successors will result in the delay affected by its predecessors. Thus, the

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.22

total execution time will beyond the required time-bound without providing good QoS [4].

For online mode algorithms, tasks are scheduled when arriving in the cloud. Because the cloud computing environment is a heterogeneous system, and the computing performance of each node are quite different, the online scheduling algorithms are more appropriate for the cloud computing. However, the existing research online mode algorithms pay little attention on the associated tasks scheduling. There has been some recent research on the time-bound constraint for various scheduling strategies for parallel tasks. This work fall in two categories. In decomposition-based strategies, the parallel tasks is decomposed into a set of sequential tasks and they are scheduled using existing sequential scheduling algorithms. In general, decomposition-based strategies require explicit knowledge of the structure of the DAG off-line. In non-decomposition based strategies, the program can unfold dynamically since no advent knowledge is required. Hariri et al. proposed a heterogeneous earliest-finish-time algorithm HEFT, to solve the scheduling delay problem in the heterogeneous environments [5]. Du et al. adopted the directed Acyclic Graph to represent the dependency among the associated tasks, and applying cluster algorithm to reduce the resource searching cost, and furthermore, shorten the execution time for all of the tasks [6]. Wilmer et al. presented the scheduling algorithm to select the appropriate resource, by setting the completion time in each tasks layer with the decomposition-based method [7].

All of the mentioned algorithms, there is no research on the associate tasks scheduling considering the execution time bound. In this paper, a hierarchical task model was discussed and the associated task scheduling algorithm based on delay-bound constraint (ATS-DB) was proposed. The proposed hierarchical task model can improve the task execution concurrency, and ATS-DB can reduce the execution time.

III. TASK SCHEDULING MODEL

In this paper, all of the mentioned tasks are referred to the computing tasks. The correlation of the tasks are reflected on the execution order of tasks. That is to say, each one task has its own previous related tasks and/or the successive related tasks. For each one task, it is ready to be executed just after all of its predecessors have been executed. In order to explicitly describe the correlation of the associated tasks, we consider a general model for associated tasks, namely the DAG model. Each task is characterized by its execution pattern, defined by a directed acyclic graph (DAG). In a DAG, each node represents an associated task and each directed edge represents dependency between tasks. As shown in Fig.1, the directed edge connected node 1 and 3. It represents that the task τ_1 is the predecessors of task τ_2. Task τ_2 should execute after the tasks τ_1 has finished.

To order explicitly describe a group of associate tasks, their dependency, and communication cost, a three tuple is applied to represent it.

Definition 1: $G = \{T, \Lambda, C\}$ is defined to describe a group of the associate tasks and their correlations.

1) Given n associated tasks, set $T = \{\tau_1, \tau_2, ..., \tau_n\}$ is all of associated task in a set. $\forall \tau_i \in T$, $i \in [1,n]$, τ_{i_id} and τ_{i_length} represent the id and the length of task τ_i, respectively. In the same condition, the tasks with longer length will cost the more execution time.

2) Λ represents the set of execution order for the adjacent tasks $< \tau_i, \tau_j >$, in which τ_i is the predecessors of task τ_j

3) $C = \{c_1, c_2, ..., c_k\}$ represents the set of communication cost between the adjacent tasks $< \tau_i, \tau_j >$. If the adjacent tasks τ_i, τ_j are dispatched to the same virtual node, it can be considered the communication cost for the adjacent tasks $< \tau_i, \tau_j >$ is zero.

To clear illustrate the Definition 1, Figure 1 shows a DAG containing 10 associated tasks. In this DAG, the first executed task is τ_1, and the last task is τ_{10}. The label of the edge represents the communication cost between two adjacent tasks.

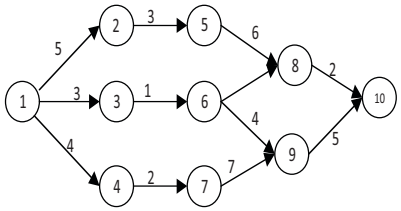


Figure 1. An example of the associated tasks (DAG)

Definition 2: Resources Set. In the Cloud computing system, there are m resources, including computing, storage, and networks. The resource set is represented as $R = \{r_1, r_2, ..., r_j, ..., r_m\}$, in which $r_j (j \in [1, m])$ denotes the j^{th} resource in the set. For $\forall r_j \in R$, r_j has four characteristics: id, computing capacity, storage capacity, and network bandwidth.

They are denoted as r_{j_id}, $r_{j\_computing}$, $r_{j\_storage}$, and r_{j_bw}, respectively. Meanwhile, r_j denotes Vector $r_j = [r_{j\_id}, r_{j\_computing}, r_{j\_storage}, r_{j\_bw}]$.

IV. ATS-DB SCHEDULING ALGORITHM

A. Hierarchical Decomposition Method

In order to improve the concurrency of the computing tasks, and reduce the execution time delay, the hierarchical decomposition method is adopted. We decompose the associate tasks into different tasks layer based on the correlations among the associated tasks. In each hierarchical layer, the tasks are independent without any the correlation in the execution order. Thus, the tasks in the same hierarchical layer will be grouped into one tasks set. These tasks in the same layer can be concurrently scheduled. The details of the hierarchical decomposition steps are as follows:

a) The dispatcher receivers a task set T of n associated tasks $\{\tau_1, \tau_2, ..., \tau_n\}$;

b) Based on the dependency order of tasks' execution, these associated tasks are used to establish the corresponding DAG;

c) Adopting the DAG traversal method from the starting task to the last task, the structure of task hierarchy is also constructed. In each one task hierarchy, there is no dependency among these tasks. That is to say, all of the tasks are concurrent tasks in the same layer. If two tasks are in the adjacent layers, the two tasks are considered as the associated tasks;

d) Based on the tasks hierarchy, the tasks sets are established. The tasks in the same task layer are grouped into one set.

Adopting the hierarchical decomposition method for the associated tasks, it can greatly improve the concurrency performance and fully utilize the system's resources when executing the task scheduling algorithm. Thus, it can ensure all of the tasks in one set finished at the delay-bound. In order to demonstrate the hierarchical decomposition method, we adopts data from Fig. 1 and Table 1 to give an illustrative example. From Fig.1, the DAG can be divided into four layers. The first layer just has one node τ_1. $\{\tau_2, \tau_3, \tau_4\}$ and $\{\tau_5, \tau_6, \tau_7\}$ belong to the second and third layer, respectively. τ_2, τ_3, τ_4 are the successor of τ_1, and the predecessors of τ_5, τ_6, and τ_7, respectively. The detail is shown in Table 1.

TABLE I. AN INSTANCE OF TASKS HIERARCHY

layer	Task sets
L_1	$\{\tau_1\}$
L_2	$\{\tau_2, \tau_3, \tau_4\}$
L_3	$\{\tau_5, \tau_6, \tau_7\}$
L_4	$\{\tau_8, \tau_9\}$
L_5	$\{\tau_{10}\}$

B. Calculation Delay Bound

As to the task scheduling in Cloud computing system, all of resources referring to the computing, storage, and bandwidth capacities, are quite different. When one task is dispatched to the different resources, the execution time may be much different due to the different computing capacities. Therefore, it should compute the execution time based on the computing capacities allocated before executing the scheduling algorithm. It can ensure the total execution time within the delay-bound. In order to accurately compute the execution time for the specific task, some definitions are given as follows.

Definition 3: Processing capacity P_{ij}. P_{ij} is denoted as the processing capacity of resource r_j for the task τ_i. The greater P_{ij}, the shorter execution time. Considering the heterogeneous resources, P_{ij} is defined as the ratio of the length of task $\tau_{i.length}$ to the computing capacity of resource $r_{j.computing}$, and can be calculated as Formula (1).

$$P_{ij} = \frac{\tau_{i.length}}{r_{j.computing}} \tag{1}$$

Definition 4: Average processing capacity of task τ_i, $\overline{P_i}$. If there are *m* resources to process the task τ_i, $\overline{P_i}$ can used as the estimation of every processing capacity for the task τ_i. Average processing capacity of task τ_i, $\overline{P_i}$ is calculated as:

$$\overline{P_i} = \frac{\sum_{j=1}^{m} P_{ij}}{m} \tag{2}$$

Definition 5: Average processing capacity of all tasks $\overline{P}$. If there are *n* tasks for scheduling, $\overline{P}$ can defined as the average processing capacity for all task. For every one task, it can calculated from Formula (2). $\overline{P}$ is calculated as:

$$\overline{P} = \sum_{j=1}^{n} \overline{P_i} \tag{3}$$

According to the above definitions, for the specific task τ_i, if the average processing capacity $\overline{P_i}$ is greater, task τ_i can obtain the shorter execution time. Thus, the processing capacity of the task can indirectly represent the execution time delay.

Definition 6: Delay-bound of the k^{th} task layer Γ_k. If the set of associate tasks has *n* tasks, the corresponding DAG is divided into *K* task layers. The average number of the tasks in every layer z_k, the average processing capacity $\overline{P}$, the delay bound in the k^{th} task layer Γ_k can be calculated as:

$$\Gamma_k = \frac{z_k}{n} \overline{P} \tag{4}$$

C. Detail of ATS-DB Algorithm

To schedule the associate tasks to meet the delay bound, the associate scheduling algorithm based on the delay-bound constraint (ATS-DB) was proposed in this section. The main idea of ATS-DB algorithm is as follows:

a) When the system received all of the task scheduling requests, based on their dependency in the execution order, all of the associated tasks construct the corresponding task association graph, namely DAG.

b) Two resource queues, resource ready queue and resource waiting queue, are established. All of the resources (computing, storage, and network) are sorted in descending order based on the resources processing capacity. Meanwhile, all of the resources are grouped into the ready queue.

c) Adopting the hierarchical decomposition method, the DAG is divided into multiple tasks layers. The tasks in the same layer are included as the corresponding task set.

d) For every task layer, the corresponding delay-bound Γ_k can be calculated. Due to no dependency among the tasks in the same layer, they can be concurrently scheduled.

e) The dispatcher will search the suitable resource from the ready queue to execute those tasks.

f) After those tasks have been completed, the occupied resources are released and put into the waiting queue.

The pseudo-code of ATS-DB scheduling algorithm is shown in Fig. 2.

```
Input: G={T,Λ,C} ;
Divide the DAG to L layers
For each τ_i in L layers
    Task set T_k ← τ_i
        Compute Γ_k
        Calcuate the priority of Task set T_k
While ( T_k is not empty) do
    τ_i ← tasks from Task set T_j
    S_j ← resources from resource ready queue
 If ( d_k < Γ_k )
    Send τ_i to S_j
  else
  Search resource from resource waiting queue
     Send τ_i to resource
  End
Allocate the resources to the resource waiting queue
```

Figure 2. The pseudo-code of ATS-DB scheduling algorithm

V. SIMULATION EVALUATION

To evaluate the better performance on tasks' executing time for the proposed ATS-DB, in this section, we compare ATS-DB with HEFT in terms of the time span and time delay for all of the associated tasks.

A. Experiments Settings & Methodology

The simulation tool, CloudSim, is used to simulate the procedure of task scheduling. In the CloudSim, each virtual node represents one resource and has one processor. All of the requested tasks are executed on the virtual nodes. The computing capacity of virtual nodes varies from 500 to 1500. The storage capacity is set from 1,000,000 to 2,000,000, and the bandwidth of the network is set between 10,000 and 20,000.

For the associated tasks were established by DAGs. All the DAGs are generated by the DAG graph random generator, in which the task numbers $|V|=\{50,100,150\}$, the number of task layer is $|K|=\{7,10,13\}$, respectively. The computer resources pool can be randomly generated from the interval [5, 10], while the task executing time is randomly generated from the interval [5, 30] and the corresponding cost is inversely proportional to the time. In the experiments, we set 50 virtual nodes to establish a heterogeneous computing environment. Adopting ATS-DB and HEFT algorithm to evaluate the execution performance with different number of the associated tasks.

To evaluate the scheduling performance of the proposed ATS-DB, the experiments uses two merits to indicate the scheduling performance. One is the execution time span, denoted as $\Delta span$. $\Delta span$ is defined as the time difference from the first task execution time τ_{i_start} to the last task completion time τ_{j_end} among a group of the associated tasks.

$$\Delta span = \tau_{i\_start} - \tau_{j\_end} \qquad (5)$$

The smaller is the execution time span, the better performance has the scheduling algorithm.

The second is the delay ratio of task in k^{th} task layer, denoted as $Ratio_{k\_delay}$. $Ratio_{k\_delay}$ is defined as the ratio of the real execution time T_{k_real} to the specified execution time T_{k_spec} for all tasks in each task layer.

$$Ratio_{k\_delay} = \frac{T_{k\_real}}{T_{k\_spec}} \qquad (6)$$

The smaller is the delay ratio of task, the better performance has the scheduling algorithm. If $Ratio_{k\_delay} > 1$, it can be considered as the delay when executing all of the tasks in each layer.

B. Simulation Results

1) Time Span: We evaluate the scheduling efficiency in terms of the execution time span under a varying number of associated tasks, ranging from 50 to 150. Figure 3 and 4 illustrate the execution time span by applying the proposed ATS-DB and HEFT scheduling algorithm with 20 and 40 virtual nodes, respectively. As Figure 3 and 4 shown, the proposed ATS-DB scheduling algorithm can obtain shorter time span, compared with HEFT algorithm in the different number of asociated tasks. On the other hand, with the increase of the number of virtual node, the average number of tasks in one virtual node decreases. Thus, the average time span are also reduced. Moreover, the time span increases with the number of associated tasks.

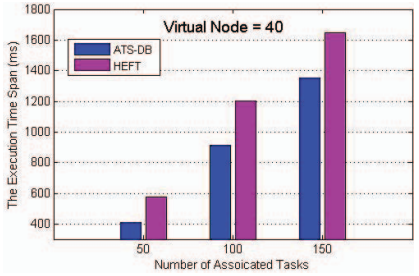


Figure 3. The execution time span with virtual node = 40

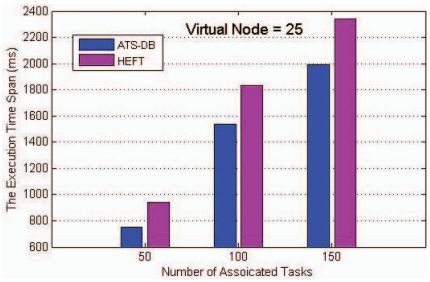


Figure 4. The execution time span with virtual node = 25

2) Delay Ratio: To evaluate the execution efficiency, we compare ATS-DB with HEFT in terms of the delay ratio of task in different number of associate tasks, 50, 100, and 150. In the three groups of associate tasks, the number of task layers in three DAGs are 7, 10, and 13, respectively. Figure 6 illustrates the delay ratio of ATS-DB and HEFT with 50, 100, and 150 associate tasks, respectively. From the results in Figure 5, 6, and 7 shown,

ATS-DB has smaller delay ratio than that of HEFT when the number of associate tasks is the same. ATs-DB can reduce the execution delay ratio by 10% to 50%. On the other hand, with the increase in the number of associate tasks from 50 to 150, the execution delay ratios of two scheduling algorithms also increase. The reason is that the increase of the number of associate tasks can increase the execution time delay, which results in the increase of the delay ratio.

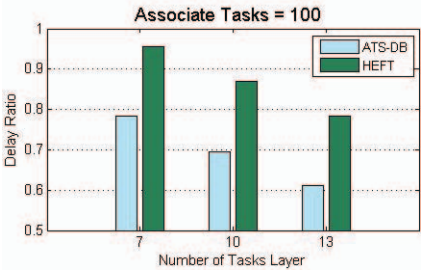


Figure 5. The delay ratio with associate tasks = 100

Furthermore, Fig. 5 shows the execution delay ratio with 7, 10, and 13 task layers when the number of associate tasks is 100. As Fig. 8 shown, with the increase in the number of task layers, the number of concurrent tasks in each layer decreases. Thus, the execution delay ratio can be reduced. From the simulation results, we can obviously find that the proposed ATS-DB algorithm can get a better performance that HEFT algorithm.

VI. Conclusion and Future Work

Task scheduling is one of the most important issues in the cloud computing environments. In the cloud systems, the main goal of the task scheduling algorithms is to balance the workload among the computing nodes and maximize the utilization while meeting the bound of the total execution time. Concerning the delay of the associated tasks scheduling in cloud computing, a hierarchical task model was discussed and the associated task scheduling algorithm based on delay-bound constraint (ATS-DB) was proposed. The associated tasks and the task execution order were represented by one directed acyclic graph (DAG). The proposed hierarchical task model is to improve the task execution concurrency. The independent tasks in each layer was grouped into the corresponding task set belonging to the task layer. Through the calculation of the total tasks execution time-bound in each task layer, the associated task was dispatched to the resources with the minimum execution time. Extensive experimental results demonstrated that the proposed ATS-DB algorithm can achieve better performance than HEFT algorithm in the terms of the total execution time and resource utilization. In this paper, the communication costs among the associate tasks are ignored. In the future work, the communication costs will be considered to meet the requirements of real applications.

Acknowledgment

This research is partially supported by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant No. 2013BAB06B04; Key Technology Project of China Huaneng Group under Grant No.HNKJ13-H17-04; Nature Science Fund of Jiangsu Province under Grant No. BK2012584, and Open Fund of Huaian Research Institute of Hohai University.

References

[1] A. Delavar, M. Javanmard, and M. Shabestari, and M Talebi, “Reliable Scheduling Distributed in Cloud Computing”, in International Journal of Computer Science, Engineering and Applications (IJCSEA), Vol. 2, No. 3, June 2012.

[2] R. Patel and S. Patel, “Survey on Resource Allocation Strategies in Cloud Computing”, International Journal of Engineering Research & Technology (IJERT), Vol.2, No. 2, Feb. 2013, 1-5.

[3] L. Guo, S. Zhao, S. Shen, and C. Jiang, “Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm”, Journal of Networks, Vol. 7, No. 3, March 2012, 547-553.

[4] G. Yan, J. Yu, X. Yang, “Workflow scheduling strategy based on the reliability in Cloud computing”, Computer Applications, Vol.34, No.3, March 2014, 673-677.

[5] Hariri S, Wu M. Performance-effective and low-complexity task scheduling for heterogeneous computing [J]. Parallel and Distributed Systems, IEEE Transactions on, 2002, 13(3): 260-274.

[6] X. Du, C. Jiang, G. Xu, and Z. Ding, “A Grid DAG Scheduling Algorithm Based on Fuzzy Clustering”, Journal of Software, Vol. 17, No. 11, November 206, 2277-2288.

[7] Wilmer D, Klos T, Wilson M. Distributing Flexibility to Enhance Robustness in Task Scheduling Problems[C], Proceedings BNAIC, 2013: 344-351.

[8] J. Huang, “The Workflow Task Scheduling Algorithm Based on GA Model in the Cloud Computing Environment”, Journal of Software, Vol. 9, No. 4, April 2014. 873-880.

[9] Tabatabaee H, Akbarzadeh-T M R, Pariz N. Dynamic task scheduling modeling in unstructured heterogeneous multiprocessor systems. 2014.

[10] Li J, Saifullah A, Agrawal K, et al. Capacity Augmentation Bound of Federated Scheduling for Parallel DAG Tasks. Tech. Rep. WUCSE-2014-44, Washington University in St Louis, USA, 2014.

[11] Liu Z, Qu W, Liu W, et al. Resource preprocessing and optimal task scheduling in cloud computing environments . Concurrency and Computation: Practice and Experience, 2014.

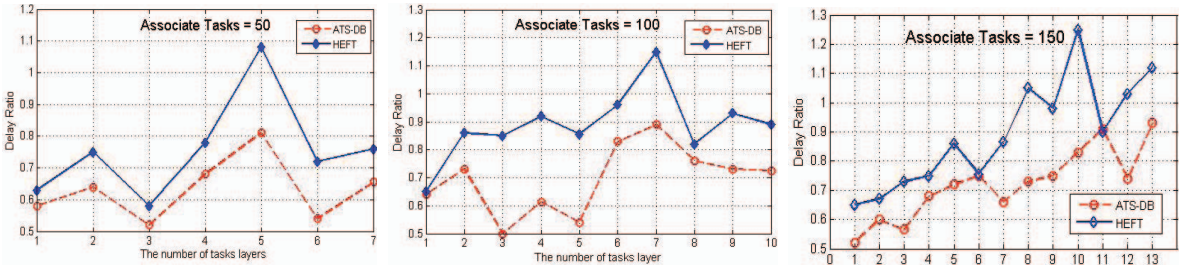


Figure 6. The delay ratio with associate tasks = 50, 100, 150, respectively

Topic Detection in Twitter Based on Label Propagation Model

Dongxu Huang
School of Automation
Northwest Polytechnical University
Xi'an, China
e-mail:huangdongxu21@mail.nwpu.edu.cn

Dejun Mu
School of Automation
Northwest Polytechnical University
Xi'an China
e-mail: mudejun@nwpu.edu.cn

***Abstract*—Many kinds of huge amount of tweets about real-world events are generated everyday in Twitter. However, the disorganization messages required to be classified by topics and events are one of challenges to get knowledge effectively. To solve the problem, we propose a novel method that combines the cluster algorithm with label propagation algorithm to detect topics in twitter. First, we use canopy cluster algorithm to cluster tweets, canopy cluster algorithm could divides a tweet into different clusters, and the tweet which only belongs to one cluster will be labeled. Second, the mechanism of label propagation is used to label the tweets that in the overlapping of different clusters. In order to evaluate our algorithm, we use two baseline algorithms, LDA (Latent Dirichlet Allocation) and Single-Pass cluster algorithm. We apply three algorithms on tweet dataset with three topics and some noisy data, and experiment results show our method outperforms other algorithms on precision and recall rate.**

Keywords-topic detection; twitter ; cluster algorithm; label propagation model

I. INTRODUCTION

A variety of messages are posted in twitter everyday, and twitter trends to show the hottest terms in most recently. However, we would like to know which topics are discussed by people in twitter recently. A topic is related with one or more events, and described as a set of terms generally.

Many previous works have attempt to detect topics in twitter use different models, such as term frequency model[1-3], probabilistic model[4-7], vector space model[8,9], graph model[10-11] etc.. Most algorithms have been proposed divided tweets into different clusters which have no overlapping, as shown in Fig. 1(a), a tweet only belongs to one cluster. However, the meaning of some tweets is confused, and these tweets are hardly divided into a cluster exactly. The probability of the word in a document is computed by probabilistic model used to detect topics in twitter [4-7, 12], however, a tweet is a small document which no more than 140 characters, and the probability of the word in a tweet is not accurate.

In this paper, we propose a novel method named C-LPA algorithm based on the combination of canopy cluster algorithm and label propagation algorithm for topic detection, and it is never considered as far as we know. In canopy cluster algorithm, only the tweets which belong to just one cluster are labeled, while the other tweets in the overlapping of different clusters are unlabeled, as shown in Fig. 1(b). Labeled tweets consist a topic certainly, however unlabeled tweets are not sure. In C-LPA algorithm, we computes the probability that the unlabeled tweet belongs to each label by LPA (Label Propagation Algorithm), and this process is iterated till the probability is converged. Finally, the label with highest probability will be the choice for the tweet.

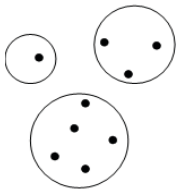


Figure 1(a). Traditional cluster algorithm

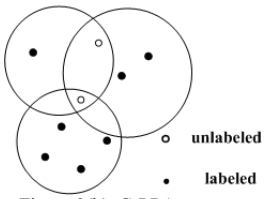


Figure 2(b). C-LPA

The contributions of this paper are two folds: (1) the model that combines canopy cluster algorithm and label propagation algorithm for tweet topic detection is proposed; (2) the label propagation algorithm is implemented in a distribution system named Hadoop, and experimental results shows our method is not only more precision than other algorithms, but also achieves high recall rate.

The remaining of the paper is organized as follows: Section 2 provides an overview of state-of-the-art approaches for twitter topic detection. Section 3 presents the proposed canopy cluster algorithm and label propagation model for topic detection. We report the experimental results in Section 4. The conclusion is presented in Section 5.

II. RELATED WORK

In the scope of topic detection, as an classic algorithm LDA has good performance with rich corpus[12], the academic works based on LDA for tweet topic detection can be found in [3-5,13]. However, the tweet which contains no more than 140 characters is considered as a document, and the difficulty lies in the probability of each word appears in a document is not accurate. Therefore in C-LPA, we computer the similarity among tweets, rather than compute the probability of each word appears. Single-Pass cluster algorithm is a good method for clustering tweets in topics with high efficiency [14], whereas it is sensitive to the order of inputs. Some other cluster algorithms divide a tweet into a topic exactly such as [15, 16], but the fact is that a tweet maybe belongs to different topics. To improve this, C-LPA

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.23

uses canopy cluster algorithm to cluster tweets. Term frequency model detects a certain topic using the mechanism of counting the co-occurrence words. However, some popular words which often appear in hottest tweets will be recognized as keywords of a topic mistakenly. To overcome this problem, in C-LPA, popular words are given low weights in feature selection phase. In addition, Hassan Sayyadi adopts a graph model to detect topic communities and finds inter-words in two topic communities [10]. This algorithm has to computes the shortest paths between all pairs of nodes during every iteration with high time complexity. In C-LPA, we use LPA and compute similarity matrix between all tweets only once.

III. TOPIC DETECTION IN TWITTER BY C-LPA

A. Feature Selection

Traditional feature selection algorithm such as TF-IDF (Term Frequency-Inverse Document Frequency) has drawbacks in selecting features from tweets. More specifically, tweet is a short text, and the term frequency of some terms is usually equal to one, approximate to a fixed value. Moreover we notice that the keywords of topics may appear in many tweets, so the inverse document frequency of keywords is low. We weight terms using the approach introduced in [3], where the weight of keywords within a topic is high and of others words is low. In this method, we have two corpuses, the first corpus for detecting topic named general corpus, the second corpus for reference named reference corpus.

We select candidate terms in tweets by Lucene[1], which helps us to segment words, filter out special characters and stop words.

We denote c as the corpus, $T = \{t_1, t_2 ... t_m\}$ as the set of terms, and $P(t_i \mid c)$ as the frequency of t_i in c, that is,

$$P(t_i \mid c) = \frac{N_{t_i} + \delta}{\sum_{j=1}^{m} N_{t_j} + \delta m} (1 \le i \le m)$$

Where N_{t_i} is the number of t_i, and δ is a smooth parameter. We set it to 0.5 in our algorithm and explain the reason in Section 4.

The weight of t_i in general corpus is computed by:

$$w_{t_i} = \frac{P(t_i \mid c_{\text{gen}})}{P(t_i \mid c_{ref})}$$

Where c_{gen} is the general corpus, c_{ref} is the reference corpus, $P(t_i \mid c_{gen})$ is the frequency of t_i in c_{gen}, $P(t_i \mid c_{ref})$ is the frequency of t_i in c_{ref}. Then the space vector of tweet is described as $tw = ((t_1, w_{t_1}), (t_2, w_{t_2}) ... (t_k, w_{t_k}))$.

B. Label Tweets by Canopy Cluster Algorithm

We first apply the canopy cluster algorithm to label tweets before the label propagation algorithm is used to detect topics. We label a tweet if the distance from this tweet to a certain cluster is smaller than D_1, and to other clusters is bigger than D_2, where D_1 and D_2 are the threshold, otherwise the tweet is unlabeled.

The Canopy cluster algorithm for cluster tweets is detail described as follows:

Input: Initial tweet set $TW = \{tw_1, tw_2 ... tw_n\}$, choose two threshold D_1 and D_2, $D_1 < D_2$, initial cluster set $C = \{\}$, a cluster is consist by some tweets.

Output: Cluster set C.

Step 1. Select a tweet tw from TW, Compute the cosine distance between tw and the center of all clusters. (If the cluster set is null, the first cluster consists of tw).

for $i = 1:1:|C|$ ($|C|$ is the size of cluster set)

$d_i = cosine(tw, h_i)$ (h_i is the center of C_i, $h_i = \sum_{tw_j \in C_i} tw_j / |C|$)

Step 2.

if($d_i \le D_1$) than remove tw from TW, and move tw to C_i,

if($D_1 < d_i \le D_2$), than only move tw to C_i,

Step 3. Repeat Step 1 and Step 2 until TW is null.

We describe the method of calculating D_1 and D_2 as follows: we prepare three corpuses manually, and every corpus contains 1000 tweets belonging to the same topic. We randomly extract 300 tweets from each corpus in a group, and divide each group into three parts with the same size. Compute cosine distance of any two tweets from different parts. The max cosine distance between two tweets in the same group is D_1, the min cosine distance between two tweets in different group is D_2. We repeat this process ten times, and get the mean of D_1 and D_2 which are showed in table 2. Group one is consist of part 1,2,3, group two is consist of part 4,5,6, group 3 is consist of part 7,8,9. We only show the max distance between tweets from different parts which in the same group, and the min distance in different groups, because the distance between two parts is symmetrical, we discard duplicate value. From the Table 2, we get $D_1 = 0.06$, $D_2 = 0.1$ (written by italic).

[1] http://lucene.apache.org/

TABLE 2 THE MAX/MIN DISTANCE BETWEET TWEETS

parts	1	2	3	4	5	6	7	8	9
1		*0.017*	*0.015*	0.356	0.425	0.361	0.425	0318	0.254
2			*0.036*	0.298	0.216	0.198	0.248	0.264	.0314
3				0.249	0.215	0.317	0.241	***0.101***	0.281
4					*0.023*	*0.018*	0.194	0.316	0.321
5						***0.060***	0.213	0.205	0.341
6							0.105	0.211	0.187
7								*0.029*	*0.041*
8									*0.012*
9									

C. Label Tweets by Label Propagation Algorithm

Since there are some labeled tweets and some unlabeled tweets, we use LPA to label the unlabeled tweets. $TW=\{tw_1,tw_2,...tw_n\}$ is the tweet vector set, $L=\{l_1,l_2,...l_m\}$ is the label set, the label vector of tw_i is $f_i=\{p_{i1},p_{i2}...p_{in}\}$, p_{ij} is the probability of t_i belongs to l_j, if the tweet t_i has been labeled as l_k, than $p_{ik}=1$, and the other vector components is equal to 0. The LPA for labeling tweets is detail described as follows:

Input: tweet vector set $TW=\{tw_1,tw_2,...tw_n\}$, label set $L=\{l_1,l_2,...l_m\}$, the initial label vector of tw_i, $f^0{}_i=\{p_{i1}{}^0,p_{i2}{}^0...p_{in}{}^0\}$.

Output: the convergent label vector of each tweet from TW.

Step 1. Initial tweet similarity matrix $S_{n\times n}$, weight matrix $W_{n\times n}$. S_{ij} denotes the cosine similarity between tw_i and tw_j, $S_{ij}=1-\cos(tw_i,tw_j)$, $W_{ij}=S_{ij}/\sum_j S_{ij}$.

Step 2. Compute the label vector of each unlabeled tweet iteratively.

$f_i^n=\alpha W\times f_i^{n-1}+(1-\alpha)f_i^0$, α is a factor, and $0<\alpha<1$, in each vector components,

$$p_{ik}{}^n=\sum_j W_{ij}p_{jk}{}^{n-1}.$$

Step 3. Repeat Step 2 until f_i^n convergence.

Step 4. Label the tweet with the highest probability label.

We proof f_i^n will converge to $(1-\alpha)(\mathrm{I}-\alpha W)^{-1}f_i^0$ as follows:

We prove a lemma firstly.

Lemma: Matrix $\mathbf{A}_{n\times n}$ is a positive matrix($a_{ij}>0$), if $\sum_j a_{ij}=\rho$, then $\rho(\mathbf{A})=\rho$

Proof: From the theorem of Perron-Frobenius [17], the Peron vector is the unique vector defined by

$$\mathbf{Ap}=r\mathbf{p},\ \mathbf{p}>0,\ \text{and}\ \|\mathbf{p}\|_1=1,\ r=\rho(\mathbf{A})$$

now $\mathbf{e}=(1,1...1)^T$, $\mathbf{e}>0$, since $\sum_j a_{ij}=\rho$, $\mathbf{Ae}=\rho\mathbf{e}$, thus $\mathbf{e}$ must be a positive multiple of the Perron vector $\mathbf{p}$, obviously $\mathbf{p}=\frac{1}{n}\mathbf{e}$, because of $\|\mathbf{p}\|_1=1$, Therefore,

$$\rho=r=\rho(\mathbf{A}).$$

Theorem: f_i^n will converge to $(1-\alpha)(\mathrm{I}-\alpha W)^{-1}f_i^0$

Proof: Consider $W_{n\times n}$ is a positive matrix, and $\sum_j W_{ij}=1$. From Lemma , $\rho(\mathrm{W})=\sum_j W_{ij}=1$. Because $W_{ij}>0$, so $\Delta k>0$, Δk is the *k*th order principal minors of $W_{n\times n}$. $W_{n\times n}$ is a symmetric matrix, $\Delta k>0$, so $W_{n\times n}$ is a positive definite matrix. Assume $\lambda_1,\lambda_2...\lambda_n$ is the eigenvalue for matrix $W_{n\times n}$, $\lambda_i>0$, because $W_{ij}>0$, $\sum_j W_{ij}=1$, so $W_{jj}<1$, $\sum_{i=1}^n\lambda_i=\sum_{j=1}^n W_{jj}<\sum_{j=1}^n 1=n$, and exist λ, $\lambda<1$. Assume $\mathbf{v}$ is an eigenvector for λ, $W^n\mathbf{v}=\lambda^n\mathbf{v}$, $(\lim_{n\to\infty}W^n)\mathbf{v}=\lim_{n\to\infty}W^n\mathbf{v}=\lim_{n\to\infty}\lambda^n\mathbf{v}=\mathbf{v}\lim_{n\to\infty}\lambda^n=0$, since $\mathbf{v}\neq 0$, so $\lim_{n\to\infty}W^n=0$. In label propagation algorithm, the iterative formula is $f_i^n=\alpha W\times f_i^{n-1}+(1-\alpha)f_i^0$, and we can get $f_i^n=(\alpha W)^{n-1}f_i^0+(1-\alpha)\sum_{j=0}^{n-1}(\alpha\,\mathrm{W})^i f_i^0$, because $\lim_{n\to\infty}\sum_{i=0}^{n-1}(\alpha W)^i=(\mathrm{I}-\alpha W)^{-1}$, $\lim_{n\to\infty}W^n=0$, so f_i^n will converge to $(1-\alpha)(\mathrm{I}-\alpha W)^{-1}f_i^0$.

D. Time Complexity Analysis for C-LPA

C-LPA algorithm is showed as follows:

Input: tweet vector set $TW=\{tw_1,tw_2...tw_n\}$.

Output: the convergent label vector of each tweet from TW.

Step1. Use canopy cluster algorithm to label a part of tweets.

Step2. Use LPA to label the rest of tweets.

In canopy cluster algorithm, suppose n_1 is the number of tweets, the algorithm cluster tweets into k clusters, and the

iteration repeat m_1 times, then the time complexity of canopy algorithm is $O(n_1 k m_1)$. In LPA, the size of label set is the number of clusters in canopy cluster algorithm, suppose n_2 is the number of unlabeled tweets, m_2 is the iterator times, the time complexity of LPA is $O(n_2 k m_2)$. So the time complexity of C-LPA is $O(n_1 k m_1) + \mathrm{O}(n_2 k m_2)$.

IV. EXPERIMENTS

In this section, we evaluate feature selection method in Section 3.1, and then apply our methodology on twitter dataset, and compare to LDA algorithm and Single-Pass cluster algorithm in topic precision and topic recall.

A. Dataset and Evaluation Methodology

We extracted twitter data from https://twitter.com by twitter4j. We select three topics: (1) MH370 flight missed, (2) Crimea departed from Ukraine, (3) Obama Michelle visited china. All tweets of each topic are collected by set certain query for twitter4j API, the percentage tweets of each topic is 60%, 25%, 5%, and also about 10% noisy data which posted in the past year has been collected randomly, and then they can't belong to the three topic.

We select two metrics to evaluate our algorithm,

- Topic precision: percentage of successfully detected by algorithm in the topic.
- Topic recall: percentage of the correctly detected tweets in the topic set.

B. Feature Selection Method Evaluation

We evaluate effectiveness of different smooth parameter and select the proper value firstly. We set the smooth parameter from 0.1 to 1, and use r as performance metric, and $r = \frac{1}{n_k}\sum w_k - \frac{1}{n_c}\sum w_c$, where w_k denotes the weight of top-k keywords in general corpus, n_k is the number of keywords, w_c denotes the weight of common words in general corpus, n_c is the number of common words. The experimental result is shown in Fig. 2(a), and $\delta = 0.5$ is the best choice.

In order to evaluate the feature selection method, we get the top-5 weight terms which can be considered as the keywords of each topic from three topics, as shown in Table 1, and then select most common words in English[2]. We get the frequency and the weight of all of these terms in general corpus and reference corpus. The results of keywords are showed in Fig. 2(b), the results of common words are showed in Fig. 2(c), where p(gen) denotes the curve of the frequency of terms in general corpus, p(ref) denotes the curve of the frequency of terms in reference corpus, w(t) denotes the weight of terms.

TABLE 1 THE TOP-5 WEIGHT TERMS OF THREE TOPICS

MH370	Crimea	Michelle
mh370, malaysiaairlines, prayformh370, relatives, keluarga	crimea, ukraine, ukrainian, rusia,annexation	michelle, Obama, china, beijing, xi'an

As shown in Fig. 2(b), Fig. 2(c), In two corpuses, the frequency of keywords vary widely and of common words is approximate, and the weight of keywords is high and of common words is almost to one. Experiment shows our method weights the words in general corpus effectively.

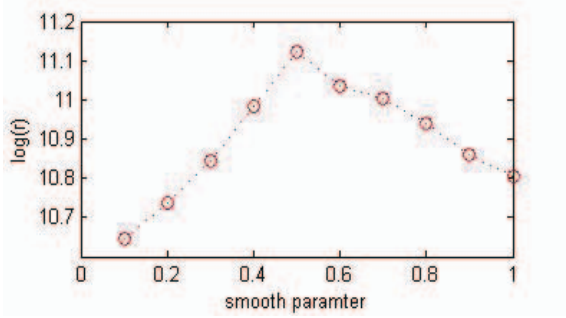


Figure 2(a). Experimental of smooth parameter

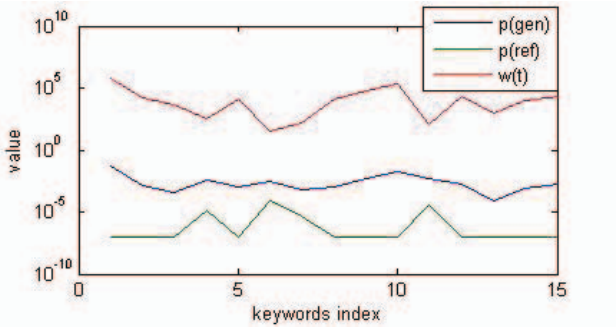


Figure 2(b). The frequency and weight of keywords

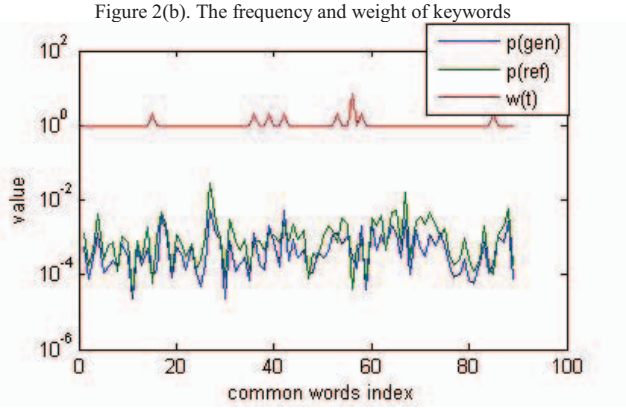


Figure 2(c). The frequency and weight of common words

C. C-LPA Evaluation

We compare precision and recall of three algorithms in the three topics. The performances as shown in figure 3 and figure 4. Experiment result shows that C-LPA is more precision than LDA and SPCA(Single-Pass Cluster

[2] http://en.wikipedia.org/wiki/Most_common_words_in_English

Algorithm), and the recall is higher. In the other hand, the precision and recall of C-LPA is stability in the three topics, however, LDA is effected by the size of dataset, the smaller of the dataset, the lower precision of LDA, the precision and recall of SPCA is change in different topics. So LDA detect small topic or subtopic hardly, and SPCA is easily affected by the order of input, in our experiment, C-LPA has good performance in different kinds of topics.

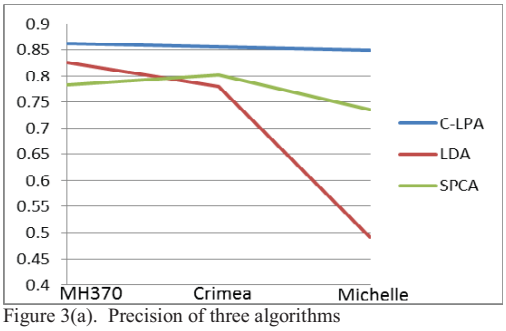


Figure 3(a). Precision of three algorithms

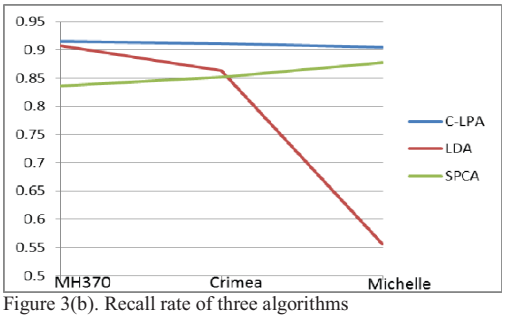


Figure 3(b). Recall rate of three algorithms

V. CONCLUSION

Topic detection in twitter is a complex task. Tweets which may be related with multi-events or multi-topics are difficult to cluster into a topic exactly by general cluster algorithm, and probabilistic model has difficulty in achieving high precision because the characters of a tweet are no more than 140. In order to deal with these challenges, we propose C-LPA algorithm. First, we adopt effective method to select features from tweets and compute similarity matrix between tweets. Second, we label the tweets that only belong to one cluster by canopy cluster algorithm. Finally, label propagation algorithm has been used for labeling the uncertain tweets in the overlapping of different clusters. Our method achieves high precision and recall rate in the experiment.

REFERENCES

[1] H.-G. Kim, S. Lee and S. Kyeong, "Discovering hot topics using twitter streaming data: Social topic detection and geographic clustering", in Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1215-1220, ACM, 2013.

[2] M. Mathioudakis and N. Koudas, "Twitter monitor: Trend detection over the twitter stream", in Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp.1155-1158, ACM, 2010.

[3] O'Connor, Brendan, Michel Krieger, and David Ahn. "TweetMotif: Exploratory search and topic summarization for Twitter." In Proceedings of the 2010 International AAAI Conference on Weblogs and Social Media, pp.384-385, ACM, 2010.

[4] D. M. Blei and J. D. Lafferty, "Dynamic topic models", in Proceedings of the 2006 International Conference on Machine Learning, pp. 113-120, 2006.

[5] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter", in Proceedings of the 2010 Workshop on Social Media Analytics, pp. 80-88, ACM, 2010.

[6] Q. Diao, J. Jiang, F. Zhu and E.-P. Lim, "Finding bursty topics from microblogs", in Proceedings of the 2012 Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pp. 536-544, Association for Computational Linguistics, 2012.

[7] L.M Aiello, G. Petkos, C. Martin, D.Corney, S. Papadopoulos, R. Skraba, A. Goker, et al, "Sensing trending topics in Twitter," Multimedia, IEEE Transactions on , vol.15, no.6, pp.1268-1282, IEEE, 2013.

[8] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: real-world event identification on Twitter," In Proceedings of the 2010 International AAAI Conference on Weblogs and Social Media, pp.438-441, ACM, 2011.

[9] J. Sankaranarayanan, H. Samet,B.E.Teitler, M.D.Lieberman, and J. Sperling, "Twitter stand: News in tweets," in Proceedings of the 2009 Intemational Conference on Advances in Geographic Information Systems, pp. 42–51, ACM, 2009.

[10] H. Savvadi, M. Hurst, and A.Mavkov, "Event detection and tracking in social streams," In Proceedings of the 2009 International AAAI Conference on Weblogs and Social Media, pp.311-314, ACM, 2009.

[11] S. Papadopoulos, Y. Kompatsiaris, and A. Vakali, "A graph-based clustering scheme for identifying related tags in folksonomies," in Proceedings of the 12th International Conference on Data Warehousing and Knowledge Discovery, pp. 65–76, Springer-Verlag , 2010.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2003.

[13] Y. Ikegami, K. Kawai, Y. Namihira and S. Tsuruta, "Topic and opinion classification based information credibility analysis on twitter", in Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, pp. 4676-4681, IEEE, 2013.

[14] H. Tu and J. Ding, "An efficient clustering algorithm for microblogging hot topic detection", in Computer Science & Service System (CSSS), 2012 International Conference on, pp.738-741, IEEE, 2012.

[15] C.-H. Lee, T.-F. Chien and H.-C. Yang, "An automatic topic ranking approach for event detection on microblogging messages", in Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, pp. 1358-1363, IEEE, 2011.

[16] J. Zhang, Y. Xia, B. Ma, J.-M. Yao and Y. Hong, "Thread cleaning and merging for microblog topic detection", in Proceedings of the 2011 International Joint Conference on Natural Language Processing, pp. 589-597, 2011.

[17] Meyer, Carl D. Matrix Analysis and Applied Linear Algebra. Vol. 2. Siam, 2000.

DEDICATION IN ONLINE COLLABORATION REDEEMS EXPERIENCE: AN ANALYSIS ON THE COMPARISON BETWEEN WIKIPEDIA AND SCHOLARPEDIA

ZhengZheng OuYang
School of Mathematic & Computer Science.
Wuhan Polytechnic University, China
oyzz@whpu.edu.cn

***Abstract*—Evaluation and performance analysis of an online collaborative project are never easy tasks because the massive human involvement and other qualitative factors are hard to assess. To figure out the relationship between human related factors and quality of collaboration outcomes, we propose an effective formal approach to estimate the human involvement in collaboration process and testify our method on 100 articles extracted from Wikipedia and Scholarpedia, the qualities of whose historical contents have been evaluated manually by volunteers of specific background. Through comparison of the human involvement and the outcome quality in these two projects, we find that the quality of collaborative products is positively related with the number, dedication and experience of collaboration participants, among which experience decides the necessary amount of human resources for a high quality, and increasing number or dedication of participants can also make up to the lack of experiences.**

Keywords-Collective Intelligence, Distributed Collaboration, Online Production

I. INTRODUCTION

"What makes an online collaborative project work?" is a pervasive question in the literature on computer-supported cooperative work and social computing. For those successfully operated online producing websites like Wikipedia and Scholarpedia, the motivations of editors, design features of the community, affordances of peer production, and other online social activities interact in complex ways at multiple levels to enable and sustain their massive collaboratively authored editing process. Among which, human factors, especially the group size, the dedication and expertise of participants, affect the performance of a CSCW project undoubtedly but yet are not studied sufficiently.

The massive and complex interaction of multiple factors during the collaboration makes evaluation of a CSCW project very difficult. Unfortunately, there are many variables and some of these are qualitative and hard to assess and thus bring obstacles for researchers to figure out how these variables affect the performance of the whole project. However, there are other variables that could be measured and with a reasonable amount of manual assessment of certain qualitative result, it is possible to estimate the effect of some human factors on the performance of a specific collaborative work.

To figure out the relationship between human involvement and quality of the outcomes generated during collaboration, we investigated the group editing process of articles in Wikipedia and Scholarpedia, and arranged a set of manual evaluations on some article content states of different phases through the editing history. Integrating the investigation results, we proposed an original formal approach to measure some human factors in online collaborative projects, and then analyzed the role of these factors playing during collaboration. Our findings suggest that although experience is the key for a collaborative project to efficiently achieve a high outcome quality, the dedication (or activity) of participants will finally drive the group work up to the same level.

Our approach does not pretend to be applicable to all CSCW projects or comprehensively evaluate all aspects of them. Nor do we focus on evaluating the CSCW groupware performances. Rather, we concentrate on the positive or negative effect that describes how the human factors have impacted on issues like the quality of the outcomes of a CSCW project.

II. RELATED WORK

Focused on the openness and valuable user resources of online collaboration projects, there are sufficient researches that have studied many aspects of their cooperation patterns, user behaviors models, communication mechanisms and other kinds of their collaboration attributes. Welser and Cosley et al. identified four key roles in Wikipedia editors: substantive experts, technical editors, vandal fighters, and social networkers, and also found that informal socialization has the potential to provide sufficient role related labor despite growth and change in Wikipedia [1]. Panciera, and Halfaker et al. studied the behavior models of those power editors in Wikipedia, Wikipedians, and suggested customizing the initial user experience to improve retention and channel new users' intense energy [2]. Lam and Riedl explored the implications of long tail to the birth and death of articles in Wikipedia [3]. Kittur and Kraut studied four coordination mechanisms on managing conflicts in Wikipedia and generalized their findings to other collaborative contexts [4]. Giles revealed that Wikipedia's scientific articles came close to the level of accuracy in Encyclopedia Britannica and had a similar rate of "serious errors" [5], and Wilkinson and Huberman found that the high-quality articles are distinguished by a marked increase in number of edits, number of editors, and intensity of cooperative behavior.

Towards traditional collaborative work, there are some explorative studied made to model the co-working process and apply the principles of coordination to extensional

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.24

applications. Belbin builds a team with several role specializations [7], and Baeza-Yates and Pino applied their theory of optimizing management for CSCW project to the collaborative retrieval process [8].Guo and Chen et al. proposed a collaborative topic prediction model for user interest recommendation in online social networks [12]. Xin and Yuhonget al. build a SVM model to predicting the stock price that influenced by human behaviors [13].

However, the evaluation on human involvement in online collaboration has not been discussed widely enough because some of the related factors are qualitative and difficult to assess. Our study aims to give a practical resolution to the problem of evaluating human factors in online collaborative projects and investigate the relationship between these factors and the quality of final products.

III. HUMAN FACTORS IN ONLINE COLLABORATION

Concerning measuring the human involvement in an online collaborative project, there are some basic human related aspects of the cooperative process that must be considered: group size, dedication (or activity) of participants, and experience of group members.

A. Group size

Group size is the number of person participating in the collaboration process. It is the most basic yet unstable measurement that related to human resource for an online project. As in online collaboration, the group work usually offers an open cooperative environment for its participants, and there is no strict time limit or task amount requirement for a group member, even that one can join or leave any time voluntarily. Therefore, the group size of a collaborative work is neither fixed nor easy to estimate during the whole process. However, since any member who has been involved in the group work is also possible to join the collaborative team again at any time, we can reasonably assume that those who might have quit still stay in the group and just become inactive, thus here we only consider the group size to increase as time passed.

Assume the group size of an online collaborative project is represented by a variable N, we expect that as N increases as the cooperative process continues, the quality of the outcomes becomes higher. As to how high quality the outcomes can finally obtain and how fast the group can achieve it, there are two more factors that need to be concerned, dedication and experience of the collaborative members.

B. Dedication

Dedication (or activity) describes how often a participant is involved in the collaboration. A dedicated group member is usually more active in making contributions to the group work and thus has a higher frequency of participating. The time gap between a participant's two contributions in a row is the reciprocal of participating frequency. The shorter the time gap is, the more frequently a group member takes part in the collaboration, thus the more actively and dedicatedly this member contributes.

Assume for a group member U_i, $1 \le i \le N$, who has participated in the collaboration for K_i times , $1 \le i \le N$, and the time stamps of whose contributions are stated as $\{T_{i,1}, T_{i,2}, \cdots, T_{i,K_i}\}$, then the dedication degree of U_i, D_i, can be defined as:

$$D_i = \frac{K_i}{T_{i,K_i} - T_{i,1}} \quad (1)$$

And the dedication degree of a group $\{U_1, U_2, \dots, U_N\}$ would be represented as following:

$$D = \frac{\sum_{i=1}^{N} D_i}{N} = \sum_{i=1}^{N} \left(\frac{K_i}{T_{i,K_i} - T_{i,1}}\right)/N \quad (2)$$

Intuitively, the more dedicated and active the collaborative working group members are, the higher quality their products would finally obtain. Therefore, here we expect that the dedication of the group will have a positive effort on the quality of collaboration outcomes.

Note that the group size and dedication degree are two factors that will make the collaborative group of higher complexity and diversity, which means, with a larger size or a higher dedication degree, a group will have a more active and complicated process, and the cost of communication and coordination would probably increase. However, for mutual and reasonable co-working groups, gradually the integration among all members will be achieved, and then their collaboration would become more comprehensive and efficient.

C. Experience

Experience can be profiled by the amount of work finished by one group member. The more work one has done before, the more experienced one become. Generally the experience of group members decides the efficiency they work. The more experienced the members are, the faster they would produce fine work outcomes. In the real world, the amount of work in a project may be hard to assess due to the diversity of its kind, however, for online collaborative process especially, work can be quantified accurately as all the contributions are recorded and saved in the same format.

Assume that by time T, one group member U_i has participated in the collaboration for K_i times, $1 \le i \le N$, and thus has made K_i contributions, which are stated as $\{C_{i,1}, C_{i,2}, \cdots, C_{i,K_i}\}$, then the experience that U_i has obtained can be calculated as $E_i = \sum_{j=1}^{K_i} C_{i,j}$, and the experience of the group will be represented as:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{K_i} C_{i,j} \quad (3)$$

Especially for online collaborative content editing project, since all the contributions that made by participants are all in the form of text, the differences between the status before a user's contribution and the status after is a measurable revision. Assume the vocabulary of V words $\{w_1, w_2, \dots, w_V\}$ has a weight distribution stated by the V-dimension vector $W = \langle v_1, v_2, \dots v_V \rangle$, and a revision between two texts is represented by the change of word frequencies, which can be written in the form of vector: $R = \langle \Delta f_1, \Delta f_2, \dots \Delta f_V \rangle$, where Δf_k is the frequency increment of word w_k, then we can compute a contribution $C_{i,j}$ of group member U_i with the revision, written as $R_{i,j}$, and the weight vector W:

$$C_{i,j} = R_{i,j} * W \quad (4)$$

Therefore, the experience of a group for a content editing collaboration process can be also represented as:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{K_i} R_{i,j} * W \quad (5)$$

We expect that the experience of the group members have a positive effect in improving the quality of collaboration results.The more experienced group generates the better products. In addition, according to a CSCW study arranged by Keegan and Gergle [9], the experienced group members make more contributions than other group members. Therefore, in our approach, the experience of the group would have an accumulative effect on the quality of outcomes and we also expect that this factor changes the tendency of quality faster than linearly increasing trend of group size.

IV. EVALUATION OF COLLABORATIVE PROJECTS

To testify the effects of human involvement on the quality of collaboration outcomes, we applied our approach of measuring the human factors into the evaluation of two real CSCW project, Wikipedia and Scholarpedia.

A. Data preparation

Wikipedia and Scholarpedia are both distinguished online encyclopedias featured by their open access to collaborative editing. They are alike in many respects: both allow anyone to propose revisions to almost any article; both are committed to the goal of making the world's knowledge freely available to all. Nevertheless, they differ from each other precisely in the collaborative editor and reviewer groups and target audience. While Wikipedia proclaimed that it is "a free encyclopedia and completely open to any edits" [11], Scholarpedia has a much higher threshold for edits to be published: "all articles in Scholarpedia are either in the process of being written by a team of authors, or have already been published and are subject to expert management." [12] This publication requirement of peer-reviewed level narrows down the range of possible editors to almost only academic circles and therefore makes the collaborative groups of its articles distinct from those of Wikipedia in all human related aspects.

We collected the editing history of 50 articles in the field of computational intelligence from Scholarpedia, the historical revisions of which must be over certain amount, and also extracted all the historical revisions of the 50 identical articles in Wikipedia. For both of these two datasets, we calculated the human factors of each editor group of the 50 articles, estimated the average level of human involvement based on the factors of these sampled articles, and then testified and analyzed relationship between the human related features of the two projects and the quality of their outcomes.

B. Evaluation on human factors

As shown in table 1, Wikipedia and Scholarpedia differs from each other in both average size and dedication degrees of editor groups. Limited within academic circles, Scholarpedia only has 9.4 editors making contributions to single article in average, while more openly, Wikipedia draws 306.7 contributors per article as an average. Also, the Wikipedia editors are proved to be more active and dedicated into the co-editing work as they have a higher average editing frequency. As another interpreting of the values inTable 1, after one edit, a Wikipedia contributor only takes a 69.1 days' break before the next edit on the same article, and for a Scholarpedia editor, this break will be 106.8 days long.

Table 1.Comparison of human factors in Wikipedia & Scholarpedia

Human Factors	Wikipedia	Scholarpedia
Average Group Size	306.7	9.4
Average Dedication Degree	0.1448	0.0094

We also illustrated the tendency of experience for both Wikipedia and Scholarpedia in figure 1. As demonstrated in the comparison, the average experience value of Scholarpedia groups rises much faster than that of Wikipedia groups when the group size increases, which indicated that Scholarpedia are continuously enlisting contributions all from skillful group members who already have plenty of experiences and the potential to earn more. This can be explained by the high level requirement for its editors' publication. In addition, with the group becoming larger, Scholarpedia has an obvious accumulating increment in the experience of whole group, which is consistent with our assumptions mentioned earlier. While the collaborative editor groups of Wikipedia have their experience improved in a comparatively stable speed, which suggests that the editors in Wikipedia are actually of the average level among common web users.

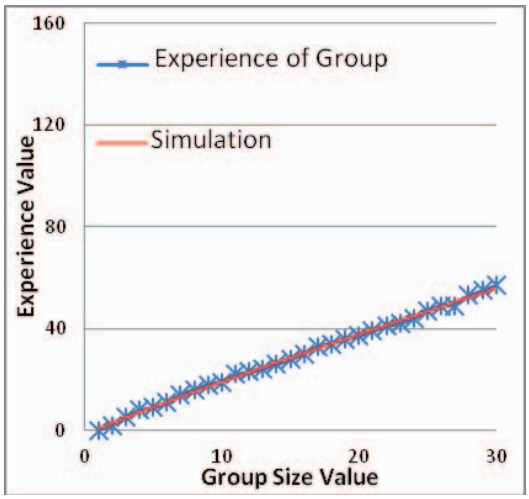


Figure 1 (a) Experience Tendency of Wikipedia Groups

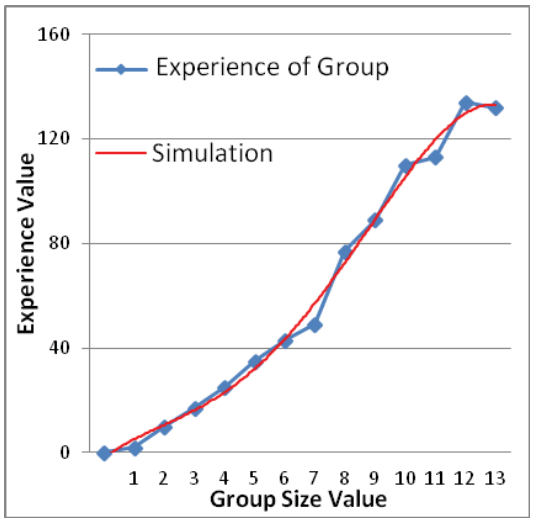


Figure 1 (b) Experience Tendency of Scholarpedia Groups

Figure 1. Experience of Groups

With all the human factors configured, the quality tendencies of the outcomes for these two projects with different types of editors should be distinct from each other. We expect two different shapes of change for these two in the next section.

V. EVALUATION ON QUALITY OF OUTCOMES

To assess the quality of the outcome articles in Wikipedia and Scholarpedia accurately, we arranged 10 graduated students of computer science background to give scores to the historical content status of the 50 articles for each encyclopedia. In order to avoid the influence of personal preferences and deviation, the scoring process went through a cross validation, and the final scores of each article was normalized into the range of 0 to 100. We give the results of evaluation in the following Figure 2:

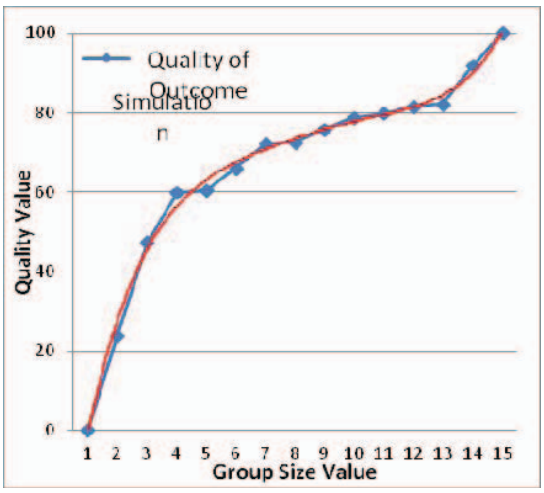


Figure 2 (a) Quality of Outcome Tendency of Scholarpedia

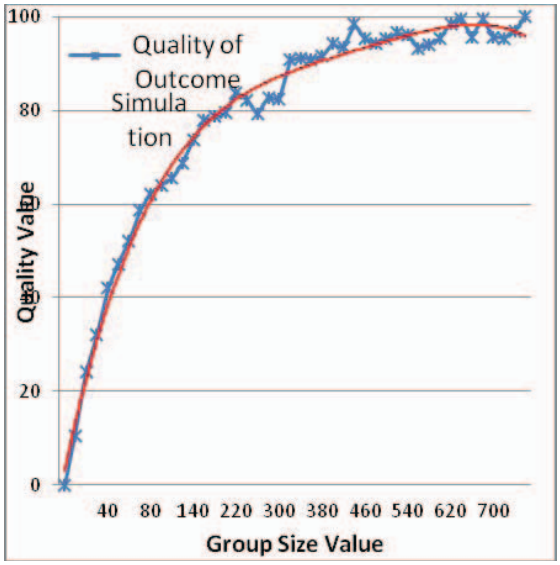


Figure 2 (b) Quality of Outcome Tendency of Wikipedia

Figure 2. Quality of Outcome

From figure 2, we can see that the quality of the collaboration outcome for Scholarpedia reaches a high level within only 15 editors' contributions, while within the same co-working group size, Wikipedia contributors can only generate outcomes of quality as 1/4 high as that of Scholarpedia contributors, and it takes 640 Wikipedia editors to working together to make the outcome quality of their collaborative work meet the same high level. This is suggesting that for Scholarpedia can save much human resources than Wikipedia for the same high quality.

VI. ANALYSIS AND DISCUSSION

As the quality evaluation of Wikipedia and Scholarpedia demonstrated, Scholarpedia can meet the same high quality standard with much less human resources than Wikipedia. Since the difference between the publication thresholds of these two online encyclopedias is remarkable, the first human factor that needs to be considered should be the experience of the collaborative editing groups. We computed the correlations between the values of group experience and the values of outcome quality for each encyclopedia, and it turns out that the correlation of group experience and outcome quality for Wikipedia is 1.479, and that for Scholarpedia is 2.396. The positive correlated connection between the group experience and the quality of outcome explains that the more experienced collaborative group in Scholarpedia have a much higher efficiency in make contributions of good quality than the groups in Wikipedia. Also, it testified the positive effect of group experience on the quality of outcomes.

Moreover, as shown in figure 2(b), despite the lack of highly experienced contributors, Wikipedia finally achieves the same high level of outcome quality as Scholarpedia after the bunch of average experienced editors' frequent contributions and collaboration. According to the good faith assumption of all online collaborative projects [10], all the volunteer contributors are believed to aim at the same goal of co-editing to

produce a mutual article of high quality content, thus with sufficient quantity of changes towards better content, the higher quality of articles will be gradually achieved. In our approach, we actually evaluated the average level of group sizes and the dedication degrees of these groups for these two projects in section 4.2. The larger groups and higher dedication degree of Wikipedia make much more contributing activities and also provide the project participants enough space and potential to improve the article quality. In this case, we believe that the group of large size or with high dedication can make up its lack of experience in generating high quality outcomes.

VII. CONCLUSION AND FUTURE WORK

We studied the possible human related factors of an online collaborative project that influence the quality of outcomes, and proposed an effective approach to estimate the human involvement within the collaboration process. We testified our method on the real data extracted from two successfully operated online encyclopedias, Wikipedia and Scholarpedia, and measured the comprehensive differences of human involvement between these two projects. To assess accurately the quality of 100 articles in our datasets, we arranged a valid manual evaluation with volunteers of specific background. And finally, we provided an analysis and discussion on the evaluation results.

Our findings suggest that the co-working group size, dedication and experience of group members, are three main human factors of online collaboration projects with positive effect on the quality of outcomes. Among them, the experience of group members decides the necessary amount of human resources for obtaining a high grade quality of outcomes, the more experienced the contributors are, the less human resource for the high quality are needed. Besides, the group size and the dedication of group members leave space for improvement of the quality and make up to the lack of experience. The larger group with more dedicated member has more potential of gradually generating the outcomes of a high quality.

For future work we would like to apply our findings to solve more practical problems such as detecting inactive contributors with low dedication and optimizing collaboration group size for human resources control. Also, we are interested in exploring new type of measurable human factors and seek proper applications for them.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Hubei Province of China (Grant No.2011CDB226).

REFERENCES

[1] Howard T.Welser, Dan Cosley, GueorgiKossinets, Austin Lin, FedorDokshin, Geri Gay, Marc Smith, Finding Social Roles in Wikipedia, In Proceedings of the 2011 iConference, pp. 122-129, 2011.

[2] Katherine Panciera, Aaron Halfaker, Loren Terveen, Wikipedians Are Born, Not Made, In Proceedings of the ACM 2009 international conference on Supporting group work, pp. 51-60, 2009.

[3] Shyong (Tony) K. Lam, John Riedl, Is Wikipedia Growing a Longer Tail? In Proceedings of the ACM 2009 international conference on Supporting group work, pp. 105-114, 2009.

[4] AniketKittur, Robert E. Kraut, Beyond Wikipedia: Coordination and Conflict in Online Production Groups, In Proceedings of the 2010 ACM conference on Computer supported cooperative work,pp. 215-224, 2010.

[5] Jim Giles: Internet encyclopaedias go head to head. Nature, vol. 438, no.7070, pp. 900–901, Dec. 2005.

[6] Dennis M. Wilkinson and Bernardo A. Huberman. “Cooperation and quality in Wikipedia”. In Proceedings of the 2007 international symposium on Wikis, pp.157-164, 2007.

[7] Meredith Belbin, Management teams: why they succeed or fail.(2nd edition), Elsevier Butterworth-Heinemann, MA. USA, 2003.

[8] Ricardo Baeza-Yates, José A. Pino. “Towards formal evaluation of collaborative work.” Information Research and International Electronic,vol.11, no. 4, 2006.

[9] Brian Keegan, Darren Gergle, Noshir Contractor. Do Editors or Articles Drive Collaboration? Multilevel Statistical Network Analysis of Wikipedia Coauthorship. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, pp. 427-436,2012.

[10] J. M. Reagle, “Good Faith Collaboration -- The Culture of Wikipedia (Web edition),” The MIT Press, Cambridge, MA. 2011.

[11] Wikipedia, “Wikipedia,” http://en.wikipedia.org/wiki/Wikipedia.Mar. 6th, 2010.

[12] Liang Guo, Qiumiao Chen, Wenwen Han,Ye Tian,Yidong Cui, Wendong Wang, Alvin Chin, Xia Wang, "Collaborative Topic Prediction Model for User Interest Recommendation in Online Social Networks", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 6, No. 23, pp. 62-73, 2012.

[13] Jin Xin, Kang Yuhong, Zhang Keyi, "Stock Price Predicting Using SVM Optimized by Particle Swarm Optimization Based on Uncertain Knowledge", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 6, No. 23, pp. 216 ~ 221, 2012.

A Database Middleware Architecture for MPI-based Cloud Platform

Lin Zhu, Yucheng Guo*
Department of Computer Science, School of Computer Science and Technology
Wuhan University of Technology, Wuhan, China
zlinhp@163.com; ycheng.g@gmail.com

Abstract*—To solve the poor performance of existing cloud platforms in high-performance computing, our research team developed a high-performance cloud computing platform based on MPI. As a cloud platform, massive data storage is a key technology. In the research, we mainly studied how to build a MPI cloud platform that supports massive data storage, and implement a distributed cluster constructed by MySQL. This paper introduces the feature of MPI cloud platform, how to build the cloud platform database and the implementation of related SQL operations on the database. Based on this, we test the related operations in the database, and elaborate the shortcomings and challenges of the distributed database in this cloud computing platform.

Keywords-MPI; cloud computing; distributed database; MySQL clusters

I. INTRODUCTION

Cloud computing is closely linked to distributed parallel computing and distributed database, and MPI is the most effective method and tool to implement parallel computing. But MPI don't support distributed file storage, this research aims to solve this problem to build and implement a MPI cloud platform. The research uses MySQL cluster for data storage, logic semantics is mainly achieved by MPI, thus building a database middleware based on MPI.

II. RELATED WORK

A mature cloud platform product is Hadoop developed by Apache Foundation, users can develop their distributed application even though they don't know the underlying details. But the MapReduce programming model of Hadoop is applicable for I/O-intensive tasks, rather than CPU-intensive tasks[1]. Implementing MapReduce in MPI can solve this problem and improve the efficiency at the same time, therefore it has great significance to build a cloud platform based on MPI.

In China, many scholars and research institutions proposed using MPI to build cloud platform. The parallel computing laboratory of University of Science and Technology of China proposed using MPI to implement MapReduce[2], In their research, they proposed communication mode based on shared memory and network transmission, and proposed to separate Map and Reduce to support multiple rounds cycle of Map and Reduce. But there is no database supporting in their prototype, which therefore can't deal with massive data. There're some papers mentioned MPI cloud computing models in China, but mostly limited to theoretical analysis, the research of distributed database of cloud platform based on MPI is rarely mentioned.

Main function of cloud platform is to deal with massive data through distributed data storage system, by now relatively mature distributed databases are BigTable and HBase[3-4]. BigTable is a non-relational distributed database designed by Google; HBase is a distributed, column-oriented open source database.

In our parallel computing lab, several members collaborated together to design and implement a relatively simple MPI cloud platform architecture earlier[5-8]. This platform is optimized for map-reduce computing model and provides fault tolerance, but it can't deal with massive data due to the lack of database support.

By integrating a database middleware into the MPI-based cloud platform, we can satisfy the demand of massive data storage and solve the problem of MPI's lacking support of file system.

III. CLOUD PLATFORM DATABASE ARCHITECTURE

There're large differences between distributed storage and centralized storage, data is not only stored in a single node in a distributed storage, but divided into multiple nodes; the nodes may in the same LAN, and even in different networks.

A. Idea and Plan of Building Database

This research designed a database middleware to integrate into the cloud platform constructed by MPICH-2 and socket communication.

In our research, we build a database cluster by MySQL to store data in the MPI cloud platform[9-10]. To deal with massive data storage, the data stored in each node isn't the same; data in each node together compose a complete database. The discretization of data storage makes the operations to the database very different from that of the traditional database.

The architecture of this platform is single master plus layered multiple workers. Workers are divided into main workers and slave workers, each main worker corresponds with at least one slave worker; the data in main worker and its corresponding slave workers is the same, but different from the other main workers. All the main workers together compose the integrated database.

B. Database Cluster Architecture

The cluster architecture of distributed database is shown in Figure 1.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.25

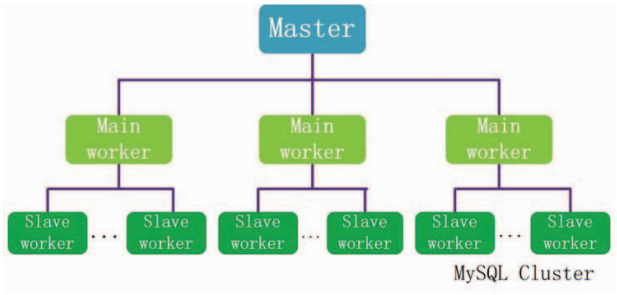


Figure 1. Architecture of Database Cluster

This architecture belongs to shared-nothing (SN) type, which is composed of relatively independent nodes essentially. There are no shared resources or cache between each node; each node has its own disk resource, memory and CPU. The links between nodes and master construct high-speed communication network; each node deals with its own data. The features of the architecture are: (1) each node of the whole cluster has an independent database; all nodes together compose an integrated database. (2) Each worker node uses its own resources (I/O, memory), doesn't need to access shared resources against other nodes, thus avoiding overhead caused by resource synchronization. (3) One advantage of SN architecture is its good extensibility—the system performance can be improved highly by increasing the number of worker nodes.

C. Database Middleware Technology

Database middleware is in the layer between client and database. The data communication between middleware and each client is achieved by a specific data communication protocol (such as TCP/IP, UDP etc.). Multiple connections can be completed by multiple processes or threads, which can improve the user concurrency and access efficiency.

The core part of database middleware is data connection manager, which is a service routine used to connect the top user layer and the bottom data storage. Generally, user sends an access request to the database, and then database middleware finds an available connection to the database for the user, and transmits the access request to the target database. After the target database finishes the SQL statement sent by user, it will return the results to the client by database middleware.

The database middleware module is located in master node layer, the upper layer of master node is client, and the lower layer is worker nodes used for data storage and comparison. Request submitted by user is received by database middleware rather than directly by the worker nodes, and then the middleware decides how to assign the request to the worker nodes after semantic analysis.

By building a database middleware, the cloud platform can deal with massive data.

D. Setting Up the Database

Before building database middleware, a database cluster is needed to be set up in the current system platform. The specific process is as below:

Operating system is Linux, version of MySQL is 5.1, and MySQL is installed by compiling source package.

Deploy MySQL for each node to constitute a MySQL cluster. When deploying MySQL on one node (default port is 3306) is finished, transmit it to other nodes through scp command, and then start up the MySQL cluster.

The last step is to modify configuration profile my.cnf in each MySQL node, comment out the statement "bind-addr=127.0.0.1" in the file, and restart MySQL, log in to the MySQL terminal, execute two commands as bellow:

Grant all privileges on *.* to 'root'@'%' identified by 'yourpassword' with grant option;

Flush privileges;

It will lead to "connection refused" error when user commits a data request if above-mentioned two commands haven't been executed. After executing these two commands, user can access MySQL database in the form of IP address. After comment out bind-addr, MySQL node can listen to all external IP address that access this database, rather than just the loop IP (127.0.0.1).

E. Data Storage Strategy

Cluster architecture of our platform is SN architecture, and data stored in each sub database is different, so data should be partitioned.

As to data partition, frequently-used methods are Round-Robin, Range-Partition and Hash method.

This paper proposed another strategy: hash-based improved data partitioning strategy. This partitioning strategy applies Consistent Hashing algorithm and relies on hash function to divide data; it will make data evenly stored into each node as far as possible.

A value is mapped to a key in commonly-used hash methods, the value ranges from 0 to INT_MAX-1, and we can view this region as a circle ring, the starting point of this ring is 0 and ending point is INT_MAX-1. Each record is mapped to a value in the hash ring after it has been processed by hash function, then mapping each node to the ring by hashing its IP address. Records still need to be mapped to the node after establishing the mapping relationships of records and nodes respectively; then hashed data records can find which main worker node to be mapped to.

In this paper, we search the hash ring clockwise to find the nearest hashed IP address for records. For example, if hashed value of record A after processed by hash function is 100, the values of three nodes (node A, node B, node C) in the cluster after processed by hash function are 1000, 500, 10000 respectively. Then node B is chosen to associate with record A because the nearest value to 100 is 500 in clockwise direction, so record A should be partitioned into node B.

There're two kind of data structure chosen to store hash value of IP address. Cycle singly linked list is chosen as data structure when nodes number is small, hash value of each node is stored in this linked list, after sorting the list, we can find which node should be mapped to for each hash value of record in only once traversal, the time complexity is o (n); red-black tree is chosen as data structure to quickly find out mapping nodes when nodes number is big, the advantage of

red-black tree is high searching efficiency, and the time complexity is only related with tree height, it is o(logN) even in the worst case.

F. Logical Semantic Controller

The logical semantic controller of database middleware has two functions, one is to parse the SQL statement submitted by user, and the other is to decide which nodes to execute the SQL statement.

The middleware of Master will deal with the SQL statements after they are submitted to the Master by user.

The database middleware will parse the statement and hand out it to the worker nodes, the SQL statements mainly divide into four kind of operations, they're SELECT, INSERT, DELETE and UPDATE. The format of job submitted by user is like this: SELECT * FROM A: compare_function, the Master node will decompose the statement after it receive a statement, in front of the colon is common SQL statement, behind the colon is the compare function which should be called by worker nodes from local dynamic link library after finish executing the SQL statement.

IV. Key Points of the Database Implementation

1) Implementation of SQL query statement

The SQL query statements, for example, "SELECT id, name from TABLE_A", are stored in a buffer at first, and then transmitted to worker nodes through MPI.

The master decides how to reduce the final records based on whether the SQL statement has a colon. If SQL statement doesn't have a colon, then master gets query result from the work nodes through RMA (Remote Memory Access). By RMA technology, the database nodes just need to put data in a specific memory area, and then master node accesses this memory area through MPI_Get, this method applies new function of MPI-2—unilateral communication. In current architecture, data acquired by the master is just the initial data, for example, when submitted SQL statement "SELECT * FROM TABLE_A WHERE id>5 ORDER BY id DESC", only in each node data is in descending order, data acquired by the master is not really in descending order, that is to say the data should be reduced by the master node before it is transmitted to the user. Master stores feedback result of worker nodes in local buffer at first; this buffer is a dynamically created two-dimensional array of strings, master then corresponds the keywords with their associated records, for example, id is the field name in descending order and its type is INT, there are four unordered records named A, B, C, D, master associates each record with its keyword and stores them in a STL multimap, then the structure of this multimap becomes like this: <id of A, A>, <id of B, B>, <id of C, C>, <id of D, D>. Multimap is based on red-black tree, so the time complexity of finding a node is related with the height of tree, which is o (logN). When all records have been scanned, multimap is in an orderly state, the results can be obtained in only once traversal.

2) Implementation of other SQL statements

Compared with SELECT statements, the situation is different when parsing and dealing with INSERT statements. When executing INSERT statements, data only needs to insert into single database node because data between each node is different. It is similar when executing DELETE statement, for example, statement submitted is "DELETE FROM A WHERE key>100", only nodes that meet this qualification need to do the deletion; situation is the same when executing UPDATE statements too, find the nodes that meet the qualification to do the modify operations.

3) Implementation of SQL semantic parsing

SQL statements usually contain keywords like ORDER BY, GROUP BY; the database middleware can parse out these keywords. Master node firstly find what kind of keywords when it receives SQL statements, then store these keywords in a specific memory, finally it determines which APIs to call to finish the reduce operation based on these keywords.

In addition, the logic semantic controller needs to make an additional judgment when user submits requests, if SQL requests have incidental information (compare function), then the name of comparison function should be parsed out. If there has no incidental information, the worker nodes can immediately return results to the master when they finish executing the corresponding operations, and then the master do the final reducing operation; if there has incidental information, the worker nodes couldn't return results to the master immediately, the results should be reduced secondly before they are returned to the master, usually the secondary operation is comparison operation.

V. Experimental Evaluation

The system was tested in the perspective of integrity, robustness and performance respectively.

A. Experimental Environment

Software environment: operating system of the physical machine is windows 2003 data center R2 64-bit version. There are 6 virtual nodes in the physical machine; each node is configured with Ubuntu 10.10 Desktop (64-bit) + mpich2-1.4.1p1.

Hardware environment: the physical machine has the symmetric multi-processor architecture, contains two 2.4GHz CPU (E5620), each CPU has four physical cores and supports hyper-threading (can be opened up to 8), memory capacity is 16GB, Dual Gigabit LAN. Each Linux virtual node has 2.5GB disk space and 350MB memory.

B. Experimental Method

The system testing plan is as follows:

Create 6 virtual machines, two slave worker nodes correspond to two main worker nodes, their IP are 192.168.145.201, 192.168.145.202, 192.168.145.206, 192.168.145.203, top management node is 192.168.145.205, master node is 192.168.145.204, user submission process runs in the physical machine (192.168.145.1).

Firstly start the monitoring process(monitor_ad.py) on the top management node, monitoring process listens to the main nodes and slave nodes that connect to it. And then start test.py on main nodes and slave nodes, letting the main and slave nodes to establish connection with the top management

node. Then start recv_ex.py on top management node. There is a long TCP connection between user and top management node after user start the submission process sender.py, the submission of jobs and return of results are achieved through sender.py and recv_ex.py.

we did three kinds of tests on the cluster to verify the performance of this system: to verify correctness of consistent hashing algorithm, we tested whether the data storage capacity of each main node is relatively balanced; verify one function of the database middleware, if job submitted by user don't need a secondary comparison, query results return to the user immediately; verify another function of the database middleware, if the job submitted needs a secondary comparison, the query results will store in a file firstly, and then execute this file with compare function to generate final result.

C. *Experimental Result*

When submitted job don't need a secondary comparison, the submission format is "select name from record where gender = female: 0: return", the second field is job priority, the third field means the job don't need a secondary comparison. when top management node acquires the query results, it sends the results to the user.

In order to validate the efficiency of distributed database implemented by database middleware, we ran another test case, we use another node and import all records into this node, and do the same SQL execution through MySQL. The SQL statement is "select id from record order by id limit 10: 0: return". The test results are shown in Table 1 and Figure 2.

TABLE I. THE TEST RESULT

	Group 1	Group 2	Group 3	Average
Middleware	17.89	18.16	17.92	17.99
MySQL	20.33	20.28	21.01	20.54

By comparing several tests, the time of ORDER BY query through database middleware is 12% less than that of query directly through MySQL in one node.

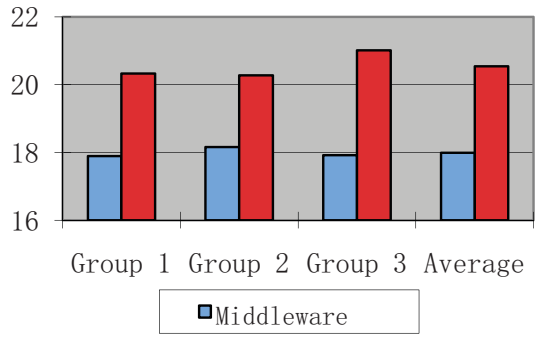


Figure2 The Performance comparison Between Middleware and MySQL

When submitted job needs a secondary comparison, the submission format is "select name from record: 0: hide: xxxx: 3", the fourth field is the target string to be compared, the last field is the level of fault tolerance.

Through the testing results, the distributed database can do all query operations, and can return correct results in a relatively quick time; the database can be proved efficient and reliable after repeated tests.

VI. CONCLUSION

MPI is applied in a wide range in the field of scientific computing, but it doesn't provide a specific file system to store user data, so it hasn't been applied to the development of cloud platform product in the academic or commercial field. In this paper we combine a distributed database built from MySQL and MPI technology, to build a cloud platform based on MPI.

This paper mainly introduces the distributed database of the cloud platform. The contents involved are database middleware technology, building of the database, database cluster architecture, data storage strategy, and logic semantic controller, and give the screenshots of part of the results.

Of course, this study has a number of shortcomings which need to be further improved. For example, only a few basic data types (INT, VARCHAR, CHAR, DATE) are supported in the database middleware, the database can't support all MySQL data types when executing the SQL statements.

REFERENCES

[1] M Bhandarkar. MapReduce Programming with Apache Hadoop[J]. 2nd IEEE International Conference on Cloud Computing Technology and Science. 2010, 721-726

[2] Qilong Zheng. The Application of HPMR in Parallel Matrix Computation[J]. Computer Engineering, 2010, 36(8):49-51.

[3] Chang F, Dean J, Ghemawat S, et al. BigTable: A distributed storage system for structured data. ACM Trans[J]. on Computer Systems, 2008, 26(2):1−26.

[4] HBase [EB/OL]. http://hbase.apache.org/

[5] Peng Wu. Research of Key Technology of Fault-tolerant and Disaster Recovery in MPI cluster[D].Wuhan: Wuhan university of technology,2012. (in Chinese)

[6] Dongxu Hu. Researcch of Key Technology of MPI-based Multilayered Fault-tolerant high performance cloud platform[D]. Wuhan: Wuhan university of technology, 2013. (in Chinese)

[7] Yuchao Zhang. Research of Key Technology of Calculated Dependence under MPI-base cloud platform[D]. Wuhan: Wuhan university of technology, 2014. (in Chinese)

[8] Yucheng Guo, Dongxu Hu, Peng Wu. MPI-based Heterogeneous Cluster Construction Technology[C]. DCABES 2012, Guilin, 2012: 120-124.

[9] A MySQL Technical White Paper [EB/OL]. [2004-02-09]. https://confluence.oceanobservatories.org/download/attachments/16418744/mysql-cluster-technical-whitepaper.pdf

[10] Michael Kofler. The Definitive Guide of MySQL5[M]. 2006. P 17-20.

* Corresponding Author: Yucheng Guo

Estimation of Cloud Computing System Construction Scale

KE Zun-You
School of Earth Sciences and Engineering, Hohai University, & Information Engineering Dept., Nanjing Institute of Mechatronic Technology, Nanjing, PRC
2265255075@qq.com

DIAO Ai-Jun
Information Engineering Dept., Nanjing Institute of Mechatronic Technology, Nanjing, PRC
236981178@qq.com

LI Xiang-Juan
College of Business Administration, Nanjing University of Traditional Chinese Medicine
Nanjing, PRC
xjlee2002@126.com

***Abstract*—Based on system dynamics modeling method, the nonlinear model of the cloud computing virtual machines scheduling is built. With system dynamics methodology and the use of simulation tools, from a system perspective on cloud computing service requirements and building scale, the key factors affecting the number of cloud computing machines are identified. Furthermore, mathematical statistical theory and methods are used to find out optimal estimation and implementation of the construction scale with regarding to the decisive factor. The regression equations are submitted and the credibility of the regression equations is verified.**

Keywords-Cloud computing scales; Construction scale; Estimate decision; System dynamics; Mathematical statistic

I. INTRODUCTION

Cloud computing is widely applied in Information and Communication Technology (ICT) systems. It distributes computing tasks in a large resource pool consisting of computers, so that various applications can access computing power as needed, as well as the storage space and the information services. Virtualization means services are run directly on the basis of virtual machines rather than physical ones. It's simplified to expand the capacity of the hardware and deploy the software with virtualization technology. A single physical CPU can simulate several parallel computing virtual CPUs with the computing virtualization technology, which makes it possible to run multiple operating systems on a platform. Then the software applications are run in the independent space simultaneously. Thus the efficiency of normal computers is significantly improved.

In 2006, Google, Amazon, IBM and other companies proposed the cloud computing concept. Typically, an information data center (IDC) is composed of a computer cluster in the cloud computing and delivered as a service to users, so that the subscription and application of cloud computing resources are simplified in the same way as using water or electricity on demand. The cloud computing usually adopts a large number of regular architecture X86 PC servers, which are combined with cloud computing system software to form into a large-scale sharing resource pool. The cloud computing reduces operation costs by sharing of resources to improve the system reuse ratio and work efficiency.

The application of the cloud computing virtualization technology makes the system computing power with the following adjustments [1]:

a) Pooling of computing resources. Resources are uniformly managed by way of a shared resource pool. Resources are shared to different users with virtualization technology. It's transparent to the user for the resource placement, management and allocation policy.

b) On-demand computing services. Users are supplied in the form of services, including computing resources, data storage, applications, infrastructure and other resources. The cloud computing system is to allocate resources automatically on demand.

Cloud computing is the integration and development of the distributed computing, the Internet technologies and the large-scale resource management technology. It is a systems engineering of the cloud computing related research and application [2] 。 The cloud computing virtualization technology and its systems are applied in lots of industry information systems.

The affecting factors to the construction scale of cloud computing systems will be analyzed, which can be used to estimate the cloud computing virtual environments construction scale and optimize the system for Green Power. The methodology and results involved might give guide to promote the construction of cloud computing theory and practice at the same time. This paper studies systematic analysis of cloud computing services, using the system dynamics (SD) model and mathematical statistics theory and methods, explores and studies the estimation foundation and the simulation analysis of a virtual machine computing resource pool construction scale.

II. CLOUD COMPUTING SERVICE MODELS AND PROBLEMS OF THE CONSTRUCTION SCALE

A. Cloud computing service models

Cloud computing core services can generally be divided into three sub-layers: infrastructure as a service (IaaS) , platform as a service (PaaS), software as a service (SaaS). As shown in Figure 1.

IaaS supplies hardware infrastructure deployment services, to provide a physical or virtual computing, storage and networking resources on demand for users. In the process of using IaaS layer service, the user is required to provide the information of the configuration, program code and related user data operating in the infrastructure for the IaaS layer infrastructure service provider [3][4].

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.26

Each physical machine contributes several virtual machines for virtualized resource pool after running virtualization software. Users or administrators can use and manage virtual machines through the portal. Computer components run in the virtual machines rather than on the basis of real basis. The virtualization technologies bring about elastically scalable computing power and system reliability to meet the requirements [5].

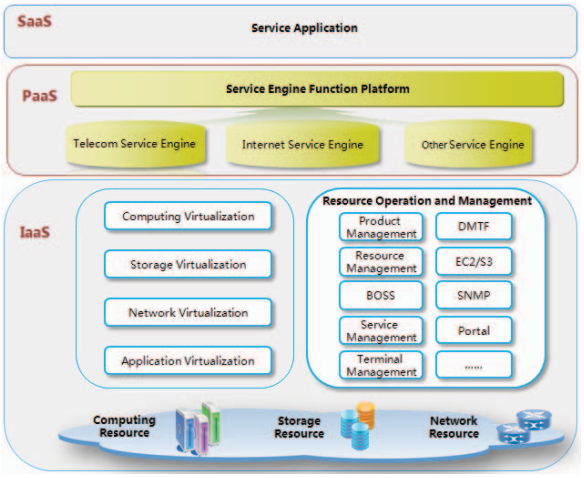


Figure 1. Core cloud computing service architecture.

CPU virtualization technology can simulate single-CPU as multi-CPU parallel computation, which allows a platform to run multiple operating systems and applications independently of each other in space, and significantly improves the efficiency of computers. The cloud computing is rapidly developed worldly as an intensive, large-scale and community-oriented ICT service delivery and usage patterns. As shown in Figure 2.

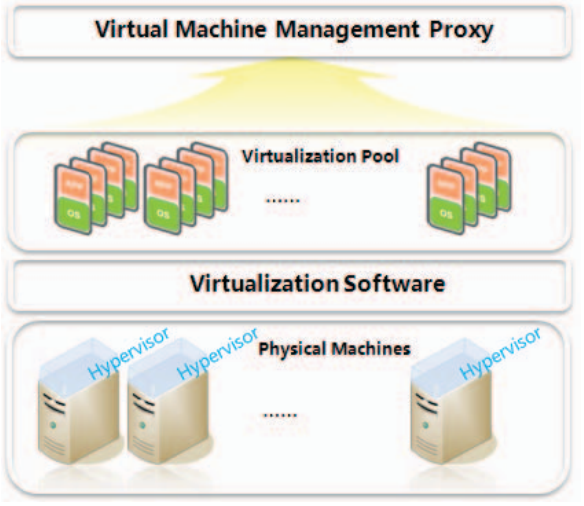


Figure 2. Cloud computing virtualization architecture.

B. *Construction scale of cloud computing services in virtual machines*

Virtual Machine elastic computing resources provide the basis for wide range of applications in the emerging large IDC with flexible application deployment, high reliability operation, and flexible deployment. Virtual machines can be used as the enterprises cloud, the private cloud or the public cloud. With the virtual machine transfer technology, the active virtual machines are moved to some corresponding physical machines, and other physical machines are shut down on which virtual machines is idle to meet the energy saving requirements [6].

However, it is not easy to fully consider, estimate and manage the scale number of on-line virtual machines in practical applications, especially in the large-scale commercial public cloud systems [7]. For public cloud applications are complex and user group compositions are diverse. Moreover, the cloud services environment complexity makes it difficult to know well the scale number of virtual machines required. The virtual machines construction scale is still far from meeting the requirements with the simple linear estimation.

In this paper, SD is research to identify the impact of cloud computing key factors in the number of virtual machines from a system perspective on cloud computing needs. Furthermore, the mathematical statistics theory and methods are applied on the decisive factor in the calculation of the scale of the resource pool construction. The estimation analysis and realization of the construction scale are proposed. Green Power can be achieved better, so as to promote a guide about the theory and practice of cloud computing.

III. CLOUD COMPUTING VIRTUAL MACHINES SCHEDULING MODEL

A. *Dynamical systems and applications*

SD is a kind of feedback control theory. Cause loop diagrams and stock-and-flow diagrams are used to describe the interrelated system. Computer simulation technology commonly is used to study the complex socio-economic system as a means of quantitative methods [8].

Public cloud virtual machine scheduling problem is actually the supply chain, and it's a nonlinear complex system, for there is time lags between the variables of the system. For example, the number of stock virtual machines, the number of active virtual machines and the number of idle virtual machines, etc., they all show significant nonlinear relationship. Moreover, the stock virtual machine number, virtual Machine recycling rate and virtual machine application rate, etc., they are changing over time. Accordingly their associated supply chains also change dynamically over time. These nonlinear factors greatly limit the effectiveness of the general mathematical research [9]. On the other hand, it is difficult to quantify the relationship between certain parameters, or there are insufficient data.

However, due to the structure of the SD model is based on a feedback loop. The existence of multiple feedback loops makes the system behavior is not sensitive for most of the parameters. Although the public cloud is lack of sufficient research data as a new application of large scale systems, some research work can be done with SD, as long as the estimated parameters are within their tolerance [10][11]。

B. *Cloud computing virtual machine scheduling system dynamics modeling*

From the upper service model view of the cloud computing virtual machine, the core of cloud computing system is virtual machine resources. Cloud computing virtual machines scheduling systems analysis and flow diagram analysis are shown in Figure 3, in which the SD hybrid diagram is used for the supply of public cloud virtual machines.

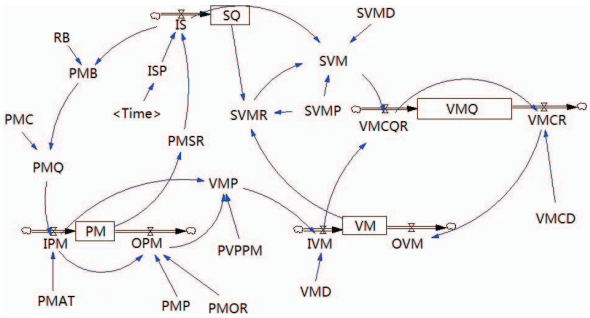


Figure 3. Public cloud dynamics hybrid model.

Among them, based on the assumption of the services carried, the physical machines and virtual machines construction scale demand are estimated respectively. Also, the multi-service reuse technology is considered in the parallel computing. Namely the service and the stock of the virtual machine affect the virtual machine reuse ratio, wherein there are a positive feedback loop of physical machines quantity and a negative feedback loop of virtual machines recovery on the service in the diagram. It is negligible about the slight impact of the physical machines scale on the virtualization efficiency, and the impact of the physical machines scale and virtual machines scale on virtual machines recovery efficiency [12].

Assumed that the amount of expected service provided (The items of abbreviations and corresponding meaning phrase are used as a whole, the same as below.):

Internet Service Provided (ISP) = WITH LOOKUP (Time,([(0,-4)-(100,4)],(0,1),(10,2),(20,4),(30,4),(40,3),(50,3),(60,2),(70,2),(80,1),(90,1),(100,1)))

Use the following table function of the analysis model:

(01)Service Quantity (SQ) = INTEG (Internet Service developed (IS), 1)

(02) Physical Machine quantity (PM) = INTEG (Input Physical Machine (IPM)-Output Physical Machine (OPM), 10)

(03) Virtual Machine quantity (VM) = INTEG (Input Virtual Machine (IVM)-Output Virtual Machine (OVM), 200)

(04) Service used Virtual Machine (SVM) =DELAY1 (Internet Service developed (IS)*normal Service Virtual Machine Permitted (SVMP)*(1-Service Virtual Machine Reused ratio (SVMR)), Service Virtual Machine Delay permitted (SVMD))

(05) Service Virtual Machine Reused ratio(SVMR) = WITH LOOKUP (Service Quantity(SQ)* Virtual Machine quantity(VM)/ normal Service Virtual Machine Permitted (SVMP),([(0,0)-(100000,1),(1,1)],(1,0),(70,0.07),(150,0.12),(250,0.18),(350,0.22),(480,0.27),(602.446,0.3),(1000,0.35),(10000,0.4),(100000,0.5)))

(06) Internet Service developed (IS) = Physical Machine scale to Service affect Ratio (PMSR)* Internet Service Provided (ISP)

(07) Physical Machine Quantity constructed (PMQ) = Physical Machine construction Budget (PMB)/ Physical Machine construction Cost (PMC)

(08) Input Physical Machine (IPM) = DELAY1 (Physical Machine Quantity constructed (PMQ), Physical Machine Quantity constructed (PMQ)* Physical Machine construction Time (PMAT))

(09) Physical Machine construction Budget (PMB) = Internet Service developed (IS)*physical machine construction expenditure ratio (RB)

(10) Output Physical Machine (OPM)=DELAY1(Input Physical Machine(IPM), Physical Machine elimination Period(PMP))* Physical Machine Output affect Ratio(PMOR)

(11) Virtual Machine recycled Quantity (VMQ) = INTEG (Virtual Machine increment for recycling (VMCQR) - ratio of Virtual Machine to be recycled (VMCR), 200)

(12) Output Virtual Machine recycled (OVM) = ratio of Virtual Machine to be recycled (VMCR)

(13) ratio of Virtual Machine to be recycled (VMCR) =DELAY1(Virtual Machine increment for recycling (VMCQR) , Virtual Machine recycle delay(VMCD))

(14) Virtual Machine increment for recycling (VMCQR) = MAX (Virtual Machine increment ratio (IVM) - Service used Virtual Machine(SVM),0)

(15) Virtual Machine increment ratio (IVM) =DELAY1 (Virtual Machine Produced ratio(VMP) , Virtual Machine on-line Delay(VMD))

(16) Virtual Machine Produced ratio (VMP) = (Input Physical Machine (IPM) - Output Physical Machine (OPM)) * normal physical machine virtualization rate (PVPPM)

(17) Physical Machine scale-Service affect ratio(PMSR) =WITH LOOKUP (Physical Machine quantity(PM), ([(0,0) - (10000,10)],(1,1),(10,1.1),(100,1.5),(1000,2),(10000,5)))

C. *Model initialization and parameter settings*

Delay initialization parameters settings and scale-related initialization parameters are listed in Table 1.

TABLE I. SYSTEM SIMULATION INITIALIZATION PARAMETERS

No.	Delay initialization parameter settings		Scale-related initialization parameters	
1	Service Virtual Machine Delay permitted (SVMD)	6	Physical Machine Output affect Ratio (PMOR)	0.1
2	Virtual Machine recycle delay (VMCD)	1	normal Service Virtual Machine Permitted (SVMP)	200
3	Virtual Machine on-line Delay (VMD)	1	normal physical machine virtualization rate (PVPPM)	30
4	Physical Machine elimination Period (PMP)	60	Physical Machine construction Cost (PMC)	1e+5
5	Physical Machine construction Time (PMAT)	1	physical machine construction expenditure ratio (RB)	1e+6

IV. SYSTEM DYNAMICS SIMULATION RESULTS AND ANALYSIS

A. System dynamics simulation results

Vensim is the most widely used modeling and simulation software in SD field. It contains almost all SD standard functions.

Vensim can describe and record the causality of system variables with the text and the arrows. Then the formula editor is used to establish the simulation model. During the Vensim modeling process, the established model can be analyzed, including the application of variables, their causality relationships and causality loops. After the simulation model is established, the model behavior can be deeply studied.

Through the above analysis about the public cloud SD modeling, the simulation tool - Vensim is applied.

Vensim simulation results of key indicators of the cloud dynamics virtual machine system are shown in Figure 4.

From the simulation results in Figure 4, it can be drawn that as time goes, in figure (a) the amount of Internet Service developed (IS) accumulates in the service development, and service quantity is increased, in Figure (b) Service Quantity (SQ) increases, in Figure (c) the number of physical machines - Physical Machine quantity (PM) and in Figure (d) the number of virtual machines VM - Virtual Machine quantity (VM) increases.

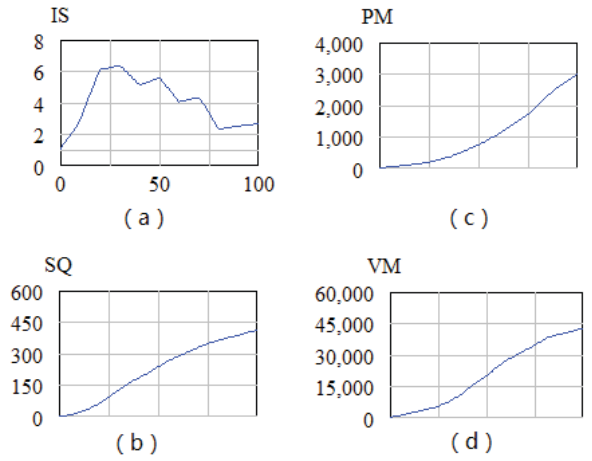


Figure 4. Public cloud virtual machine system dynamics simulation results: (a) Service increment to time, (b) Service quantity to time, (c) Physical machine quantity to time, (d) Virtual machine quantity to time.

B. System dynamics simulation results analysis

According to the above Vensim SD simulation result, the physical machines amount and virtual machines amount are qualitatively related with the number of service gradually growth, which is actually consistent with the actual operation situation of the service system. But this is not a simple linear correlation.

In order to estate the cloud computing construction scale, Matlab is applied to produce a scatter diagram of this data dependency analysis. Simulation program as:

```
>>x=[1 2 3 5  6 8 10 12 14 17 19 22];% Service Quantity(SQ) data. Limited to the length of the article, the following data is omitted.
>>y=[10 20 28 36 43 50 56 63 69 76 83 89];% Physical Machine quantity(PM), Virtual Machine quantity(VM). Limited to the length of the article, the following data is omitted.
>>plot(x,y,'r')
>>plot(x,y,'+')
```

The scatter diagram results of simulation are as Figure 5 and Figure 6:

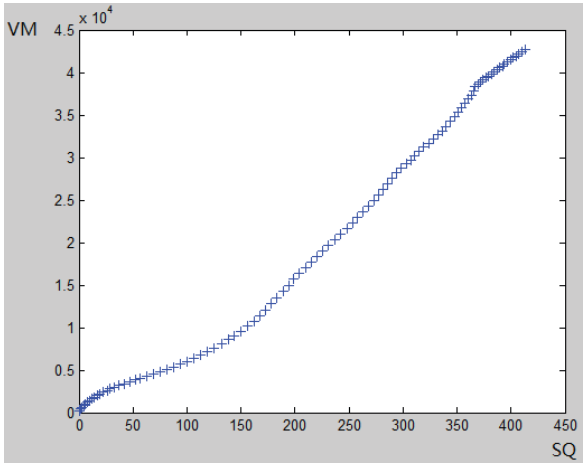


Figure 5. Service Quantity(SQ) and Virtual Machine quantity(VM) scatter diagram.

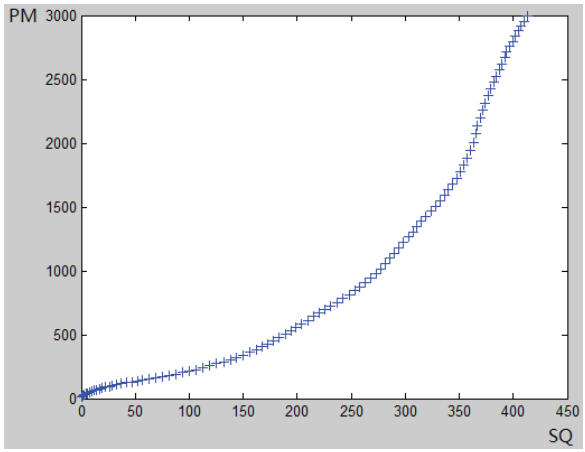


Figure 6. Service Quantity(SQ) and Physical Machine quantity(PM) scatter diagram.

It's concluded that the service quantity is leading to changes of the physical and virtual machines quantity, but it is not a simple linear proportional relationship.

In other words, it is untenable to calculate the physical and virtual machines construction scale linearly based on the service quantity. Or it will inevitably result in deviation from the real requirements of service systems and then constant allocation correction of real construction basis, the relevant human, financial and other social resources waste seriously.

V. CLOUD CONSTRUCTION SCALE ESTIMATION AND ARGUMENTATION

Based on the above nonlinear analysis, the cloud computing construction scale regression and estimation are analyzed in the following.

A. Cloud scale regression analysis

Since the Service Quantity(SQ) is the root causes to increase Virtual Machine quantity(VM), also in order to simplify the complexity, the Virtual Machine quantity(VM) y1 is considered relevant only to the Service Quantity(SQ) x, and the best fit is obtained by using the least squares method with existing simulation output data [13].

$$y_1 = m_n * x^n + \cdots + m_2 * x^2 + m_1 * x + b \qquad (1)$$

The simulation procedure is as follows by using of Matlab polyfit regression function:

>> x=[1 2 3 5 6 8 10 12 14 17 19 22 26 29] % Service Quantity(SQ) data. Limited to the length of the article, the following data is omitted.

>>y=[10 20 28 36 43 50 56 63 69 147 207 283 380 499] % Virtual Machine quantity(VM). Limited to the length of the article, the following data is omitted.

>>[p,s]=polyfit(x,y,n) % Use n-order polynomial fitting

>>plot(x,y,'*',x,f,'-') % View renderings n order polynomial fit

Different order of n is tested in order to minimize the order in Matlab simulations. It can be learned that the simulation fitted effects of the 2-order (n = 2) regression in Figure 7.

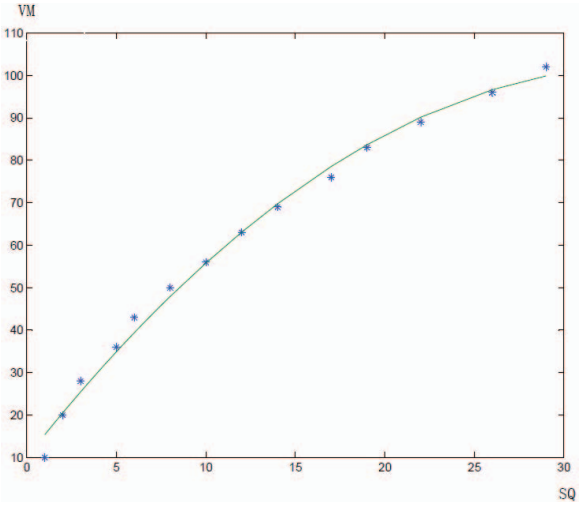


Figure 7. Virtual Machine quantity(VM) on a Service Quantity(SQ) of 2-order regression effects.

B. Estimation confidence argumentation

The following argues the cloud virtualization system regression equation to estimate the scale of construction in line with the confidence level α, to verify the regression model. Set confidence level α = 0.05, and use Matlab polyconf function to analyze fit situation of order n = 2. The simulation program is:

>> x=[1 2 3 5 6 8 10 12 14 17 19 22 26 29] % Service Quantity(SQ) data. Limited to the length of the article, the following data is omitted.

>>y=[10 20 28 36 43 50 56 63 69 147 207 283 380 499] % Virtual Machine quantity(VM). Limited to the length of the article, the following data is omitted.

>>[p,s]=polyfit(x,y,2) % Order n=2 polynomial fit

>> [yh,w]=polyconf(p,x,s,0.05)

>>plot(x,yh,'k-',x,yh-w,'k--',x,yh+w,'k--',x,y,'ks',[x;x],[yh;y],'k-')

Figure 8 is the simulation results. It can be concluded that the 2-order fit is able to achieve the confidence required.

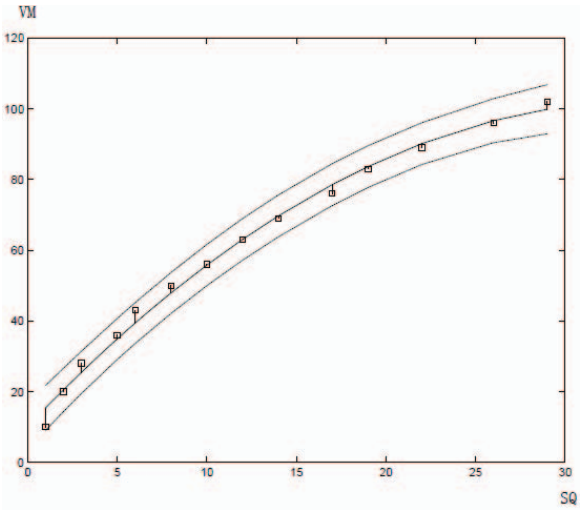


Figure 8. Virtual Machine quantity(VM) on a Service Quantity(SQ) of 2-order regression effects.

Thus, the 2-order fit is sufficient for the estimation model. By using the same simulation method for all samples data, that is, too: m2=0.14, m1=48.86, b=521.13, substitute into equation (1):

$$y_1 = 0.14*x^2 + 48.86*x + 521.13 \qquad (2)$$

Similarly, the regression equation Physical Machine quantity(PM) y2 and Service Quantity(SQ) x can be drawn as:

$$y_2 = 0.023*x^2 - 2.93*x + 171.61 \qquad (3)$$

In addition, the regression analysis method can also be applied with LINEST function of Microsoft Office Excel [14], the related procedure is omitted here.

VI. ARGUMENTATION AND CONCLUSION

The system dynamics relevant methods are applied accordingly above, the ICT service requirements and the construction scale are researched on IDC cloud computing from the system perspective，the key factor is found out which affects the quantity number of cloud computing virtual machines.

Further more, Matlab simulation tools are applied to identify the factor. The regression equation is proposed,

and the regression equation confidence is verified with theories and methods of mathematical statistics.

To be noted that Input Physical Machine (IPM) is never negative for lack of some data in the system simulation. This does not exactly reflect the full life cycle situation of service development, which is that Internet Service developed (IS) is negative while some services are out of market, so that Input Physical Machine (IPM) is negative. Subsequent researches will be continued on this topic.

On the whole, the cloud computing system construction scale is analyzed effectively with above approaches. The cloud computing virtual environment construction scale can be estimated successfully in this way, and the cloud computing system is optimized for Green Power. The methodology and results involved might give guide to promote the construction theory and practice of cloud computing at the same time.

ACKNOWLEDGMENT

First of all, we would like to extend sincere gratitude to the paper reviewer and English teacher Ms. Wang who help us to modify the paper for making it more perfect.

We thank to the National Ministry of Education project fund (Project No. 11YJC630106). We also acknowledge the facilities made available by Nanjing Institute of Mechatronic Technology (Project No. ZJ1303).

REFERENCES

[1] LUO Jun-zhou, JIN Jia-hui, SONG Ai-bo, DONG Fang, "Cloud computing: architecture and key technologies," Journal on Communication, vol.32, no.7, pp. 3-18, July 2011.

[2] CHEN Kang, ZHENG Wei-Min, "Cloud Computing : System Instances and Current Research," Journal of Software, vol.20, no.5, pp. 1337-1348, May 2009.

[3] ZHENG Pai, CUI Li-Zhen, WANG Hai-Yang, XU Meng, "A Data Placement Strategy for Data-Intensive Application in Cloud," Chinese Journal of Computers, vol.33, no.8, pp. 1472-1480, Aug. 2010.

[4] XU Xiao-long, WU Xiao-long, WU Jia-xing, YANG-Geng, CHEN Chun-ling, WANG Ru-chuan, "Mass data processing system based on large-scale low-cost computing platform," Application Research of Computers, vol.29, no.2, pp. 582-585, Feb.2012.

[5] ZTE Corporation, "Cloud Computing White Book," ZTE Corporation Nanjing Division, CN:ZTE Corporation Nanjing Division, 2012.

[6] Toby Velte, AnthonyVelte, Robert C. Elsenpeter, "Cloud Computing: A Practical Approach," U.S.A. Osborne/McGraw-Hill, 2009.

[7] John Rhoton, "Cloud Computing Explained," U.S.A. Recursive Limited, 2009

[8] CHEN Guo-wei, JIN Jia-shan, GENG Jun-bao, "Application Research Overview of System Dynamics," Control Engineering of China, vol.19, no.6, pp. 921-928, Nov.2012.

[9] LIN Chun-tao, SU Bao-cai, YU Jian-hui, "A Study on the Im pact of Oolong Tea Consumers Experience on Brand Loyalty with the M oderator of Gender," Science Technology and Industry, vol.12, no.9, pp. 32-36, Sep.2012.

[10] TANG Xu, ZHANG Bao-sheng, DENG Hong-mei, FENG Lian-yong, "Forecast and analysis of oil production in China based on system dynamics," Systems Engineering-Theory & Practice, vol. 30, no.2, pp. 207-211, Feb.2010.

[11] WU Guang-mou, SHENG Zhao-han, "Systems and Systems Methodology ," Nanjing: Southeast University Press, 2000.

[12] ZHOU De-qun, "Generality of Systems Engineering," 2nd edt., Beijing: Science Press, 2010.

[13] SHENG Zhou, XIE Shi-qian, PAN Cheng-yi, "Probability Theory and Mathematical Statistics," 4th edt., Beijing: Higher Education Press, 2010.

[14] Excel Home, "Excel 2010 Application Mannual," Beijing: People's Posts and Telecommunications Press, 2011.

Spectral domain decomposition method for physically-based rendering of photochromic/electrochromic glass windows

Guillaume Gbikpi-Benissan, Patrick Callet, Frédéric Magoulès
Ecole Centrale Paris, France
Email: frederic.magoules@hotmail.com

***Abstract*—This paper covers the time consuming issues intrinsic to physically-based image rendering algorithms. First, glass materials optical properties were measured on samples of real glasses and other objects materials inside an hotel room were characterized by deducing spectral data from multiple trichromatic images. We then present the rendering model and ray-tracing algorithm implemented in Virtuelium, an open source software. In order to accelerate the computation of the interactions between light rays and objects, the ray-tracing algorithm is parallelized by means of domain decomposition method techniques. Numerical experiments show that the speedups obtained with classical parallelization techniques are significantly less significant than those achieved with parallel domain decomposition methods.**

***Keywords*-Image rendering; Physically-based rendering; Optical; Ray-tracing; Domain decomposition; Parallel computing;**

I. Introduction

Nowadays, virtual reality is a powerful tool to design and preview real world manufacturing. This often implies to make up 3D computer-aided models displaying objects inside an environment, and to evaluate some particular behaviors according to some predefined properties. The scope of this study is to simulate the visual aspect of a room, depending on natural lighting and materials optical properties, and in particular glasses. The complexity of the interaction between light and materials is quite hard to entirely reproduce because of the heterogeneity of their optical behavior. Therefore, physically-based rendering of the visual aspect of a lighted scene can not be done by means of a simple trichromatic description of objects color. Instead, a model including spectral properties allows to simulate, for instance, the nuances induced by the photochromic adaptation of glasses. Commonly used physically-based rendering engines [1], [2], [3], are designed about the only extrinsic properties of materials, such as spectral reflectance and transmittance. A more complete approach additionally takes into account optical constants which are intrinsic properties. That is the case inside Virtuelium, the open source rendering software on which we experienced our study.

Beside the accuracy of a physical model of interactions between light and objects, a higher degree of computational complexity is introduced. As a consequence, ensuring fast simulations requires to compromise on the fineness of the rendering processes, hence on the quality of the resulting images. That is a particular concern in the context of real-time application where high level rendering is difficult to achieve. Parallel computing could be a privileged way to overcome the inherent time consuming issue of the physically-based rendering. Basic parallelization techniques, like splitting the set of pixels to be processed, raise a certain amount of efficiency problems. While dynamic load-balancing could mitigate the drawback related to the nonuniform distribution of objects and light sources in the whole scene, data size remains a limitation, principally due to data replication. With Domain Decomposition Methods (DDM) [4], [5], [6], [7], we benefit by data parallelism principles and techniques of information sharing based on interface conditions [8]. In [9], a ray-tracing domain decomposition method was proposed for accelerating the simulation of the propagation of acoustic waves. Given the good speedup results presented in [10], our proposal is to study the impact of the same approach in the context of physically-based image rendering.

In the following we first describe the optical properties considered for our simulation, and the specific methodology of acquisition on glass samples and trichromatic images depicting the 3D scene. Then, after the global rendering equation, we present rendering algorithms including those used in Virtuelium, mainly based on ray-tracing. Third, our parallel domain decomposition method for accelerating these algorithms is given. At last, we close our demonstration with a discussion about the speedups we obtained, and an example of rendered image.

II. Measurement of optical properties

Our study mainly focused on the effects of glass materials. With samples of glass, we can acquire reflection and transmission properties by means of spectrophotometry [11]. More precisely, for glass materials, this technique allows to obtain a Bidirectional Transmittance Distribution Function (BTDF). The process consists in determining the spectral response of a lighted material by repeating, for a known emission spectrum and for several points on the surface of the object, measurements of energetic quantities along a wavelength range, while varying both incident and view angles. Intrinsic properties called optical constants can be determined by a technique based on spectroscopic ellipsometry [12], [13]. It offers a way to measure the complex index of refraction defined by

$$\tilde{n}(\lambda) = n(\lambda) + ik(\lambda) = n(\lambda)(1 + i\kappa(\lambda)) \qquad (1)$$

The index of refraction, according to a wavelength λ, depends on the optical index $n(\lambda)$ and on the index of absorption $\kappa(\lambda)$. Unlike extrinsic properties which only

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.27

define a spectral response, optical constants represent the electronic behavior of dielectric materials. Yet, the characterization by optical constants is more adapted to materials satisfying Fresnel conditions of non-scattering and homogeneity. Metallic surfaces, for instance, are often described by means of optical constants for the simulation of their visual appearance [14], [15], [16].

The other kinds of materials inside the room scene were described by a completely different method, as there were no real samples of the objects to be represented. Indications about these materials were given by a set of images depicting their trichromatic appearance under various lighting conditions. There is no exact method to deduce spectral information from given trichromatic images, due to the fact that a given RGB value can be produced by infinity of different spectra. However, the Matrice-R theory, defined by Cohen and Kappauf in 1982 [17], is a possible solution to achieve accurate RGB-to-spectrum conversion. According to this theory, every spectrum can be decomposed into two components:

- the fundamental function that is unique and contains the color stimulus of the spectrum,
- a metameric black function that gives X=Y=Z=0 when converting to CIE XYZ.

An infinite number of metameric black functions exists. In term of calculation, the metameric black space is orthogonal to the color stimulus space. For this reason, Cohen defined a matrix equation we can use to compute the fundamental function from every RGB values. The practical accuracy of this method could be evaluated by computing spectra from a set of representative RGB values, then reconverting these spectra into RGB space, for qualitative comparison.

III. Rendering model

Our rendering equation is deduced from the Radiative Transfer Equation (RTE) and described by Kajiya [18] as follows:

$$L_r(\vec{\omega_o}) = \int_{\Omega} F_r(\vec{\omega_i}, \vec{\omega_o}) L_i(\vec{\omega_i}) \vec{n}.\vec{w_i} d\omega_i \qquad (2)$$

where Ω is a dome of incident lights. Knowing the Bidirectional Reflectance Distribution Function (BRDF), F_r, a remitted light L_r, in a direction $\vec{\omega_o}$, is computed from every incident light L_i emitted by the dome. The reasoning for the BTDF is the same except that we do not only consider a dome for incident lights but an entire sphere since the studied surface is non-opaque. On an applicative level, we only consider point or directional light sources, hence we simplify the equation (2) as follows:

$$L_r(\vec{\omega_o}) = \sum_{s1}^{N} F_r(\vec{\omega_s}, \vec{\omega_o}) L_s(\vec{\omega_s}) \vec{n}.\vec{w_s} \qquad (3)$$

where N is the number of light sources, L_s, the emission spectrum of the current light source s and $\vec{\omega_s}$, the incident direction.

Naturally, given that objects inside a 3D scene reflect a part of the light emitted from light sources, these objects should all be responsible for the illumination of the whole scene. That is what we call Global Illumination (GI). However, such a precise computation is sometimes avoided, as it is much simpler and faster to estimate only interaction between objects and light sources. That is local illumination. Our rendering software, Virtuelium, observes both local and global illuminations through two algorithms of each kind, respectively the "Scanline rendering" and the "Photon Mapping".

IV. Rendering algorithms

Local illumination: The "Scanline rendering" [19] is based on the inverse ray tracing algorithm [20]. Given that the image to be computed can be viewed as a matrix of pixels, a light ray is emitted from each pixel, orthogonally to the image plane. When a ray intersects the closest object on its path, we have to evaluate the received luminance at the given viewed direction. For that, new rays are shot from the hit point toward each light source, thus determining all the needed incident directions. Then, secondary rays are thrown regarding to reflection and/or refraction laws and the process is repeated. The algorithm stops when the energetic value of the ray goes bellow a threshold, or after the ray has bounced a predetermined number of times. The main difference between ray tracing algorithms lies in the way polygons of objects are sorted. In "Scanline rendering", every polygons are projected onto the image plane. Then, the image is computed line by line, from top to bottom, determining the color of each pixel by considering the closest polygons around. Another very common algorithm is the Z-buffer technique [21] which is nowadays implemented by default on graphic cards. The main advantage of the "Scanline rendering" is that each pixel is evaluated only once. In return, the memory cost is high because all the polygons of the scene must be loaded at the same time, leading to bad performances for scenes with complex geometries.

Global illumination: GI is a major progress in the quest of photo-realism, and a lot of very different techniques have been developed. "Radiosity" methods [22], [23] transform the phenomenon into a system of linear equations, solved either by direct method [24] (very effective but with a high complexity), or by iterative algorithms [25]. In another direction, the stochastic algorithm of "Monte-Carlo" [26] is sometimes used despite its slower convergence. "Path Tracing" methods [27] launch random rays from pixels of the image plane until one hits an object. It can be bi-directional (rays are shot from camera and sources simultaneously). "Metropolis Light Transport" algorithm (MTL) [28] optimizes "Path Tracing" by replacing the random shooting by heuristics. The "Photon Mapping" algorithm implemented in Virtuelium was first defined by Jensen in 1996, and is improved since this date [29], [30]. It decomposes the rendering process into two steps which are executed sequentially. In the pre-rendering step, the position of photons (light rays launched from a light source) hitting objects are stored in several appropriate structures (photon maps). At least, two photon

maps are needed, one for the global illumination itself and one for caustics. Then, a next step consists in evaluating four different contributions based on the fact that $L_r(\vec{\omega_o})$ can be decomposed into a sum of different integrals. First, the direct and specular contributions are computed the same way than in "Scanline Rendering". Then, the caustic and indirect diffuse contributions are deduced from the two photon maps. Unlike this version of "Photon Maping", most recent versions are progressive [28], [31].

V. Parallel computing

Commonly, a parallel image rendering algorithm decomposes the image grid, such that each pixel can be treated separately without any interaction. However, because of the heterogeneous spacing of objects, materials and lights-sources in the scene, it could be longer to compute some area of the image. Thus, a dynamic job-balancing mechanism is required to ensure that faster threads work more and that there is no inactivity period for any of them. The same idea can be applied to the set of light sources, to the light rays, or even to the spectral data when dealing with a full spectral rendering. But such a distribution requires to replicate the whole scene geometry onto each computational node. Indeed, on one hand, predetermining the whole light path of a ray is nearly impossible, and on the other hand, each polygons in the scene can be hit several times by different rays. Shared memory can be used to assure that only a single copy exists on a computational node but the problem remains on multiple-node architectures. Thus, we propose to apply the ray-tracing domain decomposition method introduced in [9].

By splitting a global domain into several small sub-domains, domain decomposition methods [4], [5], [6], [7] allow to load input data and to gather results in a parallel way, as each sub-domain can be associated to a unique processor. The method described in [9] takes advantage of some efficient domain decomposition techniques [32], [8], [33]. Besides the splitting of the global geometry itself, information along interfaces is shared between computational units which are processing neighboring sub-domains [34], [35]. A continuous approach [36], [37], [38], [39], [40], [41], [42] can be used to design efficient interface conditions. Similarly a discrete approach [43], [44], [45], [46], [47] can be used, which may increase significantly the performance of the algorithm. The link between the continuous and discrete interface condition can be established like in [48].

The method used in this work is based on the domain decomposition methods [49], [50], [51], [52], where here the interface conditions assure the continuity of the light ray properties (such as direction, amplitude, angle, etc.) from one sub-domain to another one. Yet, unlike classical domain decomposition methods, a computational unit does not process only one sub-domain. In order to cover load balancing issues, each processor starts by loading a certain number of sub-domains, according to memory limitation. When it remains few light rays to be handled in a sub-domain, this sub-domain is unloaded if there is still other currently not handled sub-domains with a lot of rays not processed. Then the processor starts loading one or more of these sub-domains while handling another sub-domain already available in the memory. Unloading sub-domains allows doing most of the results gathering during the processing of other rays. This overlapping gathering and processing is efficient since gathering mainly uses the communication system. A more complete description of an efficient implementation can be found in [9].

Table I
Speedup of the Virtuelium DDM program (Ethernet) with respect to the number of threads and sub-domains.

	16 threads	32 threads	64 threads	128 threads
1 sub-domain	10.6	16.7	25.4	20.2
2 sub-domains	11.9	22.1	34.3	45.4
4 sub-domains	10.4	22.3	35.1	50.7
8 sub-domains	11.2	24.2	39.8	66.9

VI. Results and discussions

An image rendered with Virtuelium is presented in Figure 1. The presented scene of a hotel room is visually a simplification but the glasses behavior, and the role of filter the windows are playing for the sunlight, are very accurate because of the direct use of Fresnel definition. Glasses are defined by using Fresnel indices of borosilicate glass Schott (BK7). Replacing these transparent glasses by colored glasses in Virtuelium simply means adapting the Fresnel definition of the material.

Speedups of Virtuelium execution are shown in Table I. They are very closed to results presented in [10] for the beam-tracing acoustic simulation software. Simulations were run on a hybrid, both distributed and shared memory, computational platform consisting of 4 nodes containing a quad core processor (a total of 16 cores). Each node were provided with 8 Gigabytes RAM (Random Access Memory). As we were expecting, DDM techniques significantly improved the performance of the parallelization. Although the speedups from acoustic simulation were quite better [10], we can notice that in both cases, from 16 to 128 threads, 8 sub-domains decomposition allowed to multiply the speedup by nearly 6, while classical parallelization only reach a factor less than 2. On another hand, for a fixed number of threads, the speedup keeps increasing as the number of sub-domains do.

VII. Conclusion

In this paper, we proposed an original ray-tracing domain decomposition method for image rendering with natural lighting. According to domain decomposition methods principle, light rays characteristics have been matched as interface constraints between neighboring sub-domains. We presented a test case on a model of a hotel room where we particularly deal with glass material properties. It outlined the performance and efficiency of our method, relatively to multi-core architectures.

Figure 1. Illustration of the image rendering of the interior of a hotel room.

AKNOWLEDGEMENTS

The authors acknowledge partial financial support from the Callisto-Sari project of the Pôle de Compétitivité Advancity, Ville et Mobilité Durables, Cap Digital Paris Region, France.

REFERENCES

[1] M Pharr and G Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010.

[2] LuxRender, 2014. Available online at http://www.luxrender.net/en_GB/index.

[3] P Shirley, R. K Morley, P-P Sloan, and C Wyman. Basics of physically-based rendering. In *SIGGRAPH Asia 2012 Courses*, SA '12, pages 2:1–2:11, New York, NY, USA, 2012. ACM.

[4] B.F. Smith, P.E. Bjorstad, and W.D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, UK, 1996.

[5] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, Oxford, UK, 1999.

[6] A. Toselli and O. Widlund. *Domain Decomposition methods: Algorithms and Theory*. Springer, 2005.

[7] J Kruis. *Domain Decomposition Methods for Distributed Computing*. Saxe-Coburg Publications, 2007.

[8] Y. Maday and F. Magoulès. Absorbing interface conditions for domain decomposition methods: a general presentation. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3880–3900, 2006.

[9] F. Magoulès. Décomposition de domaines pour le lancer de rayons. France Patent no. 1157329. 12 August 2011.

[10] F. Magoulès, R. Cerise, and P. Callet. A beam-tracing domain decomposition method for sound holography in church acoustics. In *Distributed Computing and Applications to Business, Engineering Science (DCABES), 2013 12th International Symposium on*, pages 61–65, Sept 2013.

[11] M. Bass. *Devices, measurements, and properties*. Number 2 in Handbook of Optics. McGraw-Hill and Optical Society of America, 1995.

[12] E.D. Palik. *Handbook of Optical Constants of Solids*. Number 1. Elsevier Science, 1985.

[13] P. Callet. *Couleur-lumière, couleur-matière: interaction lumière-matière et synthèse d'images*. Sciences en actes. Diderot éditeur, Arts et sciences, 1998.

[14] K Berger, A Wilkie, A Weidlich, and M Magnor. Modeling and verifying the polarizing reflectance of real-world metallic surfaces. *Computer Graphics and Applications*, 32(2):24–33, March 2012.

[15] J A. Woollam, W.A. McGaham, and B. Johs. Spectroscopic ellipsometry studies of indium tin oxide and other flat panel display multilayer materials. *Thin Solid Films*, 241(12):44 – 46, 1994. Papers presented at the European Materials Research Society 1993 Spring Conference, Symposium C: Ion Beam, Plasma, Laser and Thermally-Stimulated Deposition Processes, Strasbourg, France, May 47, 1993.

[16] G.E. Jellison and F.A. Modine. Optical constants for silicon at 300 and 10 k determined from 1.64 to 4.73 ev by ellipsometry. *Journal of Applied Physics*, 53(5):3745–3753, May 1982.

[17] J B Cohen and W E Kappauf. Metameric color stimuli, fundamental metamers, and wyszecki's metameric blacks. *The American journal of psychology*, 95(4):537–564, 1982.

[18] J T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, August 1986.

[19] C Wylie, G Romney, D Evans, and A Erdahl. Half-tone perspective drawings by computer. In *Proceedings of the November 14-16, 1967, Fall Joint Computer Conference*, AFIPS '67 (Fall), pages 49–58, New York, NY, USA, 1967. ACM.

[20] J. Arvo. Backward ray tracing. In *ACM SIGGRAPH 86 Course Notes - Developments in Ray Tracing*, pages 259–263, 1986.

[21] E E Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces.* PhD thesis, 1974. AAI7504786.

[22] J R. Wallace, M F. Cohen, and D P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *SIGGRAPH Comput. Graph.*, 21(4):311–320, August 1987.

[23] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '89, pages 335–344, New York, NY, USA, 1989. ACM.

[24] J Bu and Ed F. Deprettere. A vlsi system architecture for high-speed radiative transfer 3d image synthesis. *The Visual Computer*, 5(3):121–133, 1989.

[25] M F. Cohen, S E Chen, J R. Wallace, and D P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, pages 75–84, New York, NY, USA, 1988. ACM.

[26] E P. Lafortune. *Mathematical models and Monte Carlo algorithms for physically based rendering*. PhD thesis, Department of Computer Science, K.U.Leuven, Leuven, Belgium, February 1995.

[27] A Pajot, L Barthe, M Paulin, and P Poulin. Combinatorial bidirectional path-tracing for efficient hybrid cpu/gpu rendering. *Computer Graphics Forum*, 30(2):315–324, 2011.

[28] T Hachisuka, S Ogaki, and H W Jensen. Progressive photon mapping. *ACM Trans. Graph.*, 27(5):130:1–130:8, December 2008.

[29] H W Jensen. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, pages 21–30, London, UK, 1996. Springer-Verlag.

[30] H W Jensen. A practical guide to global illumination using ray tracing and photon mapping. In *ACM SIGGRAPH 2004 Course Notes*, SIGGRAPH '04, New York, NY, USA, 2004. ACM.

[31] T Hachisuka and H W Jensen. Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph.*, 30(5):114:1–114:11, 2011.

[32] F. Magoulès, K. Meerbergen, and J.-P. Coyette. Application of a domain decomposition method with Lagrange multipliers to acoustic problems arising from the automotive industry. *Journal of Computational Acoustics*, 8(3):503–521, 2000.

[33] F. Magoulès and F.-X. Roux. Lagrangian formulation of domain decomposition methods: a unified theory. *Applied Mathematical Modelling*, 30(7):593–615, 2006.

[34] C. Farhat, A. Macedo, M. Lesoinne, F.-X. Roux, F. Magoulès, and A. de la Bourdonnaye. Two-level domain decomposition methods with Lagrange multipliers for the fast iterative solution of acoustic scattering problems. *Computer Methods in Applied Mechanics and Engineering*, 184(2–4):213–240, 2000.

[35] A. de la Bourdonnaye, C. Farhat, A. Macedo, F. Magoulès, and F.-X. Roux. A non-overlapping domain decomposition method for the exterior Helmholtz problem. *Contemporary Mathematics*, 218:42–66, 1998.

[36] B. Després. Domain decomposition method and the Helmholtz problem.II. In *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993)*, pages 197–206, Philadelphia, PA, 1993. SIAM.

[37] S Ghanemi. A domain decomposition method for Helmholtz scattering problems. In P. E. Bjørstad, M. Espedal, and D. Keyes, editors, *Ninth International Conference on Domain Decomposition Methods*, pages 105–112. ddm.org, 1997.

[38] P. Chevalier and F. Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.

[39] M.J. Gander, L. Halpern, and F. Magoulès. An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. *International Journal for Numerical Methods in Fluids*, 55(2):163–175, 2007.

[40] Y. Maday and F. Magoulès. Optimized Schwarz methods without overlap for highly heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1541–1553, 2007.

[41] Y. Maday and F. Magoulès. Improved ad hoc interface conditions for Schwarz solution procedure tuned to highly heterogeneous media. *Applied Mathematical Modelling*, 30(8):731–743, 2006.

[42] Y. Maday and F. Magoulès. Non-overlapping additive Schwarz methods tuned to highly heterogeneous media. *Comptes Rendus à l'Académie des Sciences*, 341(11):701–705, 2005.

[43] F. Magoulès, F.-X. Roux, and S. Salmon. Optimal discrete transmission conditions for a non-overlapping domain decomposition method for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 25(5):1497–1515, 2004.

[44] F.-X. Roux, F. Magoulès, L. Series, and Y. Boubendir. Approximation of optimal interface boundary conditions for two-Lagrange multiplier FETI method. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Proceedings of the 15th International Conference on Domain Decomposition Methods, Berlin, Germany, July 21-15, 2003*, Lecture Notes in Computational Science and Engineering. Springer-Verlag, Haidelberg, 2005.

[45] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approach to absorbing boundary conditions for the Helmholtz equation. *International Journal of Computer Mathematics*, 84(2):231–240, 2007.

[46] F. Magoulès, F.-X. Roux, and L. Series. Algebraic way to derive absorbing boundary conditions for the Helmholtz equation. *Journal of Computational Acoustics*, 13(3):433–454, 2005.

[47] F. Magoulès, F.-X. Roux, and L. Series. Algebraic Dirichlet-to-Neumann mapping for linear elasticity problems with extreme contrasts in the coefficients. *Applied Mathematical Modelling*, 30(8):702–713, 2006.

[48] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3742–3759, 2006.

[49] Y. Maday and F. Magoulès. Optimal convergence properties of the FETI domain decomposition method. *International Journal for Numerical Methods in Fluids*, 55(1):1–14, 2007.

[50] F. Magoulès, P. Iványi, and B.H.V. Topping. Convergence analysis of Schwarz methods without overlap for the Helmholtz equation. *Computers and Structures*, 82(22):1835–1847, 2004.

[51] F. Magoulès, P. Iványi, and B.H.V. Topping. Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 193(45–47):4797–4818, 2004.

[52] F. Magoulès and R. Putanowicz. Optimal convergence of non-overlapping Schwarz methods for the Helmholtz equation. *Journal of Computational Acoustics*, 13(3):525–545, 2005.

An Efficient Algorithm for Solving Eigenproblem

Huirong Zhang
Laboratory of Parallel Software and Computational Science of Software
Institute of Software Chinese Academy of Sciences
Graduate University of Chinese Academy of Sciences
Beijing, China
zhang06.happy@163.com

Jianwen Cao
Laboratory of Parallel Software and Computational Science of Software
Institute of Software Chinese Academy of Sciences
Beijing, China
caojianwen@gmail.com

***Abstract*—In this paper, we consider second order elliptic ODE eigenproblems on general grids . We construct an efficient algorithm for computing the eigenvalue by using weighted mean combination of the linear finite element method and corresponding 2nd-order finite difference method .We first take the arithmetic mean of the two methods. Then we compute the quasi-optimal combined parameters for different eigenvalues to improve our efficient algorithm . The algorithm we construct convergence faster and have higher accuracy than the linear finite element method and corresponding 2nd-order finite difference method . Some numerical examples tested on both uniform meshes and nonuniform meshes are given to illustrate the computational cost of different numerical methods for solving eigenvalue problems.For efficiency, all the matrices use sparse storage in our algorithm.**

***Keywords*-Efficient algorithm; combinatorial; quasi-optimal; eigenproblem.**

I. Introduction

Eigenproblems have many applications in both applied mathematics and engineering ,such as quantum mechanics,eigenfaces in image processing, vibration analysis , principal components analysis about large data sets etc. In engineering applications there are many solving methods, therein linear finite element method or 2nd-order finite difference method ,which always involve inevitably massive meshes and large-scale computing [8] [7] [1].

Currently, many large-scale scientific computing problems use domain decomposition for parallel computing. In specific computation one often adopt enough mesh refinement to meet accuracy requirement. Alternatively, one can design efficient algorithm for specific problem to obtain required precision. This is just our theme in this paper. At present ,there are many research on constructing high accuracy algorithms , such as [3] [4] [5] [9] [10] [12] [11] [13]. In this paper we will give two efficient algorithms for the general ODE eigproblem (10) on general meshes.

$$\begin{aligned} &Lu = (-pu')' + qu = \lambda u \qquad x \in (a,b) \\ &(u^2)'|_{\partial\Omega} = 0 \end{aligned} \tag{1}$$

where $p \in C^1(I)$, $p(x) \geq \min_{x\in I} p(x) = p_{min} > 0$, $q \in C^1(I)$ $\overline{I} = [a,b]$.

In this paper , denote $(f,g) := \int_0^1 f(t)g(t)dt$.

As is well-known that the (10) eigenvalue problem $Lu = \lambda u$ can be rewritten as a variational problem. The Rayleigh quotient is defined as

$$R[v] := \frac{(Lv,v)}{(v,v)} = \frac{a(v,v)}{(v,v)}. \tag{2}$$

Where $a(v,v)$ is a weak form of the original energy form. For $(Lv,v) > 0$, eigenvalues satisfy the following min-max principle

$$\lambda_1 = \min_{v\in H} R[v], \qquad \lambda_k = \min_{E_k} \max_{v\in E_k} R[v], \tag{3}$$

$$0 < \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_k \to \infty, \tag{4}$$

where H is called admissible space and E_k is a k- dimension subspace of H.

For a given finite-dimensional approximation subspace V^h in H, the standard discrete Rayleigh-quotient is defined as

$$R[v^h] := \frac{a(v^h,v^h)}{(v^h,v^h)}. \tag{5}$$

The min-max principle also holds for the discrete form

$$\lambda_k^h = \min_{V_k} \max_{v^h\in V_k} R[v^h], \tag{6}$$

$$0 < \lambda_1^h < \lambda_2^h \leq \cdots \leq \lambda_k^h \leq \cdots \leq \lambda_{DOF}^h, DOF = dim(V^h). \tag{7}$$

Here V_k ranges over all k-dimensional subspace in V^h.

For a given basis $\{\phi_1,...,\phi_N\}$ in V^h and $v^h = \sum q_j\,\phi_j$, the discrete Rayleigh-quotient becomes

$$R[v^h] = \frac{q^T\,K^h\,q}{q^T\,M^h\,q}, \tag{8}$$

where K^h and M^h are called as stiffness matrix and mass matrix, respectively. K^h is symmetry and M^h is symmetry positive definite because it is the Gram matrix for the linear independent vectors $\{\phi_j,...,\phi_N\}$. Then, the critical points of this discrete quotient are the solutions to the generalized matrix eigenvalue problem with real eigenvalues as follows

$$K^h\,Q^h = \lambda^h\,M^h\,Q^h, \quad K^h = (a(\phi_j,\phi_k)), \quad M^h = ((\phi_j,\phi_k)). \tag{9}$$

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.28

In [3], Sun presented two fourth-order accuracy schemes for eigenvalue problem on uniform mesh.In [4] Sun even gave sixth-order schemes by linear combination of the fourth-order accuracy schemes with a proper combination coefficient.But all those high-order accuracy schemes are just applicable to uniform meshes.In [9],they generalize Sun's idea to nonuniform meshes and gave two fourth-order accuracy schemes and two sixth-order accuracy schemes .But all these high-order accuracy schemes only aim at eigenvalue problem (1) with constant coefficient in the eigen-equation,namely p=1 and q=0.

In this paper, we generalize the combinational method to construct our efficient algorithm. It is applicable to general eigenvalue problem (1) with variable coefficient q and constant p in the eigen-equation.Without loss of generality,we take the constant $p = 1$ henceforth.that is,

$$\begin{aligned} &Lu = (-u')' + qu = \lambda u \qquad x \in (a,b), \\ &(u^2)'|_{\partial\Omega} = 0. \end{aligned} \tag{10}$$

Although our efficient algorithm can not guarantee high-order accuracy, they both convergence faster than the linear finite element method and the corresponding finite difference method on the same meshes. Some numerical examples tested on both uniform meshes and nonuniform meshes are given to illustrate the computational cost of different numerical methods for solving eigenvalue problems.For efficiency, all the matrices use sparse storage in our algorithm.

Typically,we conduct all our numerical experiment about the eigenproblem (10) when the boundary condition is $u(0) = u(1) = 0$. We compute the error for every different variable coefficients q on both uniform mesh and nonuniform mesh.

II. An efficient algorithm for ODE eigenvalue problems

In this section we construct our efficient algorithm by taking weighted mean combination of the linear finite element method and corresponding 2nd-order finite difference method.

Firstly ,we suppose u is the eigenfunction of problem (10),s is the linear interpolation of u on mesh:

$$a = x_0 < x_1 < \cdots x_n = b. \tag{11}$$

Henceforth,we let h_i denotes $x_i - x_{i-1}$, $mesh[h]$ denotes such a mesh where h is the average step length of the mesh.We define vector U as $U(i) = u(x_i)$. Then,the eigenvalue expression with finite element method is as below:

$$\lambda_F := \frac{(s',s') + (qs,s)}{(s,s)}. \tag{12}$$

Then we define the finite difference scheme corresponding the above linear finite element scheme as below.

$$\lambda_D := \frac{(s',s') + (qu,u)}{(u,u)}. \tag{13}$$

From [2],we know the numerical eigenvalues computed with finite element scheme discreted by finite element method is greater than the genuine eigenvalue of (10).Conversely, the corresponding difference scheme is less than the genuine eigenvalue of (10).Based on this fact ,we know any combinational scheme in the form of

$$\lambda_C = \alpha\lambda_D + (1-\alpha)\lambda_F, \quad \text{for}\, \alpha > 0 \tag{14}$$

can obtain better approximate solution for (10) than individual finite element scheme λ_F or the corresponding difference scheme λ_D.

We can choose proper combination coefficient α to construct efficient algorithm for solving eigenproblem (10).

A. Theoretical analysis

We first take the arithmetic mean of (12) and (13) to construct efficient algorithm and give the error analysis and some numerical experiments.

In [9] and [3] they both take arithmetic mean combination of λ_D and λ_F ,namely $\alpha = \frac{1}{2}$,to construct fourth-order accuracy schemes for the model eigenproblem with constant coefficient q and p. Similarly,we also take mean combination of λ_D and λ_F for the eigenproblem with variable coefficient q. Differently,we can't directly use the Rayleigh-quotient form (13),(12) to compute the numerical eigenvalues just as [9] and [3] ,because we don't know the genuine eigenfunctions.It is practically impossible to find some kind of meshes satisfy $U^TU = (u,u)$. In this case,we can only compute the numerical eigenvalue (λ_F) by solving a generalized matrix eigenvalue problem in the form of (9).

We define the corresponding difference scheme λ_D

$$\widetilde{K}^h U^h = \lambda^h \widetilde{M}^h U^h, \tag{15}$$

where $\widetilde{K}^h$, is obtained by lumping the mass component in $(a(\phi_j,\phi_k)$ of K^h which stems from qu of (10). $\widetilde{M}^h$ is obtained by lumping M^h to a diagonal matrix.

Next ,we present our algorithm .

B. Algorithm structure

In our algorithm ,we assemble K^h with two part A and C , assemble $\widetilde{K}^h$ with A and CD ,where CD is the lumping matrix of C with proper boundary treatment. Specifically, $K^h = A + C, A = (a_{i,j})) = (\phi_j', \phi_i'), C = (c_{i,j}) = (q\phi_j, \phi_i)$,where $\phi_i, i = 1, \ldots$ is the basis of s.$\widetilde{K}^h = A + CD$. $M = (b_{i,j}) = (\phi_j, \phi_i)$,$D$ is the lumping matrix of M with proper boundary treatment. For efficiency and memory limit,we assemble all the matrix with sparse storage.Then use ARRACK software to solve the corresponding generalized eigenvalue problem with compressed matrix . Our algorithm structure see Algorithm 1.

Algorithm 1 MCM(n,m)

Require: Two integer $n \geq 0, m \geq 0$,Real $x1, x2$,Bool sp.
Ensure: The numerical eigenvalues of $\lambda_1 \cdots \lambda_m$.
1: $ld = FDM(n,m)$;$lf = FEM(n,m)$;
2: $la = (lf + ld)/2$;
Return la.

The complete iterative structure see Algorithm 2.

Algorithm 2 ITMCM(n, m, eps, λ_0)

Require: Two integer $n_0 \geq 0, m \geq 0$,Real eps, λ_0.

Ensure: The numerical eigenvalues of $\lambda_1 \cdots \lambda_m$,Meshes number n.

1: $la = MCM(n, m)$;
2: $er = la - \lambda_0$; $error = norm(er, 1)$;
3: **while** $(error > eps)$
$\lambda_0 = la$**;**
$n = 2n$;
$la = MCM(n, m)$**;**
$er = la - \lambda_0$**;**
$error = norm(er, 1)$**;**

Return la**,**n**.**

III. Improved efficient algorithm

In this section we will give a improved algorithm for the algorithm given in above section by computing the quasi-optimal combined parameters for every eigenvalue.Firstly, we compute $ld1 = FDM(n, m)$,$lf1 = FEM(n, m)$ and $ld2 = FDM(2n, m)$,$lf2 = FEM(2n, m)$.Then ,define

$$\mu := -(ld2 - ld1)./(lf2 - lf1),$$
$$\alpha := 1./(1 + \mu).$$

α is just the quasi-optimal combined parameters for (14) . Lastly, we iterate above steps and update α until meet accuracy requirement . The improved algorithm see Algorithm 3.

Algorithm 3 QOCM(n, m, eps, λ_0)

Require: Two integer $n_0 \geq 0, m \geq 0$,Real eps, λ_0.

Ensure: The numerical eigenvalues of $\lambda_1 \cdots \lambda_m$,Meshes number n.

1: $ld1 = FDM(n, m)$;$lf1 = FEM(n, m)$;
2: $n = 2n$,
$ld2 = FDM(n, m)$;$lf2 = FEM(n, m)$;
3: alf1=-$(ld2 - ld1)./(lf2 - lf1)$;
a1=1./(1+alf1);
4: **For** $i = 1, \ldots, m$
$lc(i) = (1 - a1(i)) * lf2(i) + a1(i) * ld2(i)$;
5: $er = lc - \lambda_0$;
$error = norm(er, 1)$**;**
6: **while** $(error > eps)$
$\lambda_0 = lc$**;**
$lf1 = lf2$;
$ld1 = ld2$; $n = 2 * n$;
$ld2 = FDM(n, m)$**;**$lf2 = FEM(n, m)$
alf1=-$(ld2 - ld1)$**./**$(lf2 - lf1)$**;**
a1=1./(1+alf1);
For $i = 1, \ldots, m$
$lc(i) = (1 - a1(i)) * lf2(i) + a1(i) * ld2(i)$;
$er = lc - \lambda_0$**;**
$error = norm(er, 1)$**;**

Return lc**,**n**.**

Algorithm 4 FEM(n, m)

Require: Two integer $n \geq 0, m \geq 0$,Real $x1, x2$,Bool sp.

Ensure: The numerical eigenvalues of $\lambda_1 \cdots \lambda_m$.

1: **If** $sp = 0$
n=n-1;
If $sp = 1$
n=n+1;
2: **col=zeros(3*n-2,1); lin=col;**
A**=double(zeros(3*n-2,1));**
$B = A; C = A$;
3: **Obtain the Interval subdivision of** $[x1, x2]$**.**
$[x, H] = mesh(n, x1, x2)$;
4: **Assign col,lin;**
Left boundary treatment;
5: **For** $i = 1, \cdots, n - 2$**.**
Assign C **by computing** $c_{i,i-1}, c_{i,i}, c_{i+1,i}$**,**
Assign A **by computing** $a_{i,i-1}, a_{i,i}, a_{i+1,i}$**,**
Assign B **by computing** $b_{i,i-1}, b_{i,i}, b_{i+1,i}$**,**
6: **Right boundary treatment;**
7: $AF = A + C$;
K**=sparse**$(lin, col, AF, n - 1, n - 1)$**;**
M**=sparse**$(lin, col, B, n - 1, n - 1)$**;**
8: lf**=eigs(**$K, M, m,' sm'$**);**
lf**=sort(**lf**);**

Return lf**.**

Algorithm 5 FDM(n, m)

Require: Two integer $n \geq 0, m \geq 0$,Real $x1, x2$,Bool sp.

Ensure: The numerical eigenvalues of $\lambda_1 \cdots \lambda_m$.

1: **Update** n **according** sp**;**
2: **Initialize vectors** $A, CD, D; col, lin$;
3: **Obtain the Interval subdivision of** $[x1, x2]$**.**
$[x, H] = mesh(n, x1, x2)$;
4: **Assign** col, lin**;**
Left boundary treatment;
5: **For** $i = 1, \cdots, n - 2$
Assign CD**by computing** $cd_{i,i}$**,**
Assign A **by computing** $a_{i,i-1}, a_{i,i}, a_{i+1,i}$**,**
Assign D **by computing** $d_{i,i}$**,**
6: **Right boundary treatment;**
7: $AD = A + CD$;
$\widetilde{K}$**=sparse**$(lin, col, AD, n - 1, n - 1)$**;**
$\widetilde{M}$**=sparse**$(lin, col, D, n - 1, n - 1)$;
8: ld**=eigs(**$\widetilde{K}, \widetilde{M}, m,' sm'$**);**
ld**=sort(**ld**);**

Return ld**.**

IV. Numerical examples

In this section ,we give some numerical examples to test our efficient algorithm MCM(n,m)and QOCM(n,m) Then, compare to the algorithms FEM(n,m),FDM(n,m) computed with finite element scheme and the corresponding finite d-ifference scheme. Typically,we conduct all our numerical

experiments about the eigenproblem (10) when the boundary condition is $u(0) = u(1) = 0$.Calling the algorithms FEM(n,m),FDM(n,m),MCM(n,m)and QOCM(n,m),we respectively compute the smallest 4 eigenvalues ld, lf, la, lc for the cases $q(x) = 2cos(2x)$ and $q(x) = x^3$ on both uniform mesh and nonuniform mesh.We take $x \in [0, \pi/2]$, the max iterative error $max_i \lambda_i$ of the 4 eigenvalues to be less than $eps = 1.0e-4$, initial eigenvalues vector $\Lambda_0 = \overrightarrow{0}$, mesh number starter $n = 20$. With the same accuracy requirement as above, we test the meshes number n and time cost of the four algorithms.All the example are tested on four kinds of meshes as following:

NO.1 $x_j = \pi(1 - cos(j\pi h))/4, h = 1/n, j = 1 \cdots, n.$
NO.2 $x_j = \pi tan(j\pi h/4)/2, h = 1/n, j = 1 \cdots, n.$
NO.3 $x_j = \pi(jh)^2/2, h = 1/n, j = 1 \cdots, n.$
NO.4 $x_j = \pi(jh)/2, h = 1/n, j = 1 \cdots, n.$

Case1 : $q(x) = 2cos(2x)$.In this case,the eigenequation is just the Mathieu equation The smallest 4 numerical eigenvalues are $\Lambda = (3.9170, 16.0330, 36.0143, 64.0079)$.
The iterative errors for the four algorithms tested on NO.1-NO.4 meshes with accuracy requirement $eps = 1.0e-4$ see Fig 1,time cost see Fig 2 and mesh number need see Table II.

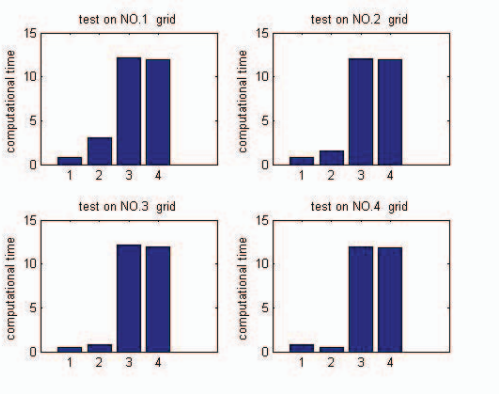


Fig. 2. time cost for $q(x) = 2cos(2x)$

TABLE I
MESH NUMBER NEED FOR $q(x) = 2cos(2x)$

Mesh NO	QOCM	MCM	FEM	FDM
NO.1	320	2560	10240	10240
NO.2	320	1280	10240	10240
NO.3	160	640	10240	10240
NO.4	320	320	10240	10240

Case2 : $q(x) = x^3$. The smallest 4 numerical eigenvalues are $\Lambda = (10.4672, 29.0874, 51.6191, 80.0499)$.
The iterative errors tested on NO.1-NO.4 meshes with accuracy requirement $eps = 1.0e-4$ see Fig3,time cost see Fig4.

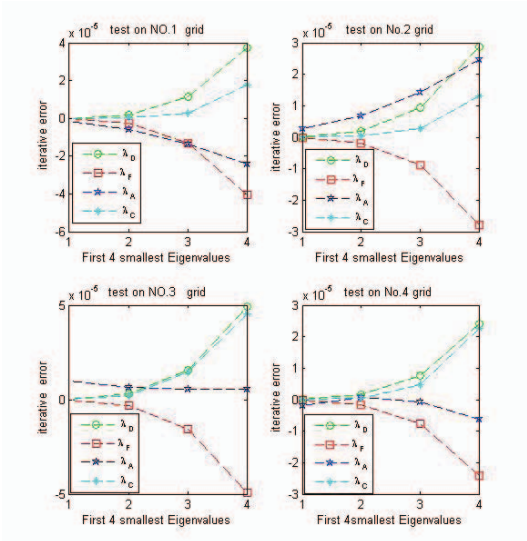


Fig. 1. Error Results for $q(x) = 2cos(2x)$

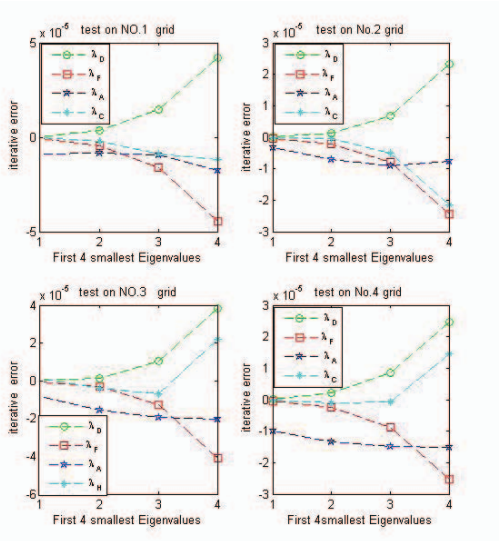


Fig. 3. Error Results for $q(x) = x^3$

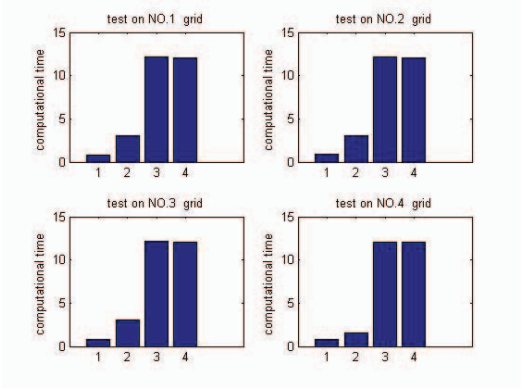


Fig. 4. Test Results for $q(x) = x^3$

TABLE II
MESH NUMBER NEED FOR $q(x) = 2cos(2x)$

Mesh NO	QOCM	MCM	FEM	FDM
NO.1	320	2560	10240	10240
NO.2	320	2560	10240	10240
NO.3	160	2560	10240	10240
NO.4	320	1280	10240	10240

From the figures and tables we can know ,with the same error requirement,for different variable coefficient $q(x)$ and different kinds of mesh ,the improved algorithm $QOCM$ is always much better than the algorithms FEM and FDM. Algorithm $QOCM$ has much lower time cost than all the other three algorithm but has the same level of accuracy.Moreover,the mesh number that Algorithm FEM and FDM need nearly 30 times of $QOCM$ .Although the algorithm MCM is not better than the $QOCM$, it is also more efficient than the algorithms FEM and FDM.Particularly, on the uniform mesh, MCM has the same efficiency with $QOCM$ or even better than the latter.

V. Conclusion

Although our efficient algorithm can not guarantee high-order accuracy, it convergence faster than the linear finite element method and the corresponding finite difference method on the same meshes. The numerical experiments in above section show that ,with the same error requirement,for different variable coefficient $q(x)$ and different kinds of meshes,the algorithm $QOCM$ and MCM take less time and need less mesh number than the algorithms FEM and FDM.The mesh number that Algorithm FEM and FDM need nearly 30 times of $QOCM$.Namely,our two algorithms really more efficient than algorithms FEM and FDM and are applicable to compute several eigenvalues at the same time. Moreover, the algorithm is easy to parallelize and the corresponding parallel algorithm has good scalability.

References

[1] Strang G, J. G. *An Analysis of the Finite Element Method*,2nd edition. Wellesley-Cambridge, Englewood Cliffs, N.J., 2008.

[2] George E, Wolgang R. *FINITE DIFFERENCE METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS.* John WIley & Sons,Inc.,1960.

[3] Sun J C. *New schemes with fractal error compensation for PDE eigenvalue computations.* Science China mathmatics, Vol.57,No.2,pp.221-244, 2014.

[4] Sun J C. *Multi-Neighboring Grids Schemes for solving PDE eigenproblems.* Science China mathmatics,Vol.56,No.12,pp.2677-2700,2013.

[5] Lin Q, Xie H H. *Extrapolation of the linear elements on general meshes.* International Journal of numerical analysis and modeling, Vol 10,No.1,pp.139-153,2013.

[6] A.B. Andreev ,V.A. Kascieva ,M. Vanmaele. *Some results in lumped mass finite-element approximation of eigenvalue problems using numerical quadrature formulas*. Journal of Computational and Applied Mathematics,Vol 43,pp.291-311,1996.

[7] E. Hinton, T. Rock and O.C.Zienkiewicz. *A note on mass lumping and related processes in the finite element method* . Earthquake Engineering and Structural Dynamics,Vol.4,pp.245-249,1976.

[8] O.C.Zienkiewicz,R.L.Taylor. *Finite element method*,5th edition. Butterworth-Heinemann 2000.

[9] Zhang H R.,Cao J W., Sun J C. *Calculation and analysis of several high accuracy scheme for solving ODE eigenproblem on non-uniform grid.* Submited to journal on numercal methods and computer applications,Vol.35,No.2,pp.131-152,Jun.2014.

[10] Barth T J. *Recent developments in high order k-exact reconstruction on unstructured meshes.* In: AIAA 93, AIAA-93-0668, AIAA, Reno Nevada, pp.1-15,1993.

[11] Dai X, Xu J, Zhou A. *Convergence and optimal complexity of adaptive finite element eigenvalue computations.* Numer. Math. vol.110,pp.313-355.2008

[12] Haider F, Brenner P, Courbet B, Croisille J P. *Parallel implementation of k-exact Finite Volume Reconstruction on Unstructured Grids.* High Order Nonlinear Numerical Methods for Evolutionary PDE's (HONOM), pp.18-22,March 2013.

[13] Lin Q, Xie H H. *New expansions of numerical eigenvalue for* $-\Delta u = \lambda \rho u$ *by linear elements on different triangular meshes.* International Journal of Informatics and systens sciences, Vol.6,pp.10-34,2010.

Non-iteration Parallel Algorithm for Frequent Pattern Discovery

Chun Liu
School of Computer Science and Technology
Wuhan University of Technology,
Wuhan, China
e-mail:liuchun_0206 @163.com

Yuqiang Li
School of Computer Science and Technology
Wuhan University of Technology,
Wuhan, China
liyuqiang@whut.edu.cn

Abstract—**For the high time overhead problems of Apriori algorithm while solving for the long length frequent patterns, using the MapReduce distributed programming ideas, the paper breaks the original idea of Aproiri which discovers the frequent item sets through gradually increasing the element numbers in the frequent item sets. It proposes a new non-iteration parallel algorithm about frequent pattern discovery, which can get arbitrary length frequent pattern at random. The experimental results show that the proposed algorithm has better time performance than such parallel algorithms which are under the ideas of traditional Apriori algorithm.**

Keywords- frequent pattern discovery; parallel algorithm; non-iteration;MapReduce

I. INTRODUCTION

MapReduce[1], BigTable[2] and GFS[3] proposed by Google have made an indelible contribution to the development of cloud computing technology. Especially the MapReduce distributed programming model provides a new parallel computing solutions for massive data sets, due to its simple and intuitive, good scalability, and easier to implement load balancing.

Another important application of Web mining is the Web usage mining. Web usage mining, also known as Web log mining, can find the user's behavior patterns to help understanding user's behaviors, improve the site's structure and provide users with a personalized service by analyzing the access logs of Web server. [4] With the advent of Web2.0, Web server log data is showing massive growth trends, and traditional frequent pattern discovery algorithm cannot satisfy the growing demand for huge amounts of data processing; so the research on parallel algorithm for frequent pattern discovery is an important direction at the age of big data.

II. FREQUENT PATTERN DISCOVERY SURVEY

A. Overview of Frequent Pattern Discovery Algorithm

In 1994, Agrawal presented the classic algorithm of frequent item sets discovery-Apriori algorithm. The core ideas are that [5]: (1) All non-empty subsets of frequent item set must be frequent; (2) Supersets of non-frequent item sets must not be frequent item sets.

Apriori algorithm completes frequent item sets discovery by gradually increasing the element numbers in item sets. It is a "bottom-up" discovery algorithm, which first finds 1-frequent item sets L_1, then 2-frequent item sets L_2.Until the element numbers in frequent item sets cannot be added any more, the algorithm stops. In the k^{th} loop, the algorithm first produces the k-length candidate item sets C_k; then counts the number of each candidate item set by traversing the transaction records database; at last, filters some candidate item sets which do not meet the conditions of the setting support threshold to get the k-frequent item sets L_k.

B. Reviews of Current Research

Classic Apriori algorithm exists two problems: (1) Apriori algorithm is an iterative search algorithm, to get L_k, it needs to scan the transaction database **D** for **k** times repeatedly, and judges whether each element in candidate item sets C_k is added into L_k; (2) In the apriori_gen (L_{k-1}) function, the self-connection operation of L_{k-1} will produce large candidate item sets, from L_{k-1} to C_k, the number of elements is in exponential growth.

In the view of the problems in classic Apriori algorithm, scholars also have conducted a lot of studies to the optimization and improvement of Apriori algorithm. The famous one is Han's frequent pattern growth (FP-Growth) algorithm [6] [7] .Using the divide and conquer strategy, through mapping the data sets to a frequent pattern tree (FP-tree), all mining process operations are carried out on the frequent pattern tree. FP-growth algorithm does not produce candidate item sets, so the efficiency is improved by an order of magnitude higher than Apriori algorithm. But it is a memory-based approach, and the FP-tree's generation is a recursive process, so the time efficiency is at the cost of space efficiency. It cannot apply to the current flood of Web data.

Along with the application of association rules mining technology in the area of Web mining [8] [9], frequent pattern discovery algorithms often face the problem of processing massive databases, therefore, parallel computing frequent item sets attracts more and more researcher's attentions. Current parallel algorithms of frequent pattern discovery are the following three main types proposed by Agrawal: CD (Count Distribution), DD (Data Distribution) and CaD (Candidate Distribution)[10].However, no matter what kind of parallel algorithm, processors synchronization and communication are the two core issues to be thought about, which make the researchers need to consider the various aspects of problems about communication, fault tolerance, load balancing, and other problems of distributed parallel systems while designing the parallel algorithms, and increases the design difficulty of parallel algorithms.

MapReduce parallel programming model for cloud computing, through putting the parallel processing, fault tolerance, load balancing, data distribution and load balance of distributed parallel system into a library, gives a good solution to the above problems. Currently many

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.73

scholars use the MapReduce parallel programming model to design the parallel algorithm of machine learning, literature 11 and 12 [11] [12] respectively proposed the parallel Apriori algorithm based on MapReduce distributed programming model. But these algorithms are still used a layer by layer iterative way to produce k-frequent item sets, in turn producing 1-frequent item sets, 2-frequent item sets,..., k-frequent item sets. Parallel operation parts are mainly used in the processing of generation L_k from C_k. Because of the using of iterative method, they still need multiple scans of the transaction database.

III. NON-ITERATION PARALLEL ALGORITHM FOR FREQUENT PATTERN DISCOVERY

A. The Idea of Non-iteration Parallel Algorithm for Frequent Pattern Discovery

Based on the above analysis, the paper proposes a non-iteration parallel algorithm of frequent pattern discovery. It uses a totally different way to find the frequent patterns. Every transaction record is regarded as a unit, and the candidate patterns are generated according to every transaction record.

The core ideas of non-iteration parallel algorithm for frequent pattern discovery are described as following:

(1) According to the rules, the whole transaction database is horizontally divided into **N** data subsets with a certain amount of transaction records, and these data subsets will be sent to the **N** computing nodes.

(2) Each node scans its local transaction data subsets to produce the local candidate item subset record<candidate pattern, local_count>. The transaction records in the local transaction data subsets are processed one by one. As for one transaction record, arbitrary length candidate patterns are generated and each given a count mark with the initial value 1, so every candidate pattern record is in the form of < candidate pattern, 1>. After all of the transaction records in its local transaction data subsets have been processed, the algorithm will accumulate the same candidate item sets and count their quantity to get the candidate item set record with the form of <candidate pattern, local_count>.

(3) The distributed system collects all candidate item subset records produced by **N** nodes, divide them into **R** different blocks and sent them to the **R** different nodes. The requirement of blocking is that the candidate item sets with the same pattern should be arranged in the same block.

(4) R different nodes separately collect their candidate item sets, count and merge the candidate item set with the same pattern to get the final candidate item set record with the form of < candidate pattern, global_count >. After that, it will compare the global_count to the min support to decide where the candidate pattern can be added into the frequent item sets. After filtering out the candidate patterns which do not meet the requirements of the min support, the frequent patterns are generated.

From the above description of non-iteration parallel algorithm of frequent pattern discovery, it can be found that the parallel technology is not only used in counting the element numbers of candidate item sets to generated frequent item sets (currently most improvement for Apriori algorithm mainly concentrated in this aspect, such as literature 11 and 12), but also applied in generating the candidate pattern of the transaction record, which has greatly improved the executive efficiency of algorithm.

There are no iterative operations in this algorithm and the operations of traversing the database can be distributed to different nodes for parallel running, which greatly increases the speed of processing. This kind of idea is not suitable for single system, because of overload large numbers of the temporary candidate item sets, but in parallel computing environments; data processing cost is far less than the cost of accessing IO. The algorithm is very suitable for parallel operation because it divides the computing task into different groups which compute their tasks respectively according to transaction records.

B. Random Length Patten Generation Algorithm

Traditional Apriori algorithm is a kind of "bottom-up" search algorithm. The long frequent patterns are derived from the short frequent patterns, and when the frequent patterns are very long in length, the algorithm's efficiency becomes very low. In order to solve this problem, the paper proposes an algorithm to generate arbitrary-length item pattern, which can be generated with arbitrary length according to user needs. It is well known, binary logic operations have simple and rapid characteristics, and the “AND” operation of two binary numbers can reflect whether the two binary numbers are equal. As for the item pattern discovery, its essence is comparing the current item pattern record to the transaction record to find whether every item in item pattern record exists in transaction record, which just can be expressed with the binary "1" and "0" characters.

Therefore, random length pattern generation algorithm maps the transaction record as a binary number, and converts the item pattern discovery processing to the binary numbers processing. The method is as following: assuming the transaction record containing N items, then the transaction record is mapped to an N bit binary number M, each bit of the binary number M represents one item in the transaction record. If the i^{th} bit in M is “1”, then it expresses that the item in such position is included in the required item pattern, while on the contrary, if the i^{th} bit in M is “0”, then it expresses that the item in such position is not included in the required item pattern. The total number of all item patterns produced by one transaction record is 2^n-1, and arbitrary an item pattern can be expressed with an integer K which is between 1 to 2^n-1. First the given integer K is converted into binary form. Then according to the natures of binary "AND" operation, the items can be extracted from the transaction record at the positions where the values are “1” according to the binary form of K. And all of the number of bits which the values are “1” in K is the length of item pattern. The algorithm is similar to memory addressing, as long as the K is given, the appropriate item pattern can be get directly, so that’s why it’s called random length item pattern generation algorithm. Algorithm is described by a concrete example as following.

Example 1: Suppose a transaction record is like this {T1, (I1,I2,I3,I4,I5)}, then the transaction record can be mapped as a 5-bit binary number, with the digits distribution being "I1,I2,I3,I4,I5". If the given item patterns number K is "11010", then the corresponding item pattern is {I1,I2,I4}.

In the course of practical application, generally pattern K will not be given straightly, but a value of length is often given to get all of the patterns collection that matched the length. For example, if the value of length is 4,then the possible values for K are "01111", "10111", "11011", "11101", "11110", respectively, and the corresponding item patterns are {I2,I3,I4,I5}, {I1,I3,I4,I5},{I1,I2,I4,I5},{I1,I2,I3,I5},{I1,I2,I3,I4}. So, there are two problems to be solved in random length pattern generation algorithm. (1) According to the given transaction record with the length of **N**, get all possible M length item pattern K set, and N≥M (2) Get the item sets from transaction according to the pattern K.

Random length item pattern generation algorithm - algorithm 1, is described as following.

Algorithm 1
(Random length pattern generation algorithm)

```
Input:Transaciton Record-R[N];
        The length of required pattern -M
Output:all M length item pattern sets-Cm
ItemSet[] Random_Mode_M(R[N], int M)
{   Int[] Ksets= Mode_Find(N, M);
    for(each K in Ksets)
     {  itemK=Mode_Change(R[N],K);
        Cm.add(itemK);   }
     return Cm;  }
```

Problem 1 is equivalent to finding the combinations of M elements in N elements, and the solution can be described as following. Define an array of N length with its subscript from 1 to N. If the value of one array element is "1", it indicates that the subscripts represent the item is selected, while "0" indicates that is not selected. First the front M elements in the array are set to 1, which illustrates that the first combination is the front M elements in the array. Then scans the "10" combination of the element values in the array from left to right, after finding the first "10" combination, exchanges it into "01" combination, at the same time, move all "1" on the left side of the combination "10" to the leftmost array. When the first "1" moves to the N-M position of the array, that is, all of the "1" have been completely moved to the far right of the array, and then get the last combination.

M length item pattern discovery algorithm - algorithm 1.1, is described as following.

Algorithm 1.1
(M length pattern discovery algorithm)

```
Input:Transaciton Record length-N; Pattern length -M
Output:the M length pattern sets-Ksets
int[] Mode_Find(int N, int M)
{   for (i = 0; i < M; i++) b[i] = 1;
    do {   K = 0;
           for ( i = 0; i < N; i++)
        {  if (b[i] == 1)
           {   S = 1; for (int j = 1; j <= i; j++) S = S * 2;
               K = K + S;  }
           result.add(K);   }
     bFound = false;
    for (i=0; i< N-1;i++)
    {   if ((b[i]==1)&&(b[i+1] ==0)&&(b[0]!=0))
        {   b[i]=0;b[i +1]=1;bFound = true; break; }
       if ((b[i]==1)&&(b[i+1]==0)&&(b[0]==0))
        {   b[i] = 0; b[i + 1] = 1; bFound = true; q = 0;
            for (int j = 0; j < i; j++)
                if (b[j] == 1) q++;
            for (int j = 0; j < i; j++)
                if (j < q) b[j] = 1; else b[j] = 0;
            break;  }  }
    }while (bFound == true);
    return result;  }
```

For problem 2, given a pattern integer K, it only needs to convert it into binary form, and then respectively makes the binary "AND" operation with the every position of the transaction record. At last, it checks the result value "1" or "0" to judge whether the item at that position in the transaction record is in the candidate set, and then gets the corresponding item pattern set according to K. The mode change algorithm-algorithm 1.2 is described as following:

Algorithm 1.2
Mode change algorithm

```
Input:Transaction record-R[N];Pattern code-K
Output:the corresponding item set –itemK
ItemSet[] Mode_Change(R[N], int K)
{   ItemSet[] itemK;
    for(i=1;i<N;i++)
    {  if(K&(1<<i))  itemK.add(R[i])  }
     return itemK;   }
```

As for a transaction record containing N items, the sum of the number of all possible length item subsets (including 1-item set, 2-item set,..., n item set) is 2^n-1. If the user wants to obtain all length item subset, is only needs to traverse K from 1 to 2^n-1, and seek out each corresponding pattern to the K. All pattern discovery algorithm-algorithm 2 is described as following.

Algorithm 2
All pattern discovery algorithm

```
Input:Transaction  record-R[N];
Output: all possible length pattern sets-Cm
ItemSet[] Random_Mode_all(R[N])
{     ItemSet[] Cm;
      for(K=1,K<(1<<N),K++)
      {     itemK=Mode_Change (R[N],K);
            Cm.add(itemK);   }
      return Cm;   }
```

C. Non-iteration Parallel Algorithm for Frequent Pattern Based on MapReduce

According to the analysis results of 3.1 and 3.2, the paper gives non-iteration parallel algorithm for random length frequent pattern based on MapReduce-algorithm 3 and non-iteration parallel algorithm for all frequent pattern based on MapReduce-algorithm 4. In Algorithm 3,the parallel process consists of 3 core components: Map function, Combine function and Reduce function.

Map function completes the scanning of local transaction data subsets. As for every record in local transaction data subsets, it generates the given M length candidate item sets with the form<key, value>, while key is the candidate pattern, and the value is the count of the candidate pattern with initial value 1. The detail design is shown in algorithm 3 Step 2.1.

Combine function completes the statistics and merge of local candidate item sets. It aggregates all the <key, value> record passed from the local Mapper procedure in accordance with the same key value, and adds up all of candidate patterns with the same key value to get the local aggregation record of the key value with the form <key, localcount>. The detail design is shown in algorithm 3 Step 2.2.

Reduce function completes statistics and merge of global candidate item sets, and generate the frequent item sets. It aggregates all the <key, localcount> records passed from Combine functions in all of the nodes in accordance with the same key value, and adds up all of candidate patterns with the same key value to get the global aggregation record of the key value with the form<key, globalcount>.Then according to the support set in advance , it filters some candidate items whose global count are less than the support, and outputs the candidate items whose global count are greater than or equal to the support. At last, the frequent patterns wanted by the user are generated. The detail design is shown in algorithm 3 Step 2.3.

Algorithm 3
Non-iteration parallel algorithm for random length frequent pattern

```
Algorithm starts.
Step1:Input frequent pattern length: M, and Support: S.
Step2: Parallel computing starts.
Step2.1:(Map)
while(lines.hasMoreElements())
{   String everyline=lines.nextToken();
    String t[]=everyline.split("\t");
    ArrayList<String>
    pbyt=nfp. Random_Mode_M (t[1],M);
    for(int i=0;i<pbyt.size();i++)
        context.write(new Text(pbyt.get(i)),
        new IntWritable(1));   }
Step2.2:(Combiner)
Iterator<IntWritable> ite= values.iterator();
int sum=0;
while(ite.hasNext()) sum += ite.next().get();
context.write(key, new IntWritable(sum));
Step2.3:(Reduce)
Iterator<IntWritable> ite= values.iterator();
int sum=0; int k=0;
while(ite.hasNext()) sum += ite.next().get();
if(sum>NoneFPConstant.tsum*NoneFPConstant.s)
    context.write(key, new Text(String.valueOf(sum)));
Parallel computing ends.
Algorithm ends.
```

In Algorithm 4， the parallel process also consists of 3 core components: Map function, Combine function and Reduce function. The combine function and Reduce function have the same design with the algorithm 3. The difference is in the Map function, in algorithm 3,it calls the **Random_Mode_M** () function to generate M length frequent patterns, while in algorithm 4 ,it calls the **Random_Mode_all()** function to generate all frequent patterns. The detail design is shown as following.

Algorithm 4
Non-iteration parallel algorithm for all frequent pattern based on MapReduce

```
Algorithm starts.
Step1: Input the support S.
Step2: Parallel computing starts.
Step2.1 :( Map)
while(lines.hasMoreElements())
{   String everyline=lines.nextToken();
    String t[]=everyline.split("\t");
    ArrayList<String>
    pbyt=nfp. Random_Mode_all (t[1]);
    for(int i=0;i<pbyt.size();i++)
        context.write(new Text(pbyt.get(i)),
        new IntWritable(1));   }
Step2.2:(Combiner):The same with algorithm3
Step2.3:(Reduce):The same with algorithm3
Parallel computing ends.
Algorithm ends.
```

IV. EXPERIMENT AND DISCUSSION

In the laboratory using 8 machines with different models and configurations (desktops and laptops), a Hadoop distributed computing environments is set up. Each computer has been installed Ubuntu 12.04 Linux operating systems, Java Development Kit JDK1.7.045 and hadoop1.0.3. Algorithm 3, algorithm 4 and the algorithm proposed by literature 11 and 12 will be carried out in such a distributed computing environment, in order to verify whether algorithm 3, 4 can effectively enhance the algorithm efficiency.

A. Experimental design

There are two experiments to verify the proposed algorithms. Experiment one is designed to compare the running time between algorithm 4 and algorithm proposed by the literature 11 and 12 to generate all the frequent pattern sets. Experiment two is designed to compare the run time between algorithm 3 and algorithm proposed by

the literature 11 and 12 to generate m length frequent pattern sets.

The test data are selected from the file SogouQ(Version 2008, size1.85G) provided by the Sogou lab. After cleaning and extracting the SogouQ file, 6 groups of transaction data are obtained, respectively containing the number of transaction records for $10,10^2,10^3,10^4,10^5,10^6$; and the average length of each transaction record is 10.

B. Experimental results

1) The efficiency verification for all frequent pattern discovery

In experiment one, the above 6 transaction datasets are respectively carried out with algorithm 4 and the algorithm proposed by literature11and12. In order to better illustrate the problem and ensure each transaction data set can generate at least 2 length frequent item sets, the support S have been set differently, the running results are shown in Table1.

TABLE I. THE RUNNING RESULT OF ALL FREQUENT PATTERN DISCOVERY

The number of transaction record	support	Max-length of frequent pattern	Algorithm 4	Algorithm proposed by literature11&12
			Run time(s)	Run time(s)
10	0.5	4-FP	7.9	38.1
10^2	0.5	3-FP	7.7	32.1
10^3	0.1	3-FP	7.9	71.1
10^4	0.2	3-FP	13.9	8178.8
10^5	0.25	2-FP	66.9	4934.9
10^6	0.3	2-FP	691.77	14712.3

Through the analysis of the results of Table 1, it is easy to obtain the following conclusions:

(1) From the results of the running time showed in Table 1, it can be found that no matter what kind of transaction data sets, and no matter what the max length of frequent patterns generated by the transaction record according the support, the performance of the algorithm 4 are much higher than the algorithm proposed by literature 11 and 12.

(2) As for algorithm 4, when the transaction dataset are not very big, no matter what length of frequent patterns, the time consumptions are roughly consistent. For the 3 groups data whose record numbers are respectively $10,10^2$ and 10^3, the running times are all nearly 7 seconds. When the data set is further increased, the running time of algorithm 4 also increases with data sets increasing.

(3) As for algorithm proposed by literature 11 and 12, it is not difficult to find that the max length of frequent pattern has a greater impact on the running time of the algorithm than the number of transaction records in datasets. For example, the running times of the transactions with records number 10 and 10^2 are respectively 38.1 seconds and 32.1 seconds. And the running times of the transactions with records number 10^4 and 10^5 are respectively 8178.8 seconds and 4934.9 seconds. Though the datasets become bigger, but on the contrary the running time become shorter. The reason is the max length of frequent pattern becomes shorter, which means the number of iterations reduce, so the running time becomes shorter. Thus, the idea of using traditional Apriori algorithm to generate frequent patterns is not suitable for distributed environment，though the parallel technology is used in the process of every scanning the database to generate L_k from C_k, but if the iteration characteristics of the algorithm does not change, the enhance of algorithm efficiency is limited.

2) The efficiency verification for arbitrary length frequent pattern discovery

In experiment 2, select the dataset with 10^3 records to be carried out with algorithm 3 and the algorithm proposed by literature11 and 12 in order to get the running time of different length frequent pattern through setting different value to S. The running results are showed in Table2.

TABLE II. ARBITRARY LENGTH FREQUENT PATTERN GENERATION RUNNING RESULT

The number of transaction record	Support	the length of frequent pattern	Algorithm 4	Algorithm proposed by literature11,12
			Run time(s)	Run time(s)
10^3	0.3	1-FP	7.897	15.113
	0.103	2-FP	7.909	64.202
	0.09	3-FP	8.019	111.844
	0.08	4-FP	8.055	802.976
	0.07	5-FP	7.986	2848.357
	0.06	6-FP	7.858	13584.389

Through the analysis of the results of Table 2, it is easy to obtain the following conclusions:

(1) When seeking the frequent patterns of arbitrary length, all of the times spent of algorithm 3 in the above 6 group datasets are almost the same. While the time spent of algorithm proposed by literature 11 and 12 subsequently increases with the frequent pattern length increasing. Therefore, using the traditional Apriori algorithm to generate the long length frequent patterns, the efficiency is very low, but the algorithm 3 gives a good solution to the problem.

V. SUMMARY

Using MapReduce distributed programming concepts; the paper restructures the calculation order of Apriori algorithm, and breaks the original idea which is through gradually increasing the elements numbers of the frequent item sets to discover the frequent item sets. It proposes a new non-iteration algorithm about frequent mode discovery, which has two features: (1) It can get the needed frequent patterns, with scanning the transaction database only once, greatly improve the processing efficiency; (2) It breaks the "bottom--up" frequent pattern discovery idea, seeks out arbitrary k-frequent pattern at

random, that is to say, calculates the k-frequent pattern directly, and solves the high time overhead problems of the Apriori algorithm and FP-growth algorithm for maximal frequent patterns

REFERENCES

[1] Jeffrey Dean,Sanjay Ghemawat. MapReduce: simplified data processing on large clusters[Z]. Comm.ACM,2008,51(1):107-113.

[2] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung.The Google file system[Z]. 19th ACM Symposium on Operating Systems Principles,2003.

[3] F Chang, J Dean, S Ghemawat, W C Hsieh, D A Wallach, M Burrows, T Chandra, A Fikes, and R.E.Gruber. Bigtable:a distributed storage system for structured data[Z]. ACM Transactions on Computer Systems,2008,26(2):1-26

[4] Nagi, M.et al. Association Rules Mining Based Approach for Web Usage Mining[C].2011 IEEE INTERNATIONAL CONFERENCE ON INFORMATION REUSE AND INTEGRATION (IRI).2011 : 166-171

[5] Bing Liu.Web Data Mining[M].Beijing:Tshinghua University Press,2009

[6] J Han, J Pei Y Yin.Mining frequent patterns without candidate generation[C]. ACM SIGMOD Record.ACM,2000,29(2)1-12

[7] J Han, M Kamber.Data Mining Concepts and Techniques[M]. Beijing:Beijing High Education Press,2001

[8] Yang Xinyue,Liu Zhen,Fu Yan.MapReduce as a programming model for association rules algorithm on hadoop[C].Proceedings of 2010 3rd International Conference on Information Sciences and Interaction Sciences.Chengdu:IEEE,2010:141-143J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[9] Lingjuan Li,Min Zhang.The Strategy of Mining Association Rule Based on Cloud Computing[C].2011 International Conference on Business Computing and Global Information.2011:476-478

[10] Li L,Zhang M.The strategy of mining association rule based on cloud computing[C]. Proceedings of 2011 International Conference on Business Computing and Global Information, Shanghai,2011:475-478

[11] Yang X, Liu Z, Fu Y .MapReduce as a programming model for association rules algorithm on Hadoop[C].Proceedings of 2010 3rd International Conference on Information Sciences and Interaction Sciences,Chengdu,2010:99-102

[12] Ning Li, Li Zeng, et al.Parallel Implementaion of Apriori Algotithm Based on MapReduce[C].the 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence,Networking and Parallel/Distributed Computing, 2012:236-241

Parallel Computing for the Radix-2 Fast Fourier Transform

Gang Xie
Institute of computer applications
China Academy of Engineering Physics
MianYang, China
e-mail: xieg@caep.cn

Yang-chun Li
Institute of computer applications
China Academy of Engineering Physics
MianYang, China

***Abstract*—The fast Fourier transform (FFT) is a speed-up technique for calculating the discrete Fourier transform (DFT), which in turn is a discrete version of the continuous Fourier transform. The Fast Fourier Transform is used in linear systems analysis, antenna studies, optics, random process modeling, probability theory, quantum physics, and boundary-value problems, and has been very successfully applied to restoration of astronomical data. This paper formulates the one dimensional and two dimensional continuous and discrete Fourier transform, especially the fast Fourier transform, considers their parallel algorithms and reports the speed up of parallel computing in both shared memory and distributed memory modes.**

Keywords-Discrete Fourier Transform, Fast Fourier Transform, parallel computing, Message Passing Interface, OpenMP;

I. INTRODUCTION

Fourier analysis converts time (or space) to frequency and vice versa. The discrete Fourier transform (DFT) is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields but computing it directly from the definition is often too slow to be practical. A fast Fourier transform (FFT) is an algorithm to compute the discrete Fourier transform (DFT) and its inverse more quickly by factorizing the DFT matrix into a product of sparse(mostly zero) factors [1]: computing the DFT of N points in the naive way, using the definition, takes O(N2) arithmetical operations, while a FFT can compute the same DFT in only O(N log N) operations. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to N / log(N). This huge improvement made the calculation of the DFT practical. As a result, fast Fourier transforms are widely used for many applications in engineering, science, and mathematics, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers. The basic ideas were popularized in 1965, but some FFTs had been previously known as early as 1805. Fast Fourier transforms have been described as "the most important numerical algorithm[s] of our lifetime" [2-10].

Multiprocessors fall into two general categories: shared-memory multiprocessors and distributed-memory multiprocessors. As their names imply, they are distinguished by whether each processor can directly access the entire memory available, or whether the memory is partitioned into portions which are private to each processor. For shared-memory architecture, the main challenge in parallelizing a sequential algorithm is to subdivide the computation among the processor in such a way that the load is balanced, and memory conflicts are kept low. For FFT algorithms, this is a relatively simple task. In terms of the design of algorithms, distributed-memory machines impose the additional burden of requiring that the data, as well as the computation, be portioned. In addition to identifying parallelism in the computation, and assigning computational tasks to individual processors, the data associated with the computation must be distributed among the processors, and communicated among them as necessary. The challenge is to do this in such a way that each processor has the data it needs in its local memory at the time that it needs it, and the amount of communication required among the processors during the computation is kept acceptably low.

Due to the important role of Fourier transform in scientific and technical computations, there has been great interest in implementing FFT on parallel computers and on studying its performance. Swarztrauber [11] describes many implementations of the FFT algorithm on vector and parallel computers. Cvetanovic [12] and Norton and Silberger [13] give a comprehensive performance analysis of the FFT algorithm on pseudo-shared-memory architectures such as the IBM RP-3. They consider various partitionings of data among memory blocks and, in each case, obtain expressions for communication overhead and speedup in terms of problem size, number of processes, memory latency, CPU speed, and speed of communication. Aggarwai, Chandra, and Snir [14] analyze the performance of FFT and other algorithm on LPRAM (Local-memory Parallel Random Access Machine) – a model for parallel computation. This model differs from the standard PRAM (Parallel Random Access Machine) model in that remote accesses are more expensive than local accesses in an LPRAM. Parallel FFT algorithm and their implementation and experimental evaluation on various architectures have been pursued by many other researchers [15-18].

The most notable work on parallel FFT was done by an Intel team [19]. What they figured out was a new way of

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.29

managing the decomposition by convoluting the sample to increase the number of sample points, then sparsely processing it conventionally, and combining results via a single all-to-all transpose. The result is a family of O(N log N) DFT factorizations that are twice as fast as well-known competing algorithms. While the Intel team focused on the 1-D FFT, we are now mainly addressing the problem of how to efficiently run the 2-D FFT in parallel in both shared and distributed memory modes.

II. DFT

For a continuous function of one variable f(t), the Fourier Transform F(f) will be defined as:

$$F(f) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi i f t} dt$$

and the inverse transform as

$$f(f) = \int_{-\infty}^{+\infty} F(f) e^{2\pi i f t} df$$

where i is the square root of -1 and e denotes the natural exponent

$$e^{i\theta} = \cos(\theta) + i\sin(\theta)$$

The corresponding discrete Fourier transform is defined as:

$$x_k = \sum_{n=0}^{N-1} x_n W_N^{nk} , \qquad k=0, 1,\ldots, N-1,$$

where

$$W_N = e^{-2\pi i/N} .$$

Of course although the functions here are described as complex series, real valued series can be represented by setting the imaginary part to 0. In general, the transform into the frequency domain will be a complex valued function.

While the DFT transform above can be applied to any complex valued series, in practice for large series it can take considerable time to compute, the time taken being proportional to the square of the number on points in the series. A much faster algorithm has been developed by Cooley and Tukey around 1965 called the FFT (Fast Fourier Transform). The only requirement of the the most popular implementation of this algorithm (Radix-2 Cooley-Tukey) is that the number of points in the series be a power of 2. The computing time for the radix-2 FFT is proportional to

$$N \log 2^N$$

So for example a transform on 1024 points using the DFT takes about 100 times longer than using the FFT, a significant speed increase. Note that in reality comparing speeds of various FFT routines is problematic, many of the reported timings have more to do with specific coding methods and their relationship to the hardware and operating system.

The N point DFT can be reduced to two N/2 point DFTs, hence a recursive algorithm for the DFT can be derived. Let

K = 2r, $0 < r < N/2 - 1$,

m = n - N/2,

then

$$\begin{aligned} x_{2r} &= \sum_{n=0}^{N-1} x_n W_N^{2nr} \\ &= \sum_{n=0}^{N/2-1} x_n W_{N/2}^{nr} + \sum_{n=N/2}^{N-1} x_n W_{N/2}^{rn} \\ &= \sum_{n=0}^{N/2-1} x_n W_{N/2}^{nr} + \sum_{m=0}^{N/2-1} x_{m+N/2} W_{N/2}^{r(m+N/2)} \\ &= \sum_{n=0}^{N/2-1} (x_n + x_{n+N/2}) W_{N/2}^{nr} \end{aligned}$$

In the same way, for K = 2r+1， we get

$$\begin{aligned} x_{2r+1} &= \sum_{n=0}^{N-1} x_n W_N^{(2r+1)n} \\ &= \sum_{n=0}^{N/2-1} x_n W_N^n W_{N/2}^{nr} + \sum_{n=0}^{N/2-1} x_{n+N/2} W_N^{(2r+1)(n+N/2)} \\ &= \sum_{n=0}^{N/2-1} (x_n - x_{n+N/2}) W_N^n W_{N/2}^{nr} \end{aligned}$$

Set

$$e_n = x_n + x_{n+N/2} , \qquad n = 0,1,...,N/2-1 ,$$

$$f_n = x_n - x_{n+N/2} , \qquad n = 0,1,...,N/2-1 ,$$

then

$$x_{2r} = \sum_{n=0}^{N/2-1} e_n W_{N/2}^{nr}$$

$$x_{2r+1} = \sum_{n=0}^{N/2-1} f_n W_N^n W_{N/2}^{nr}$$

The above two formulas indicate that the N point DFT can be reduced to the N/2 point DFT, while the N/2 point DFT can again be reduced to the N/4 point DFT, and so on. This gives a recursive algorithm for DFT, that's the famous fast Fourier transform (FFT) algorithm.

III. FFT

The above recursive algorithm for FFT can be transformed into an iterative format, as follows

n	number of elements in the vector "a".
a[0,1, … , n-1]	the vector to be transformed.
y[0,1, … , n-1]	transform result.

y ← bit_reverse(a)
for j ← 1 to log n
  $d = 2^j$
  $\omega_d \leftarrow e^{2\pi i/d}$
  $\omega = 1$
  for k← 0 to $d/2-1$
    for m ← k to n-1 step d
      $\omega \times y[m + d/2] \rightarrow t$
      x ← y[m]

```
            y[m] ← x + t
            y[m+d/2] ← x - t
          Endfor
          ω = ω × ω_d
      Endfor
Endfor
Return y
```

IV. FFT UNWRAPPED INTO TWOFOLD LOOPS

When considering parallel algorithms of FFT, to break the data dependence, we sometimes need to increase the workload and memory use (to introduce new variables) on purpose. To facilitate parallel computing we unwrap the above FFT in threefold loops into the following form of twofold loops:

```
Iterate FFT(a,n)
Parameters:
n                 number of elements in the vector "a'.
a[0,1, … , n-1]   the vector to be transformed.
y[0,1, … , n-1]   transform results.
y ← bit_reverse(a)
for j ← 1 to log n
    d = 2^j
    ω ← e^(2πi/d)
    ω_0 = 1
    For k ← 1 to d/2 − 1
      ω_k = ω_(k−1) × ω
    Endfor
    for i ← 0 to n-1
      r ← i mod d
      if r < d/2 then
        ω_r × y[i + d/2] → t
        x ← y[i]
        y[i] ← x + t
      else
        ω_(r−d/2) × y[i] → t
        y[i − d/2] → x
        y[i] ← x - t
      endif
    endfor
Endfor
Return y
```

V. PARALLEL ALGORITHM FOR FFT

In the above algorithm, the outer loop represents an iterative process, hence not suitable for parallelization. While the inner loop "for i ← 0 to n-1" actually updates the n components of the vector one by one independently, by using the whole vector obtained from the previous iteration step, hence can safely be run in parallel. In a distributed memory mode, suppose we have p processes, then each process can be responsible for computing number $n/p = g$ of components of the vector. Suppose p is also the power of 2. When $g \geq d$, no communication is needed. Contrarily, let $d/g = m$ ($m \geq 2$), then the processes p and $p + m/2$ need to exchange computing results before continuing to update the components of the vector.

VI. TWO DIMENSIONAL DFT

For a continuous function of two variables f(x,y), the Fourier Transform F(u,v) will be defined as:

$$F(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-2\pi i(ux+vy)}dxdy$$

and the inverse transform as

$$f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)e^{2\pi i(ux+vy)}dudv$$

The corresponding two dimensional discrete Fourier transform is defined as:

$$x(k,l) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x(n,m)W_N^{nk}W_M^{ml}$$

where

$k = 0,1,...,N-1$

$l = 0,1,...,M-1$

In the most general situation a 2 dimensional transform takes a complex array. The most common application is for image processing where each value in the array represents to a pixel, therefore the real value is the pixel value and the imaginary value is 0.

2 dimensional Fourier transforms simply involve a number of 1 dimensional Fourier transforms. More precisely, a 2 dimensional transform is achieved by first transforming each row, replacing each row with its transform and then transforming each column, replacing each column with its transform. Thus a 2D transform of a 1K by 1K image requires 2K 1D transforms. This follows directly from the definition of the Fourier transform of a continuous variable or the discrete Fourier transform of a discrete system.

The transform pairs that are commonly derived in 1 dimension can also be derived for the 2 dimensional situation. The 2 dimensional pairs can often be derived simply by considering the procedure of applying transforms to the rows and then the columns of the 2 dimensional array.

To prove, let's set

$$y(n,1) = \sum_{m=0}^{M-1} x(n,m)W_M^{ml}, \quad l = 0,1,...,M-1,$$

then

$$x(k,1) = \sum_{n=0}^{N-1} y(n,l)W_N^{nk}, \quad k = 0,1,...,N-1.$$

So the two dimensional DFT can be reduced to the one dimensional DFT. Namely, the two dimensional DFT can be done by first doing the one dimensional DFT by row and then doing the one dimensional DFT by column.

VII. PARALLEL ALGORITHM FOR 2-D FFT

As has been said, the two dimensional DFT can be reduced to a series of one dimensional DFT, which are independent of each other. Hence the two dimensional DFT can safely be run in parallel. In a distributed memory mode, if we are to do two dimensional DFT for a $N \times M$ matrix and we have got p CPUs to use, we can divide the matrix into a $P \times P$ block matrix, where each

block is a $(N/P)\times(M/P)$ submatrix. Firstly, process 0 sends the i'th row of the block matrix to the i'th process (0<i<p). Secondly, each process does FFT by row for the $(N/P)\times M$ submatrix it obtained from the process 0. Thirdly, the process j (0<=j<p) sends the i'th (0<=i<p) block of the row of the block matrix it responsible for to the process i. Finally, the process j (0<=j<p) does FFT by column for the j'th column of the block matrix and sends the results back to process 0 who assemblies them all into the whole result matrix.

VIII. SPEED UP

TABLE1. SPEED UP OF SHARED MEMORY PARALLEL COMPUTING (OPENMP)

CPU	time (seconds)	speed up	efficiency
1	307.834	1	100%
2	167.213	1.84	92%
4	84.9262	3.6	90%
8	49.1987	6.3	79%
16	32.5234	9.5	60%
32	35.2783	8.7	29%
64	23.1415	13.3	21%
128	20.3856	15	12%

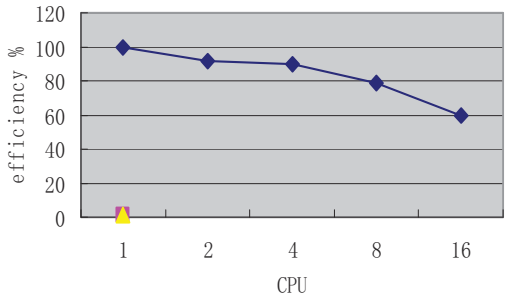


Figure 1. CPUs to efficiency

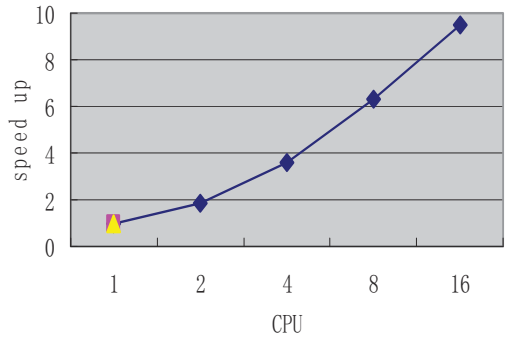


Figure 2. CPUs to speed up

TABLE2. SPEED UP OF DISTRIBUTED MEMORY PARALLEL COMPUTING (MPI)

CPU	time(seconds)	speed up	efficiency
1	331.341	1	100%
2	173.27	1.9	95%
4	94.5375	3.5	88%
8	50.6769	6.5	82%
16	31.1929	10	66%
32	23.6373	14	44%
64	16.57	20	31%
128	12,27	27	21%

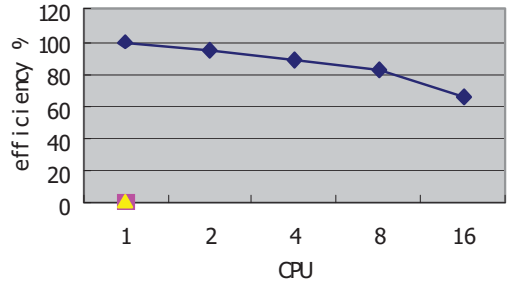


Figure 3. CPUs to efficiency

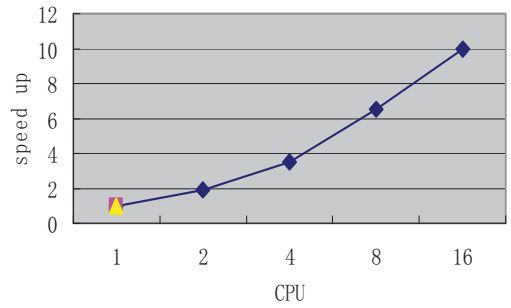


Figure 4. CPUs to speed up

IX. CONCLUSION

In principle, the FFT can be thought as a data concentrated application with a large data volume and a relatively small computation volume, so the proportion of time consumed in the memory access and communication is relatively high. This can badly restrict the efficiency of the parallel processing, although the parallelism of the problem is perfect. Yet, the above tables show the speed up is considerable.

REFERENCES

[1] Charles Van Loan, Computational Frameworks for the Fast Fourier Transform, SIAM, 1992.

[2] Strang, Gilbert, "Wavelets", American Scientist 82 (3): 253. Retrieved 8 October 2013.

[3] A. Edelman, P. McCorquodale, and S. Toledo, The Future Fast Fourier Transform?, SIAM J. Sci. Computing 20: 1094–1114, 1999.

[4] Steve Haynal and Heidi Haynal, "Generating and Searching Families of FFT Algorithms", Journal on Satisfiability, Boolean Modeling and Computation vol. 7, pp. 145–187 (2011).

[5] T. Lundy and J. Van Buskirk, "A new matrix approach to real FFTs and convolutions of length 2k,"Computing 80 (1): 23-45, 2007.

[6] Kent, Ray D. and Read, Charles. Acoustic Analysis of Speech. ISBN 0-7693-0112-6. Cites Strang, G. (1994)/May–June). Wavelets. American Scientist, 82, 250-255, 2002.

[7] Rokhlin, Vladimir; Tygert, Mark, "Fast algorithms for spherical harmonic expansions". SIAM J. Sci. Computing 27 (6): 1903–1928, 2006.

[8] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price, "Simple and Practical Algorithm for Sparse Fourier Transform" (PDF), ACM-SIAM Symposium On Discrete Algorithms (SODA), Kyoto, January 2012.

[9] D. F. Elliott, & K. R. Rao, Fast transforms: Algorithms, analyses, applications. New York: Academic Press, 1982.

[10] Q. F. Stout and B. A. Wagar. Passing messages in link-bound hypercubes. In M. T. Heath, editor, Hypercube Multiprocessors, 1987,251-257. SIAM, Philadelphia, PA, 1987.

[11] P. N. Swarztrauber. Multiprocessor FFTs. Parallel computing, 5:197-210,1987.

[12] Z. Cvetanovic. Performance analysis of the FFT algorithm on a shared memory architecture. IBM Journal of Research and Development, 21(4):435-451,1987.

[13] A. Norton and A. J. Silberger. Parallelization and Performance analysis of the FFT algorithm for shared memory architectures. IEEE Transactions on Computers, C-36(5):581-591,1987.

[14] A. Aggarwai, A. K. Chandra, and M. Snir. Communication complexity of PRAMs. Technical Report RC 14998(64644), IBM T. J. Watson Research Center, Yorktown Hights, NY, 1989.

[15] D. H. Bailey. FFTs in external or hierarchical memory. Journal of Supercomputing , 4:23-35, 1990.

[16] R. Blumofe, C. Joerg, B. Kuszmaul, C. Leiserson, K. Randall, and Y. Zhou. Cilk: An efficient multithreaded runtime system. In Proceedings of the 5th Symposium on Principles and Practice of Parallel Programming, 1995.

[17] A. Gupta and V. Kumar. Performance properties of large scale parallel systems. Journal of Parallel and Distributed Computing, 19:234-244, 1993.

[18] A. Gupta and V. Kumar. The scalability of FFT on parallel computers. IEEE Transactions on Parallel and Distributed Systems, 4(8):922-932, 1993

[19] Ping Tak, Peter Tang, Jongsoo Park, Daehyun Kim and Vladimir Petrov. A Framework for Low-Communication 1-D FFT, SC'12, IEEE Press, 2012.

Multi-index Evaluation Algorithm Based on Locally Linear Embedding for the Node importance in Complex Networks

Fang Hu[1,2], Yuhua Liu[1], Jianzhi Jin[1]
1 Department of Computer Science, Central China Normal University, Wuhan, 430079, China
2. Information Engineering Institute, Hubei University of Chinese Medicine, Wuhan 430065, China
Email:naomifang@126.com, yhliu@mail.ccnu.edu.cn, jz.jin@qq.com
Correspondence authors: Yuhua Liu

***Abstract*—Evaluation of node importance in complex network is significant, so it is important to seek and protect important node, which is ensure the security and stability of the entire network. At present, most algorithms of important node evaluation are according to the single-index, which can't reflect the whole condition of complex network. In this paper, synthesizing multi-index factors of node importance, including degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, mutual-information, etc., a new multi-index evaluation algorithm based on Locally Linear Embedding (LLE) for the node importance in complex network is proposed. In order to verify the validity of this algorithm, a series of simulation experiments have been done. Through comprehensive analysis, the simulation results represent that the new algorithm is rational, effective, integral and accurate.**

***Index Terms*—Complex Network, Node importance, Multi-index evaluation, Locally Linear Embedding**

I. Introduction

During the process of various fundamental researches on complex networks, it has great practical value to evaluate the node importance and to identify important nodes in complex networks. Node importance is a basic measure in characterizing the structure and dynamics of complex networks [1-2]. Such projects as identifying important nodes, and improving the reliability of complex networks by focusing on protecting these important nodes have been a critical research task in complex networks.

Various centrality measures have been proposed over the years to rank the nodes of a graph according to their topological importance [3]. The single-index analysis method is evaluating node importance in complex networks by analyzing characteristic indexes of nodes, such as degree, betweenness, closeness, eigenvector, mutual-information etc. Degree algorithm is simple and intuitive, and convenient in calculation, but it is inaccurate [4]. Betweenness algorithm evaluates the node importance from network traffic, which reflects the dynamic characteristic of the network, but its computational complexity is too high [5]. Closeness algorithm evaluates the centrality of nodes by considering the entire network topology, but it is not suitable for regular graph and random networks [6]. Eigenvector algorithm evaluates node importance with considering the importance of the neighboring nodes, but it just linearly superposes the parameters of each node, and overly simplifies the actual situation [7]. Mutual-information method evaluates node importance with revealing the characteristics of network topology structure, but the calculation method is too simple and can't apply to all of the networks [8].

These centrality measures and their applications have been proposed for identifying important nodes. However, most of them focused on only one centrality measure. As mentioned above, there are incomplete and limited [9]. So, researchers propose many evaluation algorithms based on multi-index to evaluate important nodes in complex networks. For example, Yu et al. [10] propose a multi-attribute decision-making method, in which, each node is regarded as a solution, and each importance evaluation criterion as one solution's attribute. Du et al. [11] propose a new evaluation method based on technique for order performance by similarity to ideal solution (TOPSIS) approach. This algorithm is utilized to aggregate the different centrality measure Jin et al. [12] propose a new multi-index evaluation algorithm based on principal component analysis. This algorithm no parameter restrictions to represent the features of the data, and synthesize the topological characteristics of the network.

Therefore, inspired by multi-index analysis algorithms, in this paper, after synthesizing multi-index factors of node importance, including degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, mutual-information centrality, etc., a new multi-index evaluation algorithm based on LLE for the node importance in complex network is proposed. In order to verify the validity of this algorithm, a series of simulation experiments have been done. Through simulation and comprehensive analysis, the simulation results show that the new algorithm is effective, and can improve the computational accuracy.

II. LLE Introduction and Index Concept Definitions

A. LLE Introduction

Scientists interested in exploratory analysis or visualization of multivariate data face a similar problem in dimensionality reduction [13]. Locally linear embedding

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.30

(LLE) is a nonlinear dimensionality reduction technique recently proposed by Roweis and Saul [14]. LLE maps its inputs into a single global coordinate system of lower dimensionality, and its optimizations do not involve local minima. In other words, LLE is a manifold learning technique which aims at mapping high-dimensional data into a low-dimensional manifold space by preserving neighbors [15]. By exploiting the local symmetries of linear reconstructions, LLE is able to learn the global structure of nonlinear manifolds, such as those generated by facial recognition [16],image-processing [17], fault diagnosis [18] and so on.

The problem involves mapping high-dimensional inputs into a low-dimensional "description" space with as many coordinates as observe modes of variability. LLE algorithm eliminates the need to estimate pairwise distances between widely separated data points, and recovers global nonlinear structure from locally linear fits. A example of LLE is as following.

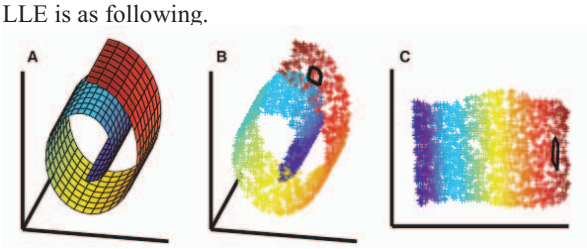


Fig. 1. Example of Locally Linear Embedding

As shown in figure 1, The problem of nonlinear dimensionality reduction, as illustrated for three-dimensional data (B) sampled from a three-dimensional manifold (A). An unsupervised learning algorithm must discover the global internal coordinates of the manifold without signals that explicitly indicate how the data should be embedded in two dimensions. The color coding illustrates the eighborhood-preserving mapping discovered by LLE; black outlines in (B) and (C) show the neighborhood of a single point [14].

B. Index Concept Definitions

A large number of centrality measures have been proposed to identify important nodes within a graph and a complex network. Typical examples are degree centrality [6], closeness centrality [6], betweenness centrality [6], eigenvector centrality [19], and Mutual-information [8], etc. A larger researches and experiments prove that these indexes can efficiently reflect node importance in different perspectives. Therefore, these indexes are chosen as the parameters for multi-index evaluation in this paper. The degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, and mutual-information are defined as follows.

1 Degree *Centrality*

Definition 1 The degree centrality of node v, denoted as $C_D(v)$, is defined as

$$C_D(v) = \frac{\deg(v)}{N-1} \tag{1}$$

where $\deg(v)$ is the degree of node v, which is defined as the number of ties that node v has. This value $N-1$ is used to normalize the degree centrality value, N is the total number of nodes.

2 Betweenness *Centrality*

Definition 2 The betweenness centrality $C_B(v)$ of node v, denoted as $C_B(v)$, is defined as

$$C_B(v) = \frac{\sum_{s \neq v \neq t \in V} \frac{\delta_{st}(v)}{\delta_{st}}}{(N-1)(N-2)/2} \tag{2}$$

where δ_{st} is the number of the shortest paths between node s and node t, and $\delta_{st}(v)$ is the number of those paths that go through node v. This value $(N-1)(N-2)/2$ is used to normalize the betweenness centrality value, N is the total number of nodes.

3 Closeness *Centrality*

Definition 3 The closeness centrality of node v, denoted as $C_c(v)$, is defined as

$$C_c(v) = \frac{\sum_{t \in V \setminus v} d_G(v,t)}{N-1} \tag{3}$$

where $d_G(v,t)$ is the shortest path between node v and node t. This value $N-1$ is used to normalize the closeness centrality value, N is the total number of nodes.

4 Eigenvector *Centrality*

Definition 4 For node v, the eigenvector centrality score is proportional to the sum of the scores of all nodes which are connected to it, i.e.

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{N} A_{i,j} x_j \tag{4}$$

where x_i denotes the score of the node i, A is the adjacency matrix of the network, N is the total number of nodes, and λ is a constant. In vector notation, this can be rewritten as $X = \frac{1}{\lambda} AX$, or as the eigenvector equation $AX = \lambda X$.

5 Mutual-*information*

Definition 5 The mutual-information of node v is the sum of the mutual information between node v and other nodes which are connected to it, i.e.

$$C_I(v) = \sum_{j=0}^{N} (\log \deg(v) - \log \deg(j)) \tag{5}$$

where $\deg(j)$ is the degree of node j. Mutual-information uses information theory to assess the importance of nodes $C_I(v)$ which represents the amount of information each node contains.

III. Multi-index Evaluation Algorithm Based on LLE for the Node Importance

A. Algorithm thought

Recently, most algorithms evaluating node importance are according to single-index in complex network. Because single-index is one-sided and unstable, it is difficult to reflect the whole situation in complex network. In this paper, synthesizing multi-index factors of node importance, including degree centrality, betweenness centrality, closeness centrality, eigenvector centrality, mutual-information, etc., and applying idea the of multi-objective optimization, a new multi-index evaluation algorithm based on LLE for the node importance in complex network is proposed. In this algorithm, high-dimensional data is mapped into a low-dimensional space by preserving neighbors.

B. Steps of the Algorithm

The principle of LLE algorithm is that it is given a set of points $X = \{x_1, x_2, \ldots\ldots, x_n\} \in R^{D \times n}$ denote n points in a high D dimensional space, the LLE will find a new set of coordinates in a low D dimensional space, satisfying the same neighbor-relations as the original points'. After some improvements, the multi-index algorithm based on LLE can be summarized as follows,

Step 1 According to the index definitions above, calculate the value of each index vector in complex networks and construct matrix X, i.e.,

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

where m is the number of nodes, n is the number of evaluation indexes in complex networks.

Step 2: For each data point $x_{ij}(i = 1, 2, \ldots\ldots, m; j = 1, 2, \ldots\ldots, n)$, find its k nearest neighbors by using Euclidean distance, which is the length of the line segment connecting two points.

Step 3: Compute the weights w_{ij} that best linearly reconstruct each data x_{ij} from its k neighbors $x_{ij}^{\,l}(l = 1, 2, \ldots\ldots, k)$, minimizing the following cost function,

$$e(W) = \sum_{i=1}^{m} \left| \sum_{j=1}^{n} x_{ij} - \sum_{l=1}^{k} w_{ij}^{\,l} x_{ij}^{\,l} \right|^2 \tag{6}$$

under the constraints that each vector of weights w_{ij} sums to unity.

Step 4: Construct the optimal low dimensional embedding S for X, in which the local linear geometry of the high-dimensional data is best preserved by the reconstruction weights W of the data X in R^D. This step is accomplished by minimizing the following cost function for the fixed weights W,

$$\phi(S) = \sum_{i=1}^{m} \left| \sum_{j=1}^{n} s_{ij} - \sum_{l=1}^{k} w_{ij}^{\,l} s_{ij}^{\,l} \right|^2 \tag{7}$$

subject to the following constraints,

$$\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} s_{ij}^T s_{ij} = I, \sum_{i=1}^{m} \sum_{j=1}^{n} s_{ij} = 0 \tag{8}$$

To optimize the embedding error, we can rewrite it in the following quadratic form,

$$\phi(S) = \sum_{i=1}^{m} \sum_{j=1}^{k} m_{ij} s_{ij} s_{ij}^T \tag{9}$$

based on inner products of the outputs s. The square $m \times m$ matrix M a sparse, symmetric and semi-positive matrix, is given by,

$$m_{ij}^{\,l} = \delta_{ij}^{\,l} - w_{ij}^{\,l} - w_{ji}^{\,l} + \sum_{l=1}^{k} w_{ij}^{\,l} w_{ji}^{\,l} \tag{10}$$

for which $\delta_{ij}^{\,l}$ is an element of the identity matrix. The constrained minimization problem can be converted to solving and eigen-decomposition of the matrix M as calculated below,

$$M = (I - W)(I - W)^T \tag{11}$$

for which the eigenvectors associated with the bottom d nonzero eigenvalues constitute the final embedding outputs S.

C. Computable Complexity

The computable complexity of calculating degree centrality and mutual-information is $O(N)$, where N is the number of nodes in complex network; calculating eigenvector centrality is $O(N^2)$; calculating betweenness centrality and closeness is $O(N^3)$. In LLE algorithm, choosing neighbors needs $O(mN^2)$, where m is the dimension of high-dimensional sample; calculating the reconstruction weights is $O((m+k)k^2N)$, where k is the number of neighbors; getting d-dimensional embedding is $O(dN^2)$; so the computable complexity of LLE algorithm is $O((m+k)k^2N)$. Finally, based on the analysis above, the computable complexity is $O(N^3)$.

IV. Simulation and Analysis

A. Simulation Example

In this paper, Windows 7 and MATLAB (R2010b) is the simulation software environment for this new algorithm.

1) ARPA NETWORK

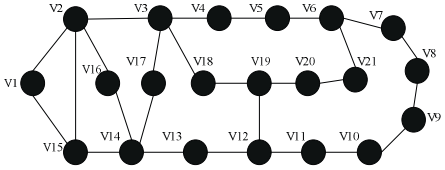


Fig. 2. ARPA's topology

In this paper, ARPA network topology (as shown in figure 2) is used to analyze and illustrate the multi-index evaluation algorithm based on LLE. ARPA is the trunk topology in North America, composed of 21 nodes and 26 edges. Although an arbitrary node is removed from ARPA, the network is still connected.

2) KARATE CLUB

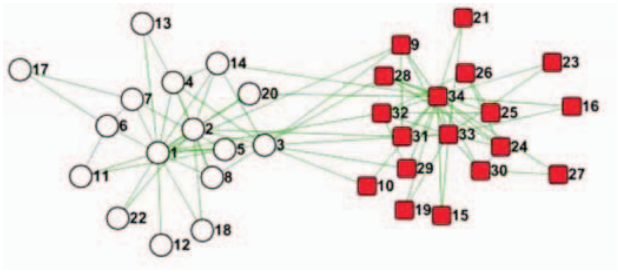


Fig. 3. Karate's topology

Wayne Zachary observed social interactions between the members of a karate club at an American university. After a long study, he built the network consisting of 34 members of the karate club as nodes and 78 edges representing friendship between the members of the club (as shown in figure 3) [20].

B. Analysis

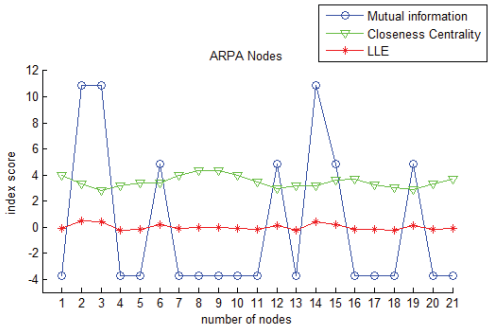


Fig. 4(a) Nodes' importance contrast line graph according to mutual-information, closeness, and LLE in ARPA

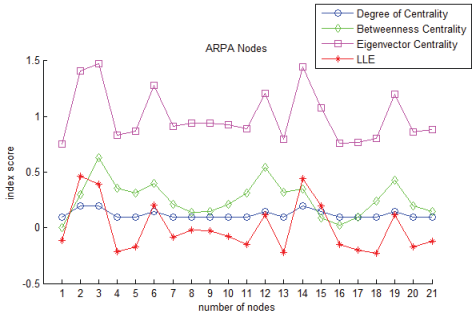


Fig. 4(b) Nodes' importance contrast line graph according to degree, betweenness, PageRank, and LLE in ARPA

Figure 4 show that, in this paper, after synthesizing the five single-indexes described above, the final conclusion is as follows.

The result of this proposed algorithm is that the most important nodes are v_2, v_3 and v_{14}, which is identical to the results of many single-index algorithms, such as degree centrality, eigenvector centrality and mutual-information. Betweenness centrality and closeness centrality can identify correctly that the most important node is v_3. The secondary important nodes are v_6, v_{12}, v_{15} and v_{19}, which is identical to the results of eigenvector centrality, degree centrality and mutual-information. The arrangement result of betweenness centrality represents that many nodes communicate with others via v_{12}; closeness centrality represents that v_{19} is closer to the center of network.

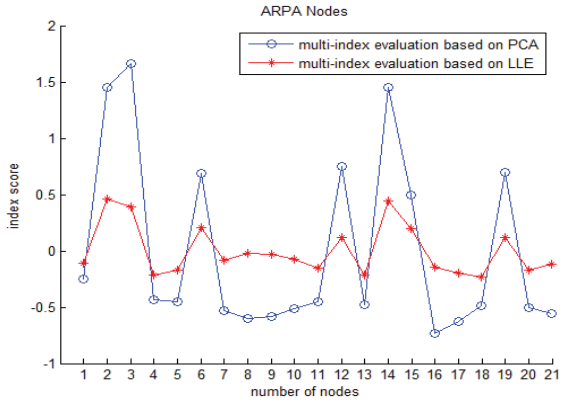


Fig. 5 Nodes' importance contrast line graph according to PCA and LLE in ARPA

Figure 5 shows the index score of each node in ARPA network comparison of multi-index evaluation based on LLE and PCA [12]. The evaluation results of these two methods are basically the same, and the former is more accurate.

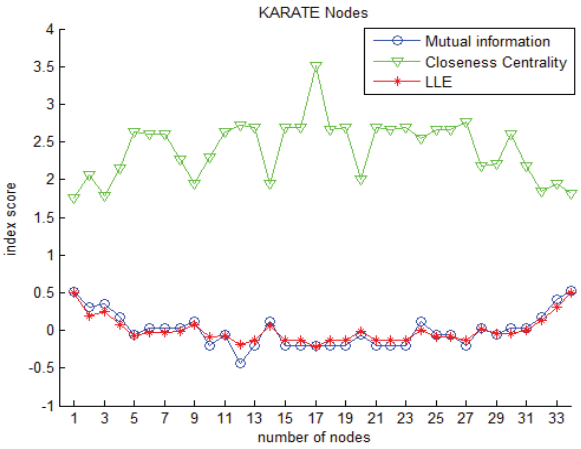


Fig. 6(a) Nodes' importance contrast line graph according to mutual-information, closeness, and LLE in Karate club

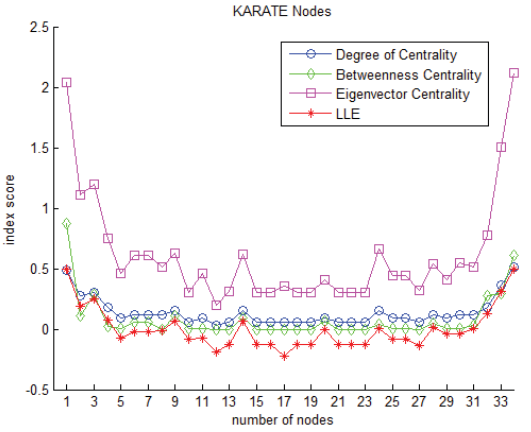


Fig. 6(b) Nodes' importance contrast line graph according to degree, betweenness, PageRank, and LLE in Karate club

Figure 6 show that, in this paper, after synthesizing the five single-indexes described above, the final conclusion is as follows.

The result of this proposed algorithm identifies the most five important nodes are v_1, v_2, v_3, v_{33} and v_{34}, in which the v_1 and v_{34} are generally considered as the most two important nodes and usually as the core nodes to community detecting communities in complex networks. This result of this new algorithm is identical to the results of the results of the degree centrality, betweenness centrality, eigenvector centrality and mutual-information. The closeness centrality can not identical the most two important nodes accurately, because the node v_3 is a wrong identification.

Through the simulation and analysis above, it represents that the result acquired from this proposed algorithm is basically identical with, or to some extent, is more careful and reasonable than other single-index algorithms and multi-index evaluation algorithm based on PCA. reasonably.

V. Conclusion

In this paper, a new multi-index evaluation algorithm based on Locally Linear Embedding for node important in complex networks is proposed, which is simple and effective, and synthesizes the statistic characteristics of nodes in complex network. This proposed algorithm maps high-dimensional data into a low-dimensional space by preserving neighbors. By example analysis and simulation experiment, it shows that this algorithm can effectively reflect the differences of node importance, accurately and efficiently find the important node in complex network. This method can be extended to directed weighted networks in future work.

References

[1] D. Chen, L. Lü, M.S. Shang, Y.C. Zhang, T. Zhou, "Identifying influential nodes in complex networks", Physica A, vol. 391, April 2012, pp. 1777–1787.

[2] J.G Liu., Z.M. Ren, Q. Guo, "Ranking the spreading influence in complex networks", Physica A, vol. 392, Sep. 2013, pp. 4154–4159.

[3] V. Nicosia, R. Criado, M. Romance, G. Russo, V. Latora, "Controlling centrality in complex networks", Scientific Reports, vol. 2, Sep. 2011, pp. 218–223.

[4] D. S. Callaway, M. E. J Newman, S. H. Strogatz, "Network robust-ness and fragility: percolation on Randon graphs", Phys. Rev. Lett., vol. 85, Dec. 2000, pp. 5468-5471.

[5] M.E.J. Newman, "A measure of betweenness centrality based on random walks", Social Networks, vol. 27, Jan. 2005, pp. 39-54.

[6] L.C. Freeman, "Centrality in social networks conceptual clarification", Social Networks, vol. 1, Jan. 1979, pp. 215–239.

[7] R. Poulin, M. C. Boily, B. R. Masse, "Dynamical systems to define centrality in social networks", Social Networks, vol. 22, July 2000, pp. 187-220.

[8] Y. H. Liu, J. Z. Jin, Y. Zhang and C. Xu, "A new clustering algorithm based on data field in complex networks", Journal of Supercomputing, vol. 67, March 2014, pp. 723-737.

[9] T. Opsahl, F. Agneessens, J. Skvoretz, "Node centrality in weighted networks: generalizing degree and shortest paths", Social Networks, vol. 32, July 2010, pp. 245–251.

[10] H. Yu, Z. Liu, Y. J. Li, "Key nodes in complex networks identified by multi-attribute decision-making method", Acta Phys. Sin., vol. 62, Jan. 2013, pp. 1-9.

[11] Y.X. Du, C. Gao, Y. Hu, S. Mahadevan, Y. Deng, "A new method of identifying influential nodes in complex networks based on TOPSIS", Physical A, vol. 399, April 2014, pp. 57-69.

[12] J. Z. Jin, K. H. Xu, N. X. Xiong and Y. H. Liu, "Multi-index evaluation algorithm based on principal component analysis for node importance in complex networks", IET networks, vol. 1, Sep. 2012, pp. 108-115.

[13] A.B. Buja, D.F. Swayne, M.L. Littman, N. Dean, H. Hofmann, L. Chen, "Data Visualization with Multidimensional Scaling", Journal of Computational and Graphical Statistics, vol. 17, 2008, pp. 444-472.

[14] S. Roweis and L. Saul, "Nonliear dimensionality reduction by locally linear embedding", Science, vol. 290, Dec. 2000, pp. 2323-2326.

[15] N. Qi, Z.Y. Zhang, Y. H. Xiang, P. B. Harrington, "ocally linear embedding method for dimensionality reduction of tissue sections of endometrial carcinoma by near infrared spectroscopy", Analytica Chimica Acta, vol. 724, March 2012, pp. 12-19.

[16] X.M. Zhao, S.Q. Zhang, "Facial expression recognition using local binary patterns and discriminant kernel locally linear embedding", Eurasip Journal on Advances in Signal Processing, vol. 20, April 2012, pp. 1-9.

[17] J. Han, J. Yue, Y. Zhang, L. Bai, "Kernel maximum likelihood scaled locally linear embedding for night vision images", Optics & Laser Technology, vol. 56, Jan. 2014, pp. 290-298.

[18] Z.Q. Su, B.P. Tang, J.H. Ma, L. Deng,"Fault diagnosis method based on incremental enhanced supervised locally linear embedding and adaptive nearest neighbor classifier", Measurement, vol. 48, Feb. 2014, pp. 136-148.

[19] P. Bonacich, "Some unique properties of eigenvector centrality", Social Networks, vol. 29, Oct. 2007, pp. 555–564.

[20] W.W. Zachary, "An information flow model for conflict and fission in small group", Journal of Anthropological Research, vol. 33, Dec. 1977, pp. 452-473.

Parallelism Analysis and Algorithm design of Petri Net

Ze-yu Tang,Wen-jing LI , Xuan Wang
College of Computer and Information Engineering
Guangxi Teachers Education University
Nanning, 530001,China
e-mail:liwj@gxtc.edu.cn

Weizhi LIAO
Guangxi Key laboratory of hybrid computation and IC design analysis
Nanning , 530006,China
e-mail:weizhiliao2002@yahoo.com.cn

***Abstract*— To solve the parallelism algorithm of Petri network system with parallel feature and implementing paralleling control and execution of Petri network, we propose the analysis and algorithm design of Petri network process. Based on employing P-invariants to partition the function of Petri network system, we firstly analyzed the situations of parallelism between processes, checked the effects of the transition behavior on processes' parallelism, and analyzed the transition behavior in the internal processes under various forms of process. Finally, by combining Parallelism of transition between inter-process and inner-process, we designed the parallelism algorithm for the processes of Petri network system, validated and analyzed it in practical examples. The experiment showed that the analysis and algorithm of Petri network processes' parallelism are feasible and effective so that it is a useful method to control and execute Petri network processes concurrently.**

***Keywords*- Petri Nets; Cross-Process; Internal Processes; Changes in Transition; Parallelism**

I. INTRODUCTION

Petri net is a modeling and analyzing tool for distributed system. It has a unique advantage in describing the process of system or sequence, concurrent, conflict and synchronous relationship in internal process. For any System, Petri net can describe it hierarchically, stepwise refinement. But if the system is too complex, stepwise regression analysis can be too long and inefficient . Aiming at this problem, domestic and foreign scholars put forward some methods For example: the literature [1]put forward resolve and research method of Petri net; the literature [2]put forward the decomposition method of Petri net based on the index of place; the literature [3]put forward functional classification method of Petri net based on P-invariant. However, these methods lack comprehensive and detailed analysis of the process of division. Therefore, this paper divides the function in the use of P-invariants， further detailed concurrency, internal division after the process of conflict and process shared place, sharing the transition of parallelism analysis. At last, we scheme out the parallel algorithm and realization it with programming.

* Supported by the National Natural Science Foundation of China (61163012); open fund of Guangxi Key laboratory of hybrid computation and IC design analysis[2012HCIC01];the University Scientific Research Project of Guangxi(2013YB147); Innovation Project of Guangxi Graduate Education(YCSZ2014187)

II. THE FUNCTIONAL PARTITIONING TECHNIQUE OF PLACE

If you want to know about the specific content of Petri nets, please refer to literature 1 and literature 3, the author does not introduce in detail here.

III. PARALLEL ANALYSIS BETWEEN PROCESSES

Several possible situations and corresponding solutions of Parallel analysis between processes are listed as following, which mainly analyzed the presence of a shared place and shared transition between two processes.

A. Sharing transition between processes

If there exists shared transition between any two processes, and there are no shared places, that is $\Sigma 1 \cap \Sigma 2 = \{t_i\} \neq \phi$, then there are synchronization transitions between the two processes. Transition

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.31

represents behavior in Petri net systems. Shared transition connect two processes which have to exchange information with the shared transition thus the transition play a role of a server.

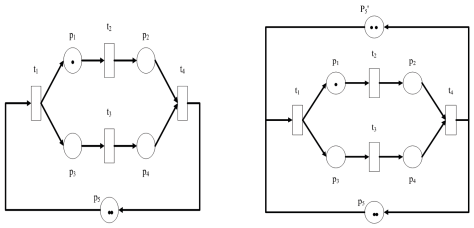
The detailed analysis process and example are discussed in reference 3.

B. Sharing place between two processes

1) Both the input and output of the shared place Pi are 1

If there is a shared place between any two processes, and the both the input and output of the shared place Pi are 1, that is $\Sigma 1 \cap \Sigma 2 = \{p_i\} \neq \phi$, $\left|{}^{\bullet}p\right| = \left|p^{\bullet}\right| = 1$, the place plays the role of resource sharing. However, in this case, the corresponding process of the place does not satisfy the dividing condition of the equation (3). Therefor，this paper proposes the following solutions:

First of all, locate the shared place pi of P-invariants' support set Xi and Xj in Petri net (i<m), then copy P_i (i<m) and its directed edge automatically to generate place p_{i1} (i<m) which is the same as the original place and the same edge. One of the support set and the original place form a sub-branch stand the new place and another support set constitutes another subnet, which will generate two independent subnets, as shown in Figure 1



a there is a shared place　　b copy the shared place

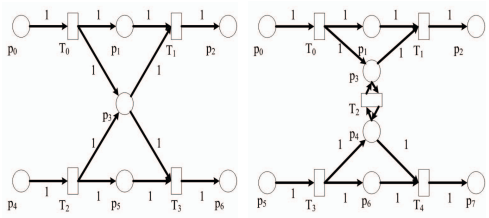
Figure 1 there is a shared place of P / T nets and copy the shared place after the P / T Network

From Figures 1b, the original process can be turned into the process $t_1p_1t_2p_2t_3p_5'$ and $t_1p_1t_2p_2t_3p_5$ of t_1 and t_4 which share transition. These two processes are parallel processes with server t_1 and t_4.

2) The input and output of shared place P_i are n (n>1)

If there exits a shared place between any two processes, and the input and output of the shared place P_i are n (n>1),that is $\Sigma 1 \cap \Sigma 2 = \{p_i\} \neq \phi$, $\left|{}^{\bullet}p\right| = \left|p^{\bullet}\right| = 1 (n > 1)$,the corresponding process of the place does not satisfy the divided condition of the equation (3). This paper proposes the following solutions:

Locate the shared place, and add the n-1 place -transition pair to generate a shared transition between two processes. The role of the transition into decide the original resource place into two resource places, thus resources can be allocated fair and reasonably. Example: Figure 2



a the input and output of shared place Pi are n　　b add the n-1 place -transition pair

Figure 2 P / T network of the input and output of shared place P_i are n and P / T network of added the place -transition pair

Figure 2a does not satisfy equation (3) obviously, so function can't be divided. We use the above method to add a place-transition pair P3-T2 (Figure 2b), and get the two processes running in parallel.

Let's solve and prove parallelism by P-invariants : according to Definition 1, the correlation matrix of Petri nets

$$D = \begin{bmatrix} 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix}$$

Obtained P-invariants were as following: $X_1=[2,1,2,1,0,0,0,0]^T$,$X_2=[0,0,0,0,1,2,1,2]^T$. Match divided condition, therefore, the system shown in Figure 2 can be divided into two parallel processes.

IV. PARALLEL ANALYSIS OF TRANSITION IN THE INTERNAL PROCESSES

Take the place in the internal processes as a resource or a state during the execution of the process and transition as an operating behavior or an action in the process of implementation when giving analysis to Petri nets. Here's analysis to the behavior of transition inside the process:

A. In case of the process of transition t meet •t = t• = 1

If the process transition meet •t = t• = 1,(there is only one input and one output place in the transition), the

transition is the local behavior or local action of process. This transition behavior occurs within the same process, and it's a child process of the process, the processor where the child process is achieving serial.

B. Process of any two transition t1 and t2 concurrent in the case of M

If for any two transition t1 and t2 in the processes, there exists an identification M, making $\boldsymbol{M}[t_1>\boldsymbol{M}_1 \rightarrow \boldsymbol{M}_1[t_2>$and $\boldsymbol{M}[t_2>\boldsymbol{M}_2 \rightarrow \boldsymbol{M}_2[t_1>$, then between the transition t1 and t2 are independent on causation . They are mutually independent events. So the transitions can be executed concurrently, and they can be achieved with a distributed work pool policy. As shown in Figure 3

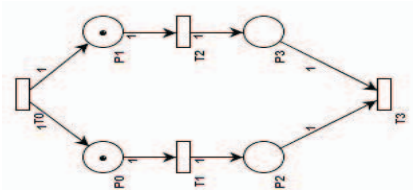


Figure 3 P / T Network of two concurrent transitions

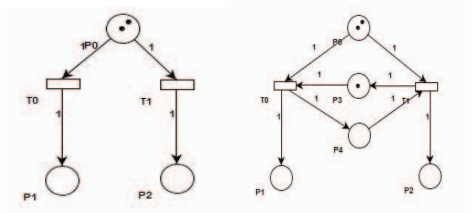
T0 occurs in marking M, so P0, P1 get new identities that T1, T2 can occur simultaneously. At the same time the main process assigns tasks to T1, T2, so that the two sub-processes run simultaneously, thus achieving load balancing, which is a distributed work pool strategy.

C. the situation of any two transition t1 and t2 conflict in M in the process

If for any two transitions t0 and t1 in the process there exists a mark M, making $\boldsymbol{M}[t_0>\boldsymbol{M}_0 \rightarrow \neg\, \boldsymbol{M}_0[t_1>$ and $\boldsymbol{M}[t_1>\boldsymbol{M}_1 \rightarrow \neg\, \boldsymbol{M}_1[t_0>$, transition t0 and t1 in the process conflict at M. For the two transitions in conflict, when one occurs first, the other transition will produce a conflict and lose occurring rights, vice versa. We can resolve the conflict of transition by applying an external control.

D. The situation that concurrent and confliction Coexistence any two between t1 and t2 in Process at the M

Example:



a Concurrency and conflict are saved in the P / T Network b the P / T Network increasing the control loop

Figure 4 Concurrency and conflict are saved in the P / T Network and the P / T Network increasing the control loop

The situation of concurrency and conflict coexistence is shown in Figure 4a. with the external control, we obtain Figure 4b. Easy to know, the effective conflict of t0 and t1 has been eliminated, and the control device has played a role that enables transition t0 and t1 to share their public resources in turn. Also, the distributed work pool policy can also be achieved, making it load balancing to achieve parallelism.

V. PETRI NET PROCESS PARALLEL ALGORITHM DESIGN

In the circumstance of multinuclear clusters, we use MPI+ OpenMP parallel programming pattern to proceed concurrent design, based on the function and parallelism analysis of inter process and internal process.

A. MPI+OpenMP Parallel programming model group

In MPI+ OpenMP Parallel programming model group, the frequently-used Parallel programming model is MPI+ OpenMP hybrid paradigms.It is parallel processing with two stages： upper layer MPI, namely parallel between nodes(processes) and under layer OpenMP, parallel in nodes(threads).Here are the advantages:(1)MPI can solve the problem of process communication between multi-core processors, and multithreading established by OpenMP can solve the problem of data exchange between inter-processor of any multi-core processor to decrease the expenditure of communication.(2) Data processing is mainly composed of various (nodes) through internal multithreaded ways, which can reduce the data transmission between the process(nodes) as well as the communication over head. (3)With processor number unchanging, the number of MPI processes significantly decreases.

B. Petri net process parallel algorithm based on multi-core clusters

Based on the function division of Petri net above and analysis of inter-process and internal process of Petri net, Petri net parallel algorithm design system are as follows under the circumstance of multinuclear clusters:

Input: the incidence matrix of Petri net system

Output: Petri nets model of parallelization

Start:

Step1: Get several processes Petri nets by Functional partitioning algorithm of Petri nets.

Step2: If there are the shared transitions between the division process , and no shared places, that is $\Sigma 1 \cap \Sigma 2 = \{t_i\} \neq \phi$, find shared transition and regard it as server process, or else enter the next step.

Step3: If there is shared places pi between process by division, and both numbers of places of input and output of pi are 1, that is, $\Sigma 1 \cap \Sigma 2 = \{p_i\} \neq \phi, \left| {}^{\bullet} p \right| = \left| p^{\bullet} \right| = 1$, find the shared place Pi, copy the pi and it's directed edge to generate a new pi' and set it with another branch constitute another subnet ,so two separate sub-processes are generated. The two new shared transitions of formation are regarded as two server processes .Otherwise, turn to next step.

Step 4: If there is shared place pi between process by division, and both numbers of places of input and output of pi are n(n>1),that is $\Sigma 1 \cap \Sigma 2 = \{p_i\} \neq \phi, \left| {}^{\bullet} p \right| = \left| p^{\bullet} \right| = 1 (n>1)$, find shared place Pi and add a place - transition pair for it so that the original process can turn into two separate processes with shared transition. The new shared transitions of formation are regarded as server processes and play a role of allocating shared resources fair.

Step 5: For each of the divided processes, repeat steps 2,3 and 4 until it can no longer be divided. Then we could get several processes.

Step 6: All processes are mapped onto different nodes in the cluster multicore, using MPI Programming model to solve process communication between multicore (nodes) . Each Petri nets process completed by each node, the server process in master node achieve transmission of information between processes, equitable sharing of resources.

Step 7: Inside each process, program with OpenMP and then use the decentralized work pool strategy to achieve concurrency.We can control the transition conflict through external conditions, add place to form a control loop, then realization implementation by distributed work pool.

Step 8: Each process is a sequential state machine, which could be seen as a programming task model. When you reach no successor state places, the loop ends.

Step 9: Internal processes transition actions are completed by each node processor through the implementation of internal processes sequential programming.

Step 10: The transition server shared by processes is responsible for information exchange between processes and resource allocation process ,etc.

VI. APPLICATION EXAMPLES AND EXPERIMENTS CHECKING

Finally, we examine the functional partitioning and parallel algorithms of Petri net system through a simulated student management system of determining whether a college can graduate or not. Student management system business flow is showed in Figure 5.

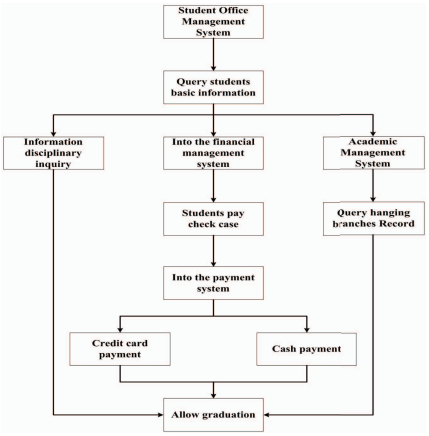


Figure 5 college management system

To solve these practical problems, we put the instances of the resources or process during the execution of a state as a place; The operation of the process execution behavior, or an action as a transition. We constructed Petri net model based on Figure 5 and obtained Figure 6.

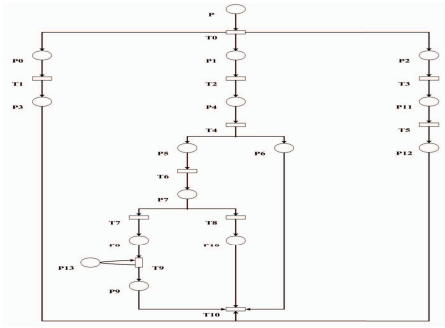


Figure 6 college management system Petri nets model.

Among them, P indicates the student management system; P0 represents the student information interface; P1 represents financial management information system; P2 represents Senate management system shows students inquiry basic information; T1 represents a query disciplinary records; T2 represents into the financial management system; T3 represents into the educational management system; T4 indicates that the query payment of premiums; T5 represents a query hanging branches record; T6 represents enter the payment system; T7 represents select credit card payment; T8 represents cash payment; T9 represents carry credit card payment; T10 represents obtain graduate qualifications.

Though the partition of system, several parallel processes can be obtained:

Process one, query disciplinary record; Process two, check payment of premiums, and enter the payment system to pay; process three, enter educational management system and query the hanging branches record .

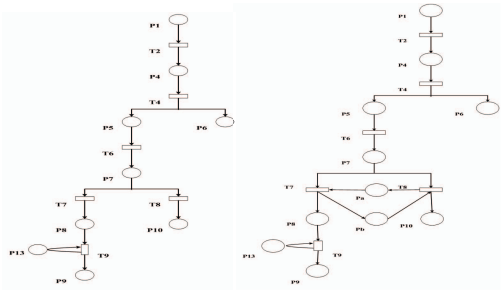


Figure 7 Conflict in the payment system and the corresponding solutions

As shown in figure 7, through the analysis of Petri model of query Payment situation we know that conflict will occur when students choose the same payment type. We can solve the problem by applying an external control, adding places p3 and p4, So that p3, t0, p4 t1 form a control loop. When a student chooses to pay by credit card, the next person must select cash transactions, or wait. So that achieved parallel effect.

In addition to the above mentioned internal processes in parallel, we can also see the parallel implementation between process1, 2, 3, by the division of the system.

In Linux 8.0 + MPI, CPU is P4, memory is 1G, JAVA programming language environment simulation laboratory Petri nets functional partitioning and parallel algorithms, and to the division results of the subnet of the application examples verified.

The results were as following:

Enter Petri nets Σ incidence matrix D [11] [13] =

$$\begin{bmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Initial identification M0= { 1,0,0,0,0,0,0,0,0,0,0,0,0}.

Experimental results and analysis:P-invariants are $X1^T$=(0,2,0,0,2,1,1,1,1,1,1,0,0),$X2^T$=(0,0,1,0,0,0,0,0,0,0,0,0,1,1),$X3^T$=(1,0,0,1,0,0,0,0,0,0,0,0,0);X1,X2, X3 are support sets. Subset Γ_p1={X1, X3} and Γ_p2={X2, X3} satisfy the conditions of Theorem 1.

These results indicated that the number of subnets of this algorithm result is the same with the result from theoretical analysis, thus verifying the feasibility of parallel.

VII. CONCLUSION

This article divided Petri nets function by p-variables, then analyzed the the division processes within and between processes through examples and programming. Verify the feasibility of functional partitioning and parallel algorithms of Petri net system. The following work is the practical applications in related fields, thereby increasing efficiency.

REFERENCES

[1] WuZheHui,“Introduction to Petri Net [M],”Beijing Machinery Industry Press,2006.

[2] ZengQingTian, WuZheHui, “The Petri nets decomposition methods based place Indicators,”Computer Science,pp.29(4) : 15-23,2002.

[3] LiWenJing, LiaoWeiZhi, WangRuLiang, “Functional partitioning and parallel algorithms of Petri net system,”ComputerEngineering,vol.35(21), pp.48-50,2009.

[4] ZhaoYongHua, ChiMeiBing, “Based on SMP clusters hybrid MPI + OpenMP programming model and the effective realization,”Microelectronics and Computer, pp.22(10) :7- 11,2005.

Study on error-detecting approach for fault tolerance recomputing oriented parallel digital terrain analysis

Shoushuai Miao [1], Wanfeng Dou [1,2], Yan Li[1]

(1.School of Computer Science and Technology, Nanjing Normal University, Nanjing, 210023)
(2.Jiangsu Research Center of Information Security & Privacy Technology, Nanjing, 210097)
e-mail:ms_shuai@126.com; douwf-fly@163.com; 467094983@qq.com

***Abstract*—In recent years, the research of parallel digital terrain analysis has become a hot spot. Using the parallel computing technology to solve data intensive problems, and it has become a development trend in digital terrain analysis. On the other hand, with the development of hardware technology and new applications, how to ensure the reliability of the computing results is a one of the key problems. We can improve the system's ability to provide the right service by adopting fault tolerance technology properly. The paper proposes a parallel error-detecting approach which is based on a mixed mode of shared memory and distributed memory. By adopting the parallel comparison, it can improve the efficiency of fault tolerance recomputing. According to the error-detecting analysis of slope algorithm, it proves the effectiveness of error-detecting approach based on fault-tolerant recomputing and achieves minor overhead.**

Keywords- parallel digital terrain analysis; parallel computing; fault tolerance; error detections

I. INTRODUCTION

With the rapid development of software and hardware technology, the rising popularity of multi-core processors, the parallel computing has been used to solve more complex problems in the many fields. In order to solve the problem of huge amounts of spatial geographic data analysis, serial to parallel computation becomes an inevitable trend for the computing architecture of the digital terrain analysis. At the same time, the utilization of multi-core or multi-node computing resources can improve the ability that can handle the complex algorithms in the digital terrain analysis. At present, the research of the parallel technology has become a new developing direction and a hot topic in the digital terrain analysis [1, 2].

Recently, as the updating and developing of hardware technology, the speed of parallel computation is faster. But the reliability of the parallel computation is becoming a new challenge. We must apply the appropriate fault tolerant technique to improve system's ability, which can provide the right services. In the digital terrain analysis, the grid DEM(Digital Elevation Model) has a huge amount of data. We need the GB level of space to store the DEM data. Under the circumstances, the traditional software fault tolerance cannot meet the demand and we introduce parallel recomputing[8] to solve a lot of computation intensive and data intensive applications. However, the study of the errors detection approach based on the parallel recomputing is rare. According to the characteristics of digital terrain analysis, the paper proposes the approach based the parallel recomputing. This method adopts a separation mode. By using the threads to compare the execution results and processes to calculate the algorithm of digital terrain analysis. Because of this separation, we can compare the results by using the multi-thread method when one process is a failure. Besides, there is no load balance problem in the multi-thread, so the parallel recomputing can greatly enhance the performance. By executing both original results set and results copies on the threads, and achieve error detection by comparing the execution results.

II. THE RELATED WORK

The basis of fault tolerance is errors detection. In parallel system, there are two kinds of common faults, hardware and software failure. A hardware failure can be further divided into two categories: data errors and control flow errors [3]. The data errors refer to a hardware failure, which caused changes of content in data storage unit. The data errors usually do not immediately cause the system failure. The execution of the program spread other parts and it eventually leads to system failure. The control flow errors refer to the hardware failures make the instruction errors in the relevant control unit, which can lead to program flow to the wrong branch. In some cases, the control flow errors can be detected by the system's structure and reported to the system.

Neumann [4] first put forward the ideas of the redundancy. The redundancy contains hardware redundancy, software redundancy, time redundancy and information redundancy. Nahmsuk introduced EDDI [5], which is a method of instruction-level redundancy. Through compile-time complicated instruction, it can obtain a redundant copy and insert the compare instruction before store or branch instruction. By comparing original program and calculation results of the redundant copy of the program, if the results are not the same, then there is a hardware failure. EDDI can achieve a high error-detecting rate. But the copy instruction will bring more overhead. Then, Nahmsuk et al. put forward the new hardware error detection technology which based on redundant instruction, namely ED4I [6]. This is also a method of instruction level redundancy. Using the difference transformation in the complex instruction, which the number of instruction operand and a copy of the program of instruction operand keep k times relationship and k values has a decisive influence on error detection rate.

Fu et al. put forward a kind of error detection by redundant processes for MPI programs, namely REDReP[7]. The method dynamically creates process copy for every MPI process and the intermediate results are compared at the same time. But for huge amounts of spatial data, the big data will be divided into many data blocks and we need

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.47

more processes to handle these data blocks. For inter-process communication and storing the intermediate state, it can bring more communication overhead and memory overhead. For new fault tolerant method, namely the parallel recomputing [8], it also copied simple calculation to idle processes. By comparing the calculation results of the partial, it can achieve the errors detection. In view of the parallel error-detecting approach of the digital terrain analysis, Song [9] et al. presented an approach based on the adjacent process, which realize the error detection by comparing the block redundancy rows. The error-detecting overhead of the approach is much lower than redundant processes. It can be very good to detect calculation errors, but also can be found out the errors according to the checking result set. The paper is based on parallel data intensive characteristics of digital terrain analysis and puts forward a kind of parallel digital analysis of parallelism error-detecting methods, which using the redundant processes and multi-threading technology to realize data sets parallel processing.

III. ORIENTED PARALLEL RECOMPUTING OF THE ERROR-DETECTING

A. The parallel recomputing

The recomputing was first used for the online fault detection in an arithmetic logic unit. Patel and Fung [10] proposed the shift operation of time redundant in 1982. The principle is that compared the calculation results of before and after the shift. The fault detection ability depends on the shifts of digits. The basic idea of the parallel recomputing is that the program needs to store the calculation variable results. In case of an error, the program will be rolled back recently to save state of computing. After, the approach made use of fault-free processes to complete the calculation of fault recovery [11]. The recovery technology is based on the parallel recomputing and can greatly improve the fault tolerant performance of parallel program.

The main points of the parallel recomputing include the segmentation and parallel scheduling problem. According to the length of the parallel, a range of the variables and the constraints, we start to divide the task. The partition is based on the user's guide of initial correction. Clearly, the partition requires the experiences of the developers and the complexity of the specific procedures. The scheduling bookkeeping records the loop blocks that executed by every thread in the execution of the program. Each thread has been out of circulation block. When a thread failure occurs, it can make sure the collection of loop blocks in accordance with the scheduling book-keeping of thread. Then, it builds parallel recomputing scheduling code.

B. The parallel error detection approach

The digital terrain analysis is facing the huge amounts of data in scientific computation. The data parallel is one of the effective methods to realize in the parallel digital terrain analysis. Through the parallelization of data, it can improve the efficiency of system processing. The grid DEM data as the source for all kinds of terrain factors can calculate the parameters of the surface topography, terrain morphology analysis and statistical features of the terrain analysis. Due to resource limitation of competition, it will lead to failure for the processes and the computed results may include errors. How to detect the error is the focus of the paper.

The paper is based on multi-thread way of error detection. Reading, writing and comparing the data will be done in the thread node. The process will handle the calculation of the data and send the result to the corresponding thread. The work flow of parallel error detection is shown in figure 1. The specific process is as follows:

Step 1: Data partition. The data is divided into blocks. According to the size of DEM, the data will be divided by row or column. The number of blocks is decided by the idle threads.

Step 2: Reading the data. Each data block will be read by the corresponding thread. Afterwards, the new data will be sent to the process and the copy of the process.

Step 3: Data calculation. The data block will be processed by the process and the copy of the process.

Step 4: Error detection. The calculated results and a copy of the calculation results will be sent to the master node. The paper takes the fuzzy electoral law and compares the calculated results. The error point will be controled within the limit of the fault-tolerant coefficient, namely: ε. In the difference of terrain analysis algorithms, we can set different values to ensure correctly operation of the fault-tolerant mechanism.

Step 5: Judgement. If the value is beyond the reasonable scope, it will turn to the first step and carry on the parallel recomputing.

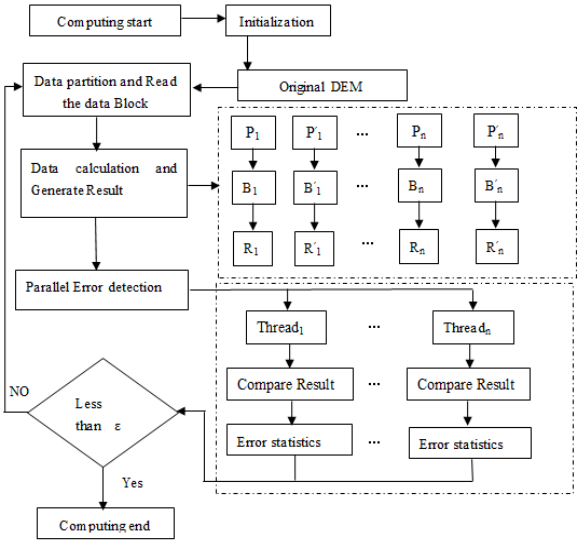


Figure 1. Work process of parallel error detection

In the parallelization of error detection workflow, the paper has do some work. First of all, the original of data block must carry on the reasonable division, which can improve the memory usage and effectively control the load balancing problem. Second, the computation results of each node will be detected. So, these works can effectively prevent the occurrence of failure.

Before the error detection work, the priority is to access and storage the data set. Secondly the copy of the process and the process will deal with the data. Because of the large amount of raster data, it will need more resources

overhead by reading the data from hard disk files into the memory. The paper adopts the model of multi-thread, every time a thread reads a data block and uses the mutual mechanism to avoid the conflicts of the reading and writing data at the same time.

(1) Select data blocks

The number of data blocks is determined by the number of threads; each thread alone reads and writes a block of data. The data level of DEM reachs megabytes (MB) or gigabytes (GB) in parallel digital terrain analysis. Assuming the number of threads is M, the size of the data block is MSize, the divided blocks are equal in size except the last data block, the size of the data block is calculated as follows (except for the last one data block) :

$$FTSize = \frac{MSize}{M}$$

The size of the last data block:

$$LTSize = MSize - (M-1) \times FTSize$$

Assume we have a data set including n data blocks B = $\{B_1, B_2,..., B_n\}$, where B_i is the data block which the process i will deal with. The calculation result set R = $\{R_1, R_2,..., R_n\}$, where R_i is the result that the data block B_i is calculated by the process i. The whole results are gained by performing a terrain analysis.

(2) The error-detecting of the results set

In parallel digital terrain analysis, grid DEM is to describe the surface with regular grid unit. Generally, the raster data stores the data with two-dimensional array. The value of array represents the center of the grid elevation values. Through terrain analysis algorithms, there is often computing error in certain intervals. To judge the detection error, we can figure out the comparing result whether the result in the error range. When the data is divided into several small data blocks, the data block B_i is calculated by the corresponding process P_i. At the same time, B_i' (a copy of the data block B_i) is calculated by P_i'(a copy of the process P_i). The calculation results will be returned the master node and the master node uses multithread technology to compare the returned set. The calculation results (R_i and R_i') are compared line by line and count the error point. Specific process is as follows.

(a) Two detection threads set to be started at the same time and threads will receive the data from the master node. In the master node, the threads will compare the data by line and statistical error point. The results of difference comparison by line are shown in figure 2.

$$\tau = \sum_{i=1}^{i=m} \sum_{j=1}^{j=m} \left(R_{ij} - R'_{ij} \right)$$

(b) Calculation error-point ratio.

$$r = \frac{C}{i \times j}$$

C stands for the number of different data in the result set.

*(c) W*hether error point within a reasonable range.

If $\tau < \varepsilon$, it means the comparison results in the reasonable range, otherwise the program will be taken fault-tolerant processing.

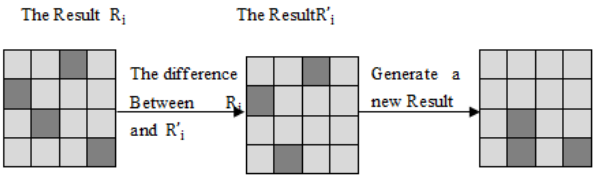


Figure 2. Process of difference comparison by line

(3) The Error detection overhead

In this paper, the overhead of error detection approach contains two aspects: communication overhead and comparison overhead. The process between sending and receiving data will generate the communication overhead. In dual-redundancy error detection mode, the processes not only produce communication overhead, but also have comparison overhead. The master node uses the multi-threaded mode in the paper. Reading and writing the data will be finished in the thread. Computing the data will be handled in the processes. Ignoring the overhead of threads creation, the comparison overhead in connection with the size of the data block and the comparison approach. Assuming the total number of processes is 2 N + 1 in the program. The number of threads is M. D stands for the communication between two processes. The comparison of each intermediate results is $d_1, d_2, \ldots, d_M$. Simplify the discussion, assuming that a data comparison and data communication take a unit time, so that we can use the data to measure the amount of overhead.

Under normal circumstances, regardless of the communication, computing overlapping and the time of threads start and stop, the time of program includes the communication time and compared time, namely:

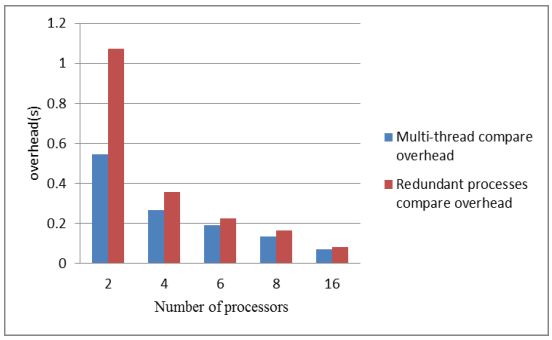
$$T = T_c + T_m = 2ND + \sum_{i=1}^{M} d_i$$

IV. THE EXPERIMENT AND ANALYSIS

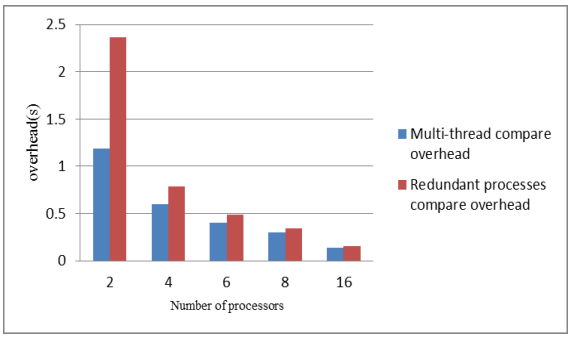
The experiment finishes the error detection performance on a small scale cluster system. Configuration is as follows: processor configuration XeonE5645 2.8 GHz quad-core processors and 8G memory, using Gigabit Ethernet connectivity between nodes. In the cluster system, we use the master-slave model. A primary node is responsible for the data distribution and error detection. Software environment: GDAL 1.6.1, OpenMP 1.5.4, GCC 4.4.7, MPICH2. Data configuration: DEM data dimension is 31492 * 13717 and 17087 * 11412. The data type is floating-point. The size of the former is 1.61 GB and another one is 376MB, TIFF format.

In software fault tolerant approach, the effective approach of parallel program error-detecting is to use redundant processes [12] [13]. If we use the process comparison, one process needs to send data to another process and another process will compare the data to the original data. The process will be more communication and may lead to load imbalance. In the paper, we adopt the redundant processes and threads mixed using. Through dividing the data block, it will speed up the data processing ability and parallel comparison by using the threads at the same time. It can improve the efficiency of error detection. In order to verify the efficiency of error detection, according to the slope algorithm to verify the ability of

error detection. The overhead of parallel slope algorithm is shown in figure 3. The paper proposes the approach differs from redundant processes, as shown in Figure 3(a), and the data size is 374MB.The results show that the more the division number, the smaller the amount of data blocks. At the same time, the threads are the less overhead, but communication overhead will increase. As the increasing number of the data blocks, the number of threads is increasing rapidly and two kinds of approach tend to be equal. Figure 3(b) looks like Figure 3(a), so data size is 1.61GB and includes a lot of computation and detects more results. It is shown that error detection approach based on parallel recomputing can achieve the best performance. We adjust the number of threads. If the number of threads exceeds a certain value, it will spend more time. The number of threads M < = (N - 1) / 2 (N presents process number) and M presents the number of data blocks.



(a)



(b)

Figure 3. Overhead of parallel slope algorithms with different data size

V. CONCLUSION

The digital terrain analysis has the characteristics of data intensive, so adopting the data parallel is one of the effective strategies for parallel digital terrain analysis. The paper proposes the error-detecting approach which based on the parallel digital terrain analysis. Firstly, the approach adopts multiple threads to read and write the data blocks in the master node. Secondly, the process and a copy of process will handle the partitioned data blocks, respectively. Finally, the computing results will be returned. According to the returned results, the master node uses multi-thread technology, which based on the parallel comparison. By using the multi-thread technology can effectively avoid the failure node. In a mixed-mode, In the case of failure, the program can still run. The master node can start a new thread to replace the failure of the thread. However, the using the number of threads should not be too many. Too many threads start and stop will spend more than the running time of the program. The paper only considers the results of error detection by comparing the computing from process and its copy process, but without comparing the reading and writing data. In the comparison of the result set, the paper is mainly on the primary node. If the other nodes are also adopted the approach, it may improve the efficiency. At the same time, it will bring more overhead and may cause the load imbalance. In the future, we will further study in this respect.

ACKNOWLEDGMENT

This work has been substantially supported by the National Natural Science Foundation of China (NO. 41171298).

REFERENCES

[1] T. Li, J. Liu, and Y. He. Application of cluster computing in the digital watershed model [J]. Advances in Water Science,2006,17(6) : 841-846.

[2] J. Lv, D. Liu, and X. Jiao. Research of parallel DEM algorithm [J]. China Journal of Image and Graphics,2002,7(5) : 506-512.

[3] H. Fu. Research on fault tolerant parallel OpenMP program [D], Doctoral Dissertation of National University of Defense Technology, 2010.

[4] Neumann J V. Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components. Princeton University Press, 1956.

[5] Nahmsuk O, Philip P, Edward J. Error detection by duplicated instructions in super-scalar processors[J] . IEEE Transaction on Reliability , 2002, 51(1) : 63- 75.

[6] Nahmsuk O, Subhasish M , Edward J. Error detection by diverse data and duplicated instructions[J] . IEEE Transaction on Computers, 2002, 51(2) : 180- 199.

[7] H. Fu, W. Song, and X. Yang. Implementation of MPI program error detection by redundant processes [J]. Microelectronics and computer,2009,26(9):53-56.

[8] P. Wang, Y. Du, etc. parallel recomputing: for high performance computing new fault-tolerant method [J]. Computer science, 2009,3(36).

[9] X. Song, X. Liu, G. Tang, etc. Fault tolerant algorithm of digital terrain analysis in parallel [J]. Geography and Geographic Information Science,2013,29(2).

[10] Dubrova E. Fault tolerant design: an introduction [M]. Draft, Kluwer Academic Publishers: London, 2008.

[11] Shang Y, Jin Y, Wu B. Fault-tolerant mechanism of the distributed cluster computers [J], Tsinghua Science and Technology, 2007, 12(supplement 1):186~191.

[12] H. Fu, Y. Ding, etc. Using a parallel recomputing OpenMP fault tolerance mechanism [J]. Journal of software,2012,23(2):411-427

[13] REED D A, LU C, MENDES C L. Reliability challenges in large systems[J]. Future Generation Computer Systems,2006,22(3):293 –302.

Low energy-consuming cluster-based algorithm to enforce integrity and preserve privacy in data aggregation

Zhengwei Guo, Xiaojiao Ding
College of Computer & Information Engineering,
Institute of Image Processing & Pattern Recognition
Henan University
Kaifeng, China
Email: {gzw@henu.edu.cn, Fanxian2008@163.com}

***Abstract*—This paper proposes a low energy-consuming cluster-based algorithm to protect data integrity and privacy named ILCCPDA, which can dynamically elect cluster head by LEACH clustering protocol and take the simple cluster fusion approach to reduce the data transmission, thus reducing energy consumption. ILCCPDA can detect data integrity by adding homomorphic message authentication code and take the random key distribution mechanism for data encryption. It can solve the problem of the integrity, privacy and energy consumption in the wireless transmission of sensor data.**

Keywords—wireless sensor networks(WSN); data aggregation; privacy protection

I. INTRODUCTION

Wireless sensor networks is the bottom of Internet of Things, responsible for data collection and transmission. Data fusion techniques in [1] can remove redundant information acquired by wireless sensor and reduce the amount of data, so as to achieve the purpose of saving energy and prolonging the network life cycle. Solving the final result's safety issue in data fusion process is a hot research topic in recent years. TAG algorithm presented in [1] is a typical fusion algorithm applied in wireless sensor networks. Since the data fusion technology has no privacy protection features and wireless sensors always face insecurity in the real environment, [2] proposed the concept of privacy-preserving in data aggregation, a cluster-based private data aggregation algorithm CPDA, and privacy protection fusion algorithm SMART based on data fragmentation. For the large calculation of CPDA and the large data communications of SMART, data can be easily lost. For the defect of SMART, [3] proposed a low-power data fusion algorithm for privacy protection named ESPART, which reduces energy consumption and the data traffic. To improve the accuracy and reduce the data traffic of SMART, an optimization factor is added in [4]. CPDA has been improved in [5], to reduce the amount of data traffic. However, the data integrity that is also part of security has not been detected by these algorithms. It can prevent data from being tampered and being injected false information. In [6], attacks against data fusion operations include active attacks and passive attacks in the wireless sensor network, privacy protection mainly against active attacks and integrity protection mainly against passive attacks. Currently, there are many researches on integrity protection algorithms [7-11]. Reference [7] and [8] extend SMART algorithm and CPDA algorithm to add the integrity of the testing function, but they also inherit the shortcomings of the original algorithm, such as complexity and large communication overhead. In this paper, the computational complexity and energy costs have been improved, on the basis of improved CPDA and of adding a homomorphic message authentication code [12] mechanism to detect the integrity of the data.

II. RELATED WORK

A. Network Model

In this paper, the wireless network is represented by a connected graph $G(V,E)$. The vertex $v(v \in V)$ represents a sensor node. And an edge $e(e \in E)$ represents a wireless link between nodes. The number of nodes in wireless sensor networks is represented by $N = |V|$, Nodes include BS (Base station) nodes, cluster heads and common nodes. We define data fusion function as $y(t) = f(d_1(t),\ d_2(t),\cdots d_N(t))$, $d_i(t)$ means the data that is collected by the node i at the time t. In this paper, addition function is the object of our study [3], $y(t) = \sum_{i=1}^{N} d_i(t)$.

B. Encryption Methods

Encryption method used in this paper the same as CPDA takes the random key distribution mechanism [13]. At first, it generates a key pool with K keys. Then each node selected key from the key pool randomly. If the two nodes have a common key, there will be a secure link between them. The probability that any two nodes can share the same key is

$$p_{connect} = 1 - \frac{((K-k)!)^2}{(K-2k)!K!} \quad (1)$$

We may establish a secure multi-hop link between Nodes without a shared key. Probability that eavesdropping nodes get

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.32

keys is $p_{overhear} = k/K$ .Under normal circumstances, $p_{overhear}$ is a very small number. So the probability to be eavesdropped is very small.

C. Clustering methods

We use LEACH algorithm [14] to elect cluster head. LEACH algorithm is an adaptive clustering algorithm, and its implementation process is cyclical. Each round of the cycle is divided into phase of establishment of the cluster data and stable data communication phase. In the phase of establishment of the cluster data, adjacent nodes dynamically form clusters, and cluster heads are randomly generated. In the stable data communication phase, the node within the cluster sends the collected data to the cluster head node. The cluster node sends the data result to the base station node by using data fusion technology.

The election process of the cluster head in LEACH algorithm is the generation of a random number among 0 and 1. If this number is less than the threshold T(n), it will be the cluster head. T(n) can be expressed as:

$$T(n) = \begin{cases} \frac{p}{1-p*(r \bmod 1/p)}, & n \in G \\ 0 & \text{,otherwise} \end{cases} \quad (2)$$

P is the percentage of the cluster heads in all the nodes, r is the election round. G is the node collection where nodes are not elected as cluster heads in this round.

D. Homomorphic Message Authentication Code Mechanism

In this paper, we take homomorphic message authentication code mechanism [12]. As the MAC data block produced by traditional MD5 and SHA-1 is too large and it's algorithms are too complex to adapt to wireless sensor networks. The homomorphic message authentication code mechanism in this paper is different. Its MAC data block is only 10 or so, and has the same normality. When MAC codes integrate, keys will be changed. We use this characteristic to test integrity. Homomorphic message authentication code algorithm uses the principle of homomorphic fusion of two MAC functions. Let homomorphic message authentication code is $MAC(d_1) = MAC(\mathrm{d}_1, g, k_1) = \mathrm{g}^{d_1+k_1} \bmod M$, d_1 is the data collected by the node 1. k_1 is the key of the node. M is a large prime number, g is a shared key of base station nodes and all fusion trees in the network. Homomorphism message authentication code MAC function of d_1 and d_2 can be added to obtain the fusion result by homomorphic sex $MAC(agg) = MAC(d_1) * MAC(d_2) = g^{d_1+d_2+k_1+k_2} \bmod M$ $= MAC(d_1 + d_2, k_1 + k_2)$.And the key of fusion result become $k_1 + k_2$ from k_1.

III. ILCCPDA ALGORITHM

This paper uses the following parameters to describe ILCCPDA:

d_A is the data collected by the sensor node A.

REC_A is the slice data received from node A.

AGG_A is the integration of the result of the node A(initially zero).

$DMAC_A$ is the integration of the result of MAC, the initial value of 1.

J_A is the number of slices received by node A(initially zero).

$\oplus$ Represents some arithmetic.

A. Algorithms Preparation Phase

Key distribution: MAC key is distributed end to end between BS nodes and other nodes. Only g and K are distributed at this stage, and g is open to the entire network. k is the private key of a node, The k of all nodes is stored in the BS. Nodes in the network form the cluster by LEACH protocol.

B. Cluster Data Collusion Stage

1) Cluster data slicing stage

Let a node is divided into three fragments, one for themselves, and the others randomly for other nodes. A slice is represented by seed. Let node A sends a slice seed to node B. The data of the slice seed is $\left|seed_{AB}\right|MAC(seed_{AB}, k_A)$, then $DATA_A = DATA_A - seed_{AB}$.

2) Cluster data mixed operation

When all fragments have been sent, the nodes will begin to take the following mixed steps (take node A for example)

$$MAC_A = MAC_A \oplus MAC(d_A, k_A)$$
$$d_A = DATA_A + REC_A$$
$$AGG_A = AGG_A + d_A$$
$$DMAC_A = DMAC_A \oplus MAC_A$$

C. Data Integration Phase

1) Ordinary nodes (take A for example) upload data to the cluster head node (take E for example). Shown in figure 1, the data is $AGG_A\left|DMAC_A\right|$.

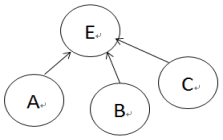


Fig.1 A cluster tree

2) Proceed as follows after the cluster head node(take E for example)receives ordinary node data:

$$AGG_E = AGG_E + AGG_A$$
$$DMAC_E = DMAC_E + DMAC_A$$

3) The cluster head node (take E for example) uploads fusion data to the BS node as follows, shown in figure 2.

$$AGG_{BS} = AGG_{BS} \oplus AGG_E$$
$$DMAC_{BS} = DMAC_{BS} \oplus DMAC_E$$

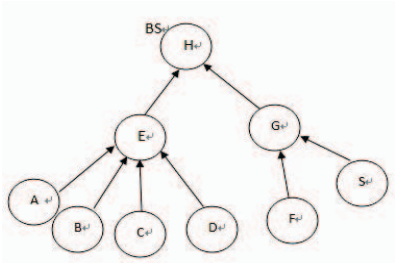


Fig.2 A complete fusion tree

D. Integrity Testing Phase

When all the cluster head nodes send fusion data to the BS node, Integrity testing starts: In the BS, $AGG_{BS} = d_A + d_B + \cdots + d_H$, AGG_{BS} is the final fusion results obtained in the network. $DMAC_{BS} = DMAC_A + DMAC_B + \cdots + DMAC_H$, $DMAC_{BS}$ is the MAC value integration of all nodes. At this point, the true key is $K_{\max} = k_A * J_A + k_B * J_B + \cdots + k_H * J_H$. $MAC(AGG_{BS})$ is obtained by using $K_{\max}$ and AGG_{BS} .Compares $MAC(AGG_{BS})$ with $DMAC_{BS}$,if they are the same, the integrity is not destroyed, vice versa.

IV. EVALUATION

In this section, we will analyze the capability of ILCCPDA from three aspects, including privacy protection, data traffic and data integrity, and compare it with other algorithms. Currently the typical algorithm includes CPDA, iCPDA and iPDA. In this paper, we use matlab to simulate.

A. Privacy Protection

Let p(q) is the probability that the information of nodes is eavesdropped, q is the probability that a link between nodes is cracked, n is the number of nodes in a cluster.

In ILCCPDA algorithm, if eavesdroppers want to steal the data of node s, they must know the data of the two slices from node s and the message from neighbor nodes. Therefore, the eavesdropper must break the link between node s and neighbor nodes acquired by information fragmentation of node s, and the link between the neighbor nodes and the cluster nodes. Then the probability that the data nodes are exposed is

$$p(q) = q^2 \times \sum_{k=0}^{n-1} p(in = k) q^k \tag{3}$$

q^2 represents the probability that the data which the two neighbors received from node s is stolen. $p(in = k) = k / N$ represents the probability that k nodes send message to node s. $\sum_{k=0}^{n-1} p(in = k) q^k$ represents the probability that all the transmitted information is Eavesdropped.

For CPDA, and its privacy protection is related to the probability P that a node randomly selected itself as the cluster. When p is smaller, the number of nodes in a cluster is greater, the probability of the message of the data is exposed is smaller and decreases exponentially. In this paper, performance of CPDA will be compared, while the p=0.25.

The privacy protection of ICPDA is the same with CPDA, for they take the same method to protect data.

Each node of iPDA points out 2L-1 pieces of data, and the number of pieces they received is unknown. Attackers at least need to break the communication links of the 2L-1 pieces of data and the links where they received them. Then they can get the original data of this node. So we can get

$$p(q) = q^{2L-1} * \sum_{k=0}^{k_{\max}} p(\mathrm{Rec}J = k) * q^k \tag{4}$$

Where $k_{\max}$ is the maximum number of slice that nodes have received. $P(\mathrm{Rec}J = k)$ is the probability that the number of slices that the node receives is k. Let x nodes received k slices in total, then $P(\mathrm{Rec}J = k) = x/N$.

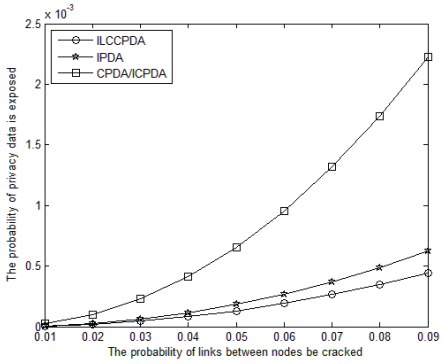


Fig.3 Privacy contrast

Figure 3 shows that privacy protection of our algorithm is a little better than other algorithms in theory.

B. Data Traffic

For iCPDA, the information will be exchanged three times after the fusion trees are established. At first，the cluster head node need to broadcast seed to the cluster members. Then each node hides their sensor data by receiving the seed, then exchange the exchanged pseudo data in the cluster. Finally, each node integrates the received pseudo data with its own data, and sends the results to the cluster head node. After received data is integrated, cluster head node will broadcast it within the cluster and send it to clusters head node at a higher level or a check terminal. Assuming a cluster contains three nodes (two member nodes and a cluster head node), and each member node needs to send two packets: a joint pseudo data packet and a fusion packet. Cluster head node needs to transmit five packets（a public seed package, a pseudo data packet, a fusion packet, a fusion packet that broadcasts to the neighboring node within the cluster and a fusion packet of the cluster head at a lower level）, in order to ensure data integrity by monitoring. Therefore, the average data communication

overhead is $O((2+3P_c)N)$, p_c is the probability that the cluster nodes choose themselves to be cluster heads. Assuming that the cluster nodes choose themselves to be cluster heads, then p_c=1, and data traffic of iCPDA is $S_{iCPDA}=5n$.

For iPDA algorithm, each node needs fragment its own data, and fragmentation number is taken as L. When two fusion trees are built, each node sends 2L-1 packets and uploads a converged packet in the integration phase. So data traffic of this algorithm is $2L\times n$, Since L is at least 2, so the minimum traffic is $S_{iPDA}=4n$.

For CPDA algorithm, the communication of data fusion in the cluster includes: n nodes broadcast seed, n nodes send encryption processing information to other n-1 nodes, and n-1 node members except the cluster head node sending the collected information to the cluster head node. Therefore, data traffic of all nodes within a cluster in CPDA is $S_{CPDA}=n+n\times(n-1)+n-1=n^2+n-1$.

For ILCCPDA algorithm, the cluster integration phases include: n nodes transmit information to two nodes, and n-1 node members except the cluster head node will sent the collected information to the cluster head node. Therefore, the data traffic of all the nodes in a cluster is $S_{ILCCPDA}=2n+(n-1)=3n-1$.

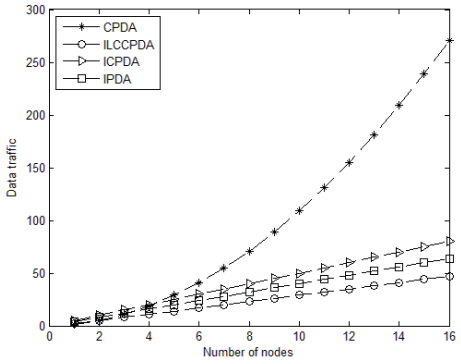


Fig.4 Data traffic contrast

Figure 4 shows that the data traffic of our algorithm is lower than other algorithms.

C. Integrity Protection

iPDA verifies the integrity by comparing the fusion results of the two fusion trees, iCPDA ensures the integrity by adding the listener nodes. ILCCPDA verifies the integrity of the data on basis of the MAC algorithm. The following analysis shows that the detection scope of ILCCPDA is more extensive.

Active attacks include data replay attacks, forged packet attacks, data tampering attacks and Camouflage node attack. We analyze the integrity of iPDA, iCPDA, and ILCCPDA from the perspective of the following ways.

1) Data replay attacks.

When the attacker attacks, for iPDA, distributions of red and blue fusion tree between this round and the last one will be different. So the replay data will change the fusion result of a single tree, leading to two different fusion results, then detecting destroyed integrity. For iCPDA, At playback, a monitor node in the cluster will find the difference with the data of the last round, and the destruction of integrity will be detected. For this algorithm, if the attack is a replay attack in slicing stage, MAC value will be changed, for the data itself does not minus node fragmentation. If an attacker starts replay data attacks in the integration phase, MAC values obtained in the last round will be obviously incorrect and illegal MAC will be detected in the base station node, for the real key to MAC of each node is different with the one of the last round in the mixing slicing stage.

2) Forged packets attack

When the attacker is forging a data packet, it may not be detected by the base station node, for iPDA, on condition that only the red fusion tree and the blue one simultaneously forge equal packets. But the red fusion tree and the blue one of each round are different, the possibility is extremely low. For iCPDA, forged packets may not be detected, on condition that monitoring data for all nodes within a cluster and base station data have become forged packets, so the possibility is also extremely low. In this algorithm, the attacker must forge the corresponding MAC to forge a packet, , otherwise it will be detected by the base station node. Meanwhile, the MAC key stored in each node is not the real key value. Even if all of nodes outside the base stations are captured, the attacker can not know the real key and can not forge the correct MAC.

3) Data tampering attacks

For such attacks, iPDA's protection is very weak. It can be detected that only fusion results of a single tree are tampered. But it cannot be detected that node data has been tampered before being sent. iCPDA node can detect the destruction of the integrity by monitorring the difference between node data and the base station. In this algorithm, it can be detected that the data and the MAC value are changed at the same time, but the possibility is extremely low.

4) Camouflage node attack

There's no detailed consideration of such attacks in iPDA, iCPDA, where disguised nodes are considered impossible to obtain a secure communication link. In this paper, it shows that the communication link is a small probability of being cracked. And through breaking the communication link of a small area, you can disguise nodes. In this algorithm, the base station node has the K of each node, and the camouflage node doesn't maintain records in the base station node, the false data of camouflage node can't be received by the base station node.

V. CONCLUSION

This paper presents a new data fusion algorithm with privacy protection and integrity detection. It can reduce communication overhead through the simple cluster fusion ways, on the basis of CPDA. And integrity detection can be completed by adding homomorphic message authentication code. To locate the cracked node accurately is our future work.

REFERENCES

[1] Madden S, Franklin M J, Hellerstein J M.TAG: A tiny aggregation service for ad-hoc sensor networks//Proceedings of the 5th Symposium on Operating Systems Design and Implementation. New York,USA,2002:131-146
[2] He W, Liu X, Nguyen H, Nahrstedt K, Abdelzaher T. PDA: Privacy-preserving data aggregation in wireless sensor networks//Proceedings of the 26th IEEE International Conference on Computer Communications.Anchorage,AK,2007:2045-2053
[3] Yang Geng, WANG An-qi, Peter Chen, An energy-saving privacy-preserving data aggregation algorithm, Journal of Computers, 2011,34 (5);. 792-800
[4] Yang Geng, Li Sen, Chen Zheng-Yu and other," High- accuracy and privacy-preserving oriented data aggregation algorithm in sensor networks", Journal of Computers, 2013,36 (1);. 189-200
[5] Feng Yan-fen, Low energy-consuming cluster-based private data aggregation, Computer Application Research, 2013,30 (3);. 885-888
[6] Li Wei, Energy-saving data aggregation algorithm for protecting privacy and integrity ,Computer Applications, 2013,33 (9);. 2505-2510
[7] He W, Nguyen H, Liu X, Nahretedt K, Abdelzaher T. iPDA: An integrity-protecting private data aggregation scheme for wireless sensor networks//Proceedings of the Military Communications Conference. San Diego, CA,2008:1-7
[8] He W, Liu X, Nguyen H, Nahrstedt K. A cluster-based protocol to enforce integrity and preserve privacy in data aggregation//Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops.Montreal,QC,2009:14-19
[9] Zhou Qing, Integrity and privacy preserving data aggregation algorithm for WSNs, Research of Computers, 2013,30 (7);. 2100-2104
[10] Chen Wei, Yang Long, Yu le, Privacy-preserving dynamic integrity-verification algorithm in data aggregation, Computer Science,2013,40 (7); 84-88
[11] XU X H,WANG Q,CAO J N, et a l. Locating malicious nodes for data aggregation in wireless networks[C]//Proceedings of the 31st IEEE International Conference on Computer Communications. Piscataway, NJ:IEEE Press,2012:3056-3060.
[12] MINAMI K, LEE A J, et al. Secure aggregation in a publish-subscribe system[C]//Proceeding of the 7th ACM Workshop on Privacy in the Electronic Society. New York: ACM Press, 2008: 95-104.
[13] Eschenauer L and Gligor V D.A key-management scheme for distributed sensor network[C].Proceedings of the 9th ACM Conference on Computer and Communications Security，Washington,2002:41-47.
[14] HEINZELMAN W R, CHANDRAKASAN A, BALAKRISHMAN H. An application-specific protocol architecture for wireless microsensor networks[J].IEEE Trans on Wireless Communications, 2002,1(4):660-670.

Improved Parallel Randomized Quasi Monte Carlo Algorithm of Asian Option Pricing on MIC Architecture

Peng Hui Yao[1,3], Yong Hong Hu[2], Zhong Hua Lu[1], Yan Gang Wang[1], Jue Wang[1]
1. Super-computing Center, Computer Network Information Center, Chinese Academy of Sciences
2. School of Statistics and Mathematics, Central University of Finance and Economics
3. University of Chinese Academy of Sciences
Beijing, China
Email: yaoph@sccas.cn, ph_yao@hotmail.com

***Abstract*—High-dimensional Option pricing, which plays an important role in complex financial activities, presents a great computational challenge in practice. Randomized Quasi Monte Carlo (RQMC) algorithm is of practical significance for forecasting option prices or other finance derivatives. In this paper, we present an improved parallel RQMC algorithm to forecast Asian option prices using Many Integrated Core (MIC) architecture. The improved algorithm employs novel data structure, independent random generator, vectorization technology, and data alignment. Numerical experiments were conducted on MIC architecture and the parallel performance was then analyzed. A speedup of 1.37 was achieved on MIC over CPU. Efficiency of 70.85% was achieved by using 64 OpenMP threads of a MIC card. An average speedup of 3.38 can be obtained by mixing the CPU and MIC computation in comparison with a single core of the CPU. Ample evidences proved the RQMC algorithm can benefit enormously from MIC architecture.**

Keywords-Quasi Monte Carlo Algorithm; Asian Option Pricing; MIC Architecture; Parallel Simulation

I. INTRODUCTION

Option pricing is one of the most complex mathematic problems in computational finance. High dimensional option pricing, such as Asian and American multi assets, presents great a computational challenge in practice. Fisher Black and Myron Scholes introduced the first complete option pricing model, known as Black Scholes pricing model, using high dimensional Partial Differential Equations (PDE) in 1973. Closed form solutions for the model are hardly acquired except for some special situations. Cox, Ross and Rubinstein proposed binomial model to value American option in 1979 [1]. The complexity of numerical methods of these two models increases exponentially in solving high dimensional option pricing. Asian basket option is a kind of path dependent derivatives, whose payoff depends on the average of the underlying assets and is popular for hedging exchange risk. The research of high dimensional Asian basket option pricing problems is divided into two classes: analytical approximation approach and Monte Carlo simulation based method, where the latter one dominates the financial applications [6].

Monte Carlo (MC) simulations are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. Pheliem Boyle firstly applied MC simulation to option pricing in 1977 [2]. In 1996, Phelim Boyle, Mark Broadie, and Paul Glasserman attempted to estimate security price by Monte Carlo methods [3]. Quasi Monte Carlo methods use low discrepancy sequences to obtain deterministic method for MC. Randomized Quasi Monte Carlo method can be also regarded as a variance reduction technique for the standard Monte Carlo method [4].

Parallel computing is generally used in high dimensional option pricing. Sak, Ozekici and Boduroglu proposed a parallel finite difference algorithm for single asset Asian option pricing [5]. Kai Huang and Ruppa K. Thulasiram developed a parallel binomial tree method for pricing American style basket Asian options [10]. Hong Xu Chang, Zhong Hua Lu, and Xue Bin Chi presented the parallel simulation of high dimensional American option pricing on heterogeneous supercomputer DeepComp7000 using stochastic mesh method [11]. Yong Hong Hu and Da Qian Chen presented the parallel randomized quasi Monte Carlo simulation to settle Asian basket option pricing using MPI mechanism on supercomputer Deepcomp6800 [12]. Monte Carlo methods have become one of the most challenging computing applications due to its universality and practicability. However, the demand of higher computing power stimulates the more rapid and accurate parallel algorithm on HPC (High Performance Computing) platforms. Graphic Processing Unit (GPU), which always improves tremendously computing power of CPU severs, has exclusive programming language including CUDA and OpenCL. MIC (Many Integrated Core) architecture is announced by Intel in 2012. Different with complicated programming languages of GPU, MIC architecture supports C, C++, and FORTRAN, and general parallel programming interface like MPI and OpenMP.

The following paper is organized as follows: The next section introduces the background of the Asian basket option pricing model and Randomized Quasi Monte Carlo simulation. Section 3 presents the improved parallel Randomized Quasi Monte Carlo algorithm on MIC architecture. Evaluations and analysis of performance are given in Section 4. Finally, conclusions are drawn in Section 5.

II. BACKGROUND

A. Asian basket option pricing model

In Black Scholes framework, the stock price follows geometric Brownian motion. The option value at time t is defined by the following model [1], [2], [3]:

$$dS = \mu S dt + \sigma S dz,\ t \in [0, T] \tag{1}$$

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \tag{2}$$

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.33

In which S is the stock price, T is the maturity of the option, the constant parameters μ is the expected return, σ is the volatility of the underlying assets, r is the risk free interest rate, and dz is the standard Brown motion following normal distribution with mean zero and variance dt. As for an option on n underlying assets, the basket option value is generated in the following model:

$$\Sigma_{ij} = \begin{cases} \sigma_i^2 & if\ i=j \\ \sigma_i \sigma_j \rho_{ij} & if\ i \neq j \end{cases}$$

$$\boldsymbol{\mu}(i) = \mu_i - \sigma_i^2 / 2,\ i = 1, 2, \cdots, n$$

$$dS_i = \mu_i S_i dt + \sigma_i S_i dz_i\ ,\ i = 1, 2, \cdots, n \quad (3)$$

$$\frac{\partial f}{\partial t} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \Sigma_{ij} S_i S_j \frac{\partial^2 f}{\partial S_i \partial S_j} + r \sum_{i=1}^{n} S_i \frac{\partial f}{\partial S_i} = rf \quad (4)$$

In which ρ_{ij} is the correlation coefficients between dz_i and dz_j, $\boldsymbol{\mu}$ is the mean vector, $\boldsymbol{\Sigma}$ is the covariance matrix that has the Cholesky decomposition $\boldsymbol{\Sigma}=\mathbf{C}\mathbf{C}^{\mathrm{T}}$. The asset dynamics (3) can also be written in the form $d\mathbf{S} / \mathbf{S} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The payoff function of the Asian basket call option on n assets with m observation times $P(\mathbf{S}, T)$ is

$$P(\mathbf{S}, T) = \max \left\{ 0,\ \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} S_i(t_j) - X \right\} \quad (5)$$

In which $\mathbf{S} = \left(S_1(t), S_2(t), \cdots, S_n(t)\right)$, $S_i(t_j)$ is the price of asset i at time t_j and X is the strike price of the basket option. Using martingale theory, Asian basket option pricing problem can also be written in the following form [12]:

$$f(\mathbf{S}, t) = \mathbf{E}\left\{ e^{-r(T-t)} P(\mathbf{S}, T) \right\}, \quad (6)$$

In which, $\mathbf{E}(.)$ is the conditional expectation under risk neutral probability measure and $e^{-r(T-t)}$ the discount factor. With this high dimensional integral, we can utilize Monte Carlo simulation based method to solve this intractable problem effectively.

B. Randomized Quasi Monte Carlo Simulation

In the classical Black Scholes framework, high dimensional option pricing problems are always modeled in the structure of high dimensional Partial Differential Equations (PDE). Monte Carlo simulation provides a powerful tool for solving these challenging option pricing problems [4-6]. Since MC is based on sequences of pseudo random numbers, which has a low rate of convergence of $O(N^{-0.5})$. Quasi Monte Carlo has a faster rate of convergence as $O(1/N)$, which uses a low discrepancy sequence such as the Sobol sequence, or the Faure sequence. While Quasi Monte Carlo method is a deterministic algorithm, the reliable error is hard to estimate. In order to analyze and estimate the variance, Randomized Quasi Monte Carlo (RQMC) uses randomized quasi random sequences for further variance reduction [6-8].

Using Quasi Monte Carlo simulation for option pricing, we can simulate the stock price dynamics by the formula

$$S_i(t_j) = S_i(0) \exp\left\{ (\mu_i - \sigma_i^2 / 2) t_j + \sigma_i Z_{i,j} \right\} \quad (7)$$

In which $\mathbf{Z_i} = \left(Z_{1,i}, Z_{2,i}, \cdots, Z_{n,i}\right)^{\mathrm{T}} = \mathbf{C} \cdot \mathbf{NR_i}$ and $\mathbf{NR_i}$ follows the multi variable standard normal distribution $N(\mathbf{0}, \mathbf{I})$. Suppose $g(u)$ is the discounted payoff function with the asset prices replaced by (7), k – dimensional random variable U follows uniform distribution $U(\mathbf{0}, \mathbf{I})$ and N is the number of simulated paths, Asian basket option value can be approximated by Monte Carlo simulation method in the form:

$$f(\mathbf{S}, t) = \mu = \int_{(0,1)^k} g(u) du = \mathbf{E}[g(U)] \approx \frac{1}{N} \sum_{i=1}^{N} g(U_i) \quad (8)$$

III. IMPROVED PARALLEL RANDOMIZED QUASI MONTE CARLO ALGORITHM

The RQMC algorithm uses a highly uniformly point set to randomize the low-discrepancy point set $P_n = \{\mathbf{u_i}\}_{i=1}^{n}$, so that each individual sequence follows the uniform distribution over $(0,1)^m$ (where m is the dimension of the random sequences) and these points are more evenly distributed. That is, let $\mathbf{v}$ be the uniform random vector in the m -dimensional space Ω, with carefully chosen randomization function $h: \Omega \times (0,1)^m \to (0,1)^m$, one can construct the randomized version $\tilde{P}_n = \{\tilde{\mathbf{u}}_\mathbf{i}\}_{i=1}^{n}$ of P_n, where $\tilde{\mathbf{u}}_\mathbf{i} = h(\mathbf{u_i}, \mathbf{v})$.

A. Parallel Algorithm

According to the above pricing model and the RQMC algorithm, the improved parallel algorithm is showed in Fig. 1. Several preparative operations, such as initializing correlation matrix and the Cholesky decomposition, are performed before the main computation starts. Correlation matrix is initialized in two-dimensional array by using loops of automation vectorization statement "#pragma ivdep". When decomposing correlation matrix to triangular matrix by Cholesky method, we use OpenMP statement "#pragma omp parallel" for external loop and automation vectorization statement "#pragma ivdep" for internal loop. The parallel segment, as shown in grey area in Fig. 1, consists of four main functions: (1) Generate random vector, which is quasi number and in accordance with normal distribution. (2) Matrix multiplication. (3) Estimate prices, which is the result of payoff function. (4) Calculate discounted payoff. The last segment is averaging the discounted payoffs under risk neutral probability measure, and estimated error.

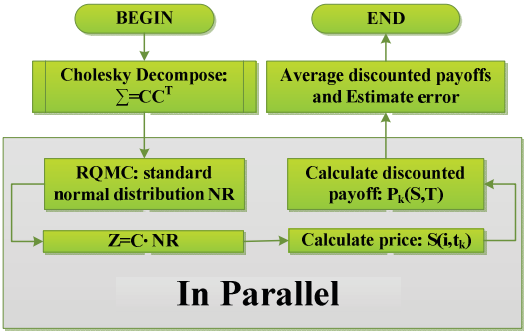


Figure 1. Parallel RQMC algorithm

The parallel segment, which is included by loop, uses OpenMP multiple threads to process four main functions and reduces option prices. The function of generating quasi random normal distribution vector consists of random function and Moro's algorithm. The latter obtains the inversion from U[0,1] to normal distribution. We create the random function by using private seed number to avoid data dependency. The function of matrix multiplication $C_m = A_{m\times m} \times B_m$ adopts sparse matrix optimization algorithm and automation vectorization statement "#pragma ivdep" to get asset prices. The payoff function of the Asian basket call option on n assets with m observation times $P(\mathbf{S}, T)$ is calculated by previous form. Then we filter discount payoff of significance. In above functions, several the procedure sequences are adjusted and declared by vectorization statement to meet vectorization and high parallel efficiency.

Using MIC co-processor as accelerator, the initialization work and control program are done by CPU. At the beginning, CPU will complete some works including constructing correlation matrix and Cholesky decomposition. Then the CPUs offload the packaged data and parameters to the MIC card. The MIC undertakes computing tasks of generating quasi random vector, matrix multiplication, and calculating option price. When computing times is enough for the given threshold, the MIC co-processor translates the final option value to CPU. At last, CPU computes the expectation of option value and estimates the variance to ensure appropriate error. Because the CPU is also responsible for a part of the computational workload, we employed static load balancing to improving performance. The task volume assigned on MIC/CPU is based on the time ratio of running Monte Carlo benchmark program.

B. *MIC Architecture*

The recent rapid ascent of Tianhe-2 to number one on the Top 500 list has now called great attention to the potential of Intel Xeon Phi, which form the backbone of the Tianhe-2 system. While the CPU+GPU based Tianhe-1 is only No. 10 in this Top 500 list. Although there's still powerful computation capability in GPU, the difficulty of programming on GPU is much more than on MIC.

The Intel MIC (Many Integrated Core) is a co-processor attached to CPU host via PCI-E bus, designed for highly parallel and vectorizing codes, using many small simple cores with wide vector units. Each core includes a VPU (Vector Processing Unit), which contains 32 512-bit vector registers. Each core has a 32 KB L1 cache and 512 KB L2 cache. Fig. 2 presents a brief overview of MIC architecture.

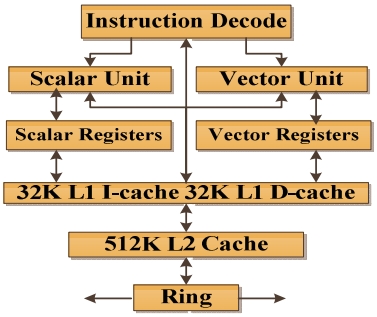


Figure 2. MIC Architecture overview

MIC architecture has two common models to execute programs: native and offload model. Under the native model, MIC co-processor serves as CPU in an independent host, compiling with -mmic and running standard C, C++, and FORTRAN source code. Under the offload model, the code begins on CPU host and parts of the code marked by offload order are accordingly offloaded to the MIC co-processor. When the data and code are transferred via PCI-E bus, MIC co-processor serves as an accelerator to CPU. In this paper, offload model is adopted for the large scale of the option pricing model.

CPU and MIC own memory space respectively and usually can be seen as independent servers. So we employ MPI+OpenMP programming model. Message Passing Interface (MPI) system is used for managing communication and computing between servers. OpenMP supports shared memory parallel programming and is used allocating multiple threads inside CPU and MIC.

C. *Optimization*

Because the basic structure of MIC cores is based on Intel Pentium architecture, there is no branch predictor, no out-of-order execution, and no memory management unit. Thus the computation capability of each core on MIC is no better than Sandy bridge core. But MIC has more computation units than Sandy Bridge, which can bring better parallelism in large scale computation. The 512 bit width SIMD instruction set is also wider than Sandy Bridge (256bit). Depending on the differences, we have explored some novel technologies.

1) Static Data Structure: Dynamic data structure, such as 'vector' array in C++ programming, allocates memory when actual data object is called in function or procedure. If the parallel procedure calls specific function for too many times, dynamic objects are frequently allocated and freed. It always wastes great proportion of total running time. In our defined data structure, the lengths of big array are limited in small range. In our parallel algorithm, the generated random vector and stock price matrix are stored in this data structure.

2) Independent random generator: Almost all current random number generators are using shared seed number to produce pseudo random number. In serial programing model, the generator works well. But in concurrent programming model, the performace is not better than the serial one. The reason is that the generator uses a shared

seed to produce each random number at a time, which cause serious seed number dependence. Our customized independent random generator adopts private seed number to produce next random number, which can eliminate the dependency between consecutive numbers. When generating quasi random vector in parallel loops, we adopt the customized random generator.

3) Vectorization technology: Vectorizing loops can lead to significant performance gains without programmer intervention, especially on large data sets. Because of the 512 bit width SIMD instruction set on MIC, MIC core can execute more loop operations than Sandy Bridge. Eliminating data dependence is the most important method for improving vectorization degree. Because there are plenty of matrix multiplication inside loops, we adjust the matrix arrays to meet vectorization requirement. According to the vectorization report, the degree of vectorization of our algorithm reaches 95%.

4) Data Alignment. Aligned data contributes to make full use of Vector Processing Unit (VPU). For example, double precision data is 8 bytes, equaling to 64 bits. So it is suitable for each thread to process 8 double precision data every time. MIC have to access the data twice for reading a little longer than 8 data of double presision. In this situation, performance gives a discount. So the data should be aligned by 64 bits to ensure the data access efficiency. We optimized the matrix multiplication and discount payoff function in paralle parts by adusting data type and length of arrays.

IV. EVALUATION AND ANALYSIS

In this section, we discuss the experimental results of implementation on CPU+MIC heterogeneous system. The experimental system includes one Intel Genuine CPU and two Xeon Phi SE10P MIC co-processors.

The peak performance of the CPUs in the machine is 332.8GFlops, while the MIC card has totally 1.073TFlops peak performance and maximum 5.5GB/s memory bandwidth. MPSS driver kit is installed for MIC co-processors. When compiling, we use "–vec-report3" to check and insure that the critical loops are vectorized. And VTune is used to analysis hotshots, vectorized instructions and the computation time distribution. The detailed parameters of CPU and MIC are listed in Table I.

TABLE I. SYSTEM PARAMETERS OF "XEON" AND "MIC"

Parameter	Xeon E5-2650	Xeon Phi SE10P
Operating System	Red Hat Enterprise Linux Server 6.3	Linux based uos
Linux Kernel	2.6.32.279.el6	2.6.38.8.g9b2c036
Complier	Intel C/C++ Compiler 13.1.1	
MPI Library	Intel MPI 4.1.0.030	
Processor	Intel Xeon E5-2670, 8 cores, 2.60Ghz	Intel MIC 61cores 1.1GHz

A. Problem Model and Accuracy Evaluation

Consider pricing the Asian basket call option underlying 15 stocks in the period of 50 days. To simplify, we assume that the initial stock price $S_i(0)$, μ_i and σ_i for all the underlying assets are the same ($i = 1, 2, \cdots, 15$), the correlation ρ_{ij} among two assets is same $(i, j = 1, 2, \cdots, 15)$, the risk free interest rate r is constant, and the observed time m is 50 days. The parameters are as follows:

$$S_i(0) = 100,\ X = 100,\ r = \mu_i = 0.05,\ \sigma_i = 0.2,$$

$$n = 15,\ m = 50,\ T = 1.0,\ \rho_{ij} = 0.1,\ i = 1, 2, \cdots, 15.$$

In order to analyze the impact caused by different numbers of sample paths, the option value is computed with 5×10^3、10^4、5×10^4、10^5、5×10^5、10^6、5×10^6 and 10^7 simulated paths. The estimated error is measured by half the length of the confidence interval at 95% level of the option price. Assuming $x_i (i=1,2,\cdots,N)$ is estimated option value and N is the number of MC simulation, we can define the sample mean value and the sample standard deviation in follow formulas:

$$\bar{x} = \sum_{i=1}^{n} x_i / N,\quad s = \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 / (N-1)}.$$

Because the significance level is $\alpha = 0.05$, the estimated error is $1.96s/\sqrt{N}$. As elaborated in Fig. 3, with the increasing simulation times, the option values are more stable and estimated errors are less. We also find that the threshold of valid simulation number is 5×10^4. The precision of estimated error reaches to 0.0023 after 10^7 simulation.

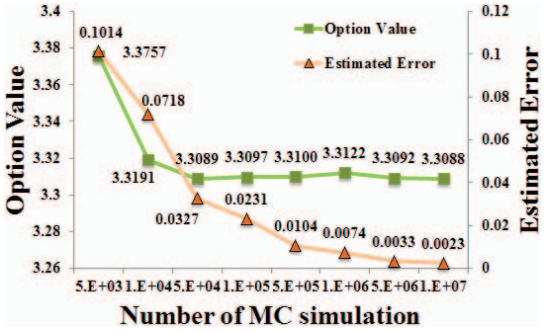


Figure 3. Estimated option values and estimated errors

TABLE II. OPTIMIZATION ANALYSIS

	MIC before Optimization			MIC after Optimization			CPU after Optimization		
Threads	Time(Second)	Speedup	Efficiency	Time(Second)	speedup	Efficiency	Time(Second)	Speedup	Efficiency
1	2398.38	1.00	100.00%	168.50	1.00	100.00%	27.85	1.00	100.00%
2	1780.48	1.35	67.35%	88.74	1.90	94.94%	13.99	1.99	99.51%
4	1118.15	2.14	53.62%	44.24	3.81	95.21%	7.12	3.91	97.77%
8	675.39	3.55	44.39%	22.18	7.60	94.95%	3.86	7.21	90.12%
16	989.58	2.42	15.15%	11.54	14.60	91.25%	2.34	11.89	74.34%
32	950.81	2.52	7.88%	6.54	25.75	80.47%	2.08	13.35	41.73%
64	1118.35	2.14	3.35%	3.71	45.34	70.85%	2.16	12.84	20.06%
128	1534.26	1.56	1.22%	2.24	75.10	58.67%	2.78	10.01	7.82%

B. *Optimization Evaluation*

We have tested the performance of the algorithm on CPU and MIC. In contrast, we set same parameters in Table II, which are 100, 000 simulation, 15 stocks and 50 periods. The speedup is defined as running time of serial algorithm dividing that of multiple threads. The parallel efficiency is defined as speedup dividing the number of threads. Without the optimization technologies, we see that the fast runtime on MIC is 675 seconds with 8 threads and the maximum speedup is 3.55. The parallel efficiencies are almost below 70%. After optimization on MIC, the speedup and the parallel efficiency increase distinctly. We even got the fastest time is 1.50 with 180 threads. But the efficiency become low, which means that many cores are not made full use of. The result of MIC indicates that 128-180 threads on a MIC would be better. From the results on CPU, the optimization is effective. Under running 32 threads, the fastest running time is 2.08 seconds and the maximum speedup is 13.35. To make use of every core, the better running threads are 8. But for faster running time, 32 threads are recommended.

C. *Evaluation with CPU system*

In the numerical experiment, we assumed that the CPU system is using CPU cores only and MIC system is in native model. The hybrid system, which has one CPU and two MIC cards, is in offload model. The speedup of CPU is always 1 as a baseline. The speedup of MIC is defined as the running time of CPU dividing that of MIC. The speedup of CPU+2MIC is defined as running time of the CPU dividing that of the hybrid system. As shown in Fig.4, the average speedup of MIC is 1.37 and the average speedup of CPU+2MIC is 3.38. The polyline of MIC illustrates that the performance of MIC is better than that of CPU. And the polyline of CPU+2MIC showed the same effect. Fig. 4 also shows that the proposed algorithm has excellent universality and transportability.

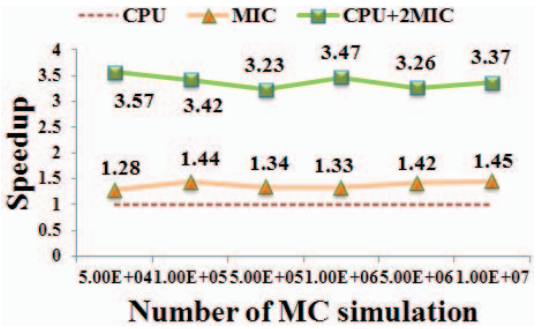


Figure 4. Estimated option values and estimated errors

V. CONCLUSIONS

The parallel algorithm of Randomized Quasi Monte Carlo methods for Asian basket option pricing based on MIC architecture is presented in this paper. Some novel optimization technologies contribute to the performance of the algorithm. We can see that the Intel Xeon Phi device can get pretty good performance, which is much better than CPU. We also implemented the program with hybrid system (CPU+2MIC), which using adequate MPI processes with some OpenMP threads together. The parallel algorithm has shown good scalability and high performance so that it can extend to deal with most kinds of high dimensional derivatives pricing. In the future, we will study more complex algorithm and simulation of computational finance problem, such as portfolio selection model.

ACKNOWLEDGMENT

This research has been financed by the National Natural Science Foundation of China grant 61272193.

REFERENCES

[1] John C. Cox, Stephen A. Ross, and Mark Rubinstein. Option pricing: A simplified approach. Journal of Financial Economics, Volume 7, Issue 3, September 1979, Pages 229-263.

[2] Phelim P. Boyle. Options: A Monte Carlo Approach. Journal of Financial Economics, Volume 4, Issue 3, May 1977, Pages 323-338.

[3] Phelim Boyle, Mark Broadie, and Paul Glasserman. Monte Carlo methods for security pricing. Journal of Economic Dynamics and Control, Volume 21, Issues 8-9, 29 June 1997, Pages 1267-1321.

[4] Russel E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. Acta Numerica, Volume 7, January 1998, Pages 1-49.

[5] Halis Sak, Suleyman Ozekici, Ilkay Boduroglu. Boduroglu. Parallel Computing in Asian Option Pricing. Parallel Computing, Volume 33, Issue 2, March 2007, Pages 92-108.

[6] Paul Glasserman. Monte Carlo Methods in Financial Engineering. Springer-Verlag, New York, 2004.

[7] John C. Hull. Options, Futures, and Other Derivative Securities, 7th edition. Prentice Hall, 2008.

[8] Pierre L'Ecuyer. Quasi-Monte Carlo Methods with Applications in Finance. Finance and Stochastics, Volume 13, Issue 3, September 2009, Pages 307-349.

[9] Christiane Lemieux. Randomized Quasi-Monte Carlo: a tool for improving the efficiency of simulations in finance. Proceedings of the 36th conference on winter simulation (WSC '04), 2004, Pages 1565-1573.

[10] Kai Huang and Ruppa K. Thulasiram. Parallel Algorithm for Pricing American Asian Options with Multi-Dimensional Assets. In 19th International Symposium on High Performance Computing Systems and Applications (HPCS'05), 2005, Pages 177-185.

[11] Hong Xu Chang, Zhong Hua Lu, and Xue Bin Chi. Large-scale Parallel Simulation of High-dimensional American Option Pricing. Journal of Algorithms & Computational Technology, Volume 6, Number 1, March 2012, Pages 1-16.

[12] Yong Hong Hu and Da Qian Chen. Parallel Randomized Quasi-Monte Carlo Simulation for Asian Basket Option Pricing. Journal of Algorithms & Computational Technology, Volume 6, Number 1, March 2012, Pages 101-112.

Beam-tracing domain decomposition method for urban acoustic pollution

Guillaume Gbikpi-Benissan, Frédéric Magoulès
Ecole Centrale Paris, France
Email: frederic.magoules@hotmail.com

***Abstract*—This paper covers the fast solution of large acoustic problems on low-resources parallel platforms. A domain decomposition method is coupled with a dynamic load balancing scheme to efficiently accelerate a geometrical acoustic method. The geometrical method studied implements a beam-tracing method where intersections are handled as in a ray-tracing method. Beyond the distribution of the global processing upon multiple sub-domains, a second parallelization level is operated by means of multi-threading and shared memory mechanisms.**

Numerical experiments show that this method allows to handle large scale open domains for parallel computing purposes on few machines. Urban acoustic pollution arrising from car traffic was simulated on a large model of the Shinjuku district of Tokyo, Japan. The good speed-up results illustrate the performance of this new domain decomposition method.

***Keywords*-Domain decomposition methods; Parallel and distributed computing; Acoustics; Ray-tracing methods; Beam-tracing methods;**

I. Introduction

Important material resources is often a requirement, either for the processing of large volumes of data, or for the fast resolution of large problems. Especially in parallel computing, the effectiveness of the designed algorithms is related to their scaling behavior on massively parallel platforms. However, for a great part of the scientific community, there remains a need for big simulations with very low resources. Then, it could be of major interest to design algorithms with good acceleration properties on very small clusters of machines.

In this study, we tackle the simulation of car traffic noise level at a whole district scale. In a lot of countries, environmental noise has become a major source of stress and of various diseases like hypertension or heart ischaemia. Fast noise simulation could be an effective solution to assess the effect of architectural configurations, in order to reduce the acoustic pollution or to maintain a maximum noise level. Given that sound propagates far away in open domains, a large scale model is required, and well designed methods are a key point to obtain fast simulations.

The more physically correct algorithms are based on numerical methods, such as the Boundary Element Method [1], the Infinite Element Method [2], [3], the Finite Element Method [4], [5], the Stabilized Finite Element [6] and the coupling methods [7]. Even if these methods accurately approximate the mathematical equations of the acoustic problems, they unfortunately need a lot of computational power and memory. Hence, they are not good candidates for our context of low resources. Furthermore, relatively to the wavelength, these methods have some difficulties to handle extremely large domains and open domains, such as ones involved in exterior acoustic problems.

Contrariwise, geometrical methods, such as the image-source method [8], [9], the ray-tracing method [10] and the beam-tracing method [11], [12], are well suited for large domains and open domains. But only simulations of high frequencies, relatively to the size of the problem, provide accurate results, due to the fact that, in such methods, the sound propagates in straight lines. Fortunately, this is not a limitation for our car traffic noise simulation.

In this study, we consider geometrical methods for the fast simulation of urban acoustic pollution within a large open area. After a brief overview of geometrical methods, we describe a hybrid method coupling beam-tracing and ray-tracing methods, with some similarities to frustum-tracing [13]. Then we present the parallelization of the algorithm by means of techniques from domain decomposition methods [14], widely used for numerical methods in acoustics [15]. Finally we illustrate the efficiency of our parallelization technique, using a few machines for simulations run on a model of the shinjuku district of Tokyo, Japan.

II. Hybrid geometrical acoustic method

Commonly, geometrical methods compute multiple paths followed by the sound emitted from a source, and measure the acoustic pressure at some points called microphones. The image-source method [8], [9] creates virtual sources each time a sound ray reflects on a surface inside the 3D model. Sound pressure level is evaluated at a microphone by adding the contribution of the sources and virtual sources for which there is no obstacle on the direct path toward that microphone.

The ray-tracing method [10] divides the energy of each source between a huge number of elementary particles (similar to their physical counterpart, the photons) which propagate as sound rays. The energy of a particle gradually decline due to air damping, and it also looses energy when it reflects on a surface. Once this energy reaches a threshold low value, the particle is deleted.

The beam-tracing method is based on a similar principle, except that sound rays are replaced by sound beams, then the source energy is split between beams, according to its power in each direction. This method is harder to implement, mainly because intersections between beams and surfaces are more complex to detect and to handle.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.34

For example, each time a beam partially hits one single surface, it needs to be split into sub-beams, what can dramatically increase the complexity of the computation, due to the recursion of this principle. However, for the same simulation quality, there is much less beams to launch, relatively to ray-tracing.

Ray-tracing methods have been extensively studied in the context of computer generated imagery. Efficient generation of hierarchical partitioning of good quality [16], [17] for the handling of intersection detection, the use of vectorial instructions [18], [17], and graphics processing unit (GPU) [19], [20] allowed to accelerate the ray-tracing. Large scale models got some attention too [18], [21], [22]. However, these optimized methods are very efficient when the rays follow similar trajectories, which is rarely the case in acoustic analysis. On the other hand, even if beam-tracing generates great quality soft shadows and anti-aliasing in computer graphics [23], the additional complexity and slowdown do not worth it.

The hybrid approach considered here applies beam-tracing principles, except that the intersections between beams and objects surface are handled as in ray-tracing method. In other words, instead of testing the whole volume of the beam against the model, only the guiding ray of the beam is tested. Hence, in this method, the beams are never split on reflection. This trade-off between precision and speed allows to use very efficient ray shooting methods. In practice, the drawback due to the approximation error can sufficiently be mitigated by processing very small beams. Moreover, a beam becomes larger far from the energy source, where its energy is significantly depleted, what would reduce the impact of an approximation error.

III. Parallel computing

Advantage of domain decomposition: In a standard implementation, the input and the output of parallel computing are sequential. According to the Amdahl's law, the global speed-up of a parallel program containing a percentage f of sequential part is bounded by $1/(1-f)$. One of the main goals of the Domain Decomposition Methods (DDM) is to allow the whole program to be parallel. The idea is to split the model into sub-domains, such that the loading, output and result gathering could be done in parallel for each sub-domain. The splitting of the model can be done only once for multiple simulations.

Another important advantage of the DDM is to allow to consider very large models and a great number of microphones. This is particularly important when generating volumetric noise map, since the number of microphones quickly becomes very large. Each process only needs to allocate memory for the sub-domains it is working on instead of allocating memory for the whole domain.

Microphones partitioning: In this kind of decomposition, only the set of microphones is split, while the complete geometry is replicated. Each sub-domain is totally independent so the program can be run in parallel without synchronization. Even if a whole part of the work is replicated, especially the intersections detection and the loading/preprocessing of the model, such a decomposition can be efficient for a simple geometry model populated with a huge set of microphones. The absence of communication between the processes is a major advantage in the context of a largely distributed or loosely connected platform. However, there is no easy way to mitigate load balancing issues when the processing time of sub-domains varies a lot.

On demand geometry and microphones loading: With prior computation of some hierarchical acceleration structure, sub-domains can be loaded only when needed, but the actual parallelization is done on the set of beams. At the beginning, first level of the hierarchy is loaded, and throughout the computation, this structure is accessed several times. If during the traversal of the structure, a node not yet loaded is reached, the corresponding data are loaded. If it is an interior node, these data are the child nodes information, if it is a leaf, the data are the corresponding mesh and microphone data. Beside the specific precomputed acceleration structure, this model of parallelization is quite hard to implement. The latency for loading a node of the hierarchical structure should be hidden, for instance by changing the current beam to one whose data are ready. The gathering of the final results is also complicated since the data are spread over all processes, and potential copies need to be merged. The main drawback is the risk for each process to load the complete model when beams largely spread inside the area. That is particularly the case in exterior acoustics problems where beams often go far and are widely dispersed.

Geometry and microphones partitioning: The original method used in this work matches more closely a domain decomposition approach [14]. Here, both the geometry and the microphones are split into multiple sub-domains. When a beam goes out of a sub-domain, it is sent to the process working on the sub-domain intersected. Interface conditions [24] are used to assure the continuity of the beam properties from one sub-domain to another one. Efficient interface conditions can be designed by a continuous approach [25], [26], [27]. Continuous optimized approaches improve the convergence of the algorithm [28], [29], [30], [31], [32], [29]. Similarly, the performance of the algorithm can be increased by using a discrete optimized approach [33], [34], [35], [36], [37], [38]. In [39], a link is established between these two approaches. In this paper we consider a domain decomposition approach as described in [40], [41], [42] but extended to beam-tracing [43]. In contrast, unlike classical domain decomposition methods, load-balancing issues can not be fixed in a static way, where a one-to-one correspondence is done between the processes and the sub-domains. Indeed, considering for instance the case where there would be only one sound source, the sub-domain containing the source would have the most computational load. This is why a more complex load balancing scheme has to be used. The idea is that each process starts with one sub-domain, but when it has few remaining beams

to handle, it starts loading one or more sub-domains still containing a lot of unhandled beams. The number of sub-domains which can be loaded simultaneously is limited by the memory size. In order to overlap the results gathering time, sub-domains can also be unloaded when few beams remain to be processed. Since the gathering operation mainly uses the communication system, this overlapping does not slow down the computation. Nonetheless, the loading and unloading of sub-domains take a long time, hence it would be preferable to avoid such operations as much as possible. For that, it is still relevant to approximate at best an average good load balancing. The solution used is to process first the exchanged beams and then to equalize the number of remaining beams in each sub-domain comparatively to the processing power affected to this sub-domain by loading or unloading sub-domains only when this ratio deviates too much from the average.

At last, the processing of a sub-domain itself is parallelized by coupling a shared memory multi-threading and a work-sharing load balancing. This allows an additional acceleration of the computation without any data replication. There is no need of gathering local outputs at the end since the output data are shared. However, it required to implement a thread-safe access mechanism. A simple fine-grained locking were implemented by setting a spin-lock mutex on each cell of the output array. Given that the multi-threaded beam-tracing generates few contention, the busy-waiting time is insignificant, hence the spin-lock is an acceptable trade-off between simplicity and efficiency for our acoustic problem.

IV. Results and discussions

In this study, we tackled the simulation of urban acoustic pollution related to car traffic. Experiments were run on a model of the Shinjuku district of Tokyo, Japan. It is an area of 2.5 km by 1.5 km around the point of coordinates $35^o41'23''N$, $139^o1'25''E$. By hosting the busiest train station of the world, multiple commercial and administrative buildings including the administration center of the government of Tokyo, and around ten skyscrapers at least 200 meters high, the Shinjuku district faces a large number of activities generating a very important car traffic. Figure 1 shows a global view of the simulation model. The whole district buildings has been represented, and as we can see on figures 2 and 3, it is a detailed designing leading to a quite large model.

As we were saying at the beginning, our goal is to study the efficiency of the proposed DDM-based hybrid geometrical method on low-computer resources architectures. The numerical experiments were run on a hybrid, both distributed and shared memory, computational platform consisting of 8 workstations, each one containing two quad core processors (a total of 64 cores). Each machine was provided with 8 Gigabytes RAM (Random Access Memory) and a Windows 7 operating system. The volume of the hexahedral bounding box of the model was populated with a total number of 13.7 millions of microphones, regularly distributed on a 3D grid. According to the dimensions

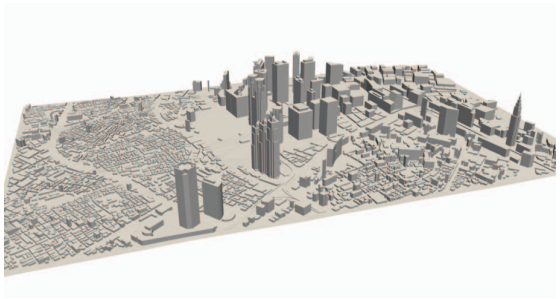


Figure 1. Global view of the virtual model of Shinjuku, district of Tokyo, Japan.

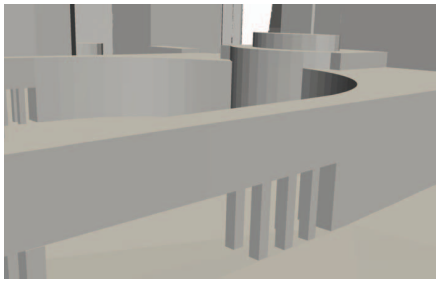


Figure 2. Close view of the virtual model of Shinjuku, just next to the Tokyo Metropolitan Government Building.

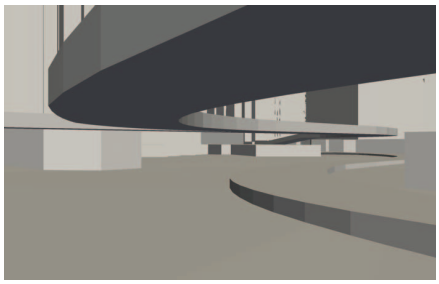


Figure 3. Close view of the virtual model of Shinjuku, under the Metropolitan Expressway No. 4.

of the district, that corresponds to a microphone every 4 meters in each of the three spatial directions. We simulated 49 sound sources, most of them placed at major crossroads, and each one launching 3 millions of beams. Figure 4 illustrates the noise level distribution obtained from such a simulation.

Speed-ups are computed using the total execution time, including the loading of the model, the acceleration structure generation and the saving of the results. A speed-up S is determined by $S = t_{ref}/t$, where t_{ref} is the execution time of the sequential version, and t, the time consumed by the parallel version. Table I shows different speed-ups relative to various DDM/multi-threading schemes. The computation was distributed such that each core runs exactly 2 threads, which corresponds to 16 threads per workstation. Then, for the 16, 32, 64 and 128 threads, we used respectively 1, 2, 4 and 8 workstations. Even if we are not in a scaling study, one could remark that

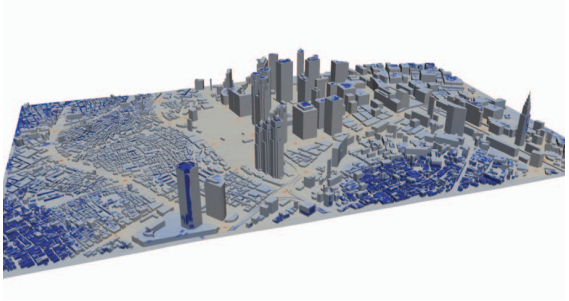


Figure 4. Simulation of a sound pressure level distribution in Shinjuku, district of Tokyo, Japan.

Table I
SPEED-UP OF THE ACOUSTIC DDM PROGRAM (ETHERNET) WITH RESPECT TO THE NUMBER OF THREADS AND SUB-DOMAINS.

	16 threads	32 threads	64 threads	128 threads
1 sub-domain	16.03	25.20	30.26	23.19
2 sub-domains	16.28	30.64	49.25	56.64
4 sub-domains	16.15	29.36	50.82	54.26
8 sub-domains	16.24	29.16	52.79	83.72

the classical parallelization without DDM underperforms from 64 threads (4 nodes) to 128 threads (8 nodes). The use of multiple sub-domains allows a bigger part of the application to be parallelized, and gives better results. Let us consider a sequential version consuming about 4 hours to complete the whole simulation. With only 4 workstations, the simple multi-threaded simulation would last about 8 minutes (speed-up for 1 sub-domain and 64 threads), while the version with 8 sub-domains would not exceed 5 minutes. However, that is the limit for the classical hybrid distributed/shared-memory parallelization. According to the last result in the table, the DDM approach would allow to reduce the maximum execution time from 4 hours to 3 minutes, using 8 workstations composed of 2 quad-core processors with 8 GB RAM.

V. CONCLUSION

Acoustic simulation software is an invaluable tool to handle noise problems, both inside closed an open areas. In this paper, we reminded some principal geometrical methods which basically consider the sound propagation as straight line paths shot from a sound source. An hybrid beam/ray-tracing algorithm was parallelized by means of domain decomposition methods coupled with a dynamic load balancing. An additional acceleration was gained by a second level parallelization based on multi-threading in shared memory environment.

Numerical simulations have been carried out to evaluate the performance and efficiency of this new method on low-resources platforms, for analyzing large scale urban acoustic pollution. This new method allows to obtain a significant acceleration on few processors by parallelizing the whole algorithm, including the gathering process. As our experiments show, it can be used on inexpensive systems, such as a cluster of workstations provided with an Ethernet interconnection, since it reduces the memory usage and the communication bandwidth.

REFERENCES

[1] J Zhang, W Zhao, and W Zhang. Research on acoustic-structure sensitivity using FEM and BEM. *Journal of Vibration Engineering*, 18(3):366–370, 2005.

[2] J.-C. Autrique and F. Magoulès. Analysis of a conjugated infinite element method for acoustic scattering. *Computers and Structures*, 85(9):518–525, 2007.

[3] J.-C. Autrique and F. Magoulès. Studies of an infinite element method for acoustical radiation. *Applied Mathematical Modelling*, 30(7):641–655, 2006.

[4] F. Ihlenburg. *Finite Element Analysis of Acoustic Scattering*. Springer, 1998.

[5] L L. Thompson. A review of finite-element methods for time-harmonic acoustics. *The Journal of the Acoustical Society of America*, 119(3):1315–1330, 2006.

[6] I. Harari and F. Magoulès. Numerical investigations of stabilized finite element computations for acoustics. *Wave Motion*, 39(4):339–349, 2004.

[7] J.-C. Autrique and F. Magoulès. Numerical analysis of a coupled finite-infinite element method for exterior Helmholtz problems. *Journal of Computational Acoustics*, 14(1):21–43, 2006.

[8] J B. Allen and D A. Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

[9] V Pulkki, T Lokki, and L Savioja. Implementation and visualization of edge diffraction with image-source method. In *Proceedings of the 112th Audio Engineering Society Convention*, 2002.

[10] D Oliva Elorza. Room acoustics modeling using the raytracing method: implementation and evaluation, 2005.

[11] T Funkhouser, N Tsingos, I Carlbom, G Elko, M Sondhi, J E. West, G Pingali, P Min, and A Ngan. A beam tracing method for interactive architectural acoustics. *Journal of the Acoustical Society of America*, 115(2):739–756, 2004.

[12] S Laine, S Siltanen, T Lokki, and L Savioja. Accelerated beam tracing algorithm. *Applied Acoustics*, 70(1):172 – 181, 2009.

[13] A. Chandak, C. Lauterbach, M. Taylor, Z. Ren, and D. Manocha. AD-Frustum: Adaptive Frustum Tracing for Interactive Sound Propagation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1707–1722, 2008.

[14] F. Magoulès and F.-X. Roux. Lagrangian formulation of domain decomposition methods: a unified theory. *Applied Mathematical Modelling*, 30(7):593–615, 2006.

[15] F. Magoulès, K. Meerbergen, and J.-P. Coyette. Application of a domain decomposition method with Lagrange multipliers to acoustic problems arising from the automotive industry. *Journal of Computational Acoustics*, 8(3):503–521, 2000.

[16] K. Subramanian and D. Fussel. A search structure based on k-d trees for efficient ray tracing. Technical Report Tx 78712-1188, The University of Texas at Austin, 1992.

[17] I Wald. *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Computer Graphics Group, Saarland University, 2004.

[18] I Wald and P Slusallek. State of the art in interactive ray tracing. In *State of the Art Reports, EUROGRAPHICS 2001*, pages 21–42. EUROGRAPHICS, Manchester, United Kingdom, 2001.

[19] J Günther, S Popov, H-P Seidel, and P Slusallek. Realtime ray tracing on GPU with BVH-based packet traversal. In *Proceedings of the IEEE Eurographics Symposium on Interactive Ray Tracing*, pages 113–118, September 2007.

[20] D R Horn, J Sugerman, Houston, and P Hanrahan. Interactive k-d tree GPU raytracing. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, pages 167–174, New York, NY, USA, 2007. ACM.

[21] G Cadet, S Zambal, and B Lcussan. Rendering complex scenes on clusters with limited precomputation. In *Proceedings of the International Symposium on High Performance Computational Science and Engineering*, volume 172, pages 79–96. Springer Boston, 2005.

[22] C Lauterbach, S-E Yoon, M Tang, and D Manocha. Reducem: Interactive and memory efficient ray tracing of large models. *Computer Graphics Forum*, 27(4):1313–1321, June 2008.

[23] R Overbeck, R Ramamoorthi, and W R. Mark. A Real-time Beam Tracer with Application to Exact Soft Shadows. In *Eurographics Symposium on Rendering*, Jun 2007.

[24] Y. Maday and F. Magoulès. Absorbing interface conditions for domain decomposition methods: a general presentation. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3880–3900, 2006.

[25] B. Després. Domain decomposition method and the Helmholtz problem.II. In *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993)*, pages 197–206, Philadelphia, PA, 1993. SIAM.

[26] S. Ghanemi. A domain decomposition method for Helmholtz scattering problems. In P. E. Bjørstad, M. Espedal, and D. Keyes, editors, *Ninth International Conference on Domain Decomposition Methods*, pages 105–112. ddm.org, 1997.

[27] P. Chevalier and F. Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.

[28] M.J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 24(1):38–60, 2002.

[29] M.J. Gander, L. Halpern, and F. Magoulès. An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. *International Journal for Numerical Methods in Fluids*, 55(2):163–175, 2007.

[30] Y. Maday and F. Magoulès. Optimized Schwarz methods without overlap for highly heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1541–1553, 2007.

[31] Y. Maday and F. Magoulès. Improved ad hoc interface conditions for Schwarz solution procedure tuned to highly heterogeneous media. *Applied Mathematical Modelling*, 30(8):731–743, 2006.

[32] Y. Maday and F. Magoulès. Non-overlapping additive Schwarz methods tuned to highly heterogeneous media. *Comptes Rendus à l'Académie des Sciences*, 341(11):701–705, 2005.

[33] F. Magoulès, F.-X. Roux, and S. Salmon. Optimal discrete transmission conditions for a non-overlapping domain decomposition method for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 25(5):1497–1515, 2004.

[34] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approach to absorbing boundary conditions for the Helmholtz equation. *International Journal of Computer Mathematics*, 84(2):231–240, 2007.

[35] F. Magoulès, F.-X. Roux, and L. Series. Algebraic Dirichlet-to-Neumann mapping for linear elasticity problems with extreme contrasts in the coefficients. *Applied Mathematical Modelling*, 30(8):702–713, 2006.

[36] F. Magoulès, F.-X. Roux, and L. Series. Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(29–32):3742–3759, 2006.

[37] F. Magoulès, F.-X. Roux, and L. Series. Algebraic way to derive absorbing boundary conditions for the Helmholtz equation. *Journal of Computational Acoustics*, 13(3):433–454, 2005.

[38] F.-X. Roux, F. Magoulès, L. Series, and Y. Boubendir. Approximation of optimal interface boundary conditions for two-Lagrange multiplier FETI method. In R. Kornhuber et al, editor, *Proceedings of the 15th Int. Conf. on Domain Decomposition Methods, Berlin, Germany, Jul.21-15, 2003*, Lecture Notes in Computational Science and Engineering (LNCSE). Springer-Verlag, Haidelberg, 2005.

[39] M.J. Gander, L. Halpern, F. Magoulès, and F.-X. Roux. Analysis of patch substructuring methods. *International Journal of Applied Mathematics and Computer Science*, 17(3):395–402, 2007.

[40] F. Magoulès, P. Iványi, and B.H.V. Topping. Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 193(45–47):4797–4818, 2004.

[41] F. Magoulès, P. Iványi, and B.H.V. Topping. Convergence analysis of Schwarz methods without overlap for the Helmholtz equation. *Computers and Structures*, 82(22):1835–1847, 2004.

[42] F. Magoulès and R. Putanowicz. Optimal convergence of non-overlapping Schwarz methods for the Helmholtz equation. *Journal of Computational Acoustics*, 13(3):525–545, 2005.

[43] F. Magoulès. Décomposition de domaines pour le lancer de rayons. France Patent no. 1157329. 12 August 2011.

The Parallel Computation of Green Function Based On the Characteristic Length of Ship

Zixiang Yu, Dan Li, Jin Shengping, Yufeng Gui , Zhang Shesheng
Association of Mathematical Modeling, Wuhan University of Technology, Wuhan, 430070
E-mail:Dan li, lidan0520@163.com

***Abstract*—The change of ship's characteristic length can change scientific researching method. The paper gains the calculation parameters of Green function and the relational expression of vessels' characteristic length by establishing the analysis theory of ships' characteristic length dimension based on the control equation of green function and gives the statistical expression for ships' characteristic length. Moreover, it constructs a parallel algorithm of green function. The numerical results show that our algorithm has high parallel calculating ratio.**

Keywords- Green function; shipping; probability; parallel computation

I. INTRODUCTION

The proportion of shipping is inevitably going up as the speed up of marine development and the development of economic globalization. However, as we all know, there will be strong oscillations when a ship moving on the surface of the water. It will not only affect the speed of the ship, but also cause the deformation by waves' dash, and emerge unimaginable consequence. So, it's essential for us to study how does waves act on ships' hydrodynamic force. When it comes to ships' hydrodynamic force, it is usually to study Green function whose computing is associated with ships' characteristic length. Therefore, to study Green function, we have to get an insight into ships' characteristic length.

By using Green Theorem, Haskind[1]solves the velocity potential with time of ship's navigation, and he derives the expression of point-source Green function. According to the boundary condition, and he solves integral equation about the velocity potential by dividing of disturbance velocity potential in flow field into diffraction velocity potential and radiation velocity–potential. Newman[2] calculates Green function with single integral by series expansion method, and he gives detailed illustration of the fundamental of rapidity, seakeeping and operability of ship motion. Zhen Chengsheng[3] computes the derived ordinary differential equations by using the numerical calculation, and improves the accuracy and efficiency of the calculation. According to the rapidity of the ship, Dai Yishan[4] has done years' research on Green function, and he respectively gave the computing method of frequency domain Green function and Time domain Green function. Meanwhile, Liu Yingzhong[5] has unique research findings, and gives the computational formula of Green function applied to ship fluid dynamics. In recent years，many scholars have been studying Green function. For example, Xing[6]gives the computing formula in consideration of the governing equation of Green function, and has a discussion about inwardness of Green function, and he discovers that shipping characteristic length parameter needs further investigation as it influences calculating efficiency of Green function.

The paper will discuss the relationship between Green function and shipping characteristic length, construct parallel computing method based on ships' length and calculate given examples.

II. DIMENSIONAL PARAMETER ANALYSIS IN GOVERNING EQUATION

It is noticed that the data size of calculating Green function is associated with the ranges of parameter in some ways. The wider the range, the more the volume of function value needs to be calculated. Only the range of Green function's parameter in engineering practice is determined can we calculate the value of Green function needed.

Firstly, we set the Green function governing equation and boundary conditions as follows:

$$\Delta\phi = \delta(P-Q) \qquad (1.1)$$

In the equation above, is potential function, P is the field point: ,Q is the source point: . All of them are in the upper half plane, therefore the boundary conditions are:

$$\frac{\partial\phi}{\partial x} + k\phi = 0 \qquad \frac{\partial\phi}{\partial x} \mp ik\phi = 0$$

In the above equation, is complex potential. Let's suppose that L is shipping characteristic length, U is characteristic reference velocity. Let

$$x = XL \qquad y = YL \qquad z = ZL \qquad \Phi = ULF \quad (1,2)$$

And governing equations become as

$$\Delta F = \delta(P-Q)$$

Boundary conditions become as

$$\frac{\partial F}{\partial X} + kF = 0 \qquad \frac{\partial F}{\partial Y} \mp ikLF = 0 \quad (1.3)$$

And let K=kL , then

$$\mathrm{Im}(\frac{dF}{dZ}) - K\,\mathrm{Re}[F] = 0 \qquad \mathrm{Re}(\frac{dF}{dZ}) \pm jK\,\mathrm{Re}[F] = 0 \qquad (1.4)$$

From the above discussion, it can be safely concluded that if we change wave number from K to kL, the equations will be converted into dimensionless equations. So, we can calculate K value by shipping characteristic length and wave number.

III. SHIPPING CHARACTERISTIC LENGTH DISTRIBUTION

The principal dimension of vessel is the geometrical parameter used to reflect the size of hull; the coefficients of form is the geometrical parameter used to reflect the shape of hull; scale ratio is the geometrical parameter used for hull's size. All of these parameters are proved to be

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.35

really useful for vessel's design, construction, operation and performance analysis.

As is known to all, the size of vessel is mainly measured by ship length, molded breadth, mould depth and draught. This paper will analyze the length of ships. The length of ships we often choose falls into three categories: overall length, length between perpendiculars and length of designed water plane. The overall length is the maximum horizontal distance parallel to the waterline from the forefront of the bow to the last end of the quarter. Length between perpendiculars, which is often abbreviated as LBP, refers to the length of a vessel along the waterline from the forward surface of the stem, or main bow perpendicular member, to the after surface of the sternpost, or main stern perpendicular member. Forward perpend- icular: make a vertical through the point of intersection of designed waterline and stem post's leading edge. After perpendicular: a vertical line through the intersection of the design waterline with the after-side of the straight portion of the rubber post of a ship. Waterline length: it denotes the length of the vessel at the point where it sits in the water, usually it refers to length of designed waterline. In the calculation of hydrostatic property, we often adopt the length between perpendiculars; while in the analysis of resistance , we choose waterline length; and when it comes to dry-docking, alongside a pier or passing a lock, overall length is our best option. This paper selects the overall length as our computational data.

We collected 2510 samples of the overall length of ships and obtained the following diagram of L distribution through Matlab by doing distribution fitting analysis,see Fig.1.

$$f(x|a,b) = ba^{-b}x^{b-1}e^{-(\frac{x}{a})^b} \tag{2.1}$$

Here a=0.000268, b=1.626. We can find that the length of the smallest ship is only 2.35m, while the longest ship has an unimaginable length of 458.45m. The span between ships is really large, so we assumed that the data covers all kinds of ships. The length was divided into 8 segments and every step size is 57m.What's more, the length of vast majority of ships is between 2.35 to 250m, which indicates that mighty ship is still in the minority in another side, and the majority ships in our daily life are all less than 250m.Their distribution roughly accord with the Weibull's Distribution.

.

IV. GREEN FUNCTION PARALLEL COMPUTATION

From the discussion above, we realized that when the ships' characteristic length is given, parameter K in Green function is determined as well. Green function is determined by formula (1.1) that can be expressed as follows:

$$G(z,\zeta,K) = \frac{1}{2\pi}\ln\frac{Z-\zeta}{Z-\bar{\zeta}} - \frac{1}{\pi}H(z,\zeta,K) + i\exp(iK(Z-\bar{\zeta}))$$
$$H(z,\zeta,K) = PV\int_0^{\infty}\frac{\exp(iu(Z-\bar{\zeta}))}{u-k}du \tag{3.1}$$

In the formula above, PV means principle value integrals. Coordinates value in [0,L],parameter K value in (0， KL). So the corresponding Green function G is the function of quintuple space . If we choose 1000 dots for every coordinate, then we need to calculate Green functions of 10^(15) dots in quintuple space, however, because the numbers of infinite integrals need computing and the amount of calculation is numerous, it's essential for us to structure parallel algorithms by means of which we will do the calculation. Its steps are as follows:

1) Given k, L and the number of processors: P.

2) Divide (0, kL) into P parts which are of the same length.

3) Calculate the value of Green function in every subinterval.

4) Output calculated results.

From the expression of Green function , we may see that to calculate Green function is to calculate H function. An example about how to calculate Green function is given below.

Example: Consider an underwater section rectangle whose depth is 0.5 and width is 1, then we took the number of panel points n as 60, which step size choose h=1/30, wave number k=2, and length of the ship L=1.The outcome is printed on figure-2. The following figure-2 represents the relationship between the real part and imaginary part of H function, which looks like a book in some ways.

V. CONCLUSIONS

The paper deal with the relation between Green function and shipping characteristic length, by adopting the method of dimension analysis on the basis of Green function governing equation. On the other hand, it decreases calculation through combining wave numbers and shipping characteristic length and forms a new parameter. Meanwhile, we dug deep to see the probability distribution of the length of ships, finding out that the length is approximate distribution of Weibull. What's more, in the article, we give an opinion to parallel computation and work out quintuple space Green function value.

ACKNOWLEDGMENT

The paper is financially supported by China national natural science foundation (No. 51379168, No.51139005). ， and Humanity and Social Science foundation of Ministry of Education of China (Grant No.12YJAZH022)

REFERENCES

[1] Haskind."The hydrodynamic theory of ship oscillation in rolling and pitching".Prikl .Mat.aMekh,Vol.10, pp,33-36.1946.

[2] Newman J N. Algorithms."For The Free-Surface Green function".Journal Of Engineering Mathematics,1985,19(1):57-67.

[3] Chengsheng Zhan, Zaojian Zou, Weitao zheng. "Numerical analysis of the far field Green function in the ship motion". Journal of Wuhan University of Technology (Transportation Science & Engineering),2002,26(4):1-4.

[4] Wenyang Duan, Yishan Dai. "Numerical analysis of the two dimensional frequency domain Green function". Research and development of water dynamics,1996, 11(3):330-334.

[5] Yingzhong Liu, Guoping Miu."Theory of the ship motion in the waves".Shanghai:Shanghai University of Transportation Press,1987.

[6] Wang Xing, Liu Chao, Sun Zhenli, Zhang Shesheng, Inherent Properties of Two Dimension Green function with Linear

Boundary Condition of Free Water Surface[J], Applied Mathematics, 2013, Vol.4, No.8A, 97-99

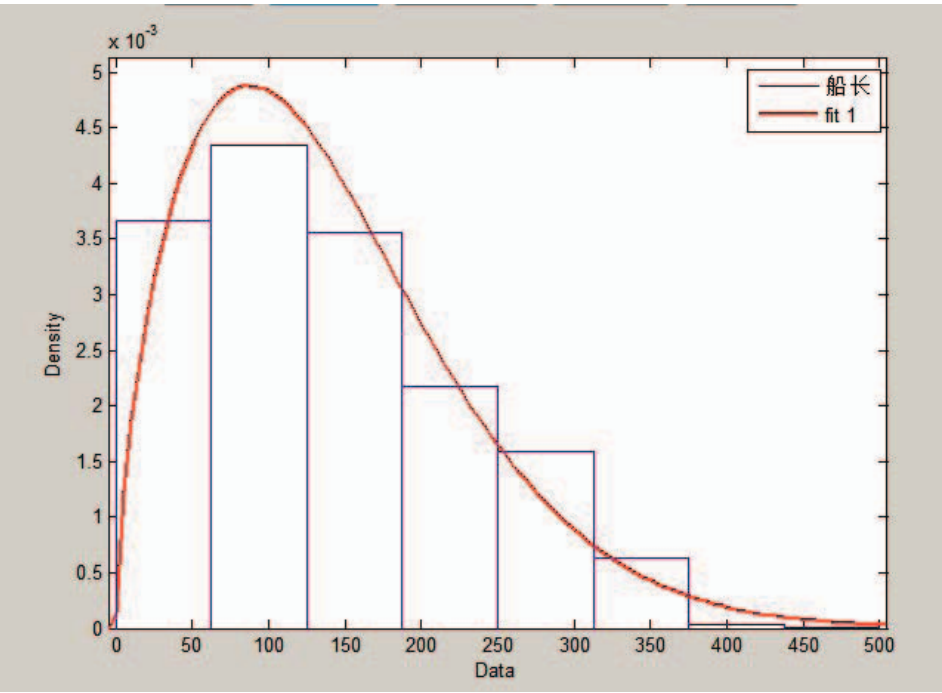


Figure 1. The distribution of ship's length.

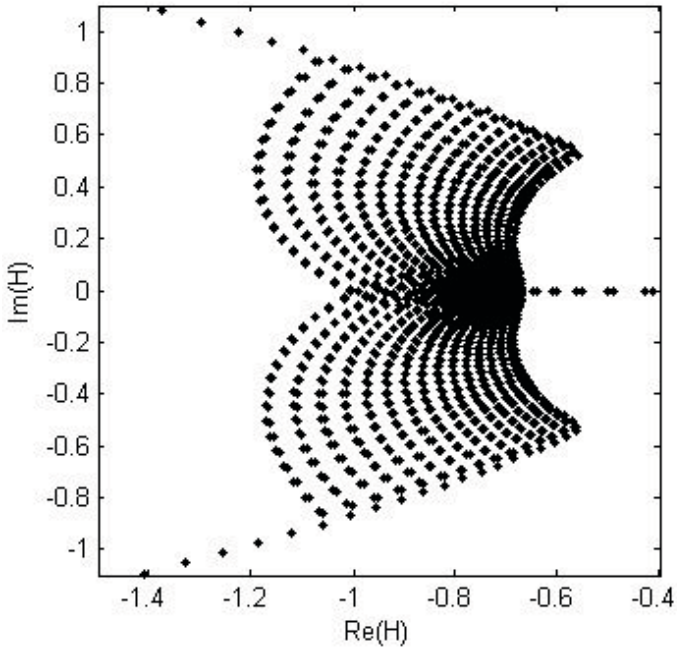


Figure 2. Im(H) varied with Re(H),and the width of underwater section rectangle is 1, depth is 0.5, and the number of panel points n=60,h=1/30,k=2.

TwigMRR: distributed XML twig query processing

Zhixue He[1,2] Husheng Liao[1] Hang Su[1]
1. Beijing University of Technology, Beijing, China
2. North China Institute of Aerospace Engineering Langfang, China
hezhixue@gmail.com liaohs@bjut.edu.cn suhang@bjut.edu.cn

***Abstract*—Twig query is considered the core query pattern in most XML query language. With the XML document size becoming larger, single site cannot deal with such volume data in storage capacity and compute ability. Partitioning the large data and distributed parallel processing query is an efficient and effective way. This paper proposes TwigMRR algorithm for evaluating XML twig query over large XML data that is encoded by Dewey, partitioned horizontally and distributed storage in a cluster. TwigMRR is based on MapReduce and extended a new model Map-Reduce-Reduce to get the final results for twig query. The experimental results show that our approach is scalable and efficient on this problem.**

Keywords- partitioned XML document; twig queries; distribute query processing

I. INTRODUCTION

XML is a commonly used format for data exchanging in multi-system environment, and becomes the de facto standard for representing semi-structured data. Due to widespread used, the problem of effectively and efficiently query processing has attracted significant effort in both research community and in commercial products. The centralized management and processing of XML query have been well studied, but with the quickly producing and aggregating information, centralized techniques are not applicable to large XML dataset.

In most XML query languages, e.g., XPath[1] and XQuery[2], queries are expressed as twig patterns. Finding all occurrences of a twig pattern query in an XML document is considered the core operation for XML query processing. In this paper, we focus on how to use MapReduce distributed framework to process and manage twig queries with large amount of XML data. MapReduce is a widely used software framework for processing huge datasets on distributable problems using a cluster of commodity computers. It defines distributed and parallel processing of Map and Reduce operations, and hens, is a simple, fault-tolerant and scalable paradigm, can be applied to huge dada sets. It has found a wide range of applications in the area of massive data analysis.

In this paper we propose an approach for twig queries processing over large XML data which is coded by Dewey, portioned and distributed stored in distribute file system. The Dewey is selected because it is more flexible in XML update than other code methods. There exist significant bodies of work on querying XML data in a centralized environment, but for distributed processing XML query evaluation, research on this problem just started. In [3], a number of plan alternatives and optimizations for the distributed processing of queries in a vertically partitioned XML database system was proposed, this work focused on how to build a execute plan, but not the query processing. Base on partial evaluation, [4] proposed distributed evaluation algorithms for Boolean XPath queries and for data-selecting XPath queries, and presented a MapReduce algorithm for processing Boolean XPath queries using partial evaluation. However, the author assumed that the XML data had been distributed stored across a number of sites, and not show how to partition large XML datasets and how to decompose the query. There are some works developed a system that simultaneously parallel processing XML queries for a massive volume of XML data with Hadoop, for example [5]. Close to this work is [6], HoX-MaRe algorithm was present for distribute XPath query evaluation, but compared with this paper, we have modified the MapReduce model, after the Reduce phase, another Reduce is joined to MapReduce and form a new Map-Reduce-Reduce program model.

The main contribution of this paper is summarized below. We present a new algorithm, called TwigMRR, which is independently developed and shares similar with HoX-MaRe. Our algorithm changes the MapReduce model to Map-Reduce-Reduce model. The new model not increases the program difficulty, but for XML query processing is very efficiency. Compared with other method, this algorithm does not change the essence of distribute processing that send the query to the data, and the data transferred between store sites are determined by the query size but not XML data size.

The rest of the paper is organized as follows. We first describe some background information in Section 2. After that we present an evaluation algorithm for twig queries in distributed architecture, referred as TwigMRR, in Section 3. An experimental study is provided in Section 4, and concludes our work in Section 5.

II. PRELIMINARIES

We next discuss the twig queries, partition of XML document, and MapReduce studied in this paper.

A. XML and Twig Query

An XML documents is modeled as a directed, rooted, labeled tree T. In an XML tree, the labels come from an infinite set $\sum$. *N(t)* and *E(t)* respectively denote the set of nodes and edges of the tree t. *label(n)* denote the label of node n, and *root(t)* is the root of t. If there an edge(n_1, n_2)

978-1-4799-4169-8 2014

DOI 10.1109/DCABES.2014.36

in *E(t)*, the node n_2 is the child of n_1 (n_1 is the parent of n_2). If there is a path from n_1 to n_2, n_2 is the descendant of node n_1 (n_1 is the ancestor of n_2). A branching node is the node which has at least two children. In the next of this paper document element and tree node are equivalence.

The core query pattern in most standard XML query languages (e.g., XPath and XQuery) is also in a tree-like structure, which is often referred as a twig. In a twig query, an edge can be either single-lined "/" or double-lined "//", which constraints the two matched nodes is either a PC(Parent-Child) relationship or an AD(Ancestor-Descendant) relationship.

The process to find all the occurrences of a twig in an XML document is called twig matching. A mach of twig Q in an XML tree T is identified by a set of mappings form the query node of Twig Q to the document node of Tree T. For mapping e: (i) label preserving, for node $n \in N(Q)$, either label(n) = * or label(n) = label(e(n)), where * is the "wildcard" sysmbol; (ii) relationship preserving, for edges$(n_1, n_2) \in E_{/}(Q)$, then $(e(n_1),e(n_2)) \in E(t)$. For edges $(n_1, n_2) \in E_{//}(Q)$, then node $e(n_2)$ is a descendant of node $e(n_1)$. The result of matching Q to T is a list of n-ary tuples, where n is the number of nodes in Q, and each tuple$(n_1, n_2, \ldots, n_n)$ consists of the document nodes $n_1, n_2, \ldots, n_n$, which identify a distinct match of Q in T.

B. Dewey labeling scheme and XML Partitioning

Dewey labeling scheme[7] is used to present the position of an element occurrence in an XML document. In Dewey, a label for an XML tree node is a concatenation of its parent's label and its local order. For element n_2, label(n_2)=label(n_1).x, where n_2 is the x-th child of n_1. Dewey supports efficient evaluation of structural relationships between elements. That is, elements n_1 is an ancestor of element n_2 if and only if label(n_1) is a prefix of label(n_2), element n_1 is a parent of n_2 if and only if label(n_1) is a prefix of label(n_2) and their length is different in one.

For example, the label for "d" in Fig.1, "1.1.1" is a concatenation of its parent's "b" label "1.1" and its local order "1". And the node "a" labeled "1" is a parent of node "b" labeled "1.1", and is ancestor of node "d" labeled "1.1.1".

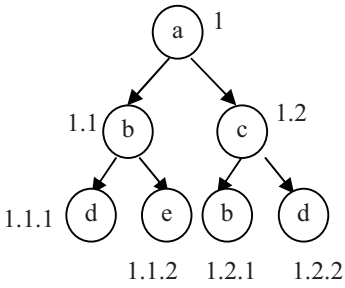


Figure 1. Example document with Dewey labeled

For a single large XML document which size beyond the memory capacity, we can split it into some splits and for each split stored in a computer of a cluster. In this paper we focus on horizontal partition, a split is a subtree of the XML tree T. For a horizontal split S of XML document T, the followings are hold: (i)for each node n_S in S, there is a node n of T having the same path from root to them; (ii)each split has a distinct leaf node which in not contained in other splits. For instance, in Fig.1, the XML tree can be split into s_0 and s_1 as depicted in Fig. 2. Maybe there are many split results, here only represents one partition form. In this paper, we only considered the situation that the size of a split does not lager than the max block size in distributed file system.

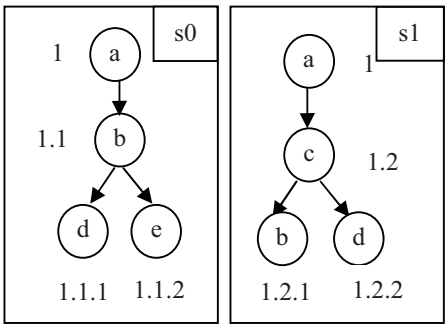


Figure 2. Partition of an XML document

C. MapReduce Model

MapReduce[8] is a model for data-intensive parallel computation in shared-nothing clusters. MapReduce hides the details of the parallel execution so that users can focus only on their data processing strategies. This programming model consists of Map and Reduce functions, and data are modeled as <key, value> pairs. The computation is expressed as follow:

map <k1, v1> → list(<k2, v2>);

reduce <k2, list(v2)> → list<k3, v3>.

The input for a MapReduce-based program first partitions the input data into input splits, and each of the splits is modeled as <k1, v1> form and assigned to a Map task. The Map task compute intermediate results i.e., (k2, v2) pairs in parallel. The key-value pairs <k2, v2> produced in the Map phase are hash-partitioned based on the key, yielding <k2, list(v2)>. Each of k2 partitions is assigned to a Reduce task. The Reduce function reads the list of all values, list(v2), producing<k3, v3>.

III. TWIGMRR ALGORITHM

A. An overview of TwigMRR

For processing a Twig query over a large amount of XML data using the MapReduce model, we present an TwigMRR algorithm, which include four step. In the first step, we parse an XML document, and label the nodes based on Dewey. In the second step, we split the XML document into some splits and read them as input to distributed storage system. Normally, this parsing step and partitioning step are only executed once for an XML document. After these two steps, the system is ready to processing twig queries in parallel style. In third step, we execute Map task to compute intermediate result for twig query in all computer stored the XML data split, using any structural join algorithms, e.g., TwigStack[9]. The last step

is Reduce task which collects all the intermediate results to get the final result of twig.

Consider the XML document and corresponding horizontal partition illustrated in Fig.1 and Fig.2, for the twig query in Fig.3: a[c]//b, where b is surrounded by double circle denoting it is the output node, if we evaluate it to s_0 and s_1 in isolated method, and then only get result node "b" labeled "1.2.1", but miss the node "b" labeled "1.1". This is because the partition of XML document destroys its structure constraint, and for one split, it does not know the overall structure information of the related nodes. For this reason, in the third step, before execute Map task, we decompose the twig query into linear path query, where linear path is a path from root to leaf, and there no branch node in the path. As in Fig.3, the twig query can be decomposed into two linear paths: a/c and a//b. Then partial results are computed for each linear path in Map function, and transformed to Reduce task to get the final result for twig query.

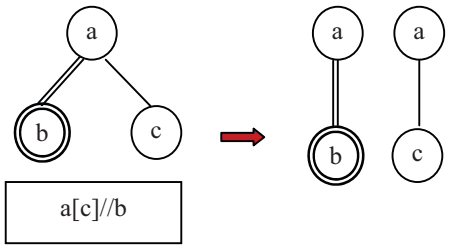


Figure 3. Twig query and linear path query

B. *Query processing in TwigMRR*

In this section, we present the TwigMRR algorithm, which is a MapReduce model style program. In this algorithm, the processing XML document is very large, and in the first two steps, we use SAX streaming technique encoded the XML node with Dewey, and then split it into blocks, which size can be varied. After this, the XML is portioned and stored in the distributed file system. Based on these works, Map and Reduce tasks can be executed on distribute file system. First, for every linear path query, map function computers the partial result.

Example1. For XML document splits in Fig.2, compute the twig query Q in Fig.3 for splits s0 and s1, we can get that, for query a//b and a/c in s_0, the branch nodes is <a> and, m(a) = 1, m(b) = 1.1 , m(c) = null; and in s_1, the branch nodes is <a>, and m(a)=1, m(b) = 1.2, m(c) = 1.2.1. Then we can get the results are<<a>, <1, 1.1, null>> and <<a>, <1, 1.2, 1.2.1>>.

After the Map task, in Reduce, we can computer the final result, but note that, maybe there are multiple intermediate Key values, so in this algorithm, we define two phases Reduce task. In the phase1, we computer intermediate results for branch nodes corresponding to the linear path queries; and then in the phase2, the Key value is defined by all the branch nodes in twig query, and the second Reduce function is executed to get the final result. Thus, the algorithm forms a new program model as Map-Reduce-Reduce for processing twig queries.

Example2. Continuing the Example1 and for Fig.3 twig query, because the query is simple, we only in task1 can get the result. Integrate the partial results <1, 1.1, null> and <1, 1.2, 1.2.1>, we know that 1.1 and 1.2 are all the results. The result "1.1" is get for a/c is satisfied in "1" and "1.2", this make the "b" labeled "1.1" satisfied the a[c]//b structure constraint.

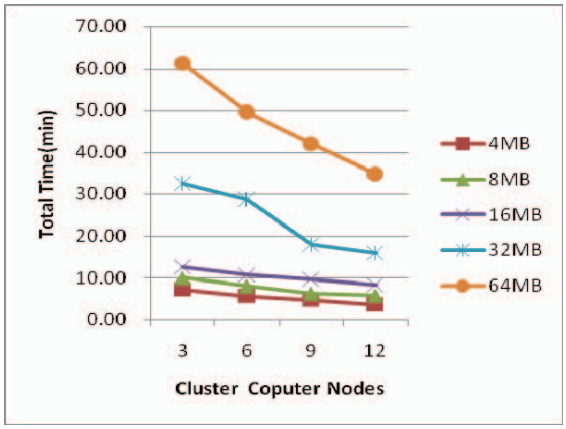
Algorithm TwigMRR query processing
Input: A twig query Q and an XML document
Output: A set of value results answering Q
1: **//step 1: parse and encode**
2: SAX to read the input document
encode it by Dewey
3: **//step 2: partition**
4: define split size, and split the XML into split blocks
5: upload the splits into distribute file system
6: **//step 3: Map task <split s, twig query Q>**
7: T_Q = getDecomposedTwig(Q); decompose Q into linear path query
8: For each query q in T_Q
9: K_q = getBranchNodes(q); get the branch nodes corresponding q as the key
10: Ms = getPatialResult(s); get the matching of linear path query q using any existing efficient structure join algorithm like TwigStack and the result is a n-ary tuples
11: mapOutput = <K_q,Ms>
12: // **step 4: Reduce phase1 <Key K, partial results M>**
13: N_Q = getBranchNodes(Q); get the branch nodes of twig query Q
14: compute the result for key K and output <N_Q, result>
15: // **step 4: Reduce phase2 <Key N_Q, result M>**
16: compute the final result for Q

Figure 4. TwigMRR algorithm

IV. EXPERIMENTAL STUDY

We provide an experimental study of our algorithm for evaluating Twig queries using TwigMRR. We implement out algorithms with Hadoop[10], a well-known open-source Java implementation of MapReduce, since MapReduce itself is not available to the public for Google's proprietary use. Like MapReduce utilizes the Google File Systems (GFS) as an underlying storage layer to read input and store output in Google, Hadoop MapReduce is the data processing layer and based on Hadoop DFS data storage layer. All experiments were performed on 3.00GHz Pentium(R) Dual-Core processor PC with 4G RAM running on Windows XP systems. We used benchmark dataset XMark[11], and by XML generator we get an XML document of 3G for testing and compare our algorithm with HoX-MaRe. For the XMark standard queries, we construct a cluster cloud environment. In this cluster, we run the code on 3, 6, 9, 12 computer nodes and get the results in Fig.5. We selected the HDFS file block is default

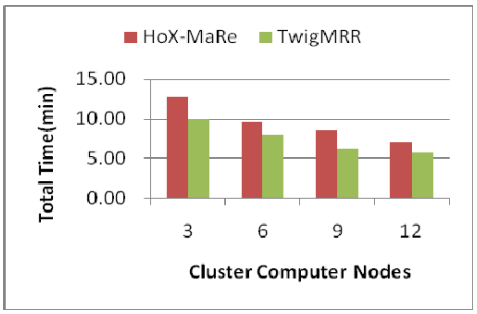
64MB, and defined the XML split size from 4MB to 64MB, and the run total time are as follows. As mentioned in section Ⅲ, the encoded parsing and split partitioning only execute once at the beginning, and in the next query processing their results can be reused, so here the total time is twig evaluation time. From the result we can conclude that with the computer node number is added, the used time is decreased. But the split size is a main factor for processing efficiency, with the split size becoming larger, the time is increased, this is because, in the Map function, TwigStack algorithm is a memory algorithm.



Nodes	4MB	8MB	16MB	32MB	64MB
3	7.25	10.00	12.67	32.50	61.33
6	5.67	8.00	10.75	28.75	49.67
9	4.75	6.25	9.67	18.00	42.00
12	3.67	5.75	8.33	16.00	34.75

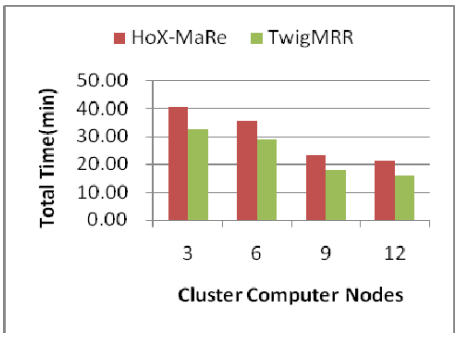
Figure 5. Total computation time for different cluster nodes

Compared with HoX-MaRe algorithm, we tested two split size situations: 8MB and 32MB, the result were show in Fig6 and fig7. On average, the efficiency is increased by about 20-25 percents. This is because in TwigMRR, “Bucket” concept was removed, and the time is decreased for no need to compute the getBuckets().



Nodes	HoX-MaRe	TwigMRR
3	12.75	10.00
6	9.67	8.00
9	8.50	6.25
12	7.00	5.75

Figure 6. Comparsion with HoX-MaRe for 8MB



Nodes	HoX-MaRe	TwigMRR
3	40.67	32.50
6	35.33	28.75
9	23.50	18.00
12	21.33	16.00

Figure 7. Comparsion with HoX-MaRe for 32MB

V. Conclusions and Future Work

In this paper, we propose a new TwigMRR algorithm to distributed processing XML twig queries using MapReduce program model, the XML document is encoded by Dewey. The algorithm can be used in the distributed cloud environment. Our experiments show that this algorithm is scalable for large XML datasets and outperforms the related works.

As the future work, we are planning to study the problem of vertical partition of XML document and other node label schemes to get a better performance.

Acknowledgment

This research was supported by the Beijing Natural Science Foundation (4082003) and Langfang Science and Technology Research and Development Project (2014011015) “Research XML database query processing method based on semantic information”.

References

[1] A. Berglund, D. Chamberlin, M. F. Fernandez, M. Kay, J. Robie, and J. Simeon. XML Path Language (XPath) 2.0. W3C Working Draft (2003)

[2] S. Boag, D. Chamberlin, M. F. Fernandez, D. Florescu, J. Robie, and J. Simeon. XQuery 1.0: An XML Query. W3C Working Draft (2003)

[3] P. Kling, M. T. Özsu, and K.Daudjee, “Generating efficient execution plans for vertically partitioned XML databases,” Proc. VLDB Endowment, 2010, 4(1), pp.1-11.

[4] G. Cong, W. Fan, A. Kementsietsidis, J. Li, and X. Liu, "Partial evaluation for distributed XPath query processing and beyond," ACM Transactions on Database Systems (TODS), 2012 37(4), 32, pp.1-43.

[5] H. Choi, K. H. Lee, S. H. Kim, Y. J. Lee, and B. Moon, "HadoopXML: a suite for parallel processing of massive XML data with multiple twig pattern queries," Proc. the 21st ACM international conference on Information and knowledge management, 2012, pp.2737-2739.

[6] M. Damigos, M. Gergatsoulis, and S. Plitsos, "Distributed Processing of XPath Queries Using MapReduce," In New Trends in Databases and Information Systems, 2014, pp. 69-77.

[7] I. Tatarinov, S. Viglas, K. S. Beyer, J. Shanmugasundaram, E. J. Shekita, and C. Zhang, "Storing and querying ordered XML using a relational database system," Proc. of SIGMOD, 2002, pp.204-215.

[8] J. Dean, S. Ghemawat. "MapReduce: simplified data processing on large clusters," Communications of the ACM, 51(1), 2008, pp.107-113.

[9] N. Bruno, N. Koudas, and D. Srivastava, "Holistic twig joins: Optimal XML pattern matching," Proc. of SIGMOD, 2002, pp. 310-321.

[10] Hadoop. Apache Software Foundation. http://hadoop.apache.org

[11] XMark. An xml benchmark project. http://www.xml-benchmark.org.

A Track Correlation Algorithm for Radar Intelligence

Jiang Surong
Fourth Department
Air Force Early Warning Academy
Wuhan, China
e-mail: yyhandy@yeah.net

Lan Jiangqiao
Fourth Department
Air Force Early Warning Academy
Wuhan, China
e-mail: sqqking@163.com

***Abstract*—Aiming at track correlation in airborne early warning radar intelligence analysis, an improved dual threshold track correlation algorithm is proposed. Taking into account that we have held all the point information after flying, we use wave gate technology instead of the hypothesis to test the hypothesis one by one, and the data have been screened for relevance to avoid extrapolation errors and history cases. It reduced the correlation computation amount. The experimental results show that the algorithm in airborne radar information processing can improve the efficiency of correlation between the ground radar tracks and airborne radar tracks.**

Keywords-airborne early warning radar;track correlation; dual threshold track correlation algorithm;intelligence analysis

I. INTRODUCTION

Airborne early warning radar is the basis for airborne warning and control system. Intelligence qualities will affect the development trend of the entire situation and command decisions directly. Therefore, airborne early warning radar intelligence analysis is one of the most important things for the early warning detection, command and control. Furthermore, it is also the rapid and effective means to improve the battle effectiveness of the airborne early warning radar.

Usually, the common way of airborne radar intelligence analysis is comparative and statistical. By comparing intelligence from airborne early warning radars with intelligence from ground radars at the same time and in the same airspace, we can draw a statistical analysis conclusion, which can be used to find out what are the advantages and disadvantages between the tracks from the airborne early warning radars and the ground radars. The methods can be put forward to improve the advantages and compensate for the lacks. For example, using an appropriate correlation algorithm for airborne radar track and ground radar track correlation can bring three benefits. Firstly, it can increase the degree of the automation of intelligence analysis. Secondly, it can ascertain the relationships of the large amounts of intelligence efficiently. Thirdly, it can improve the level of intelligence analysis.

Track correlation is one of the most important technologies of multi-sensor multi-target data fusion, also known as filtrating duplication. Several algorithms have been proposed for it. Such as weighted and modified track correlation algorithm[1], sequential track correlation algorithm[2], the nearest neighbor track correlation algorithm(NNTC)[3], dual threshold track correlation algorithm(DTTC)[1], track correlation algorithm based on fuzzy idea of sequential detection[4], based on fuzzy comprehensive multi-sensor multi-function target track correlation algorithm and distributed multi-sensor fusion fuzzy multi-target track correlation algorithm[5].

All these correlation algorithms are real-time algorithms, which require tight timeliness. They are prone to result in error and leakage correlated tracks in a dense targets environment or on the occasions of more division-merge tracks. Most algorithms above have taken several other factors into account, such as extrapolation error or historical circumstances. Otherwise, the tracks to be correlated in our post hoc analysis are all known tracks. It will add a lot of unnecessary work burden to join the correlation for airborne early warning radar intelligence analysis. So those correlation algorithms are not suitable for post hoc analysis. Relatively speaking, the dual threshold track correlation algorithm based on the idea of dual threshold signal detection is much closer to the idea of post hoc track correlation.

In this paper, we have improved the traditional dual threshold track correlation algorithm and apply it to airborne radar intelligence analysis. Experimental results show that it has better effects.

II. INDEPENDENT DTTC

Assume the amount of estimated error samples is R. Three steps are used to test whether tracks are correlated or not. Firstly, for each of the samples, χ^2 distribution is used to verify hypothesis. If it is test to accept H_0, the value of counter is incremented by 1, otherwise it is unchanged. Secondly, repeat the action R times that is comparing the value of the counter with a specified number $L(L<R)$. Thirdly, if the output of the counter is greater than or equal to L, then the determination is that tracks are correlated; otherwise, the determination is that tracks are not correlated. The following is main decision process.

For $l = 1, 2, \cdots, R$, using formula (1) to compute one by one.

$$\alpha_{ij}(l) = \left[\hat{X}_i^1(l|l) - \hat{X}_j^2(l|l)\right]' \left[P_i^1(l|l) + P_j^2(l|l)\right]^{-1} \left[\hat{X}_i^1(l|l) - \hat{X}_j^2(l|l)\right] \quad i \in U_1, j \in U_2 \tag{1}$$

In the formula (1), $\hat{X}_i^1(l|l)$ and $\hat{X}_j^2(l|l)$ are the state assessments of the track i from sensor A and track j from sensor B at time l, respectively, $P_i^1(l|l)$ and $P_j^2(l|l)$ are their covariance, respectively, U_1 and U_2 respect the sets of the tracks from sensor A and Sensor B, respectively.

If $\alpha_{ij}(l) \le \delta \quad i \in U_1, j \in U_2$,

then $m_s(l) = m_s(l-1) + 1 \quad (m_s(0) = 0)$.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.37

If $m_{ij}(l) \geq L$,

then the track i from the sensor A correlates with the track j from the sensor B; Otherwise, they are not correlated.

III. IMPROVED DUAL THRESHOLD TRACK CORRELATION ALGORITHM

The improved dual threshold track correlation is put forward to avoid the calculation errors of extrapolation. Taking into account the airborne radar intelligence analysis to its own characteristics, the improved dual threshold track correlation algorithm has been presented. It uses wave gate technology to filter the data correlated instead of the χ^2 hypothesis to test the hypothesis one by one, data were screened for relevance to avoid calculation errors and history cases, significantly reducing the amount of computation correlated with the traditional dual threshold track correlation algorithm applied to the correlated intelligence analysis tracks.

Improved dual threshold track correlation algorithm presents as follows:

Assume that airborne radar detecting targets is independent of the ground radar detecting targets. $U_x=\{1,2,\ldots,n_1\}$ represents the dot set of the track N_x from the airborne radar and $U_k=\{1,2,\ldots,n_2\}$ represents the dot set of the track N_k from the ground radar. $U_1=\{1,2,\ldots,N_1\}$ represents the track set of the airborne radar. $U_2=\{1,2,\ldots,N_2\}$ represents the track set of the ground radar. One dot of a track is represented by four coordinates (x, y, z, t);

Suppose and are one of the following events ($i \in U_x$, $j \in U_k$) :

H_0^1: The same target is detected by different sensors. The dot i and the dot j are the track points from an airborne radar and a ground radar respectively.

H_1^1: Different targets are detected by different sensors. The dot i and the dot j are the track points from an airborne radar and a ground radar respectively.

Therefore, the correlation algorithm can be converted into a hypothesis testing problem.

Formula (2) is used to compute the test statistic of their distance difference. Formula (3) is used to compute the test statistic of their time difference.

$$\Delta S_{max}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2+(z_i-z_j)^2} \quad (2)$$

$$\Delta T_{(i,j)}=|t_i-t_j| \quad (3)$$

If $\Delta S_{(i,j)}$ is less than the threshold ΔS_{max} and $\Delta T_{(i,j)}$ is less than ΔT_{max} then accepts H_0^1. That means the i^{th} dot of N_x and the j^{th} dot of N_k are correlated (the correlation degree L adds 1); otherwise accepts H_1^1 .

Suppose H_0^2 and H_1^2 are one of the following events ($N_x \in U_1$, $N_k \in U_2$):

H_0^2: A same target is detected by different sensors. N_x and N_k are tracks from different sensors.

H_1^2: Different targets are detected by different sensors. N_x and N_k are tracks from different sensors.

We take the test standard as the track N_x of the airborne radar. Formula (4) is used to compute the test statistic.

$$P=L/Sum \quad (4)$$

In (4), *sum* is the total number of already tested tracks. If P is greater than a threshold P_{min}, then H_0^2 is accepted. That means N_x and N_k are correlated. Otherwise, H_1^2 is accepted.

The concept of Spatial Correlation Region (SCR) is shown in Fig.1.

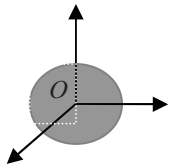


Figure 1. Spatial correlation region

The measurement point can be determined by the known location of the correlated target region, according to the spatial location information of the present testing dot O (the i^{th} dot of the track N_x), the correlation region of the ball and the current measurement of radar accuracy information. The following are detailed steps. Firstly, draw a sphere whose center is point O and radius is ΔS_{max},which is the max allowable detection error of the device. In figure 1, the grey sphere is the space correlated with the target point O. The dots in this space are determined to be correlated with it.

Similarly, the time-domain correlation interval is defined as follows. For current investigated point O (the i^{th} of the N_x) , it is a time interval [$-\Delta T_{max}$, ΔT_{max}],whose origin is the current time of point O. The length of the interval is determined by the performance of the detection equipment and the communication delay of the data link, as shown in Fig.2. Fall in this interval for trace points to determine when it is correlated with the current investigated point.

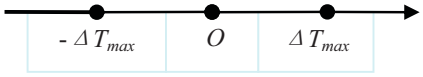


Figure 2. Time-domain correlation interval

Correlation process in detail is as follows:

1) First take out the i^{th} track (the batch number is N_x) from ground radar track data, and start with the j^{th} (initial value of j is 1) point of the track N_x. Then find out those points which are both in spatial correlation region and in time-domain correlation interval. Finally, for the corresponding batch number N_k of those points, the correlation degree L becomes L++;

2） calculate the correlation ratio $P=L/Sum$ values, which, *Sum* is the total points of N_x which have already been processed;

3） Compare of P with the threshold probability (i.e., the second threshold), if P is greater than the second threshold, the determination is correlated, i++, go to step 1); Otherwise, j++, if all the points of track N_x have not been processed, jump to step 1) to continue, if the treatment has been completed, i++ and then go to step 1) continue;

Taking these algorithms to meet the condition P> = P_{min} of track N_r for the grant with the N_x, track N_r is determined to be correlated. It means that track N_r and track N_x, belongs to the same batch of targets from the ground radar and airborne radar.

Finally, the output of the correlated results is in the form of a table, including three parameters of the correlated tracks: the batch number of airborne radar intelligence, the batch number of ground radar intelligence and their correlation ratio.

IV. APPLICATION VERIFICATION

In accordance with the improved dual threshold track correlation algorithm, we design and implement the track correlation process for the intelligence analysis of airborne early warning radar, shown in Fig 3.

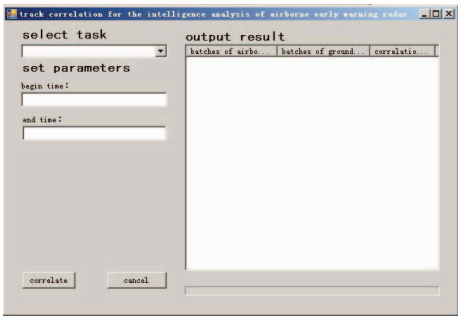


Figure 3. Parameter setting interface

By correlating a particular treatment for airborne and ground radar intelligence, we selected a certain mission, in which the ground radar has detected 688 batches of target. The correlating process consumed a total of less than three minutes through the program running, greatly reducing the correlated processing time. And the other by correlation with the use of means, will greatly reduce the workload of the late intelligence analysis. A part of the results is shown in Table 1.

TABLE I. PARTIAL CORRELATION TABLE

Airborne Radar	Ground Radar	Correlation Ratio
16	87	0.80769228
19	98	0.89130437
53	144	0.90196079
56	122	0.98245614

Comparison with the tracks playback of the same mission can find that on the target track these correlations coincide with the actual situation. For example, track 56 of airborne radar intelligence (red) and track 122 of ground radar intelligence (blue), shown in Fig 4.

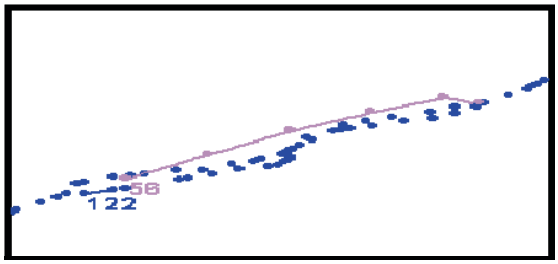


Figure 4. Playback the results screenshot

Through improved dual threshold track correlation algorithm，airborne radar information processing greatly improves the efficiency of correlation between the ground radar tracks and airborne radar tracks. It will help us to further researches on quality characteristics and laws of airborne radar intelligence for analysis. Furthermore, we can train and evaluate the airborne crews based on this correlation conclusion. Through continuous training to improve fundamentally the tactical level personnel subjective initiative into full play the role of personnel will be equipped to maximize the performance, and effectively improve the airborne early warning radar combat effectiveness.

V. CONCLUSION

In this paper, an improved dual threshold track correlation algorithm has solid theoretical foundation, and proved workable, and in the process of applying to the field of intelligence analysis, it has achieved good results. Through the dual threshold track correlation algorithm improvements, making the track correlated with the status quo in the field of intelligence analysis has been a fundamental change. This application not only improves the automation of information processing analysis of airborne radar track correlation, but also grasps the equipment performance, group training model to explore the mechanism of the crew for valuable information. Of course, the method in dealing with the short track is inadequacies, and needs to be further improved.

REFERENCES

[1] Xu Yu, Hua Zhonghe, and Zhou Yan, Radar Network Data Fusion , Beijing: Military Science Press, 2002, pp.239-246.

[2] Bar-Shalom Y, On the Sequential Track Correlation Algorithm in a Multisensor Data Fusion System. IEEE Transactions on Aerospace and Electronic Systems, vol. 44, Jan. 2008, pp.396-396, doi:10.1109/TAES.2008.4517016.

[3] Yang Wanhai, Multi-sensor data fusion and its application, Xi'an: Xi'an University of Electronic Science and Technology Press, 2004 , pp.60-101.

[4] Chen feishi, Kong Xiangwei, Zhao Kun, et al., Dual infrared sensor track correlation algorithm and fuzzy sequential simulation. System Simulation, vol. 16, Aug. 2004, pp.1652-1654.

[5] Guo Huidong and Zhang Xinhua, Based correlation track fuzzy comprehensive function algorithm and its application. Systems Engineering and Electronics, vol. 25, Nov. 2003, pp.1402-1403.

Parallel numerical model of water lubricated rubber bearing

Xin Chen, Hualing Zhao, Yufeng Gui, Shesheng Zhang
Association of Mathematical Modeling, Wuhan University of Technology, Wuhan, 430070, China
E-mail: Hualing Zhao, hualingbo324@126.com

***Abstract*—Study on water lubricated rubber bearing has important theoretical significance and applied value in military affairs. This paper builds a low noise composite rubber bearing parallel computing model, and drives the difference equations of bearing displacement. According to the step values, the parallel computing algorithm is given, and the convergence time t varied with time step length is obtained.**

Keywords- parallel calculation, Water lubricated stern tube bearing, friction, noise

I. INTRODUCTION

The stern bearing is an important part of the propulsion system of naval vessels, the role of which is to support the propeller shaft or stern bearing. In the poor lubrication state, the stern bearing produces serious friction, wear, resulting in seal failure [1]. Much poor lubrication will cause vibration noise from the shaft and bearing friction pair, seriously affecting the naval vessels safety, concealment and survival ability [2].Rubber material, is widely used in water lubricated stern tube bearing. The rubber advantages are, strong shock absorption, strong impact resistance performance, no water pollution, and an excellent concealment performance for the ship [3]. In World War II, many naval battles, especially submarine, also showed above advantages [4-5]. But the disadvantages are also obvious: low bearing capacity, low design pressure ratio that be only oil lubricated bearings 1/3. At the time of start, stop, low running speed, water lubricated stern tube bearing will produce vibration noise (Bearing noise). Such noise will harm water vehicle, and become more prominent as the electronic detection technology progress. This is an urgent problem l for us to improve rubber material, to find the mechanism of vibration noise, and to find noise reduction measures.

This paper will consider parallel computation model of water lubricated stern bearing [6], find relation between calculation error and time step .In the paper, section II gives a mathematic model of water-Lubricated rubber bearings; section III discuss analyses solution, section IV discuss parallel calculation steps, and section V shows numerical results.

II. MATHEMATICAL MODELS

Water lubricated rubber bearings, as shown in Figure 1, the friction surface of rubber strip has certain elastic degrees of freedom. In the case of bad lubrication, two friction surfaces is stick slip motion, so that the motion is discontinuous. The stick slip motion occurs in two stages, the first stage is "sticky" stage, the friction pair surface protrusions cohesive or embedding, when the friction pair relative motion trend, mutual tearing, elastic deformation; the second stage is "slip" stage, along with the movement continues, elastic deformation continue to increase, when the elastic force eventually equal to the maximum static friction force, adhesive end state, movement side effects in the elastic force of sliding phase separation occurred, enter slip state. At the same time, on the one hand, tearing adhesive or embedded generate rebound contrary to the direction of movement, on the other hand, the process of movement of new bonded or embedded is in the continuous formation. So, there is relative motion discontinuity when water lubricated rubber bearing friction pair is in the lubrication bad case, friction noise. Considering the actual condition of rubber bearings, we establish an analysis model of five degree of freedom nonlinear, as shown in figure 2. Comparing with rubber layer, the metal shaft is relative high stiffness. In the paper, it consider as rigid shaft. In the Fig.2, the shaft has 3 degrees of freedom, 3 directions displacement: horizontal displacement x3, vertical displacement of x4, and the rotation displacement θ. Water lubricated rubber bearings is simplified as a mass rigid block, that has 2 degrees of freedom, and 2 directions displacement: horizontal displacement x1, and vertical displacement x2. m1 is the quality of water lubricated rubber bearings, m2 is the quality of water lubricated shaft. c1, k1 respectively for the water lubricated rubber bearings in the direction of friction force (tangential) damping coefficient, stiffness coefficient. c2, k2 respectively for the water lubricated rubber bearings in radial direction (vertical) damping coefficient, stiffness coefficient. Ff is friction between the friction interfaces. N is positive pressure. M is the drive torque on the shaft of the role. From Fig.2, we can get the following equation:

$$m_1\ddot{x}_1 + c_1\dot{x}_1 + k_1x_1 = F_f \qquad (2\text{-}1a)$$

$$m_1\ddot{x}_2 + c_2\dot{x}_2 + k_2x_2 = \text{-}N \qquad (2\text{-}1b)$$

$$m_2\ddot{x}_3 = \text{-}F_f \qquad (2\text{-}1c)$$

$$m_2\ddot{x}_4 = N - G_2 \qquad (2\text{-}1d)$$

$$J\ddot{\theta} = \text{-}F_fR + M \qquad (2\text{-}1e)$$

The two friction surface has relative speed u, and the friction is nonlinear function of relative speed u. it can be represented as:

$$F_{\mathrm{f}} = F_f(u) = u\frac{dF}{du}$$

Here u is relative velocity of the two friction surface:

$$u = R\dot{\theta} + \dot{x}_3 - \dot{x}_1$$

F is empirical formula of friction:

$$F(u) = N[f_1 + (f_0 - f_1)Exp(-au)] \qquad (2\text{-}2)$$

Where, a is a constant, whose unit is: s/m. The static maximum coefficient is f0, and the static minimum friction coefficients is f1.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.38

III. NONLINEAR EQUATIONS

Since N, G2 are independent with x1, x3, and θ in the above equation, so that x2, x4 are also independent with x1, x3, and θ, then x2, x4 can independently solved. We can easy get solution of Eq((2-1b)

$$x_2 = \frac{-N}{k_2} + c_{21}e^{\lambda_{21}t} + c_{22}e^{\lambda_{22}t} \tag{3-1}$$

Here c_{21}, c_{22} are constants, and λ_{21}, λ_{22} are characteristic roots The solution of Eq((2-1d) is:

$$x_4 = (N - G_2)\frac{t^2}{2m_2} + c_{41}t + c_{42} \tag{3-2}$$

Here c_{41}, c_{42} are constants. The other equations are nonlinear, solved by numerical method.

We obtain:

$$\begin{aligned} &m_1\ddot{x}_1 + c_1\dot{x}_1 + k_1x_1 = F_f(u) \\ &m_2\ddot{x}_3 = -F_f(u) \\ &u = \frac{R}{J}[m_2R\dot{x}_3 + Mt + A] + \dot{x}_3 - \dot{x}_1 \end{aligned} \tag{3-3}$$

Let x=x1, y=x1', z=x3, the above equations can be written as:

$$\begin{aligned} &x' = y \\ &m_1y' + c_1y + k_1x = F_f(u) \\ &m_2z' = -F_f(u) \\ &u = \frac{R}{J}[m_2Rz + Mt + A] + z - y \end{aligned} \tag{3-4}$$

Above are first order nonlinear ordination equations, We will solve them by using difference method.

IV. PARALLEL CALCULATION

4.1 Step Euler method

From above first order nonlinear ordination equations, we may get difference equations:

$$\begin{aligned} &x_{k+1} = x_k + hy_k \\ &m_1\frac{y_{k+1} - y_k}{h} + c_1y_k + k_1x_k = F_f(u_k) \\ &m_2\frac{z_{k+1} - z_k}{h} = -F_f(u_j) \end{aligned} \tag{4-1}$$

Where h is time step length. Let h1=h/m1; h2=h/m2, above equations may be written as:

$$\begin{aligned} &x_{k+1} = x_k + hy_k \\ &y_{k+1} = y_k + h_1[F_f(u_k) - c_1y_k - k_1x_k] \\ &z_{k+1} = z_k - h_2F_f(u_k) \end{aligned} \tag{4-2}$$

It is easy to find that the numerical results are dependent on time step h witch take value on the domain [a,b], We choose P computers, and divide domain [a,b] to P subdomain. Every computer calculates difference equation on one subdomain. The initial conditions are:

$$\begin{aligned} &x(0) = x_0 \\ &y(0) = x'(0) = x'_0 \\ &z(0) = x'_3(0) = x'_{30} \end{aligned} \tag{4-3}$$

In this paper, we analyses numerical function (x,y,z) varied with step h. The parallel calculation steps are:

(1) give parameters m, c, k, R, J, M, A, and initial value;

(2) divide domain [a,b] to P sub-domain;

(3) Every computer calculates difference equation on one sub-domain.

(4) output numerical results

The convergence time varied with time is shown in the Fig.3.

4.2 fourth-order R-K method

Firstly, let

$$\begin{aligned} &f(t,x,y,z) = y \\ &g(t,x,y,z) = \frac{1}{m_1}(\mathrm{F}_f(u) - c_1y - k_1x) \\ &l(t,x,y,z) = \frac{1}{m_2}\mathrm{F}_f(u) \end{aligned} \tag{4-5}$$

From above formula, we know:

$$\begin{aligned} &x_{k+1} = x_k + f(t,x,y,z) \\ &y_{k+1} = y_k + g(t,x,y,z) \\ &z_{k+1} = z_k + l(t,x,y,z) \end{aligned} \tag{4-6}$$

Using fourth-order R-K method, we calculate convergence time varied with time, the results show in the Fig.4: Compared with the fig3 and fig.4 we can see that different numerical methods have different convergence time, and the tendency of the variation of the t with h is divorced.

V. NUMERICAL RESULTS

Let m1=2.4, k1=61081.8, c1=50.854, m2=56, k2=35643.7, c2=250, N=4338, R=0.0855, a=0.065, M=140, f0=0.14, f1=0.02. We calculate position from above difference equations with different time step length h, and output time t=t(h) if $|z|>0.001$. In this case, we think error is too large and end calculation for such time step h.

The number of process is P=5. time step h is taken value from domain [a,b]=[0.0001,0.1]. The numerical results are written in theFig.3. FromFig.3, we find that time t=t(h) is decrease as h increase.

The numerical position x1 of rubber bearing is shows in Fig.5(a), The line is go up as time t increase until t=0.185, and then decrease. There is volatility curve, it shows the axis in the process of movement, because the causes of friction, shaft vibration. The vibration amplitude of the displacement is about 7% of the position x1. If use fourth-order R-K method, the numerical position x1 of rubber bearing is shows in Fig.5(b), Compared with the fig3 and fig.5(a), it is obvious that the curve in the fig.5(b) is more smooth than the curve in the fig.5(a). It means that the fourth-order R-K method has higher precision in the beginning.

VI. CONCLUSIONS

According to the movement of water lubricated stern tube bearing, this paper establishes the physical model and mathematical model of water lubricated stern tube bearing, the motion equation to describe the water lubricated stern tube bearing displacement, and the equations of motion are divided into two categories, one category with analytical solution of motion equation, another kind is only the numerical solution of the equations of motion. On the convergence and time step numerical solution, this paper established a parallel computing method and step about relationship, convergence time and step.

ACKNOWLEDGMENT

The paper is financially supported by China national natural science foundation (No. 51379168，No.51139005), and Humanity and Social Science foundation of Ministry of Education of China (Grant No.12YJAZH022), and the Fundamental Research Funds for the Central Universities (WUT: 2014-Ia-039)

REFERENCES

[1] Heting Yang, Yuming Tang, Design of ship water lubricated stern tube bearing[J], Wuhan ship building, V.131, No.2, 2000,19-22.

[2] A.I.Krauter . Generation of Squeal/chatter in water-lubricated Elastomeric Bearings[J] . Journal of lubrication technology . 1981, 103(Compendex) : 406-413.

[3] Roy L.Orndorff Jr . Water-Lubricated Rubber Bearings,History and New Developments [J]. Naval Engineers Journal. 1985, 10 : 39~52.

[4] Harish Hirani, Manish Verma. Tribological study of elastomeric bearings for marine propeller shaft system[J] . Tribology International. 2009, 42(2) : 378-390.

[5] Roy L.Orndorff Jr , Darren G Finck . New design,cost-effective,high performance water lubricated bearings[J] . WARSHIP. 1996 : 367-373.

[6] Min Lan,Yan Huan,Shesheng Zhang , Large ship fluidsolid coupling dynamic mathematic model with explicitly form［J］, Applied Mechanics and Materials.April 19-20, 2014

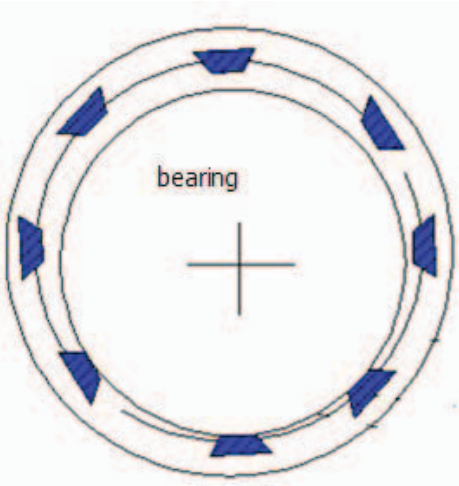


Figure 1. slab type water lubricated rubber

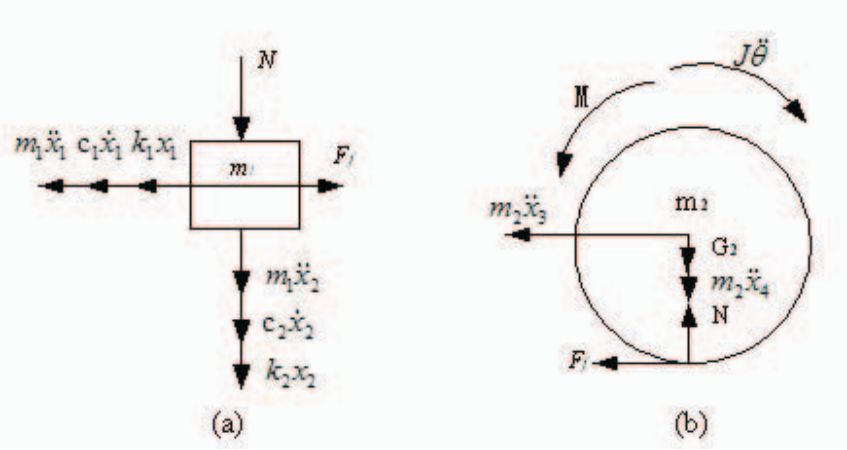


Figure 2. Force diagram of isolated water lubricated rubber bearings in the system.

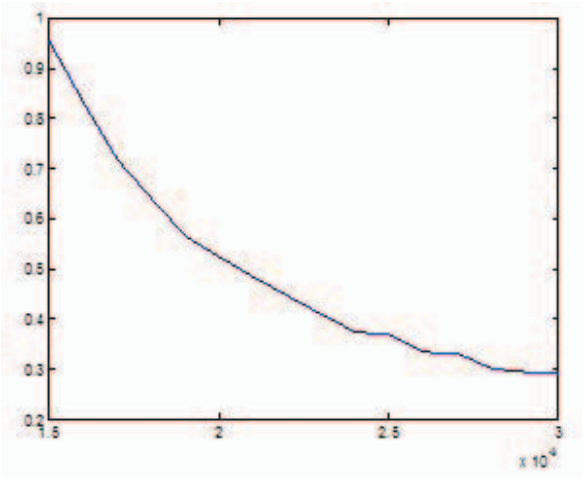


Figure 3. convergence time varied with step length h, Euler method.

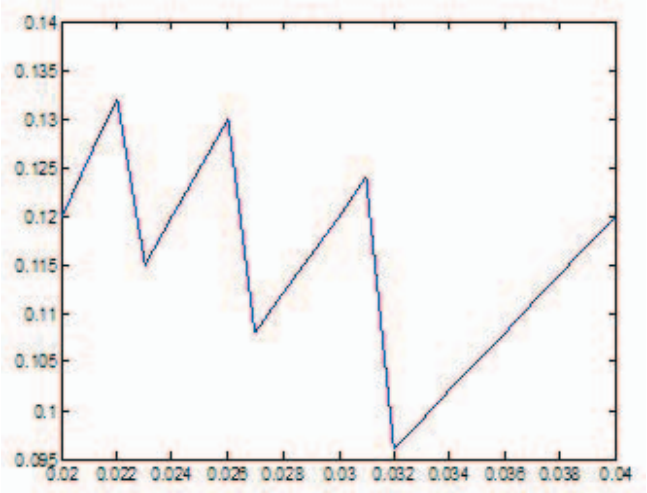


Figure 4. convergence time varied with step length h, forth R-K method

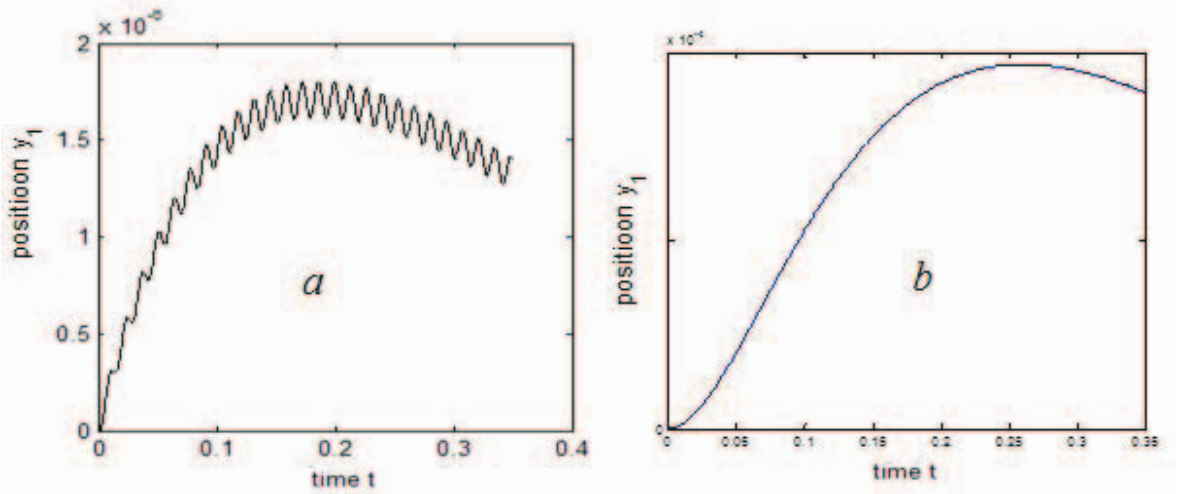


Figure 5. The position x1 of rubber bearing, (a) Euker method, (b) forth R-K method.

Research and Implementation of Petri nets parallelization model

Xuan Wang, Wen-jing LI ,Ze-yu Tang
College of Computer and Information Engineering
Guangxi Teachers Education University
Nanning, 530001,China
e-mail:liwj@gxtc.edu.cn

Weizhi LIAO
Guangxi Key laboratory of hybrid computation and IC design analysis
Nanning , 530006,China
e-mail:weizhiliao2002@yahoo.com.cn

***Abstract*—In order to solve parallel algorithm of Petri net system with concurrent functions and implement parallel control and execution of Petri net, parallel programming model of Petri net based on multi-core clusters is put forward. First, P-invariant technology is used to do the functional division of Petri net system and the parallel analysis of Petri net process. Next, based on architecture of multi-core cluster and combined with parallelism of Petri net process, Petri net system model is constructed. Simultaneously, parallel algorithm of Petri net system is raised. Finally, under the environment of multi-core cluster, the simulation experiment of parallel algorithm is done. The experimental results show that parallel algorithm based on Petri net system model is correct and feasible. The model can effectively simulate the actual operation of the system and is an effective model of Petri net parallelization system.**

Keywords- Petri Net; Parallelism; Multi-core Cluster; Parallel Model; Parallel Algorithm.

I. INTRODUCTION

Petri net is modeling and analytical tool of distributed system. For any system, Petri net can do hierarchical description and stepwise refinement. How to achieve a parallel processor after refinement and functionality of the divided Petri nets is an urgent problem. In the previous studies of Petri net, reference [1] have introduced how to implement Functional division of Petri net by using the P-invariant technology. Reference [2] on the basis of Research in [1] designs a model of the parallel Petri net based on the multi-core threads. This method partly achieves the parallelism of Petri net, but the efficiency is not optimistic for the complex Petri net. Because of the actual processing of engineering problems, electrical problems, etc complex systems, can't be described by a few simple parallel processes. In this case, we need to seek a way to solve the parallel model of complex systems. Based on this kind of demand, this paper proposes a parallel Petri net model based on multi-core cluster, which can realize level parallelism and effectively improves the speed of Petri net.

* Supported by the National Natural Science Foundation of China (61163012); open fund of Guangxi Key laboratory of hybrid computation and IC design analysis[2012HCIC01];the University Scientific Research Project of Guangxi(2013YB147); Innovation Project of Guangxi Graduate Education(YCSZ2014187)

II. PETRI NET PROCESS ANALYSIS OF THE PARALLELISM

A. The relative concepts of Petri net

The relative concepts of Petri net and p-invariant technical would reference [1] and [3].

B. Process analysis of Petri net

The parallel analysis between processes is mainly for shared places and shared transitions between two processes. Internal transition action and behavior analysis of processes is as follows.

According to the different structure of Petri net, several possible situations and the corresponding solutions are as follows.

(1) The places that has only a precursor and the subsequent must not be parallel.
(2) Two processes existed shared transition and not-shared places are parallel. The shared transition can look for the server to exchange information.
(3) The two processes between which existed shared

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.39

places and whose pi of input and output is 1 can be turned into parallel process by improving and the shared places realizes resources sharing of processes. The improved method is as follows: First, find out the shared places $P_i(i<m)$of the P-invariant support set X_i and X_j in the Petri net. Next, automatically copy the P_i(i<m) and its directed edge, meanwhile generate the same $P_i(i<m)$ and its directed edge. The old places and one of the support set consist of a parallel process, The new places and the other support set constitute another parallel process.

(4) The two processes which shared places and its input and output is n（$n>1$） can be turned into parallel process by improving. The improved method is as follows: Find the Shared places, add n-1 places-transitions pairs and the two processes who share the added transitions could be as two parallel processes.

III. THE ARCHITECTURE OF MULTI-CORECLUSTER AND THE PARALLELIZATION OF MODLE OF PETRI NET

A. The architecture of multi-core cluster

The multi-core cluster is a kind of hierarchical architecture for parallel computing. Its hierarchy is using distributed storage structure to pass the message between the nodes and applying the shared memory model in the nodes. Therefore this hierarchical cluster of architecture has not only the architectural features of multi-computer system which is loose coupling based on distributed-memory and network transmission, but also the architectural features of multiprocessor system which is tight coupling based on shared storage and bus transaction.

From the analysis of 2.2 for Petri net parallelism, we knows that Petri net system after functional division can be divided into several parallel processes and each process can be also divided into some parallel sub-processes. Here we can traverse to the sub-process as a thread. In view of this hierarchical division of Petri net system, this article selects multi-core cluster architecture as the basis of Petri net parallel model, as shown in figure 1.

In figure 1 nodes connect by switches and uses the star topology to facilitate the expansion and shrink of cluster size, and choose multi-core computers of the unified structure is used, and nods communicate by using the message passing mechanism. Between multi-core, it uses the shared memory model. Choosing one computer as the main node, other nodes in the cluster are slave the node.

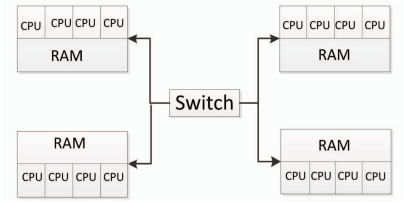


Figure .1 Multi-core clusters model of Petri net

In this paper, no storage consuming process because the parallel calculation increases. As the inter-cluster using a distributed storage model, and each node stores only a single parallel process and it's outcome，multi-core shared storage between uses, so the total cost of storage is the sum of each process takes with sequence running, while parallel computing and the storage cost of using serial computing only have a very small difference.

B. The parallelization model of Petri net

Given parallel computing problems achieve the desired effect they need, then the load balancing problem is the primary problem they should be paid attention to. The scales of the parallel process problem that have achieved by using of P-invariants technology to divide Petri network is often vary greatly. So it is necessary to design a model which not only meet the need of the Petri's hierarchy parallel, but also realize the load balance in the process of allocation. Based on these two requirements above, in this paper, we put forward the parallelization model of Petri net, which has been showed in figure 2.

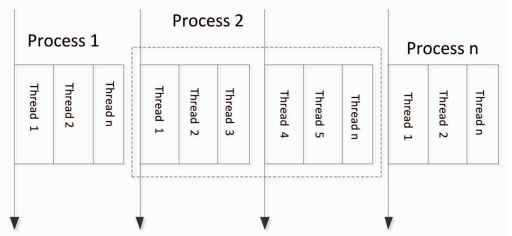


Figure. 2 The parallelization model of Petri net

The model has been showed in Figure 2, the number one process refers to the main process,and the number 2 and n processes refer to the slave process respectively. The main process is responsible for assigning tasks to each process, while the number two and three processes receive

tasks from it and summarize the results they get to the main process. The goal of the process 2 is get load balancing by the method of combining with two nodes to a complex process. Beside each process can be divided into several threads, which parallel execution on the multi-cores computer.

The mapping of process of Petri net in the parallel model can be divided into the following several cases:

(1) There are only one precursor and successor of the place, and the precursor and successor must be mapped to a process.

(2) The shared transition could be as a server and maps to the main process with playing a part in information interaction.

(3) The parallel process are mapped into several different slave processes, of which numbers are allocated by the master process accordingly.

(4) If the process scale is far higher than other process, then, according to the complexity of the problem. Combine with amount of nodes to form a node and generate a nested computer cluster. And the process is assigned to the master node of the small cluster, and then each sub-process is assigned to each slave node of the small cluster.

(5) The transition that isn't in any parallel process is mapped to the master process.

C. The programming model of Petri network parallelization model

Hybrid programming model of MPI+OpenMP has the qualities of reducing the communication cost, improving the scalability of MPI code, achieving communication and computation overlapping, improving the computational efficiency of each node and making full use of CPU as so on. Based on the parallel model of the multi-core cluster and multistage allocation strategies of both processes and threads, the programmatic way in the parallel model of Petri net adopt the MPI+OpenMP hybrid programming model, to take full use of the parallel mechanism between Intra node and nodes. Between the nodes use MPI function library based on message passing and intra node adopts the OpenMP function libraries of shared model. In order to realize the load balance of parallel programs, adding counting function in each parallel process to count the number of each sub-process within the process, if the ratio of the number of the process's sub-processes count with the average amount of each process is higher than 25%, then multiple nodes are allocated to it. The function of decision and distributing the number of nodes have been showed in formula (3.1)-(3.2):

$$\eta = \frac{C_i - \frac{C_1 + C_2 ... + C_n}{n}}{C_i} \times 100\% \tag{3.1}$$

$$\varepsilon = \begin{cases} \lceil \eta - 25\% \rceil & \eta > 1.25 \\ 1 & \eta \le 1.25 \end{cases} \tag{3.2}$$

Among them, Ci denotes the number of threads, n denotes the number of process, ε denotes number of nodes that each process needs, $\lceil \eta - 25\% \rceil$ denotes for η-25% on rounding.

IV. THE PARALLEL ALGORITHM OF PETRI NETS BASED ONMULIT-CORE CLUSTERS

According to Petri parallelization model mentioned in Section 3, the Petri process parallelization mapping algorithm is designed as follows:

Input: Input matrix and output matrix of Petri nets

Output: Parallel processes of Petri nets

Start:

Step 1:According to P-invariant technology, the Petri nets is divided by it's function and get several parallel processes in net system.

Step 2: If there are shared transition and aren't shared places existed between divided processes, then regard the transitions as a server.

Step 3: If the shared places pi exists in division processes, then change it into two parallel processes by using corresponding method, according to whether the input and output number of change is 1 or not.

Step 4: Iteratively divide the divided parallel process into small process according to step 1 to step 3, until the parallel process can't be divided.

Step 5: The transition in the same process marked it the same label as the process label, the sub-process which exist in the same thread is marked it the same label with the thread label. If the transitions is not in any parallel process, then mark it label with the master process label. For the shared changes, mark it label with the master process and server label.

Step 6: The search start from initial point of the Petri net, meantime add the thread count function and use the function to count the number of different thread label with

the same process label.

Step 7: Mapping the transitions of the Petri net to each node. If the transitions label is the main process label, then map it to the master node.

Step 8: If the process label of transitions is not the main process label, then judge the number of threads. If the number of threads is not greater than 1.25 times of the average number of threads, then the process map to one idle node randomly. Otherwise, judge the number of idle nodes of processes needed by using the [η -25%]. It should possibility map the threads of the process to each node evenly, and add the same vice process label to the thread, which will be assigned to the same node.

Step 9: Allocate threads of the process that have mapped into each node. Map the threads with the same thread label into the same core.

Step 10: Summary and output the allocation results of each process.

end

V. THE APPLIATION EXAMLE AND EXPERIMENTAL RESULTS

Finally, use a simulation system of students assessment to examine the parallelization model of Petri net. As shown in figure 3 which is the picture of data stream of this system .

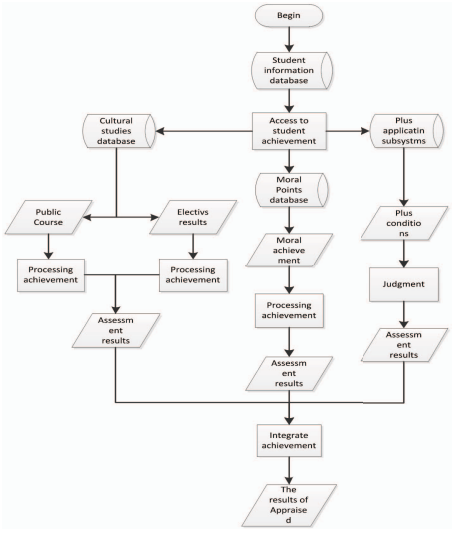


Figure. 3 The diagram of student assessment system's data stream.

According to parameters and structure of a practical problem, we structure the above problem into system model of Petri net and have a parallel partition for it, as shown in figure 4:of which P_0 represents student login information,P_1, P_2 and P_3 respectively represents database of academic grades , database of moral education grades ,database of application for bonus points,P_4, P_5 respectively represents system of general courses' grades ,system of elective courses' grades ,P_6 and P_7 represent grades of moral education,P_8, P_9 represent conditions of application for bonus points ,P_{10}, P_{11} represent grades of general courses,P_{12}, P_{13} represent grades of elective courses,P_{14} represents moral education grades ,P_{15} represent judgment result of whether the moral education meet the standard of being appraised to be the best, P_{16}, P_{17} represent the result of application for bonus points,P_{18}, P_{20} represent judgment result of whether general courses and elective courses meet the standard of being appraised to be the best or not,P_{19}, P_{21} represent grade point average of general courses, elective courses ,P_{22}, P_{23}, P_{24}, P_{25} represent final result of moral education grades, bonus points , general courses, elective courses,P_{26} represents cultural studies final grade,P_{27} represents the result of assessment.T_0 represent login assessment system,T_1 represent obtaining data of academic grades ,T_2 represents getting moral education grades,T_3 represents getting conditions of application for bonus points, T_4, T_5 represent screening general course grades, elective course grades, T_6, T_7 represent summation of moral education grades ,judging whether the moral education grades meet the requirements or not, T_8, T_9 represent auditing conditions for application for bonus points, T_{10}, T_{12} represent averaging the grades of general course and elective course, T_{11}, T_{13} represent judging whether the general course, cultural course meet the assessment standard or not, T_{14} ,T_{15},T_{16},T_{17} represent calculating the final grades of moral education, application for bonus points, general course, elective course,T_{18} represent summary culture results,T_{19} represent accumulating all the grades .

The part which is encircled by dotted box in the figure represents division of three parallelism process 1, 2, 3,among them process 1 can be divided into $T_4T_{10}T_{11}T_{16}$ and$T_5T_{12}T_{13}T_{17}$, process 2 can be divided into $P_6T_6P_{14}$, $P_7T_7P_{15}$, process 3 can be divided into $P_8T_8T_{16}$, $P_9T_9P_{17}$. the thread which is divided from process P1can be divided into $P_{10}T_{10}P_{18}$, $P_{11}T_{11}P_{19}$, $P_{12}T_{12}T_{20}$, $P_{13}T_{13}P_{21}$. Using formula 5 and 6 we can get the nodes' number of process 1

is 2, and the nodes' number of process 2, 3 is 1, So process 1, 2, 3 respectively runs in node of Petri's Parallelization model , and process 1 takes up two nodes, other changes and places are realized within the master process.

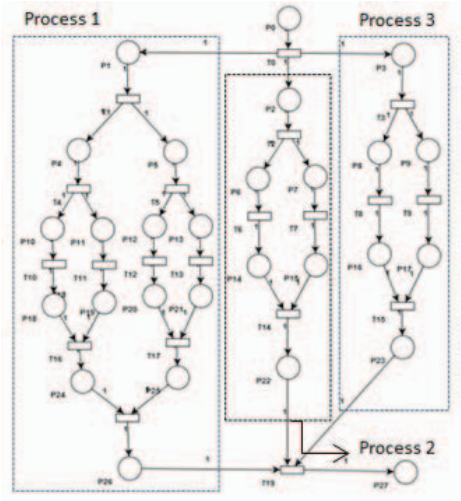


Figure. 4 Petri net system model of students assessment

According to the result of Petri net's parallel programming and situation of nodes mapping to programmed practical problems, make a parallel programmer on divided practical problems .Under the SHUGUAN cloud platform system, use four computers which has dual-core CPU and memory is 2G with windows7,Use hybrid MPI+OpenMP programming model, and use the Java programming language to realize the simulation experiment. The experimental data is randomly produced from 100 students' information and extracting 30 from them on the equal probability to conduct experiment, the result is the same as the default. The experiment's running time and the serial time are as figure 5 shown:

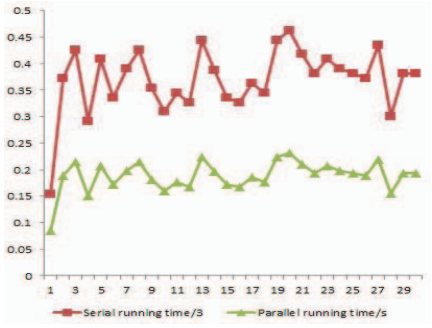


Figure .5Time line chart of serial and parallel

The average time and speedup as table 1 shown :

TABLE. 1 THE EXPERIMRNTAL SPEEDUP

the average time used by Serial procedures / s	the average time used by Parallel program / s	speedup
0.369831	0.18811	1.966036

In the figure 5, abscissa represents number of experiment, ordinate represents the running time. As we can see from the result of the experiment, the speedup is about 2 and the efficiency of Petri net's parallel computation is about half of the serial time's, which shows high computation efficiency. But the ideal speedup should be approximately 4, The causes of this situation is that there is a great gap between the problem size of each thread. This article is failed to solve this problem which need further efforts.

VI. CONCLUSION

This paper discusses the parallel model of Petri net based on multi-core cluster in detail and make the experimental verification. Thus the study of the parallel of the Petri systems is from theory to the practical and it has a significant effect in improving the efficiency of system operation. The problem that there are a large gap among the scale of the problem of each thread and which impact of parallel efficiency is our next step to focus on research.

REFERENCES

[1] WenJingLi, WeiZhiLiao,and RuLiangWang, "Functional partitioning and parallel algorithms of Petri net system," Computer Engineering,vol.35(21), pp.48-50,2009.

[2] YingLin, ZhengMeng, "The research and Implementation of a thread-scheduling algorithm under Multicore," Computer Technology and Development, , vol .23 (10) ,pp.19-26,2013.

[3] ZheHuiWu, "Introduction to Petri Ne［M］,"Beijing Machinery Industry Press,2006.

[4] ChangLingWu, "The parallel Algorithms Research of the three-dimensional FDTD Based on MPI and OpenMP," Institute of Electrical and Electronic Engineering in Huazhong University of Science and Technology ,pp.10-25,2009.

[5] L.M.Kristensen and M. Westergaard,"Automatic Structure-Based Code Generation from Coloured Petri Nets,"A Proof of Concept. In Proc. Of FMICS'10. LNCS,pp.215–230, 2010.

[6] WeiPan, LiaoYuanChen and JinHuaZhang, "The MPI + OpenMP hybrid programming model Based on SMP Cluster , "Application Research of Computers, vol.26(12),pp.4592-4594,2009.

[7] ZhiJiaLiu, WenJingLi and RuLiangWang, "Petri nets sharing synthesis and its application in parallel systems,"Computer Engineering and Design,vol.32(3),pp.968-983,2011.

[8] LinC,ChaudhuryA and WhistonAB, "etal.Log-ealInferenee of hornelauses in Petri net models JI,"IEEETrans.onKnowledgeand DataEngineering, vol.5(3),pp.416-425,1993.

Computing Green's Function for the Free Water Surface Near Ship with large parameter by using parallel computer

Xin Chen, Dan Li, Shesheng Zhang
Association of Mathematical Modeling, Wuhan University of Technology, Wuhan, 430070, China
E-mail: Dan li, lidan0520@163.com

***Abstract*— Near ship water field, computing Green's Function with large parameter is discussed, the differential equation and boundary condition are considered by using Dirac delta function, and analyzing solution is represented as integration. The convergence series is obtain for large parameter, and parallel computing results show that convergence speed is high.**

Keywords- Green's Function, free surface, representation.

I. INTRODUCTION

Green's Function is often used in the ship research. Add mass is used as we research for ship moving on the water. Such add mass is calculated usually by using boundary element method(BEM). if free water surface is considered, the kernel function expression form of the boundary integral become complicated. The kernel function is often called Green's Function in the case of that the singular point is point source. The Green's Function with free surface condition has two singular integral points, which leads to huge calculation and larger error. With the rapid improvement of the computer software and hardware, lots of scholars take researches on the Green's Function. Huan[1] proposes that concisely and precisely obtains the time-domain Green's Function and its spatial derivative is the key of the ship hydrodynamics problems. Han[2] uses multi-dimensional polynomial approximation to replace the direct numerical calculation with integral form, and this replacement can be adopted in the boundary element method calculation. Dan[3] takes some researches on the two-dimensional time-domain Green's Function and its partial derivatives calculation in ship hydrodynamic issues, proposes a new expression form with these two functions, and a new creating table interpolation method to obtain function results. Xie[4] takes integration on the limited depth complex Green's Function, and the result agrees well with the result from the Norway DNV Classification Society SEAM software. Liu[5] separate the part got by direct calculation from the integral function by reduction of a fraction. This procedure reduces the order of the left part in the integral function and reduces the calculation by using the unlimited depth Green's Function theory. Xie[6] combine the three-dimensional potential flow theory with limited depth complex Green's Function, and calculates the water elastic response of the Floating Production Storage & Offloading(FPSO). Shen[7] proposes the ordinary differential equations about depth Green's Function and its derivative, and a rapid Green's Function calculation method combining solving ordinary differential and interpolation between nodes. Liu[8] proposes a convolution calculation recurrence formula by using the Fourier transform relation between time-domain Green's Function and frequency-domain Green's Function. This method largely reduces the calculation difficulty. In all Green's Function theory, the calculating CUP time becomes longer as parameter becomes larger[9]. There are no answers to the problem until now.

Because of the rapid development of ship transportation, it is necessary to consider the numerical method of the Green's Function with large parameter.

II. TWO DIMENSIONAL GREEN'S FUNCTION

Suppose velocity potential φ satisfies Laplace equation

$$\Delta\phi = \delta(P-Q) \qquad (2.1)$$

Here P is field point Z=x+iy，Q is source point ζ = ξ +i η . The right of equation is Dirac delta function. The boundary condition is：

$$\begin{aligned} &\frac{\partial\phi}{\partial y} - k\phi = 0, y = 0 \\ &\frac{\partial\phi}{\partial x} \pm ik\phi = 0 \qquad x = \infty \end{aligned} \qquad (2.2)$$

Here is complex potential. On the free surface y=0, velocity potential satisfies linear condition. The analysis solution is called Green's Function as:

$$\begin{aligned} &G(z,\zeta) = \frac{1}{2\pi}\ln\frac{Z-\zeta}{Z-\bar{\zeta}} - \frac{1}{\pi}I + i\exp(-ik(Z-\bar{\zeta}) \\ &I = PV\int_0^\infty \frac{\exp(-iu(Z-\bar{\zeta})}{u-k}du \end{aligned} \qquad (2.3)$$

Here I is principle value integration and u=k is zero point of denominator. The integral domain is infinite. From the expression of integration, the value I is determined by the parameters of k,(x-ξ),(y+η). Let unit transfer as

$$X = -i(Z-\bar{\zeta})/R \qquad R = |Z-\bar{\zeta}| \qquad (2.4)$$

The principle value integration may be written as

$$I(a) = \int_0^\infty \frac{\exp(uRX)}{u-k}du = \int_0^\infty \frac{\exp(vX)}{v-kR}dv = H(kR,X) \qquad (2.5)$$

Where v=uR, H(a,b) is two parameters function defined as:

$$H(a,X) = \int_0^\infty \frac{\exp(vX)}{v-a}dv \qquad (2.6)$$

There is an infinite integral boundary, and one singular point v=a. They make more calculation, or long CPU calculating time. The CPU time will become longer if parameter a become lager.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.40

III. THE DIFFERENCE METHOD

According to the expression of Green's Function, we have another form representation of H function:

$$H(a,\theta)=\int_0^{\infty}\frac{\exp(vX)}{v-a}dv \qquad (3.1)$$

$$X=-ie^{i\theta}=-e^{i\delta} \qquad \delta=\theta-1.5\pi$$

It is easy to obtain a difference equation as below

$$H_\theta = iaXH - i$$
$$H_0 = H(a,\theta=1.5\pi) \qquad (3.2)$$

The above equation is calculated by using forth order R-K method

$$H_{n+1}=H_n+\frac{h}{6}[K_1+2K_2+2K_3+K_4] \qquad (3.3)$$

Here

$$K_1 = iaXH_n - i$$
$$K_2 = iaX[H_n + K_1h/2] - i$$
$$K_3 = iaX[H_n + K_2h/2] - i$$
$$K_4 = iaX[H_n + K_3h] - i$$

and h is step length, and

$$\theta_n = nh - 1.5\pi \qquad \theta_{n+0.5}=\theta_n+0.5h$$
$$H_n = H(a,\theta_n) \qquad (3.4)$$

As θ =1.5 π , or X=-1， we have initial value

$$\theta_n = nh - 1.5\pi \qquad \theta_{n+0.5}=\theta_n+0.5h$$
$$H_n = H(a,\theta_n) \qquad (3.5)$$

It is real function integral, and can be rewritten as

$$H_0=\int_0^{\infty}\frac{\exp(-va)}{v-1}dv \qquad (3.6)$$

IV. REAL PARAMETER CALCULATION

In this section, we will discuss integral with real function.

If θ =1.5 π , or X=-1， from above section, we have

$$H(r,a)=-\int_0^{r}\exp(-va)\sum_{k=0}^{\infty}v^k dv=-\sum_{k=0}^{\infty}A_k \qquad (4.1)$$

Here r=1-m/a, and

$$A_k=\int_0^{r}v^k\exp(-va)dv \qquad (4.2)$$

Lemma 1，if r<1，and k→∞，we have A_k→0

Proof: because of r<1, v<1, when k→∞, there is limit value v^k→0, according to the integral expression, we have A_k→0. #

From the expansion, it's easy to derived the convergence theorem of series：

Theorem 1， if r<1， we have (1) $|A_{n+1}|<|A_n|<1$， and (2) $A_1+A_2+A_3+...$is the convergent series.

On the basis of recursive formula: if n>0, there is:

$$A_n=\frac{-r^n}{a}\exp(-ra)+\frac{n}{a}A_{n-1} \qquad (4.3)$$

When n>a, we can use this recursive formula to calculate

$$A_{n-1}=\frac{r^n}{n}\exp(-ra)+\frac{a}{n}A_n \qquad (4.4)$$

By the above formula, getting the following theorem is easy：

Theorem 2: when r<1, there are recursive formulas：

$$A_0=\frac{1}{a}[1-\exp(-ra)]$$
$$A_n=\frac{-r^n}{a}\exp(-ra)+\frac{n}{a}A_{n-1} \qquad (4.5).$$

Table I, A_n is varied with large *a*, *r*=1-10/*a*

a=15	*a*=20	*a*=30	*a*=40
0.0042652	0.002499	0.0011116	0.00062733
0.00051873	0.00024931	7.41*E*-05	3.13*E*-05
8.71*E*-05	3.71*E*-05	7.41*E*-06	2.34*E*-06
1.77*E*-05	7.28*E*-06	9.88*E*-07	2.34*E*-07
4.05*E*-06	1.75*E*-06	1.65*E*-07	2.93*E*-08
1.00*E*-06	4.89*E*-07	3.29*E*-08	4.39*E*-09
2.62*E*-07	1.54*E*-07	7.68*E*-09	7.69*E*-10
7.14*E*-08	5.25*E*-08	2.04*E*-09	1.54*E*-10
2.00*E*-08	1.92*E*-08	6.11*E*-10	3.46*E*-11
5.74*E*-09	7.39*E*-09	2.03*E*-10	8.65*E*-12

V. PARALLEL COMPUTING RESULTS

The steps of parallel computation are:

(1) divide domain 0<a<A to P sub-domain, and choose P computers;

(2) Every sub-domain, there is a computer to calculate Green's function.

(3) Output calculating results.

This paper choose P=20 computers to calculate Green's function.

The coefficient A_n are calculated for large parameter *a*, and r=1-10/*a*. The results are shown in the table I. From table I, the ratio $[A_n/A_{n-1}]$ is about 0.5. The figure 1 shows that $\log(A_n)$ varied with sub-scribed n, the parameter *a*=40, r=1-10/*a*. From Fig.1, the value of $\log(A_n)$ is decrease with n, and near a linear function.

The real part real(H) of H function is varied with parameter a is drawn in Fig.2 for δ=-1.428, -0.8568, -0.571. From Fig.2, the amplitude of line is larger as parameter δ become smaller.

VI. CONCLUSIONS

In this paper, the calculating method of Green's Function with large parameter is discussed. The differential control equation is build near ship water field by using delta function, and analyzing solution is represented as integration. The convergence series is obtained for large parameter, and results show that convergence speed is high.

ACKNOWLEDGMENT

The paper is financially supported by China national natural science foundation (No.51139005)，and Humanity and Social Science foundation of Ministry of Education of China (Grant No.12YJAZH022), and the Fundamental Research Funds for the Central Universities(NO:2012-Ia-045,2014-Ia-038).

REFERENCES

[1] Huan Debo, time domain Green's Function and its derive numeric calculation[J], China Ship building, 1992(4),1

[2] Han Ling, Teng Bin, Guo Ying, Approximation of time domain Green's Function[J], J of hydrodynamics (A),2004(5), 929-637.

[3] Dan Wenyang, Dai Yishang, Numerical evolution of two dimensional time domain Green's Function[J], J of hydrodynamics,1996(6), 331-335

[4] Xie Yonghe, et all, Numerical calculation of finite water depth composite Green's Function[J],J of Ship Mech.,2005(1), 23-28

[5] Liu Ri-rrdng,Ren Hui-long,Li Hui, A nimproved Gauss-Laguerre method for finite water depth Green's Function and its derivatives[J], J Ship Mech.,2008(2),p188-197.

[6] Xie Yonghel，et al, The Effects of Water Depth on Hydroelastic Response of a Very Lager FPSO[J], J Shanghai Jiaotong Univ.,2006(6),p993-997

[7] Shen Liang，et al, A practical numerical method for deepwater time domain Green's Function[J], J of hydrodynamics(A),2007(3),380-386

[8] Liu C.，et al, New convolution algorithm of time－domain Green's Function[J], J of hydrodynamics(A),2010(4), 25-34.

[9] Sun Z. et al, Green's Function theory and algorithm of hydrodynamics for ship section [J], Journal of Wuhan University of Technology, Vol.36, No.1, 2014, 39-42

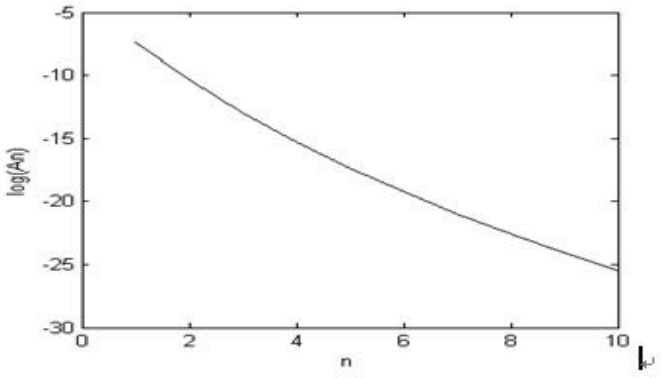


Figure 1. log(An) are varied with n，*a*=40; r=1-10/*a*

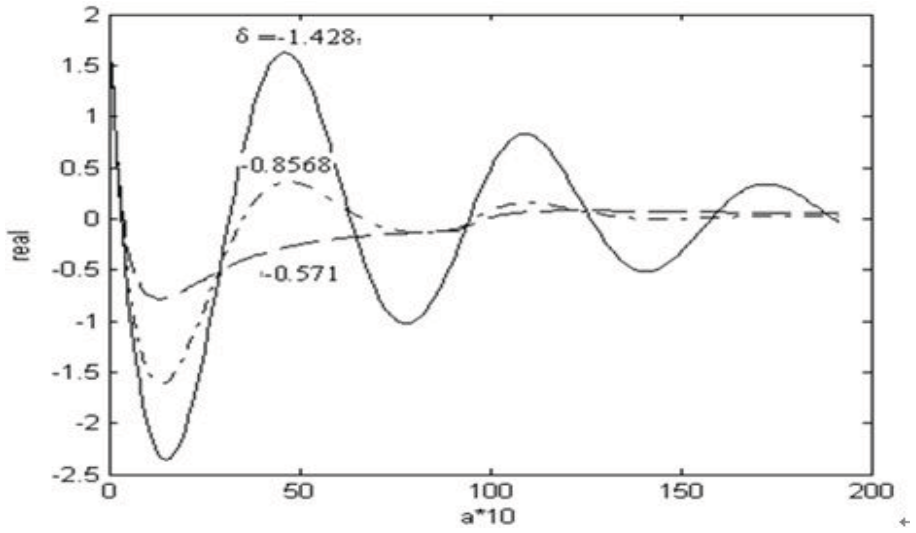


Figure 2. H of Green's Function varied with parameter *a*

A Novel Identity Authentication Scheme of Wireless Mesh Network Based on Improved Kerberos Protocol

Min Li [1], *Xin Lv* [*1], *Wei Song* [2], *Wenhuan Zhou* [1]
[1] College of Computer and Information, Hohai University, HHU
[2] Office of Jiangsu Province Flood Control and Drought Relief Headquarters
Nanjing, China
[*] Corresponding Author: lvxin.gs@163.com

Rongzhi Qi [1], *Huaizhi Su* [3]
[1] College of Computer and Information, Hohai University, HHU
[3] College of Water Conservancy and Hydropower Engineering, HHU
Nanjing, China

Abstract*—The traditional Kerberos protocol exists some limitations in achieving clock synchronization and storing key, meanwhile, it is vulnerable from password guessing attack and attacks caused by malicious software. In this paper, a new authentication scheme is proposed for wireless mesh network. By utilizing public key encryption techniques, the security of the proposed scheme is enhanced. Besides, timestamp in the traditional protocol is replaced by random numbers to implementation cost. The analysis shows that the improved authentication protocol is fit for wireless Mesh network, which can make identity authentication more secure and efficient.*

Keywords-Kerberos protocol; public key encryption; Wireless Mesh network; identity Authentication

I. INTRODUCTION

A. Background

With the increasing social requirements of flexible mobile communications services, and traditional network technologies are based on infrastructures which is unable to provide flexible mobile communication services, wireless network technology is developing rapidly nowadays. Different from the traditional wireless network, Wireless Mesh network (WMN)[1] is a new kind of technology, which allows each node to receive and transmit signals in the network. Compared to traditional wireless network, Wireless Mesh network has many advantages:1) non line of sight transmission expands the application field of wireless broadband[2]; 2)high transmission rate makes transmission distance relatively short; 3)high reliability; 4)faster network configuration and maintenance; 5)low cost. These advantages indicate that, WMN is a leap of wireless network technology which has a very broad application prospect. In wireless Mesh networks (Wireless Mesh Networks), nodes can be divided into three types according to their functions[3]: ① MP (Mesh Point), MP only supports the Mesh interconnection; ② MAP (Mesh Access Point), MAP supports Mesh interconnection and access;③MPP (Mesh Point with a Portal), MPP supports Mesh interconnection and network communication[4]. The WMN architecture is shown as figure 1.

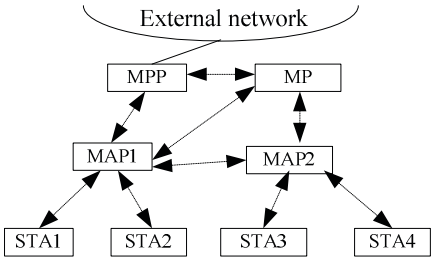


Figure 1. A Typical WMN Architecture

B. Kerberos Authentication Protocol

Kerberos is a key network authentication protocol developed by Massachusetts Institute of Technology (MIT). Based on KDC (key distribution center), a symmetric key encryption algorithm was utilized by this authentication protocol. Then, through the trusted third party KDC, both sides of network communication can be identified mutually. This authentication mechanism does not depend on the operating system and the address of the host. Kerberos identity authentication system includes a range of services, in addition to the user C, the following three parts are also included: ① Authentication server (AS)[5]. AS is used to verify the identity of the user C when login, and pass the identity authorization bill TGT to authenticated user, which is used to prove the identity to service authorization server TGS[6].②Service authorization server (TGS).Service authorization server (TGS) provides service access ticket to user C who has already been authenticated. With this service access ticket user C can apply for access to other services. ③Application Server. The application server is the final executor of service[7].The Kerberos authentication protocol process is shown as Fig. 2.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.41

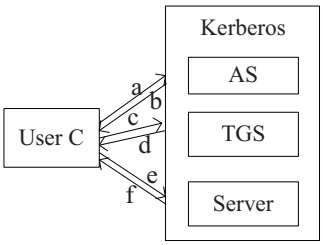


Figure 2. Flow Chart of Kerberos Authentication Protocol

The principle of Kerberos protocol is generally divided into three phases, and each phase has two steps. User C sends a message to the AS server to request an identity authorization bill TGT, which will be sent back to the user C after the secret key encryption by user. User asks the service authorization server for service access bills which will be used as proof to request permission to access specific server. The server can be accessed by the user if the identity of the user is secure[8].The specific process of Kerberos protocol is shown as follows:

(1) User C asks the authentication server AS for certification services; user C gets the identity authorization bill:

$$C \rightarrow AS : \mathrm{ID}_C \,||\, Times \,||\, Nonce_1 \,||\, ID_{tgs}$$

$$AS \rightarrow C:$$

$$ID_C \,||\, Ticket_{tgs} \,||\, E_{Kc}\left[K_{C,tgs} \,||\, Nonce_1 \,||\, ID_{tgs} \,||\, Times\right]$$

$$Ticket_{tgs} = E_{Ktgs}\left[K_{C,tgs} \,||\, AD_C \,||\, ID_C \,||\, Times\right]$$

(2) User C apply for Service authorization from TGS; user C receive Service authorization paper:

$$C \rightarrow TGS:$$

$$ID_V \,||\, Nonce_2 \,||\, Times \,||\, Authenticator_C \,||\, Ticket_{tgs}$$

$$TGS \rightarrow C : \mathrm{Ticket}_V \,||\, ID_C \,||\, E_{Kc,tgs}\begin{bmatrix} K_{C,V} \,||\, Nonce2 \,|| \\ Times \,||\, ID_V \end{bmatrix}$$

$$Ticket_{tgs} = E_{Ktgs}\left[ID_C \,||\, AD_C \,||\, Times \,||\, K_{C,tgs}\right]$$

$$Ticket_V = E_{Kv}\left[ID_C \,||\, AD_C \,||\, Times \,||\, K_{C,V}\right]$$

$$Authenticator_C = E_{K\,\mathrm{c,tgs}}\left[TS_1 \,||\, ID_C\right]$$

(3) User C asks the application server for access services; user C obtains access permissions:

$$C \rightarrow V : Authenticator_C \,||\, Ticket_V$$

$$V \rightarrow C : E_{Kc,v}\left[Subkey \,||\, Seq*\right]$$

$$Ticket_V = E_{Kv}\left[ID_C \,||\, AD_C \,||\, Times \,||\, K_{C,V}\right]$$

$$Authenticator_C = E_{Kc,v}\left[TS_2 \,||\, Subkey \,||\, ID_C \,||\, Seq*\right]$$

C. Kerberos-based WMN Identity Authentication Mechanism

In wireless Mesh network (WMN), firstly, the server will authenticate the identity of each new node MP. When the authentication is passed, the new node MP will be allowed to access the network. At the same time, MP will be authorized to access. A mutual trust relationship will be established between MP and KNAS, and the secret key will be shared. In the initial authentication phase, single sign-on (sso), unified authorization, centralized authentication methods are used to divide identity authentication and access authorization into two parts. The authentication and authorization entity complete the identity authentication and access authentication of MP. KNAS on the MPP Node is a network access server. KNAS, service authorization server and the authentication server make up the authenticator system together. AS is used to verify the identity of user C, and pass the identity authorization bill TGT to authenticated user, which is used to prove the identity to service authorization server TGS. As long as the new node MP does not leave the network, TGT will be used for a long time, thus greatly shorten the interactive authentication process in subsequent periodic access. After the authentication, TGS will inspect the authorization bill of MP. If MP passes the inspection, it will be allowed to access the network and receive an access service bill, including a key shared by each MPP node. Thus, the new node MP can also continue to interconnect with the neighbor nodes. Wireless Mesh network security mechanism structure is shown as figure 3.

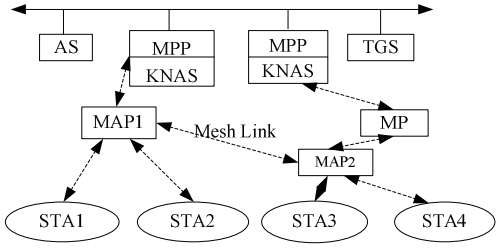


Figure 3. WMN Security Mechanism Architecture

D. The Limitations of Kerberos Protocol

Although Kerberos protocol is the major identity authentication protocol at present, and to a certain extent, Kerberos protocol can guarantee the security of the network. Though it has incomparable advantages and strong practicability, Kerberos protocol still has drawbacks and defects of its own.

- **Clock synchronization.** Timestamp was added to the bills and certifications in the Kerberos protocol, which is used to solve the problem of replay attack. Only when the timestamp differs little, data can be considered as valid. This requires the machine time of user C, authentication server (AS) and service authorization server be roughly the same. However, in the distributed network environment, such precise time requirement is difficult to achieve. At the same time, attackers could take advantage of this feature, too. Once the bill is got and then sent immediately, it will be difficult to be searched within prescribed period of time, thus increases the potential hazards of replay attack.
- **Password guessing attack.** In Kerberos protocol, the message authentication server (AS) sent to user C is encrypted by the secret key EKc of user C, while EKc derives from user's password which is encrypted by Hash function. So the attacker may collect a large number of TGS bills, calculate and analyze the user password to steal users' information.
- **Key storage.** Symmetric key algorithm is used in Kerberos, which requires establishing a high standard maintenance mechanism to share the secret key. With the expansion of Internet network and the growing number of the shared secret keys, the problems of

storage, management, maintenance of the secret key are difficult to solve[9].

- **Malicious software attacks.** The security requirements for Kerberos software are high in Kerberos authentication protocol, if security is not high enough, the attacker may execute Kerberos protocol and record the user's password to replace the Kerberos software that user uses, leading to attack.

II. WMN IDENTITY AUTHENTICATION MECHANISM BASED ON THE IMPROVED KERBEROS

A. The Improved Kerberos protocol

The Kerberos protocol model discussed in this paper is shown as Figure 4. It is similar to the traditional Kerberos protocol mode, specific process is shown as below:

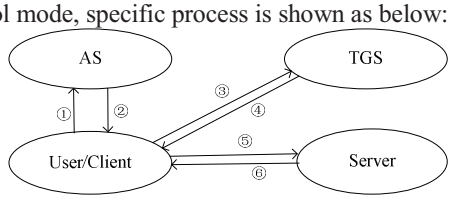


Figure 4. The Schematic Diagram of The Improved Kerberos Protocol

$\Pr vK_X$ denotes private key for X; $PubK_X$ denotes public key for X; $K_{X,Y}$ denotes shared key for X and Y; $\{m\}_k$ denotes encrypt q $\{m\}$ by k.

(1) $MP \rightarrow AS: \{\{MP, AS, TGS, K_{MP,AS}\} \mathrm{PrvK}_{MP}\} PubK_{AS}$

(2) $AS \rightarrow MP: \{\{MP, K_{MP,tgs}\} K_{MP,tgs}, \{T_{MP,tgs}\} PubK_{MP}\}$

$T_{MP,tgs}: \{AS, MP, TGS, Addr, Lifetime, K_{,\mathrm{MP,tgs}}\} \Pr vK_{AS}$

(3) $MP \rightarrow TGS: \{A_{MP,tgs}, \{T_{MP,tgs}\} PubK_{tgs}\}$

$A_{MP,tgs} = \{MP, MPP, Addr, Lifetime, N_{MP}\} K_{MP,tgs}$

(4) $TGS \rightarrow MP: \{\{T_{MP,MPP}, N_{MP}\} K_{MP,tgs}\}$

$T_{MP,MPP}: \{TGS, MP, MPP, K_{MP,MPP}, Lifetime\} \Pr vK_{tgs}$

(5) $MP \rightarrow MPP: \{\{MPP, MP, N_{MPP}\} K_{MP,MPP}, T_{MP,MPP}\} PubK_{MPP}$

(6) $MPP \rightarrow MP: \{MP, MPP, N_{MPP}\} K_{MP,MPP}$

In the improved Kerberos protocol, each new access node MP has a pair of asymmetric secret key. The open key of the new node MP can be open to all people, but the private key of the new node MP is kept by itself; A random number was introduced to replace the timestamp in the traditional Kerberos protocol in (3)(4)(5)(6) steps.

B. The Improved WMN Identity Authentication Mechanism Based on Kerberos

In order to solve the problems in the traditional Kerberos protocol, such as clock synchronization, password guessing, key storage, malicious software attacks, Smart Cards was introduced into Kerberos protocol, which can effectively solve the problem of password guessing. However, in order to support the technology of Smart Cards, new hardware device should be added to the system at the same time[10]. The problem of clock synchronization can be solved by serial number cycle mechanisms in Kerberos protocol, and the original timestamp will be replaced by the random number which is generated from user C. In a certain degree, it can solve the problem of replay attack, however, this may change the structure of the Kerberos protocol. In addition, Kerberos protocol also can be improved by public key cryptography, which is another research hotspot at present. This method can efficiently solve the problems of excessive consumption of key storage management. However, drawbacks are also obvious that it takes more encryption and decryption time of public key cryptosystem than symmetric key cryptosystem.

Based on the state of art above, a new solution based on traditional Kerberos protocol was proposed in this paper, the specific process is as below:

1. New node MP sends a message to the authentication server AS to request certification service and apply for certification authorization bill TGT, which is used to prove the identity of MP to TGS. The request message includes MP's name, AS's name, service authorization server's name, shared cryptographic key $K_{MP,AS}$ between AS and MP. The shared encryption key $K_{MP,AS}$ is generated from MP randomly, which replaces the function of timestamp in traditional Kerberos protocol.

After the certification of MP by AS, the response message will be sent back to MP. The response message will be encrypted by $K_{MP,AS}$ which ensures that the response message is generated from the AS. The message that MP sends to AS is digitally signed with the private key $\Pr vK_{MP}$ of MP, which can prove that the message is generated from MP. Then the message will be encrypted with the public key of AS, which guarantees that the message only can be decrypted by authentication server AS.

2. When authentication server AS receives the message that MP sent, the message will be decrypted by AS first, the decryption process will be accomplished by utilizing the encryption and decryption key of AS. After decryption, the authentication message will be inspected and signed by the public key in the certificate comes from MP by AS. If the signature and authentication passed, AS will response to the MP by sending a message. The response message contains the shared key $K_{MP,tgs}$ between MP and TGS. The shared key $K_{MP,tgs}$ is generated by the authentication server AS randomly, and is encrypted with the shared key $K_{MP,as}$ between MP and AS, which ensures the security of the shared key $K_{MP,tgs}$. At the same time, the message also contains an identity authorization bill $T_{MP,tgs}$, which will be sent to the service authorization server TGS to prove the access permissions. The identity authorization bill $T_{MP,tgs}$ contains the AS's name, the shared key $K_{MP,tgs}$ between MP and TGS, MP's name, TGS's name, MP's address Addr, and the effective time Lifetime of the identity authorization bill $T_{MP,tgs}$ of MP. The authorization bills $T_{MP,tgs}$ of MP is signed by the private key of AS, then the bill will be encrypted by the public

key $PubK_{MP}$ of MP, which ensures $T_{MP,tgs}$ be more secure.

When the new node MP receives messages from the authentication server AS, $\{MP, K_{MP,tgs}\}K_{MP,as}$ will be decrypted with the shared encryption key $K_{MP,as}$ between the MP and AS. After decryption, MP's name and the shared encryption key between MP and TGS will be obtained. $\{T_{MP,tgs}\}PubK_{MP}$ will be decrypted with its public key $PubK_{MP}$ by MP, and the digital signature of AS will also be removed , thus $K_{MP,tgs}$ will be acquired. This two $K_{MP,tgs}$ will be compared to confirm whether they are same. If so, the shared encryption key $K_{MP,tgs}$ between MP and TGS will be saved by MP. Then the shared encryption key will be utilized as the key to communicate between TGS and MP, and to visit TGS together with $T_{MP,tgs}$.

The identity authorization bill $T_{MP,tgs}$ sent by AS will be encrypted with public key of TGS, that is $AS \rightarrow MP:\{\{K_{MP,tgs}\}K_{MP,as},\{T_{MP,tgs}\}PubK_{tgs}\}$. Therefore, attackers may disguise as AS to send others' $T_{MP,tgs}$ or replay previous $T_{MP,tgs}$ to MP, which MP can't identify. In this paper, public key of MP is utilized to encrypt first, once MP receives response messages from AS, private key of its own will be utilized to decrypt the message upon. Then, comparing the decrypted information with that in the $T_{MP,tgs}$, it will be confirmed whether the messages is fake or replayed.

3. When accessing wireless Mesh network, new node MP will send a message to TGS first, apply for authorization bill for visiting TGS. The request message contains the authentication information $A_{MP,tgs}$ and identity authorization bill $T_{MP,tgs}$. Then public key of TGS will be utilized to encrypt the $T_{MP,tgs}$, which can ensure that MP's identity authorization bill $T_{MP,tgs}$ can only be decrypted by TGS, thus guarantee the security of $K_{MP,tgs}$ further. The authentication information $A_{MP,tgs}$ in request message contains the MP's name, node MPP's name, MP's address, the validity date of passport, and random number, etc. The authentication information is encrypted by the session key $K_{MP,tgs}$ between the new node MP and the TGS. When TGS receives the request message, $\{T_{MP,tgs}\}PubK_{tgs}$ be will decrypted with private key $\Pr vK_{tgs}$. At the same TGS will verify the digital signature of the AS inside $T_{MP,tgs}$. If the verification passes, it means that the new node MP can access the Internet. Then the TGS will receive the session key $K_{MP,tgs}$ between MP and TGS. Then, the authentication information $A_{MP,tgs}$ will be decrypted by the TGS with the session key. After comparing and analyzing the new node MP's name, address Addr, and with the corresponding information in $T_{MP,tgs}$, TGS will confirm whether the message sender is the new node MP marked by $T_{MP,tgs}$ or not.

4. When the new node MP receives a response message from the TGS, the response message will be decrypted with $K_{MP,tgs}$ by MP, then the random number information N_{MP} will be acquired. The random number N_{MP} will be compared with the random number N_{MP} that MP itself sent to the TGS. If the two numbers are equal, then it can be confirmed that this message is a new message. Then, the signatures in the passport will be verified by the new node MP, if the verification passes, then it can be proved that the passport is sent by TGS itself[11]. The bill $T_{MP,MPP}$ will be reserved by MP to obtain the right to access MPP for MP. In addition, the new node MP will reserve the session key between MP and MPP. When receiving a response message from TGS, MP will utilize his private key to decrypt the response information to gain the $T_{MPP,MP}$ and the random number N_{MP} . Then the new node MP will verify the signature of TGS. After the verification, the $K_{MP,MPP}$ will be saved and used as the shared key between MP and MPP.

5. When a new node MP apply for access to wireless Mesh network (WMN), the new node MP will send messages to the KNAS server on the MPP to request access service. The message contains the name of MPP, the name of MP, random number and the passport $T_{MP,MPP}$ requested from TGS, in which, MPP's name, MP's name and the random number are encrypted with the shared key between MP and MPP. After encryption, $K_{MP,MPP}$ will be encrypted with the public key once again. This double encryption can improve the security of the message.

When the KNAS server on the MPP receives a request message from the new node MP, the KNAS server will use its private key to decrypt the message to obtain permission $T_{MP,MPP}$. MPP will first verify the signature of $T_{MP,MPP}$. The pass of verification proves that the passport come from TGS, then MPP utilize $K_{MP,MPP}$ of $T_{MP,MPP}$ to decrypt $\{MPP, MP, N_{MPP}\}K_{MP,MPP}$. After that, MP's name and MPP's name will be compared with those that MP sent to the KNAS server. If they are the same, it shows that the sender of this message is the new node MP marked in the passport, by which MP will be identified [12]. At last, $K_{MP,MPP}$ in the $T_{MP,MPP}$ will be saved by MPP as the shared key between MP and MPP.

6. After the verification on MP by KNAS on the MPP, KNAS will send confirmation message to the new node MP. The confirmation message includes the name of the MPP, the name of MP and the random number. The message is encrypted with the shared key $K_{MP,MPP}$ between MP and MPP. The new node MP will decrypt the response information MPP sent with $K_{MP,MPP}$ .After decryption, MP will verify whether the MPP's name, MP's name and N_{MPP} are the same or not. If they are the same, it proves that MPP has permit MP to access Mesh wireless network, and MP has got the shared session key between MP and MPP. That means MP has passed the authentication, and

can interconnect with at least one MPP. Then both sides can communicate with each other with the shared key.

III. SECURITY ANALYSIS

In this paper, traditional Kerberos protocol was improved by adding public key cryptosystem, which in a certain extent overcome the shortcomings of traditional Kerberos, and improve the security of Wireless Mesh network identity authentication scheme. Compared with the traditional Kerberos protocol, the new Kerberos protocol can meet the security standards better. The new Kerberos protocol has the following characteristics:

a) Random number was introduced into the Kerberos protocol to replace the timestamp in the traditional Kerberos protocol, which avoids the clock synchronization problem in the network. When the new node MP receives the response information sent by TGS, MP will decrypt the response message with session key between MP and TGS to get Nc. Compare Nc with the random number that MP sent to TGS. If they are the same, it means that the message is new, not a retransmitted one, which can prevent the replay attacks.

b) In the Kerberos protocol discussed in this scheme, message is first encrypted with the private key of the sender and then the public key of receiver. Only after being decrypted with private key of the sender, can we know which public key should be used to get the final information; PIK technique was brought to guarantee the integrity of messages between MP and the server, which could alleviate the password guessing problem in the traditional Kerberos protocol.

c) In the improved Kerberos protocol, only public key of the new node MP is reserved in the authentication server AS. The private key of the new node MP is kept by MP itself. Therefore, even if the database is accessed illegally, not too much damage will be caused.

d) According to the improved Kerberos protocol, operations are taken only in the new node MP and authentication server AS. While in the sessions between MP and TGS, or MP and MPP, there isn't any operation. Thus, public key will not be involved into high-cost calculation. Therefore, execution efficiency will be improved. The requirements for the PKI (Public Key Infrastructure) are relatively low due to the limited quantities of the AS, TGS and MPP.

IV. CONCLUSIONS

In order to make up the shortage of the traditional Kerberos protocol, an improved authentication scheme was proposed in this paper. PKI techniques are introduced in the scheme to secure the process of authenticating, and random number is added in the interaction between entities in the scheme. The analysis indicates that the proposed scheme is practical in wireless mesh network.

In future work, we are focusing on the authentication in mobile cloud computing environment, more precisely, how to design the federal authentication protocol in cross-domain situation.

ACKNOWLEDGMENT

This paper is supported by National Natural Science Foundation of China: "Research on Trusted Technologies for The Terminals in The Distributed Network Environment" (Grant No. 60903018), "Research on the Security Technologies for Cloud Computing Platform" (Grant No. 61272543), and "The National Twelfth Five-Year Key Technology Research and Development Program of the Ministry of Science and Technology of China" (Grant No. 2013BAB06B04), "Key Technology Project of China Huaneng Group" (Grant No.HNKJ13-H17-04).

REFERENCES

[1] I. Akyildiz, and X. Wang, Wireless mesh networks, John Wiley & Sons, 2009.

[2] I.F. Akyildiz, X. Wang, and W. Wang, Wireless mesh networks: a survey, Computer networks, vol. 47, no. 4, 2005, pp. 445-487.

[3] W. Ze, W. Qi, L. Wenju, and K. Yongzhen, Scalable authentication protocol for wireless mesh network access, IEEE Press, Year Published, pp. 3051-3054.

[4] L. Wenju, S. Yuzhen, and W. Ze, A wireless mesh network authentication method based on identity based signature, IEEE, Year Published, pp. 1-4.

[5] B.C. Neuman, and T. Ts'O, Kerberos: An authentication service for computer networks, Communications Magazine, IEEE, vol. 32, no. 9, 1994, pp. 33-38.

[6] Wei He, The research of remote access VPN password authentication protocol based on improved Kerberos system, Zhejiang: Zhejiang University, 2003, pp.

[7] P.L. Hellewell, T.W. Van Der Horst, and K.E. Seamons, Extensible Pre-authentication Kerberos, IEEE, Year Published, pp. 201-210.

[8] S.T.F. Al-Janabi, and M. Rasheed, Public-Key Cryptography Enabled Kerberos Authentication, IEEE, Year Published, pp. 209-214.

[9] N.T. Abdelmajid, M.A. Hossain, S. Shepherd, and K. Mahmoud, Location-Based Kerberos Authentication Protocol, IEEE, Year Published, pp. 1099-1104.

[10] B. Wang, Y. Wang, and H. Zhang, A new secure password authentication scheme using smart cards, Wuhan University Journal of Natural Sciences, vol. 13, no. 6, 2008, pp. 739-743.

[11] Xilan Wu, Yuanzhong Shu, Zetao Jiang , and Zhihong Wu, An improved Kerberos protocol combined with PKI Technology, Computer Application and Software, vol. 26, no. 2, 2009, pp. 85-86.

[12] Peishun Liu, and Hongyu Liu, The research on the identity authentication technology of the ocean environment information of cloud computing, The Technology Journal of Huazhong University of Science (Natural Science Edition), vol. 1, 2012.

A Quantitative Analysis about the Cache Set-Level Utilization

Huang Zhibin, Zhou Feng

Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing China, 100876

huangzb@bupt.edu.cn, zfeng@bupt.edu.cn

***Abstract*—it has seen a decrease in the amount of cache per core in order to make space for more cores. The efficiency of cache space becomes an important factor. The set-level unbalanced utilization is analyzed under the fixed capacity when associativity is varied and extended by increasing the number of physical ways (referred as Direct Extension Method (DEM)). We evaluate 29 benchmarks from Spec Cpu2000/2006 in Simics, compare the set-level unbalanced utilization between different associativity quantitatively and explain the trade-off of selecting a proper associativity for the cache with a certain fixed capacity from the view of the set-level unbalanced utilization and workloads. Furthermore, we explain the inherent deficiency of DEM through the statistical analysis of three typical benchmarks' set-level unbalanced utilization.**

Keywords- set-level unbalanced utilization, associativity,way,

I. INTRODUCTION

As Multicore and Manycore architectures become the mainstream, the definite transistors are partitioned by cores and on-chip cache. The amount of cache per core is decreasing with the increasing of cores in order to make space for more cores [1]. Therefore, the efficiency of cache space becomes an important factor, especially for Last-Level cache. Associativity is one of the important parameters for the design style in cache. Its value and implementation method have a remarkable influence on cache space efficiency. In fact, the associativity of LLC in mainstream commercial processors is no more than 16, e.g. IBM POWER7 [2] has a 32MB 16-way shared L3, Oracle SPARC T5 has a 8MB 16-way shared L3 [3] 。

Higher associativity provides more flexibility in block (re)placement in the best possible manner. Moreover, some architectural proposals rely on highly associative caches. For instance, transactional memory and thread-level speculation [4], event monitoring and user-level interrupts [5] and memory consistency implementations utilize caches to buffer or pin specific data blocks. Nevertheless, different extension method of associativity has profound influence on cache delay, power, and capacity efficiency. Conventional caches improve associativity by increasing the number of physical ways; we refer this method as Direct Extension Method (DEM). DEM is simple and the basis of other extension methods. In this paper, we analyze the change characteristic of set-level utilization when associativity is varied based on DEM. And it is helpful to find the inherent deficiency of DEM and explore novel extension method of associativity. We try to probe the relationship between the associativity and set-level utilization with statistical analysis.

The increase of associativity by DEM augments the latency and energy cost of cache hits. For last-level caches, a 32-way set-associative cache has up to 3.3× the energy per hit and is 32% slower than a 4-way design [6]. Several approaches [7] have been published. And this is not directly relative to our subject and we don't concern it again.

As for the set-level utilization, prior researches focus on how to utilize the unbalanced utilization between sets under highly associative caches. For example, A. Basu et al. [8] propose Scavenger, which divides the total storage budget into a conventional cache and novel victim file architecture, and employs a skewed Bloom filter in conjunction with a pipelined priority heap to identify and retain the blocks that most frequently missed in the conventional part of the cache in the recent past. Therefore, conflict miss due to unbalanced utilization in set-level is alleviated. S. Khan et al. [9] construct the dead block from two unbalanced sets into virtual victim cache. Dyer Rolan et al. [10] propose Set Balancing Cache or SBC, which has associativity algorithm, placement algorithm and search algorithm to utilize set-level unbalanced phenomenon. D. Zhan [11] establishes an accurate metric for measuring individual sets' capacity demands by developing a group of mathematical models and presents COSET that identifies the capacity needs of individual sets, dynamically couples two sets with complementary capacity demands and reduces conflict misses. These approaches mainly focus on how to utilize under a certain high associativity rather than the relationship between the associativity and set-level unbalanced utilization. Nwachukwu[12] et al. presen a side-by-side comparison of techniques addressing the non-uniformity of accesses.

In this paper, we investigate the change of the set-level utilization for different associativity based on DEM and explain the trade-off of selecting a proper associativity for the cache with certain fixed capacity from the view of the

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.42

TABLE 1: Baseline configuration

Processor core	2 Cores; scalar in-order;1 hardware context per core; clocked at 1.5 GHz;
	L1-I cache and L1-D cache: 32kB, 64B line-size, 4-way, 3-cycle delay, LRU. The L1 caches are private to each core.
Unified Shared L2 Cache	1M,64B line-size,4-way,11-cycle delay,LRU,32-entry MSHR, 128-entry store buffer.L2 Cache L2 cache is shared among all the cores, MESI protocol,
Memory	32 DRAM banks; 200-cycle access latency; maximum 32 outstanding requests
Bus	16B-wide split-transaction bus at 4:1 frequency ratio queuing delays modeled

TABLE 2: Benchmarks from Spec Cpu2000/Cpu2006

Spec Cpu2000	164.gzip;168.wupwise;171.swim;172.mgrid;173.applu;175.vpr;177.mesa;179.Art;181.mcf; 183.equake; 188.ammp; 197.parser; 255.vortex; 256.bzip2; 300.twolf; 301.apsi;
Spec Cpu2006	433.milc;444.namd;445.gobmk; 447.dealII;450.soplex; 456.hmmer;458.sjeng;470.lbm;471.omnetpp; 473.astar;482.sphinx3

set-level unbalanced utilization and workloads. Furthermore, we explain the inherent deficiency of DEM through the statistical analysis of four typical benchmarks' set-level unbalanced utilization. It is benefit to research novel associativity extension mechanisms. We assume that LLC has a fixed capacity cache, such as 1MB, and associativity is varied by 4, 8, 16, 32 and 64 based on DEM. 29 benchmarks from Spec Cpu2000/Cpu2006 are chosen to be executed in Simics [5]. Our main contributions are: (1) from the view of set-level utilization, we explain the trade-off of selecting a proper associativity for the cache with certain fixed capacity. (2) Investigating the relationship between the associativity and set-level utilization based on DEM and try to explain the inherent deficiency of DEM.

II. EXPERIMENTAL METHODOLOGY

Table 1 shows the parameters of the baseline configuration used in our experiments. It is based on the cycle-based full-system simulator; Simics [13].The operating system of the simulator is Solaris 10.

We choose 29 benchmarks from Spec Cpu2000/Cpu2006, as presented in Table 2. We heavily modify the gcache module of Simics and all cache requests arriving at LLC are traced. Miss Count and Access Count of each core are summed group by each set. Therefore, we can analyze set-level utilization for different associativity.

For each workload, we warm the caches and branch predictors for 200 million instructions, and then perform detailed simulation for 400 million instructions per benchmark and for each 100 million instructions; the sample is performed on the various attributes of L2.

III. ANALYSIS OF CACHE SET UNBALANCED UTILIZATION FOR DIFFERENT ASSOCIATIVITY

In this paper, we focus on the set-level utilization under different associativity based on DEM. DEM is a simple and direct method that increase the number of physical ways to improve associativity.

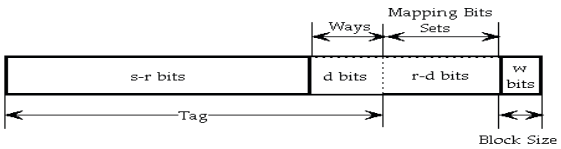


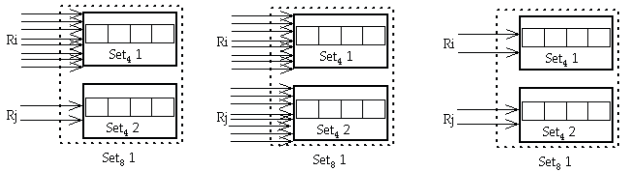
Fig1. Physical Address Segment in Cache

Addressing of DEM for set and way is simple and less overhead. As shown in Fig1, assumed that cache is indexed and tagged by physical address, physical address is segmented by bits. The length of physical address is (s+w) bits. And w bits are for offset of data block (the size of data block is 2^w bits); r – d bits for the addressing of set (the number of sets is 2^{r-d}), which is also the bits for mapping from cache request address to cache set; d bits for the addressing of way (the associativity is 2^d); and the total number of lines are 2^r. When associativity increases, then the d become larger, therefore the mapping of cache request is gathered. In our experiments, LLC has fixed capacity,1MB, and the size of block is not changed, as a result, the total number of sets are also not changed and r =14. When the associativity is varied by 4, 8, 16, 32 and 64 based on DEM, then d=2, 3, 4, 5, 6. And the number of sets(r-d) is 12,11,10,9,8. Therefore the mapping is changed, which also alters the set-level utilization.

Firstly, we qualitatively analyze the change of the conflicts miss when associativity increases. For example, when associativity varies from 4 to 8, one set in 4-way (denoted by Set_4 1) and another set (denoted by Set_4 2) are combined into one large set (denoted by Set_8 1).Although the lines in the same set increase, the cache requests are also gathered from the requests mapping to Set_4 1 and Set_4 2, as shown in Fig.2.

If the utilization of Set_4 1 is high and that of Set_4 2 is low, then the utilization of Set_8 1 may be the average of Set_4 1 and Set_4 2 and conflicts are alleviated due to low utilization of Set_4 2.Therefore it is benefit to the decrease of the whole cache MPKI, as shown in Fig2(a).

If the utilization of Set_4 1 and Set_4 2 are both high, then the utilization of Set_8 1 is still high, and the contention and pollution between Set_4 1 and Set_4 2 will intensify the conflicts rather than alleviate its conflicts, therefore, the whole cache MPKI increases, as shown in Fig2(b). If the utilization of Set_4 1 and Set_4 2 are both low, then the increase of associativity help to alleviate the conflicts so that the MPKI may decrease, as shown in Fig2(c).



a) (b) (c)

Fig2.Cache Request Conflicts
When associativity From 4 to 8

Therefore, when the associativity increase and the sets of low utilization and those of high utilization can be combined, it is benefit to decrease the MPKI and optimize the cache space efficiency. However, DEM has no mechanism to select such combination and heavily depends on the request address distribution of workingset. As a result, when associativity increases, the MPKI of some workloads may increase, that of some workloads may decrease, that of some workloads may be stable and that of some workloads may have no monotonous changes.

Secondly, we quantitatively analyze the change of MPKI of 29 benchmarks from Spec Cpu2000/Cpu2006 to validate the results of qualitative analysis. Table 3 presents the MPKI of each benchmarks when associativity is varied by 4, 8, 16, 32 and 64 based on DEM. All MPKI data is normalized to the MPKI of associativity equal to 4.

As we can see from the Table 3, within the error of 1%, there are 9 benchmarks whose MPKI are stable .And there are 4 benchmarks, 172.mgrid，188.ammp，456.hmmer and 473.astar whose MPKI increase while there are 10 benchmarks whose MPKI decrease. And the MPKI of the remaining 3 benchmarks have no monotonous changes.

We analyze the change of MPKI according to the set-level utilization. In this paper, we select the miss count of each set to the metrics of utilization rather that access count. There are three reasons: (1) the lines are replaced and removed from the cache when the request misses so that the conflict happens. Therefore, miss count can reflect the conflicts more directly. (2) For almost benchmarks, ratio of the hit count is far more than that of the miss count in each set, even to 90%. Whereas the hit only maintain the replacement status field and have no change to block, the access count has less relative to conflicts than the miss count.(3)The change of Miss count in each set has directly relative to MPKI.

In our experiments, we warm up the cache and branch predictors for 200 million instructions to alleviate the influence of code miss. And then sampling happens every 100 million instructions. As for the statistic samples of miss count for the whole cache, we calculate the minimum (denoted by Min), the maximum (denoted by Max), the mean, the standard deviation and coefficient of variation (denoted by C.V.).We select C.V. as the statistic to analyze the distribution of the miss count for all sets. Fig3. presents the results for all 27 benchmarks. We can see the following from Fig3.

(1)As for the same associativity, when the C.V. value of a benchmark is large, it does not mean that the MPKI of the benchmark is large, but that the utilization of each set is more unbalanced. For example, 188.Ammp has stable C.V. value, 0.8, when associativity is varied by 4, 8, 16, 32 and 64 based on DEM and 179.art has stable C.V. value, 0.6. From our statistical results, in 179.Art, the miss count of 50 percent sets is more than 80% total miss count. Meanwhile, in 188.Ammp, the miss count of 50% sets is more than 99% total miss count. Therefore, the set-level utilization in 188.Ammp is more unbalanced than that in 179.Art. This conforms to the comparison of these C.V.

TABLE 3: MPKI Change Ratio
(Normed To MPKI of A=4)

Benchmark	A=4	A=8	A=16	A=32	A=64
164.gzip	1	0.97	0.97	0.97	0.98
168.wupwise	1	1	1	1	1
171.swim	1	1	1	1	1
172.mgrid	1	1.03	1.04	1.05	1.05
173.applu	1	1	1	1	1
175.vpr	1	0.76	0.72	0.68	0.68
177.mesa	1	1	1	1	1
179.Art	1	0.99	1	1	1
181.mcf	1	1	1	1	1
183.equake	1	0.97	0.96	0.96	0.96
188.ammp	1	1.04	1.08	1.12	1.14
197.parser	1	1	1.01	1.01	1.01
255.vortex	1	0.97	0.95	0.95	0.95
256.bzip2	1	0.99	0.98	0.97	0.97
300.twolf	1	0.8	0.69	0.66	0.65
301.apsi	1	1	1	1	1
433.milc	1	1	1	1	1
444.namd	1	0.99	0.99	0.99	0.99
445.gobmk	1	0.93	0.9	0.89	0.88
447.dealII	1	1.04	1.06	1.05	1.05
450.soplex	1	1	0.99	0.99	0.99
456.hmmer	1	1.05	1.09	1.12	1.13
458.sjeng	1	0.98	0.96	0.95	0.94
470.lbm	1	1	1	1	1
471.omnetpp	1	0.77	0.75	0.75	0.75
473.astar	1	1.01	1.01	1.01	1.01
482.sphinx3	1	0.97	0.96	0.96	0.96

Miss count in 453.povray, 464.h264ref is so small that we pass over

(2)As for the same benchmark, when associativity is varied by 4, 8, 16, 32 and 64 based on DEM, the C.V. for almost all benchmarks except 179.art and 188.ammp decreases. This change means:

(a) The set-level utilization is more balanced when associativity increases. Quantitative analysis is provided latter.

(b)When associativity is varied from 4 to 16, there is sharp decline in C.V. for almost benchmarks except for 175.Vpr ， 179.Art and 188.ammp. Meanwhile, when associativity is varied from 16 to 64, the change of C.V. becomes smooth. It means that the utilization of each set when associativity equal to 16 is relatively balanced for almost all benchmarks. This conclusion is also evidenced by the change of MPKI presented in Table 3. When associativity is varied from 4 to 16, the MPKI has fallen sharply. And when associativity is varied from 16 to 64, the

MPKI has very smooth change. Therefore, from the view of the set-level utilization in each set, 16-way is more proper. And in fact, most of the mainstream commercial general processors have a 16-way LLC.

Thirdly, we analyze the sample data in detail for 175.Vpr, 471.omnetpp, 179.Art and 188.ammp. And try to find the inherent deficiency of DEM through quantitative analysis.

Case 1 175.Vpr. As shown in Table 3 and Fig.3(a), the MPKI of 175.Vpr has remarkable decline (the value list is 1，0.763，0.716，0.685) and the C.V. has also notable decline (the value list is 1.033，0.799，0.586，0.318) when associativity is varied from 4 to 32. And the MPKI varies from 0.685 to 0.683 and C.V. is reduced from 0.318 to 0.237 when associativity is varied from 32 to 64. Obviously, it is benefit to the decrease of MPKI for 175.Vpr when associativity increases. We analyze the distribution of miss count sample for each set in depth. And we calculate the 0.25 quantile value, referred to as Q1, the 0.5 quantile value, referred to as M, the 0.75 quantile value, referred to as Q3, the minimum (denoted by Min) and the maximum(denoted by Max). Then we calculate the ratio of miss count in the range (Min, Q1), (Q1, M), (M, Q3) and (Q3, Max) when associativity is varied by 4, 8, 16, 32 and 64 based on DEM, as shown in Fig.4. We can see that:

(1)Q1 increases, the ratio of miss count in the range (Min, Q1) increases from 1.4% to 18.7% and the ratio position of the range (Min, Q1) between Min and Max move up from 0.008% to 31%. Meanwhile, Q3 decreases, the ratio of miss count in the range (Q3, Max) decreases from 60.8% to 32.2%. The gap of Q1, M and Q3 is smaller, which can be clearly seen from Fig4. It means that the sets of low utilization and the sets of high utilization are possibly combined when associativity is varied by 4, 8, 16, 32 and 64 based on DEM. Therefore the MPKI decreases.

(2)On the other hand, from the change of C.V., we can see that the utilization of the sets is more balanced. It is benefit for the MPKI of 175.Vpr when associativity increases. This confirms to the statistical analysis.

Case 2 179.Art. As shown in Table 3 and Fig.3 (a), the MPKI and C.V. of 179.Art have minimal change when associativity is varied from 4 to 64. And we calculate Q1, M, Q3, Min and Max. Then we calculate the ratio of miss count in the range (Min, Q1), (Q1, M), (M, Q3) and (Q3, Max) when associativity is varied by 4, 8, 16, 32 and 64 based on DEM, as shown in Fig.6. We can see that:

The distribution of miss count in the range (Min, Q1), (Q1, M), (M, Q3) and (Q3, Max) is stable. And the ratio of extra high utilization sets (the miss count is in the range (Q3, Max)) is high, close to 44%.

Furthermore; the miss count of 50 percent sets is more than 80% total miss count. When associativity increases based on DEM, the utilization of each set is still more unbalanced because DEM has no mechanism to select low utilization sets and high utilization sets to combine. DEM gathers the sets by address rather than the characteristic of set-level utilization .

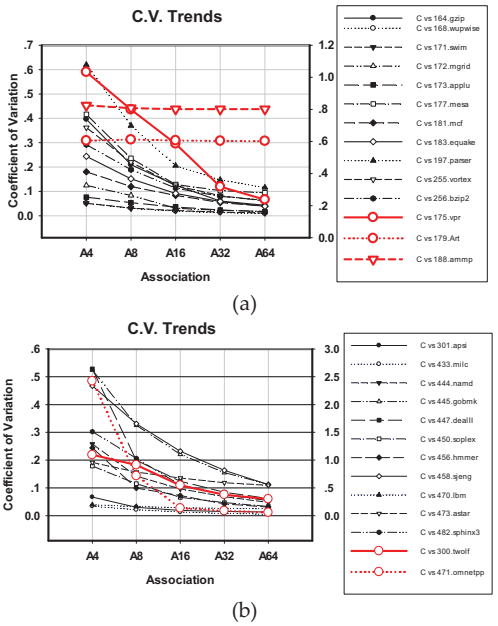


Fig.3. Coefficient of Variation for Spec Cpu2000/2006 when associativity is 4,8,16,32,64

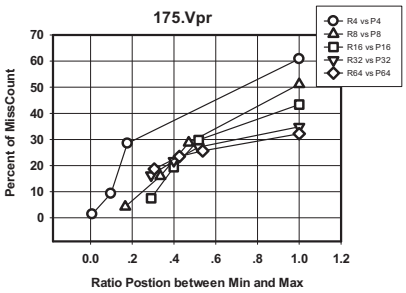


Fig.4. Miss Count Distribution in 175.Vpr

Therefore, the optimization of the set-level utilization is heavily depending on the distribution of cache request address. Meanwhile the distribution is often bursting and clustering due to spatial locality. It is the inherent deficiency of DEM.

Case 3 188.ammp. As shown in Table 3 and Fig.3 (a), the C.V. of 188.Ammp has minimal change when associativity is varied from 4 to 64. However, the MPKI has remarkable increase(the value list is 1，1.04，1.08，1.12，1.14). We calculate Q1, M, Q3, Min and Max. Then we calculate the ratio of miss count in the range (Min, Q1), (Q1, M), (M, Q3) and (Q3, Max) when associativity is varied by 4, 8, 16, 32 and 64 based on DEM, as shown in Fig.7. We can see that:

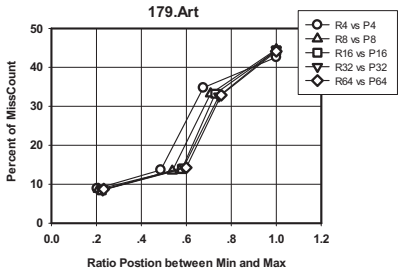


Fig.5. Miss Count Distribution in 179.Art

The utilization of each set is very unbalanced. The miss count of 50% sets is more than 99% total miss count. When associativity increases based on DEM, the ratio of the extra high utilization sets (whose miss count is in the range (Q3, Max)) declines from 54.4% to 49.0% meanwhile that of the high utilization sets (whose miss count is in the range (M, Q3)) grows up from 45.2% to 50.0%. It means that combination mainly happens between high utilization sets and extra high utilization sets, as shown in Fig2 (b). Therefore, the conflict misses do not decrease but increase so that the MPKI has remarkable increase.

In general, from our quantitative analysis to 175.Vpr, 179.Art and 188.Ammp and 175.Vpr benefit from the increase of associativity based on DEM and their set-level

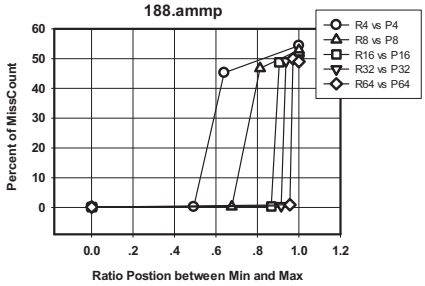


Fig.6. Miss Count Distribution in 188.Ammp

utilization is more balanced so that the conflict misses effectively decrease. Nevertheless, the set-level utilization of 179.Art and 188.Ammp is very unbalanced. And the unbalanced status is not alleviated from the increase of associativity based on DEM. Furthermore, the combination of sets using DEM possibly intensifies the conflict of cache request address rather than alleviates it, for example, 188.Ammp.It is due to the inherent deficiency of DEM. DEM has no mechanism to select low utilization sets and high utilization sets to combine. DEM gathers the sets by address rather than the characteristic of set-level utilization. Therefore, the optimization of the set-level utilization is heavily depending on the distribution of cache request address. Due to spatial locality, it is inevitable to the set-level unbalanced utilization. As a result, we need the information about set-level utilization when associativity increase so as to alleviate effectively the conflict misses and optimize the cache space utility.

IV. CONCLUSIONS

We investigate the change of the set-level utilization for different associativity based on DEM and explain the trade-off of selecting a proper associativity for the cache with certain fixed capacity from the view of the set-level unbalanced utilization and workloads. Furthermore, we explain the inherent deficiency of DEM through the statistical analysis of four typical benchmarks' set-level unbalanced utilization and try to find possible improvement approach.

ACKNOWLEDGMENT

This work was supported by Key technology research for the framework of Dealing with the scientific Big Data from Beijing Key Lab of Intelligent Telecommunication Software and Multimedia and China Postdoctoral Science Foundation funded project (Contract No. 2014M550662).

REFERENCES

[1] David Wentzlaff, Nathan Beckmann, Jason Miller,and Anant Agarwal.”Core Count vs Cache Size for Manycore Architectures in the Cloud”. MIT-CSAIL-TR-2010-008,February 11, 2010

[2] Wendel D F, Kalla R, Warnock J, et al. POWER7™, a highly parallel, scalable multi-core high end server processor[J]. Solid-State Circuits, IEEE Journal of, 2011, 46(1): 145-161.

[3] Hart J, Butler S, Cho H, et al. 3.6 GHz 16-core SPARC SoC processor in 28nm[C]//ISSCC’2013: 48-49.

[4] L. Ceze, J. Tuck, J. Torrellas, and C. Cascaval, “Bulk disambiguation of speculative threads in multiprocessors,”. ISCA’2006.

[5] V. Nagarajan and R. Gupta, “ECMon: exposing cache events for monitoring,” ISCA’2009.

[6] D. Sanchez and C. Kozyrakis. The ZCache: Decoupling Ways and Associativity. MICRO-43, 2010.

[7] <7>K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge. Drowsy caches: Simple techniques for reducing leakage power. In ISCA, 2002.

[8] A. Basu et al. Scavenger: A New Last Level Cache Architecture with Global Block Priority. MICRO-40, pages 421-432, December 2007.

[9] S. Khan et al. Using Dead Blocks as a Virtual Victim Cache. PACT-19 489-500, September 2010.

[10] D. Rol′ an, B. B. Fraguela, and R. Doallo, “Adaptive line placement with the set balancing cache,” MICRO’2009.

[11] D. Zhan, H. Jiang, and S. C. Seth, “Exploiting Set-Level Non-Uniformity of Capacity Demand to Enhance CMP Cooperative Caching,” IPDPS-24 pp. 222–233, 2010.

[12] Nwachukwu I, Kavi K, Ademola F, et al. Evaluation of techniques to improve cache access uniformities[C]// IEEE ICPP’2011: 31-40.

[13] P. S. Magnusson, M. Christensson, J. Eskilson, et al. Simics: A Full System Simulator Platform, volume 35-2, pages 50–58. Computer, 2002.

Research and Application of NetEye Network Traffic Monitoring System

Xi Liya
Institute of information science and engineering, Huazhong University of Science and Technology, Wuchang Branch, Wuhan, China
e-mail:lucy_xz@163.com

Mei Wenjun
Institute of information science and engineering, Huazhong University of Science and Technology, Wuchang Branch Wuhan, China
e-mail:xiliya@hustwb.edu.cn

***Abstract*—Through the analysis and study of common traffic monitoring methods, NetEye(referring to the name of the model put forward in the context) traffic monitoring model is put forward. The article expounds its function and the key algorithm, and system implementation by technologies such as WinPcap and so on.**

Keywords- traffic monitoring;model;WinPcap

I. INTRODUCTION

With the continuous development of information technology, network has become an indispensable part of people's life. However, in the network applications, the existing Internet Protocol address space can not meet the demands of the future network. In order to alleviate the constantly diminishing Internet Protocol address space, multiple users often form a group or a local area network (LAN) to access the Internet by one or a few Internet Protocol addresses. In this kind of application mode, if these users attack the Internet or occupy a lot of network bandwidth, other users may not access the Internet normally, which we do not want[1]. So we need to come up a way to manage and maintain this sort of network traffic, which is also the purpose of this article.

II. RESEARCH OF TRAFFIC MONITORING METHOD

There are two common methods of traffic monitoring implementation. The first one is transparently guiding the traffic of monitored host passing the monitoring host by the ARP cheating technique[2]. After capturing the traffic, monitoring host will act as a gateway to forward data[3]. Monitoring host needs to calculate captured traffic and use ARP attack technology to limit the traffic of the host. This way of traffic monitoring mainly depends on the ARP technical implementation. The running of the system needs to send plenty of ARP data packages. Therefore, when monitoring, the system also produces a lot of extra traffic. Moreover, if the monitored host opens the ARP firewall, this way of traffic monitoring will be invalid.

The other method is based on C/S structure. The monitored host must run a caught program to capture packet through its own network adapter card[4]. These caught program is a server program and runs as a background program. The captured traffic is calculated by the monitored host and sent to the monitoring host. Next, the monitoring host will collect, display and analysis them. This implementation of the traffic monitoring program will not be invalid even if the monitored host runs security procedure, and it can also detect the illegal operation from the monitored host. However, this traffic monitoring method is inflexible, and the monitoring host cannot manage the monitored hosts timely. On the other hand, the monitoring program will also be invalid when the caught program in the monitored host stops running .

III. THE MODEL OF NETEYE TRAFFIC MONITORING

A traffic monitoring system must obtain the information of all active hosts in the managed network, capture the traffic of all active hosts, limit the traffic of all active hosts, and cover the demands of most users in network application. But these applications still cannot meet the demand for today's network. Therefore, in order to prevent internal user attacking the network deliberately, we also need the traffic monitoring system to detect these attacks from intranet.

Therefore, the basic function of a traffic monitoring system needs to contain:

(1) Achieve all active host information in a monitored LAN.

(2) Capture and analysis Packets, and calculate the user traffic.

(3) Compute all users data traffic, and show the statistics in a visual representation format.

(4) Have abilities to limit upload and download speeds to users.

(5) Have some abilities of detecting attack.

(6) Have ability to record user operations.

(7) Have abilities to capture and analysis the specific packet.

Above all, NetEye model as a new traffic monitoring project in this paper is put forward ,as shown in figure 1.

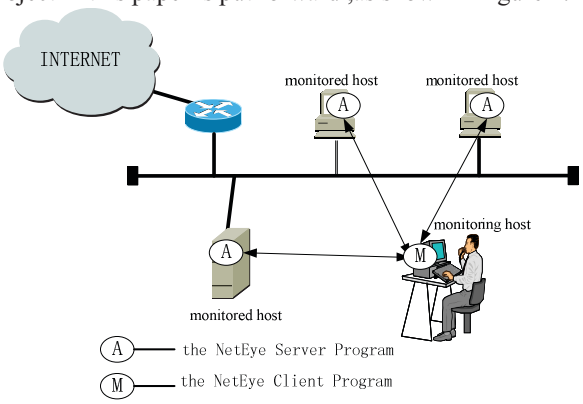


Figure 1 NetEye Traffic Monitoring Model

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.43

NetEye Traffic Monitoring Model is based on the C/S structure. Monitored hosts capture packet from their network cards by WinPcap technology[5], and these captured packet will be sent to the monitoring host to be analyzed and processed. Monitoring program, which is the client in this model, uses the free ARP packet to gain active host information in LAN, and limits the user traffic by the ARP attack technology. NetEye model has the following features:

(1) A good flexibility. Monitoring program can immediately summarize and analyze the captured data packages from the Monitored hosts, and timely control the monitored hosts according to the requirement.

(2) Reduces the generated traffic on the system operation because the technology of the free ARP and WinPcap are used in the monitoring system.

(3) Reduces the dependency of the ARP technology, and improves the reliability of the monitoring system. In addition, to improve the security, the monitored hosts which loss contact with the monitoring host are required to be off-line.

IV. THE REALIZATION OF NETEYE TRAFFIC MONITORING SYSTEM

A. General Design

This system is mainly implemented by WinPcap, ARP technology. The NetEye monitoring program summarizes, analyzes and processes the user traffic. The functions of the monitoring host modules are shown in figure 2.

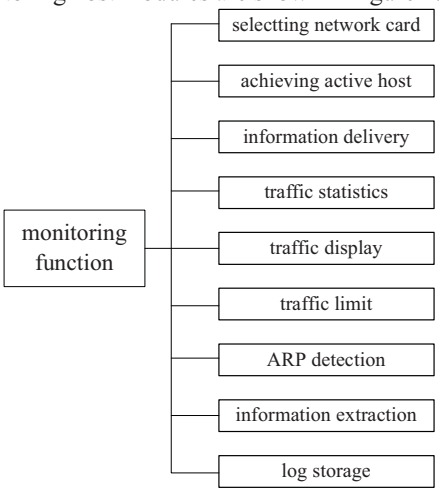


Figure 2 monitoring function modules

The monitoring host gets active host information by sending ARP request to each Internet Protocol address user in this segment and receiving ARP reply from them. The information delivery module sends the Internet Protocol address of the monitoring host and network parameters of the running client application program to each active host. The traffic statistics module summarizes and stores all the user traffic information for the traffic display module. The traffic limit module sets some parameters to limit upload or download speed for specific users. The ARP detection module judges ARP attacks from the intranet through monitoring and analyzing ARP packets. The information extraction module sends extracting traffic instructions to the monitored users.

NetEye server, also known as a client module, captures traffic through the network card of the monitored hosts, and calculates the captured traffic to send to the monitoring program. Client module is shown in figure 3.

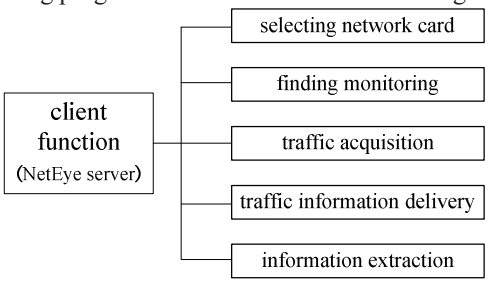


Figure 3 Client function modules

The monitoring finding module extracts the Internet Protocol address of the monitoring host and the necessary network information. The traffic acquisition module captures the Internet Protocol packets through their own network cards. The traffic information delivery module sends the statistical traffic to the monitoring program. According to the command from the monitoring host, the information extraction module makes a copy of traffic from the specific network card to the monitoring program.

B. Algorithm and Implementation of the Main Modules

In the NetEye system, the system interface is mainly implemented by MFC Picture Control, List Control and Edit Control implement, and the traffic diagram is drawn by the function of OnPaint() based on message processing mechanism[6]. The main algorithm of Monitoring modules are the following:

(1) algorithm and implementation of achieving active host module

First, we use WinPcap to get the Internet Protocol address and the mask of the monitoring host, and use the Internet Protocol address and the mask to calculate all the Internet Protocol addresses in this network segment. Then, we build the ARP packet structure, and send ARP request to all the Internet Protocol users by the function of pcap_sendpacket() in WinPcap[7]. At the same time, we listen to the ARP reply packet, which module will set the package filter conditions to ARP by the function of pcap_compile() and pcap_setfilter() in WinPcap. At last, it receives ARP packets by pcap_next_ex(), and display the information in List Control of the monitoring program[8].

(2) algorithm and implementation of traffic graph drawing module

The traffic chart updated thread needs to read a sery of data to draw traffic dynamic graph. In implementation,the front data of these data need be deleted and the end of this data need be added. The common queue cannot implement this function, so we choose the “deque” which possesses a fast speed and the operating ability of the front data and the end data. For implementation, we create two deque queues to store uploaded and downloaded traffic through deque<pair<up,down>> structure [9].

(3) algorithm and implementation of off-line module

Send ARP cheating data packages to the target host and trick the target host to change the MAC address of the gateway. So the information of target host cannot be forwarded through the gateway, and we achieve the goal of off-line.

(4) algorithm and implementation of speed limit module

If the upload or download speed of the user exceeds the speed limit, an ARP attack packet will be sent by pcap_sendpacket() in WinPcap. This packet will trick the user to rewrite the MAC address of the gateway. It will also lead that the user cannot access to the network in a short time, which indirectly achieve the goal of limit the user traffic .

The client module mainly completes the function of collecting traffic and extracting traffic.

(1) algorithm and implementation of collecting traffic module

Upload or download traffic is distinguished by the source or destination Internet Protocol address in the captured packet, and the size of data which is obtained from the length field in the captured packet will be added to count the size of upload or download traffic per second. This module is implemented by the function of pcap_compile() and pcap_setfilter() in WinPcap and the technology of Winscok[10].

(2) algorithm and implementation of extracting traffic module

When the client application receives extract commands from the monitoring program, it will change the target Internet Protocol address of captured Internet Protocol packet to the Internet Protocol address of the monitoring host. Therefore, it will guide the traffic of the monitored host flow to the monitoring host.

The running NetEye traffic monitoring system is shown in figure 4.

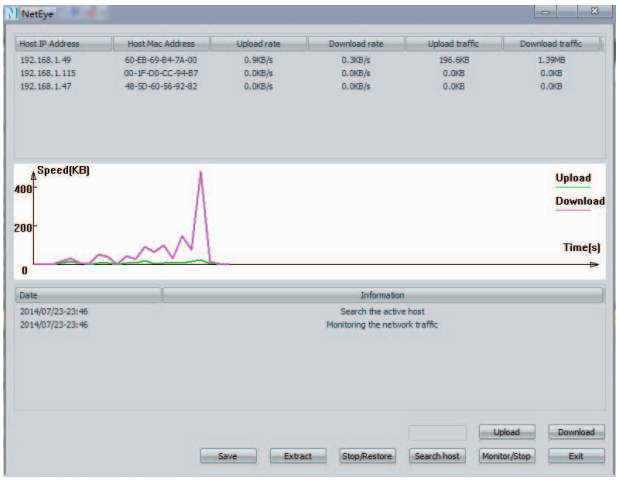


Figure 4 NetEye traffic monitoring system

V. CONCLUSION

This paper discusses the characteristics of NetEye traffic monitoring model, the key algorithm and system implementation technologies in detail. The design of NetEye modules fully considers the reliability and flexibility of application system. The design methods and ideas can provide references for the research of the network traffic monitoring to some extension. Meanwhile, the implementation methods of NetEye modules also have some reference value.

REFERENCES

[1] Xu Hong , Yang Yunjiang. The study of intranet security based on sniffer technology[J]. Microcomputer & Its Applications,2011(1):38-40,43.

[2] Sun Xianshu. The Research and Application of IP Network Traffic Measurement[D]. Beijing: Beijing University of Posts and Telecommunications, 2005.3.

[3] Zou Lianying,Li Fei,Wang Meizhen.Design of Application Software Flow Monitor System Based on the Gateway[J].Network & Computer Security,2013(5):26-29.

[4] Lin yongjian. The Research and Implementation of Data acquisition in Network traffic acquisition and analysis system[D]. Guangzhou: South China University of Technology, 2004.11.

[5] Robert M，Ido G．Detection of Unknown Computer Worms Activity Based on Computer Behavior using Data Mining [C] . Proceedings of the 2007 IEEE Symposium on CISDA，2007：169-177.

[6] Zhao Xinhui , LI Xiang.Methods of Capturing Network Packets[J].Application Research of Computers, 2004（8）:242-243,255.

[7] Wang Huaiyu, Lu Bingliang, Zhang Li. LAN Packets Processing System Based On Winpcap[J]. Microprocessors,2011, 10 (5):35-37.

[8] Liu Lijun, Xiong Wei. Implementation of Network Monitoring System Based on WinPcap[J].Computer Study,2010,5(10):56-57

[9] Hu Zhikun. Visual C++ Communication Programming Project Example Extract Solution[M]. Beijing : Mechanical Industry Press , 2007.1.

[10] Zhang Huiyong. WinSock Network Programming Meridian [M]. Beijing: Electronic Industry Press, 2012.8.

A Dynamic Topology Management Mechanism in Green Internet

Jinhong Zhang, Xingwei Wang, Min Huang
College of Information Science and Engineering
Northeastern University
Shenyang, 110819, P.R. China
wangxw@mail.neu.edu.cn

***Abstract*—In recent years, the rapid development of the Internet has provided people with more and more convenience. However, meanwhile a tremendous energy consumption problem and a low energy efficiency problem in the networks still exist. With the continuous expansion of the network scale, these problems are increasingly more severe. In order to achieve the "green" goal in the paper, first a networking model for green backbone network is built, then an innovative node structure and an innovative link structure are designed; furthermore, constitution of energy consumption in network device is analyzed, and a power model of the network device is set up; then according to the power-law principle, the values of network elements are evaluated from the three aspects of resource utilization rate, the impact degree on robustness of the network topology and the impact degree on communication quality; finally, a dynamic topology management mechanism in green Internet is proposed. Applicability and validity of the innovative mechanism with respect to energy saving and topology management are verified via prototype simulation and performance evaluation.**

Keywords-green Internet; energy saving; dynamic topology management; quantum evolutionary

I. INTRODUCTION

With the rapid development of Internet and the information industry, the proportion of energy consumption in network devices accounted for the total global energy consumption is greater and greater. The task of current topology management is to reduce the negative influence as much as possible.

The current topology management mechanisms are divided into two categories of centralized management mechanism and distributed management mechanism. The advantages of centralized management are easier to obtain global information of the whole topology and achieve a better network configuration. But the centralized management occupies more computing resources, especially for the larger size of topologies, the message overhead will be higher and also the response will be slower. On the contrary, the advantages of distributed management are that a less message overhead is required and the response will be quicker. However, due to lacking of global view, when the managed topology is larger and more complex, the result of network configuration from distributed management is worse.

Ref. [1] took advantage of new hardware technology to plan the network topology, path computing unit adopting the parallel processor to solve exponential computing in the configuration process to configure the network topology according to the network status. Based on the Ref. [1], Ref. [2] introduced the QoS and link utilization as parameters in the decision process, which achieved a trade-off between energy saving and performance in the final generated topology. Ref. [3] added the energy-aware ability to the OSPF protocol, which can not only power off the underutilized links to achieve energy saving but also minimize the influence on the current topology by confirming the impact of migrating traffic on the other links. Ref. [4] proposed a distributed energy-saving mechanism which adopted a resource reserve strategy that dynamically opening or closing the line cards in accordance with the reserved resource requirement. Ref. [5] also proposed a distributed energy-saving mechanism which only relied on traffic history record as the decision-making basis. The mechanism adjusted the weight of links via a utility function and a penalty function, and then calculated the configuration with the maximum utility value by the Q-learning method [6]. Ref. [7] proposed a topology-aware energy saving solution on the basis of knowledge on graph theory and related research on complex network, which measured topology connectivity-an evaluation criterion of topology configuration by calculating the second eigenvalue of Laplacian matrix corresponding to adjacency matrix.

A dynamic topology management mechanism in green Internet is proposed in the paper, which can achieve a goal of energy saving by changing network topology, actively putting underutilized routers and links into sleep for a low-traffic case at off-peak time and waking them up for a high-traffic case at peak time.

II. PROBLEM DESCRIPTION

A. Network Model

A model for green backbone network is simplified to a connected graph $G(V,E)$. $V=(v_1,v_2,\cdots,v_n)$ represents a set of vertexes, and a vertex represents a network node in green Internet; $E=(e_1,e_2,\cdots,e_l)$ represents a set of edges, and an edge represents a network link in green Internet.

A centralized decision node is included in the network topology as shown in Fig. 1. The other nodes are divided into multiple clusters. The centralized decision node is responsible for collecting global information and configuring the inter-cluster links. The node structure in green Internet is depicted in Fig. 2, consisting of chassis, line cards and other function modules, such as a master engine (ME), forwarding engines (FE), replication engines (RE) and so on. The master engine is in charge of routing and updating continuously the routing table; the switching matrix connects the input and output ports inside the router; forwarding engines are in charge of the routing table lookup; replication engines is used for multicast replication. A link structure in green Internet is depicted in

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.44

Fig. 3, consisting of a pre-amplifier, in-line amplifiers, regenerators, a post-amplifier and so on [8].

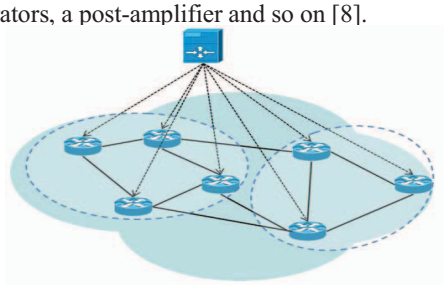


Figure 1. Network structure

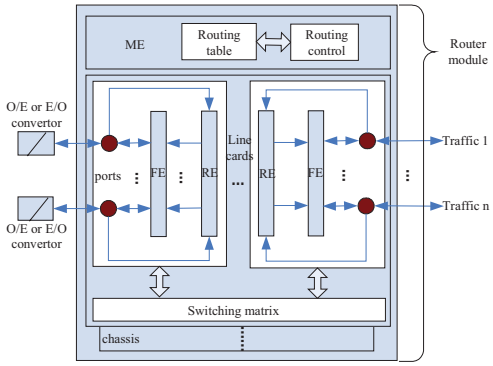


Figure 2. Node structure

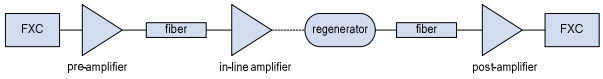


Figure 3. Link structure

B. Power Model

A node power consumption model is shown in formula (1), where $P^{i}_{n\_ctrl}$ represents the power consumed by the master engine in the core router i; $P^{i}_{n\_frm}$ represents the power consumed by a chassis in the core router i; $P^{l}_{l\_cpu}$ represents the power consumed by CPU of the line card in the core router i; $P^{l}_{l\_mem}$ represents the power consumed by memory of the line card in the core router i; $P^{l}_{l\_bus}$ represents the power consumed by bus of the line card in the core router i; P^{p}_{port} represents the power consumed by a port in the core router i; N^{i}_{from} represents the number of chassis in the core router i; N^{k}_{lcrd} represents the number of line cards of chassis k in the core router i; N^{l}_{port} represents the number of ports in line card l of chassis k in the core router i; trf_l represents the traffic flowing on line card l in the core router i; trf_p represents the traffic flowing on port p in the core router i; α and β are the constants denoting the relation between traffic and power.

$$ndpw_i = P^{i}_{n\_ctrl} + \sum_{k=1}^{N^{i}_{frm}} \left(\begin{array}{l} P^{i}_{n\_frm} \times FrmSt^{k}_{i} + \\ \sum_{l=1}^{N^{k}_{lcrd}} \left(2 \cdot \left(P^{l}_{l\_mem} + P^{l}_{l\_cpu} \left(1+\alpha \cdot trf^{\beta}_{l}\right) \right) \times LdSt^{l}_{k} + \right. \\ \left. \sum_{p=1}^{N^{l}_{port}} \left(P^{p}_{port} (1+\alpha \cdot trf^{\beta}_{p}) \times PortSt^{p}_{l} \right) \right) \end{array} \right) \quad (1)$$

A link power consumption model is shown in formula (2), where, $lkpw_j$ represents the power consumption of link j; len_{ref} represents the referential length of link (the default is 80km); len_j represents the practical length of link; P_{rely} represents the benchmark power consumption; trf_j represents the traffic in link j; α and β are the constants used to confirm the relation between traffic and power consumption.

$$lkpw_j = \left[len_j / len_{ref} \right] \times P_{rely} \times \left(1+\alpha \cdot trf^{\beta}_{j}\right) \quad (2)$$

III. Mechanism Design

A. Topology Clustering Module

An evaluation function (Q function) of network module performance is taken as the classification standard of clusters. Q function is defined in formula (3), where K represents the number of clusters in the network, m represents the total number of links in the network, m_s represents the number of links in the cluster s, d_s represents the sum of node degree in the cluster s.

$$Q = \sum_{s=1}^{K} [m_s / m - (d_s / 2m)^2] \quad (3)$$

B. Traffic Prediction Module

The traffic prediction is repeated N times. The parameter N is adjusted on the basis of prediction accuracy and interval requirement of operating mechanism. Discarding the top 5% of results prevents possible burst traffic from appearing in individual prediction results, and the maximum in the remaining results is taken as the maximum of prediction traffic within the period of prediction time.

The process of traffic prediction is shown as follows:

Step1：Check the port of node, and if the time arrives a threshold $tmth$, execute traffic prediction; else quit.

Step2：Further predict by applying the autoregression method and the Markov method. If both of them are invalid, then go to Step1; if the only one of them is valid, then take its value as the prediction result; if both of them are valid, then take their average as the prediction result.

Step3：Is it executed N time? If no, go to Step2.

Step4：Discard the top 5% of results in N results, and take the maximum in the remaining results as the final prediction result.

C. Information Awareness Model

Node awareness and traffic awareness use a consistent awareness interval (AI). AI is related to the change extent of traffic. If the traffic sharply changes, then AI should be decreased; else, increase AI until the maximum stable AI. The change extent of traffic is calculated in formula (4), where $tfch$ represents the change extent of traffic, trf_{last} represents the last traffic, trf_{now} the current traffic, θ is the constant between 0 and 1.

$$tfch = \begin{cases} \left| trf_{last} - trf_{now} \right| / trf_{last}, & trf_{last} \neq 0 \\ 0, & trf_{last} = 0, trf_{last} = trf_{now} \\ \theta, & trf_{last} = 0, trf_{last} \neq trf_{now} \end{cases} \quad (4)$$

AI is calculated in formula (5), where *tfin* represents the traffic AI, $tfin_{last}$ represents the last traffic AI, $tfin_{st}$ represents the stable AI, θ is the constant between 0 and 1, γ is an increasing coefficient of traffic AI; *pmin* represents the ratio of minimum AI to stable AI, $tfin_{st} \cdot pmin$ represents the minimum traffic AI.

$$tfin = \begin{cases} \theta \cdot tfin_{last}, & tfch \le \gamma, tfin_{last} < tfin_{st} / \theta \\ tfin_{st}, & tfch \le \gamma, tfin_{last} \ge tfin_{st} / \theta \\ tfin_{st} \cdot pmin, & others \end{cases} \quad (5)$$

A time sliding window mechanism makes the node only record the latest *n* times of awareness results. The traffic records are updated according to formula (6), where $flow_{n-1}$ and $flow_n$ represent two adjacent traffic records, θ is used to adjust the update rate and $0<\theta<1$.

$$flow_n = \theta \cdot flow_n + (1-\theta) \cdot flow_{n-1} \quad (6)$$

D. Centralized Topology Decision-making Module

The centralized decision-making problem is formulated as an integer linear program in (7)-(11), where pw_{lk} represents the power consumption in each link, x_{lk} represents the inter-cluster link state (on/off), x_l represents the state of the bridging inter-cluster bundle-link, F_l represents the prediction traffic on the bridging inter-cluster bundle-link, c_l represents the overall bandwidth of a bundle-link, L represents the set of bridging inter-cluster bundle-link, N represents the number of clusters.

$$\text{Objective function: } \min \sum\nolimits_{lk \in L} pw_{lk} x_{lk} \quad (7)$$

$$\text{Subject to: } x_{lk} = 0 \text{ or } 1 \ \ \forall lk \in L \quad (8)$$

$$x_l = 0 \text{ or } 1 \ \ \forall l \in L \quad (9)$$

$$F_l \le \eta_l c_l \quad (10)$$

$$N \le \sum_l x_l \le 2^N \quad (11)$$

The decision results are extended to 2^n due to the scope of solutions in our improved quantum evolutionary algorithm. The improved individual in quantum bits is denoted in equation (12), where α_{ki} represents the quantum bit, m is the number of links, n represents the number of divided domains, $sbw_{\max}$ is the maximum link bandwidth sum in all bridging bundle-links, $bw_{\min}$ is the minimum link bandwidth in all bridging bundle-links.

$$\Phi_{m\times n} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mn} \end{bmatrix}, \quad n = \left[sbw_{\max} / bw_{\min} \right] \quad (12)$$

$$R_{m\times n} = \begin{bmatrix} \alpha_{11}\alpha_{12}\cdots\alpha_{1n} \\ \alpha_{21}\alpha_{22}\cdots\alpha_{2n} \\ \vdots \\ \alpha_{m1}\alpha_{m2}\cdots\alpha_{mn} \end{bmatrix}, \ A_{m\times n} = \begin{bmatrix} a_{11}a_{12}\cdots a_{1n} \\ a_{21}a_{22}\cdots a_{2n} \\ \vdots \\ a_{m1}a_{m2}\cdots a_{mn} \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} \quad (13)$$

Random matrix $R_{m\times n}$ and the binary matrix $A_{m\times n}$ representing a solution obtained by comparing $R_{m\times n}$ and $(\Phi_{m\times n})^2$ are shown in equation (13). Further, D_m is obtained from $A_{m\times n}$ by converting its row vector from binary digit to decimal digit S_m is calculated in equation (14), where I denotes the matrix with all its elements are unit.

$$S_m = D_m + I \quad (14)$$

A fitness function $F(t)$ in quantum evolutionary algorithm is given in (15)-(16), where t represents iteration times, $f(x)$ represents an evaluation function for the bridging bundle-link x, pw_l represents the power saving when select the sleeping link l from the bridging bundle-link x, $kvalue_l$ represents the link value of link l, $[bw_L^x, bw_H^x]$ represents the bandwidth range of the bridging bundle-link x.

$$F(t) = \sum_{i=1}^{m} f(i) \quad (15)$$

$$f(x) = \beta \cdot \max \sum_l pw_l^{\alpha \cdot kvalue_l^{-1}}, \quad \sum_x bw_l \in \left[bw_L^x, bw_H^x \right] \quad (16)$$

Quantum evolutionary gate $G(\theta_i)$ is defined in (17)-(19), where θ_i is the rotation angle, F denotes the current fitness value, $F_{\min}$ denotes the minimum one of fitness values calculated, $F_{\max}$ denotes the maximum one of fitness values calculated, φ is a very small value. Quantum rotation Gate is defined in equation (20), where $(\alpha_i, \beta_i)^T$ is the *i*th quantum bit in certain chromosome and $(\alpha_i', \beta_i')^T$ is the corresponding updated one.

$$G(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \quad (17)$$

$$\theta_i = s(\alpha_i, \beta_i) \times \theta \times e \quad (18)$$

$$e = \begin{cases} \varphi, & F = F_{\min} \\ \kappa \cdot (F - F_{\min}) / (F_{\max} - F_{\min}), & else \end{cases} \quad (19)$$

$$\begin{bmatrix} \alpha_i' \\ \beta_i' \end{bmatrix} = G(\theta_i) \times \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (20)$$

The Improved Quantum Evolutionary Algorithm (IQEA) is shown in Algorithm 1.

Algorithm 1 IQEA.

Input: $G(V, E)$, Iter

Output: a new topology

1: classify inter-cluster links and the number of bridging bundle-link is *m*;
2: calculate the bandwidth sum of each bundle-link, $n \leftarrow sbw_{\max} / bw_{\min}$;
3: initialize $\Phi_{m\times n}$, *iterations* $\leftarrow 1$ and let $\alpha_{ik} \leftarrow 1/\sqrt{2}$;
4: **while** *iterations* < Iter **do**
5: generate a random number matrix $R_{m\times n}$;
6: calculate $A_{m\times n}$, S_m and the bandwidth range of each bundle-link;
7: **if** the network topology is not split and $F_l \le bw_H^x$ **then**
8: calculate the optimal combination of each bundle-link;
9: calculate $F(t)$, and update $F_{\min}$ and $F_{\max}$;
10: **end if**
11: update $\Phi_{m\times n}$;
12: *iterations* $\leftarrow$ *iterations*+1
13: **end while**
14: take the configuration corresponding to $F_{\max}$ as a final configuration;

E. Link Value Evaluation Module

Bandwidth utilization value $BwValue_l$ of link l is defined in formula (21), where bw_l is the practical bandwidth of link l, trf_l is the current traffic of link l, α and β are used to adjust $BwValue_l$, μ is the optimal bandwidth utilization value.

$$BwValue_l = 1/(1+\alpha(trf_l / bw_l - \mu)^{\beta}) \quad (21)$$

Power efficiency $PwUtility_l$ of link l connecting to node n is defined in formula (22), where trf_l is the current traffic of link l, trf_{total} is the current total traffic of node n, $ndpw_n$ is the total power consumption of node n, $lkpw_l$ is the current power consumption of link l connecting to node n.

$$PwUtility_l = trf_l / (ndpw_n \cdot (trf_l / trf_{total}) + lkpw_l) \quad (22)$$

Edge connectivity degree $lkcon_l$ is defined in (23), where λ is a connectivity value of the original topology, λ_l is a connectivity value of the topology excluding link l, λ_{ref} is an adjustment coefficient.

$$lkcon_l = \lambda_{ref}(\lambda - \lambda_l)/\lambda \quad (23)$$

If a node exists for the k-core but removed for (k+1)-core, then its core number is k; k-core is a subgraph excluding all the nodes whose core number is not more than k from the original graph. Core number $Core_l$ is defined in formula (24), where $CorNum_{\max}$ is the maximum core number in the network, $CorNum_l$ is the core number of current node.

$$Core_l = CorNum_l / CorNum_{\max} \quad (24)$$

Comprehensive evaluation on the topology performance is shown in formula (25).

$$TopolyInflu_l = \alpha \cdot lkcon_l + \gamma \cdot Core_l \quad (25)$$

Mediation coefficient $betw_l$, that is, the number of the shortest paths going through link l, is defined in formula (26), where σ_{ij} is the number of the shortest paths between node i and j, $\sigma_{ij}(l)$ is the number of the shortest paths not only going through link l but also between node i and j.

$$betw_l = \sum_{i=1}^{j} (\sigma_{ij}(l) / \sigma_{ij}) \quad (26)$$

Link stability $lkst_l$ indicating how often link l to switch its state is defined in formula (27), where $hibrtime_{\max}$ and $hibrtime_{min}$ are respectively the maximum and minimum link sleeping time in the betwok, $hibrtime_l$ is the total sleeping time of link l.

$$lkst_l = \begin{cases} \dfrac{hibrtime_{\max} - hibrtime_l}{hibrtime_{max} - hibrtime_{min}}, hibrtime_{max} \neq hibrtime_{min} \\ \alpha, hibrtime_{max} = hibrtime_{min} \end{cases} \quad (27)$$

Comprehensive evaluation on the communication quality $cmtinflu_l$ is defined in formula (28), where trf_l is the current traffic, trf_{ref} is the reference traffic, α and β are adjustment coefficients.

$$cmtinflu_l = (\alpha \cdot lkst_l + \beta \cdot betw_l) trf_l / trf_{ref} \quad (28)$$

Based on all above, link value is defined in formula (29), where α and β are constants, $\alpha+\beta=1$.

$$kvalue_l = (1 + TopolyInflu_l)(\alpha BwValue_l + \beta PwUtility_l)^{cmtinflu_l} \quad (29)$$

F. Sleeping Control Module

The model controls the devices to sleep or awaken, including sending, receiving, and handling the information related to negotiating before the devices to sleep, advertising the sleeping devices to awaken and noticing the current link state to the centralized topology decision-making module.

IV. Simulation And Performance Comparison

A. Simulation

The proposed Dynamic Topology Management Mechanism in Green Internet (GIDTMM) is simulated by C++ in the programming environment of VC6.0 under OS Windows7. Three kinds of link traffic models shown in Fig. 4 are used to analyze the performance of GIDTMM.

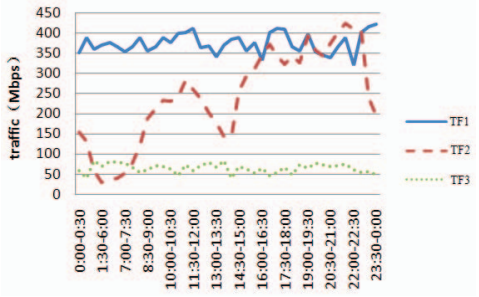
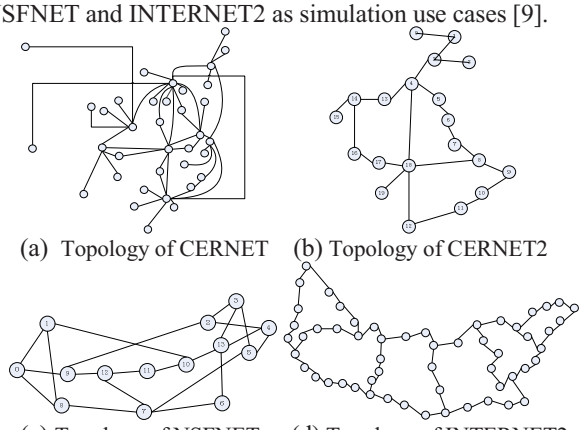


Figure 4. Traffic use cases

Take the topologies (shown in Fig. 5) of China Education and Research Network (CERNET), CERNET2, NSFNET and INTERNET2 as simulation use cases [9].



(a) Topology of CERNET (b) Topology of CERNET2

(c) Topology of NSFNET (d) Topology of INTERNET2

Figure 5. Topology use cases

CA algorithm [10] proposed by Po-Kai Tseng et al. is selected as the benchmark algorithm.

B. Performance Evaluation

1) Comparison on power consumption

For periods of regularly collecting information, both GIDTMM and CA are 15mins. In Fig. 6, it is concluded that GIDTMM is of a better power-saving effect and adaptive capacity for different combinations of topologies and traffic models. This is mainly due to two reasons: the

traffic prediction module of GIDTMM can avoid some decision-making mistakes caused by traffic fluctuation; the ability of automatically adjusting information awareness interval (IAI) of GIDTMM makes it update data timely to ensure the decision-making validity. In contrast, the IAI of CA algorithm is a fixed value, so the greater traffic fluctuation is, the worse its decision-making result is.

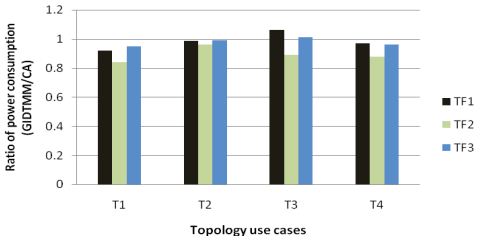


Figure 6. Comparison of power consumption (GIDTMM/CA)

2) Comparison on network performance

Take TF2 as the test model, and observe the average hops between nodes and packet loss rate (PLR) of each node every 15 minutes.

In Fig. 7, average hops in CA are less than that in GIDTMM. This is because CA can obtain global information and only the inter-cluster links in GIDTMM are managed by the centralized topology decision-making module according to the global information. Further, GIDTMM shows a relatively good stability for different topologies, especially in CERNET and INTERNET2.

In Fig. 8, PLR in CA is the higher. PLR majorly depends on the response rate to the changes of network status. PLR in GIDTMM is better than that in CA because of multiple inter-cluster links. Furthermore, the more complex the topology is, the smaller the gap of PLR is, that is because the fact more paths in a complex topology can be selected to avoid congestion.

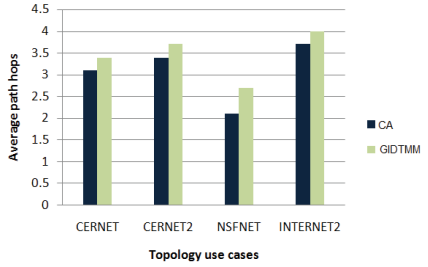


Figure 7. Average path hops

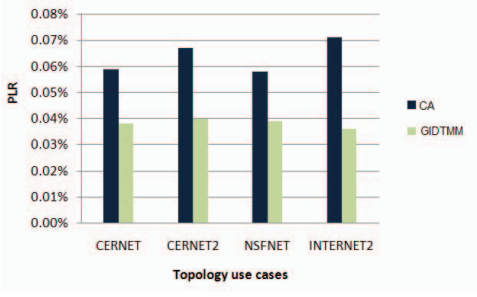


Figure 8. Network PLR

V. CONCLUSION

In this paper, a network model in green Internet is built, a node structure and a link structure are devised, the constitution of energy consumption in network device is analyzed and a power model of network devices is set up. To achieve power saving, GIDTMM consisting of six modules is proposed. Evaluate the network element as for resource utilization, topology impact and communication impact. Finally, simulation and performance evaluation demonstrate the power-saving potential and network performance guarantee of GIDTMM on the basis of the power consumption ratio, average path hops and PLR.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant No. 61225012 and No. 71325002; the Specialized Research Fund of the Doctoral Program of Higher Education for the Priority Development Areas under Grant No. 20120042130003; the Fundamental Research Funds for the Central Universities under Grant No. N110204003 and No. N120104001.

REFERENCES

[1] N. Yamanaka, S. Shimizu, and Gao Shan, "Energy efficient network design tool for green IP/ethernet networks," Proceedings of the Optical Network Design and Modeling (ONDM 2010), IEEE, Feb. 2010, pp. 1–5, doi: 10.1109/ONDM.2010.5431566.

[2] H. Yonezu, K. Kikuta, D. Ishii, et al., "QoS aware energy optimal network topology design and dynamic link power management," Proceedings of the Optical Communication (ECOC 2010), IEEE, Sept. 2010, pp. 1–3, doi: 10.1109/ECOC.2010.5621098.

[3] A. Cianfrani, V. Eramo, M. Listanti, et al., "An OSPF enhancement for energy saving in IP networks," Proceedings of the Computer Communications Workshops (INFOCOM WKSHPS 2011), IEEE, April 2011, pp. 325–330, doi: 10.1109/INFCOMW.2011.5928832.

[4] A. Coiro, F. Iervini, and M. Listanti, "Distributed and adaptive interface switch off for Internet energy saving," Proceedings of the Computer Communications and Networks (ICCCN 2011), IEEE, July -Aug. 2011, pp. 1–8, doi: 10.1109/ICCCN.2011.6005884.

[5] F. Cuomo, A. Cianfrani, M. Polverini, et al., "Network pruning for energy saving in the Internet," Computer Networks, vol. 56, Issue 10, July 2012, pp. 2355–2367, doi: 10.1016/j.comnet.2012.03.009.

[6] C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine learning, Vol. 8, Issue 3–4, 1992, pp. 279–292, doi: 10.1007/BF00992698.

[7] F. Cuomo, A. Abbagnale, A. Cianfrani, et al., "Keeping the connectivity and saving the energy in the internet," Proceedings of theComputer Communications Workshops (INFOCOM WKSHPS 2011), IEEE, April 2011, pp. 319–324, doi: 10.1109/INFCOMW.2011.5928831.

[8] Xingwei Wang, Hui Cheng, Keqin Li, Jie Li, Jiajia Sun, "A cross-layer optimization based integrated routing and grooming algorithm for green multi-granularity transport networks," Journal of Parallel and Distributed Computing, vol. 73, Issue 6, June 2013, pp. 807–822, doi: 10.1016/j.jpdc.2013.02.010.

[9] Xingwei Wang, Hui Cheng, Min Huang, "Multi-robot navigation based QoS routing in self-organizing networks," Engineering Applications of Artificial Intelligence, vol. 26, Issue 1, January 2013, pp. 262–272, doi: 10.1016/j.engappai.2012.01.008.

[10] T. Po-Kai and C. Wei-Ho, "Near optimal link on/off scheduling and weight assignment for minimizing IP network energy consumption," Computer Communications, vol. 35, Issue 6, 15 March 2012, pp. 729–737, doi: 10.1016/j.comcom.2011.12.010.

Design of Component-Oriented Centralized and Distributed-Integrative Runtime Infrastructure of Simulation System

Jian-xing GONG
College of Electromechanical Engineering and Automation
National University of Defense
Changsha, China
e-mail: 28982215@qq.com

Zhong-Jie ZHANG
College of Electromechanical Engineering and Automation
National University of Defense
Changsha, China
e-mail: 597289799@qq.com

Jian-guo HAO
College of Electromechanical Engineering and Automation
National University of Defense
Changsha, China
e-mail: hla2000@163.com

Jian HUANG
College of Electromechanical Engineering and Automation
National University of Defense
Changsha, China
e-mail: 13973166586@139.com

Yun ZHOU
College of Information Systems and Management
National University of Defense
Changsha, China
e-mail: zhouyun8007@126.com

***Abstract*—Focusing on specific domains, most traditional simulation systems have features of standalone-centralized running mode, weak interoperability with external simulation systems, difficult reusability of system components. With the technical development of the distributed simulation, the inspiration of developing reusable simulation model components satisfying the needs of simulation systems or environments in the whole simulation domain, as well as endowing the simulation system with the ability of customization, expansion and interoperability by assembling above mentioned components rapidly and conveniently have drawn significant attention within simulation community. The paper addresses a component-oriented centralized and distributed-integrative runtime infrastructure of simulation system. The system has designed the corresponding data distribution, filtering and cache concentration, operating on the subscribing and publishing mechanism, for reducing the redundancy of interactive data between models and improving the running efficiency of the system. It also proposes the simulation service based on the bridge-based design pattern, shielding implementation details between models in the application layer and underlying runtime platforms. The service enables the application of models used not only in traditional standalone centralized simulation systems, but also in distributed simulation system with no editing even compiling. Besides, through using international standard descriptive specification, namely Base Object Model (BOM), for component-oriented simulation models, the system improves the standardization and interoperability of models.**

Keywords- Component-oriented; centralized-and-distributed-integrative; Runtime Infrastructure of Simulation System; Base Object Model

I. INTRODUCTION

Established on the needs of specific domains, most traditional simulation systems have features of standalone-centralized running mode, weak interoperability with external simulation systems and hard reusability of system components. In the meantime, with the development of the distributed simulation techniques and the extensions of the simulation application fields, especially in the military domain, the scales of simulated systems have become larger and more complex, and their levels have also been higher. Therefore, it is very low efficiency to design and develop a large scale of simulation systems completely from scratch and it is difficult to ensure the reliability and accuracy of the models and simulation results. Thus, research and development of a large simulation system from scratch would be very low-efficiency, and at the same time, the validity and accuracy of the model and simulation results would be hard to be guaranteed. Therefore, the inspiration of developing reusable simulation model components satisfying the needs of simulation systems or environments in the whole simulation domain, as well as endowing the simulation system with the ability of customization, expansion and interoperability by assembling above mentioned components rapidly and conveniently would be an effective way of tackling existing problems faced by complex system simulation[1].

With the development of the technology of software components, the components based development (CBD) approach of software has been gradually accepted by the simulation domain, especially the distributed simulation software development. The large-scale complex simulation system requests the production of simulation software to be standardized, large-scale and cost-saving, as well as a transformation from individual or group-based workshop mode to socialized labor-division cooperative mode. In recent years, many develop software for simulation system based on several approaches, such as public library, production line and interoperability protocol has emerged. However, there are severe limitations in the simulation components based on above mentioned approaches and the interoperability of simulation systems due to the lack of

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.45

the unified model description specification, the open architecture and the standard interoperability protocol.

Focusing on these problems, the paper proposes a component-oriented distributed and centralized-integrative runtime infrastructure of simulation system, with the ability of dynamic loading model components and flexible construction of simulation applications. This enables the application of models used not only in traditional standalone centralized simulation systems, but also in distributed simulation system with no editing even compiling. Besides, through using international standard descriptive specification, namely Base Object Model (BOM)[2][3], for component-oriented simulation models, the system improves the standardization and interoperability of models.

II. COMPONENT-ORIENTED BASE OBJECT MODEL

With the purpose of providing a kind of mechanism promoting interoperability, reusability and composability, BOM (Base Object Model) specification raised by SISO (Simulation Interoperability Standards Organization) helps to implement the composability based on components in the field of modeling and simulation. Regarding the interoperability, reusability and composability as its core, BOM has the idea of DIY (Do it by yourself) through combining reusable simulation resources into customized simulation system according to requirements. With the standard specification describing component-oriented models, BOM provides an open architecture for extendable, composable and interoperable simulation systems.

The define of BOM is that a piece part of a conceptual model composed of a group of interrelated elements, which can be used as a building block in the development and extension of a federation, individual federate, FOM or SOM[2]. The BOM concept is based on the assumption that piece-parts of simulations and federations can be extracted and reused as modeling building-blocks or components. The interplay within a simulation or federation can be captured and characterized in the form of reusable patterns. These patterns of simulation interplay are sequences of events between simulation elements and realized through the HLA object model structure provided by BOM [3].

III. RUNTIME INFRASTRUCTURE OF SIMULATION SYSTEM BASED ON MIDDLEWARE

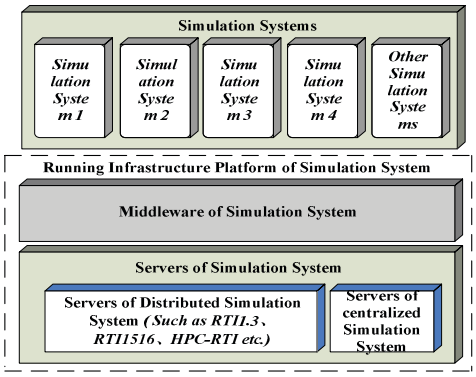


Figure 1. Architecture of Running Infrastructure of Simulation System Based On the Middleware

Multiple application software needs to be transplanted among many platforms, so can platforms support multiple application software. This requires reliable and efficient data exchange or conversion among software, hardware and application systems to ensure synergy between systems and resource sharing among different technologies. The independent software fulfilling this need is called the middleware [5]. Usually, the middleware has the following features: satisfying the need of applications, running on multiple hardware and operating system platforms, supporting distributed computing, and providing interactive functions across network, hardware and operating system and supporting standard protocol and interfaces.

As shown in Figure 1, runtime infrastructure of simulation system mainly includes two parts: simulation system middleware and services. The middleware is a narrow-sense software middleware, acting as a bridge between application system and simulation system services. This helps to isolate the realization of runtime services and the detailed application system, which enables the simulation application software developed based on this structure runnning in a distributed and centralized runtime services. Thus the infrastructure realizes the cross-platform transplantation of simulation model components and simulation system under the circumstances of little changes or even no compilation.

IV. SOFTWARE FRAMEWORK FOR THE RUNTIME INFRASTRUCTURE OF THE SIMULATION SYSTEM

A. *Logical structure of the software*

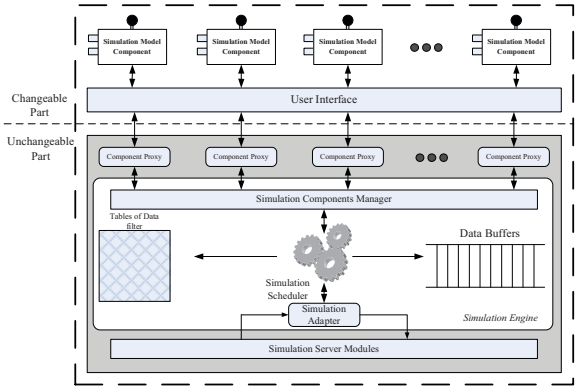


Figure 2. Logical Implementation structure of Running Infrastructure of Simulation System

The software framework for the component-oriented distributed and centralized-integrative runtime infrastructure of simulation system is shown in Figure 2. A simulation system is abstracted and divided into changeable and unchangeable parts. On the application layer of the simulation system which is user-oriented, the changeable part is mainly composed of different model components with different functions and other application components. The paper defines the unchangeable part as simulation system framework. The simulation engine provided is the core module of the framework, controlling the data filtration, cache and distribution of the simulation system. Besides, based on the description information of the simulation model, the engine also drives the harmonious advancement of model components in the

simulation system. Through the simulation adapter to use the simulation service model proposed by the system architecture, the simulation engine is the scheduling system of the simulation architecture, just like the heart of the simulation system.

The simulation system framework will not couple with any specific simulation model components and is also independent of detail simulation application. It is customized according to user's requires. This framework is just like the battery (like engine) and assemblage tool (set of tools) of the toy car. But those are not enough. Body, wheel and remote control unit (like model components) are also required to design a new toy (federate). Thus, this framework is not a specific simulation application system, but a general framework for constructing different kinds of simulation system through combination of model components. If the runtime infrastructure of the simulation system is based on the HLA distributed simulation service[1], then the simulation system based on this framework is one of federates. The paper also calls this framework as component-based federate framework.

In order to solve the runtime efficiency of the simulation system, one of key techniques is used to improve the efficiency of the data filtering, cache and distribution within the simulation system framework, thus the generation of redundant data as well as its acceptation in simulation model component would be reduced. Therefore, through applying one-to-many data sending in the publishing and subscribing mechanism among simulation model components, it helps to avoid the blindness in the data broadcast.

B. The mechanism of Data distribution\ filtering\cache

1) The mechanism of data distribution

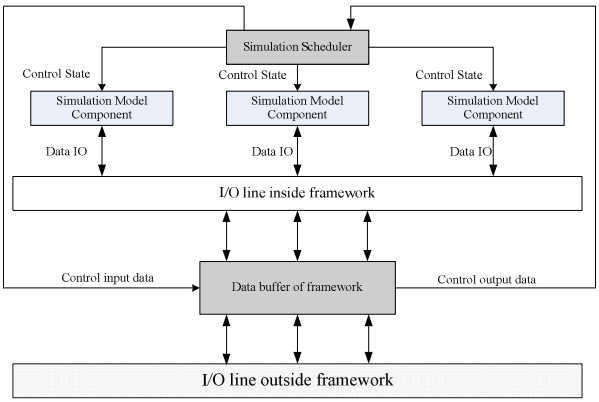


Figure 3. Logical Implementation structure of Running Infrastructure of Simulation System

In order to distribute the cached data in simulation system framework efficiently and continually, the simulation system framework must efficiently control the data. The simulation scheduler plays this role. As shown in Figure 3, the simulation scheduler not only controls the state of every model component but also data inputs and outputs in the buffer area of the framework.

[1]The distributed simulation system described by the paper is looked as the simulation system based on HLA.

The data generated in model components during simulation activities is not directly sent to its destination, but filtered by IF BOM [3] firstly. Then the data is transmitted to the data cache through the internal I/O data line within the framework. Before data caching, the simulation scheduler first controls inputs in the buffer through data input control line, ensuring the existence of valid subscriber. Otherwise, they will be filtered directly. After that, data is saved in the buffer generated by framework, thus avoiding data redundancy and improving the efficiency of the data buffer.

The simulation scheduler will keep detecting the state of model components in the simulation advancement. If one model component is at a request update state, then the scheduler would fetch interested data from data buffer through data output control line, and then send to model components through the internal I/O data line. Below is the pseudo code of internal data distribution controlled by the simulation scheduler.

```
Begin the data distribution inside the framework;
if compA is at the request state checked by simScheduler then
  simScheduler sent the data output command to fedDataBuffer;
else
  return the above step to check the state of the next component;
endif
if the data subscribed by compA is fetched by the fedDataBuffer then
  dataA is sent to compA through the io line inside the framework;
endif
dataB is produced by compA during the process of dataA;
if dataB is published by BOMA of compA then
  dataB is sent to the io line inside the framework;
endif
if the subscribers exclude compA exist the framework then
  dataB is saved in fedDataBuffer and the reference counter increase one;
else
  compA releases dataB;
endif
Moreover,
if dataB is published by the assemble bom or som of the framework then
  dataB is sent to the io line outside the framework;
endif
End the data distribution.
```

Figure 4. Pseudocode of Internal Data Distribution Controlled By the Simulation Scheduler

As showed in Figure 4, in the above pseudo code, $comp_i$ stands for the i index of simulation component, BOM_i describes the interface information of $comp_i$, and *simScheduler* is the simulation scheduler of the framework, and *fedDataBuffer* is the data buffer of the framework.

2) The mechanism of data filtering

Figure 5 shows the data filter layer based on distributed runtime mode. The I/O data line of the framework could be divided into two types: I/O data line in federation level, I/O data line in federate level (i.e. IO data line inside the framework). Data on the I/O data line in federation level is flowing among federates supported by RTI. So that, we only focus on the data filter mechanism within framework in the following.

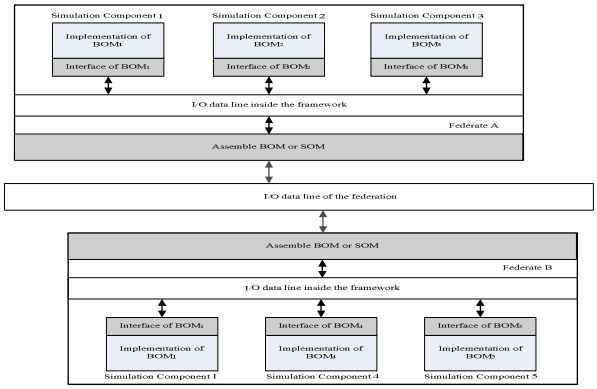


Figure 5. Data Filter Layer Based on Distributed Runtime Mode

The I/O data line within framework provides one of approaches for data interaction among simulation framework model components. The framework does not use the broadcast way of distributing data produced by

model components, while the data filtering net regarding IF BOM as its components used in data regional filtering mechanism. As show in Figure 5, the simulation model components are mainly composed of two parts: BOM implementation and BOM interface, which depicts the interaction of model components and the ability of the interaction among simulation model components and between simulation model components and the outer environment respectively. Through the BOM information provided by each simulation model component, the simulation scheduler can get all the publishing and subscribing information of model components in the whole framework and filter the data within federates effectively, reduce redundant data transmission and improve the transmission ability of I/O data line in framework.

Based on the BOM interface information of components, the simulation scheduler coordinates the data transmission of I/O data line in the framework. And further according to the BOM/SOM information, the scheduler determines the distribution of data in I/O data line to other federates (for example, sent to the network through RTI). The assembly BOM or SOM information comes from the combination of BOM describing each component, and is the data filtering net, which describes publishing and subscribing information of the framework, thus reduce the transmission of redundant data in the network. The data between federates transmit through the federation's I/O data line provided by RTI. The framework can use the data distribution mechanism of RTI, such as DDM (data distribution management) to improve the transmitting ability of the network. The following pseudo code presents data is sent and accepted between components.

```
compi begins to send the data;
if data is publishing declared by BOMi then
   data is sent to the io line inside the framework;
endif
if data is subscribing declared by BOMj then
   data is sent to  compj through the io line inside the framework;
endif
moreover;
```

```
if data is published by the assemble bom or som of the framework then
   data is sent to the io line of the federation;
endif
compi finishes the data sending;
compj begins to receive the data;
compj qureys the buffer of the framework;
If the buffer of the framework exists data subscribed by BOMi then
   data is sent to the io line inside the framework;
endif
compj finishs to receive the data;
```

Figure 6. Pseudocode of Sending and Receiving Data for Simulation Compoents

As showed in Figure 6, in the above pseudo code, $comp_i$ stands for the i index of simulation model component, BOM_i describes the interface information of $comp_i$, and $i \neq j$.

3) The center-distributed data cache mechanism

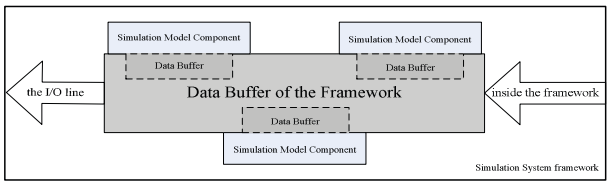


Figure 7. Center-distributed Data Cache Mechanism

The runtime infrastructure for simulation system adapts the centralized and distributed cache mechanism, in which every component has its data buffer, so does the simulation system framework itself. While the data buffer of model components is not allocated within components themselves, the framework automatically allocates data buffers for each component in the framework's buffer. So the data cache of components described in Figure 7 is depicted by dotted line. Each component can only access to the own data buffer allocated by the framework. When accepting data from internal components or external federates, the framework would distribute data according to the subscribing information of IF BOM from the component. However, direct copy of data is not used in this approach but reference copy (not the real copy, just increment using counter of data) based on reference counting. The framework must coordinate the management of all data buffers for components. This approach would not increase the difficulty of the implementation and development of components, and at the same time, help to reduce the redundant accessing to data from components, avoid multiple copies of one piece of data, lower the cost of resources and improve the efficiency of data cache as well as running performance.

C. The scheduling process

1. $LBTS_{\min} = \min\{\forall LP_i, LP_i.lookahead + LP_i.requested\_Time\}$
2. $L \leftarrow \{LP \mid \forall LP_i, LP_i.requested\_Time \le LBTS_{\min}\}$
3. $SortByPriority(L)$
4. $while\ (L \neq \Phi)\ \{$
5. $\quad LP_i \leftarrow head(L)$
6. $\quad EL_{RO} \leftarrow EL_{RO} \cup CheckOutSimEventOfRO(LP_i)$
7. $\quad EL_{TSO} \leftarrow \{e \mid \forall e \in EL_{TSO}, e.timestamp \le LBTS_{\min}\} \cup CheckOutSimEventOfTSO(LP_i)$
8. $\quad LP_i.local\_Time \leftarrow LP_i.requested\_Time$
9. $\quad LP_i.state \leftarrow update$
10. $\}$
11. $while\ (EL_{RO} \neq \Phi)\ \{$
12. $\quad EL_{RO} \leftarrow EL_{RO} - \{e\}$
13. $\quad \{IE, EE\} \leftarrow execute(e)$
14. $\quad IB \leftarrow IB \cup IE$
15. $\quad Send(EE)$
16. $\}$
17. $SortByTimeStamp(EL_{TSO})$
18. $while\ (EL_{TSO} \neq \Phi)\ \{$
19. $\quad e \leftarrow head(EL_{TSO})$
20. $\quad \{IE, EE\} \leftarrow execute(e)$
21. $\quad IB \leftarrow IB \cup IE$
22. $\quad Send(EE)$
23. $\}$

Figure 8. Scheduling Algorithm

Shown as the Figure 8, the scheduling algorithm of simulation engine is actually based on the method of BL[7] protocol which first defines the unique Lookahead for every simulation model components to ensure the visibility of all those TSO [6]events in the future. The safe execution time window, i.e. $LBTS_{min}$, can be computed by Lookahead. The simulation model components inside the time window are permitted to advance, based on the priority to determine the their calling order. Thus in the scheduling algorithm of simulation engine, if below condition is satisfied, then the event x appearing before event y could be ensured:

1.y.pred = x or

2.x.ts + lookahead<y.tsor

3.[x.ts/LBTS$_{min}$] < [y.ts/LBTS$_{min}$]

Compared with BL method, the scheduling algorithm of simulation engine considers the scheduling priority of model components (line 3 in algorithm).Though it is invalid for TSO events, but still affects the ranking order of RO events in the event queue EL_{RO}.

D. *The simulation service based on the bridge pattern*

Simulation model components could call services provided by the simulation system framework by simulation events. In order to make the design of model components not affected by simulation services, the details of implementation between simulation events and simulation services should be separated. The bridge pattern [8] of the design pattern provides an effective implementation mechanism.

As shown in Figure 9, in the implementation structure of simulation services, the left side service is the abstract interface of simulation service, the right side is the service implementation module of simulation service and the middle one is the simulation adapter which bridges simulation service abstract interface and simulation service implementation.

The simulation service of simulation system framework include two parts: simulation service abstract interface and simulation service implementation. The simulation service abstract interface is exposed to users and offers necessary data for the simulation calls. While simulation service implementation accomplishes calls from simulation services on the base of data provided by simulation service abstract interface.

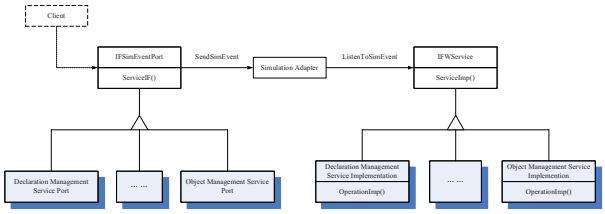


Figure 9. Implementation Structure of Simulation Services Based on the Bridge Pattern

V. THE REALIZATION OF RUNTIME INFRASTRUCTURE OF SIMULATION SYSTEM

A. *Architecture of Runtime Infrastructure of Simulation System*

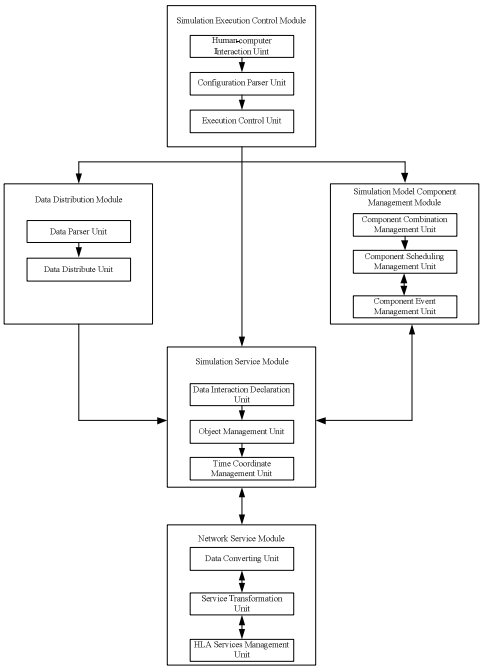


Figure 10. Modules of the Runtime Infrastructure of Simulation System

As shown in Figure 10, component-based centralized and distributed-integrative runtime infrastructure of simulation system includes: simulation execution control module, component management module, data distribution module, simulation service module and network service module.

1) Simulation Execution Control Module

This module is used to provide the human-computer interaction interface, control the state of starting, pausing, continuing and ending of simulation, and analyze the configuration information of the system, controlling commands of simulation. Moreover, through simulation service module, it establishes data interaction table among components, as well as the table between system and external federate in the distributed running mode. The simulation execution control module loads the data distribution module and the component management module, inputs the script data for data distribution module and the component initial information of system configuration for component management module respectively.

2) Simulation Model Component Management Module

The simulation model component management module includes the component combination management unit, the component scheduling management unit and the component event management unit. The component combination management unit obtains the system configuration information parsed by the execution control module, checks the completeness of component resources of simulation models, and loads components composed to build the simulation system. The component scheduling

management unit stores the component instances created by the combination management unit through the scheduling manager and manages the scheduling states of components during the simulation running. The component event management unit manages the simulation events generated in the process of scheduling management unit and calls the simulation service module to realize data interaction among model components.

3) Data Distribution Module

This module is used to parse the input simulation script data file under the centralized running model, obtain the information of simulation entities and their corresponding components. In the process of simulation running according to the time of its advancement, this module outputs temporal system configuration files, sends corresponding script data and calls the service module to distribute script data to model components by simulation events.

4) Simulation Service Module

This module is used to provide model components with simulation service set, accomplish data publishing and subscribing among components. Moreover it establishes data interaction table among components as well as table between system and external federates in the distributed running mode, maintains the registration and remove of model entities within system and coordinates time advancement of the runtime simulation system.

5) Network Service Module

The network service module is used for the transformation among the RTI serves and the framework simulation services under the distributed running mode. It can segregate model components and RTI, which ensures the data interaction between system's runtime infrastructure and external federates.

B. Distributed Application Mode

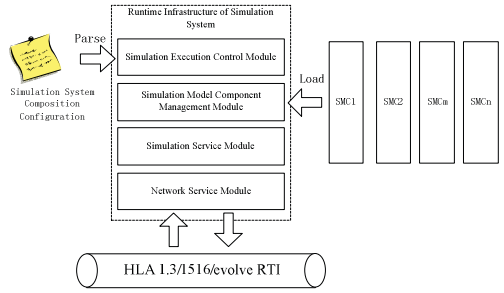


Figure 11. Distribution Running Mode

As shown in Figure 11, the Runtime Infrastructure of Simulation System reads the specified configuration files of system through human-machine interaction unit provided by Simulation Execution Module in the distributed running mode. And those files are parsed by the configuration analysis unit. The result of the parse information is used as the input of the component management module. According to the information, component management module loads simulation model components (SCMs) in specified locations to construct of simulation system. Through invoking the RTI services, successfully established system would be added to the HLA distributed simulation system as a federate in compliance with HLA specification and realizes interoperability with other federates. Because the runtime infrastructure and specified version of RTI are separated by network service module, different network service modules could be replaced with corresponding version of RTI without changing other modules of the runtime infrastructure of the simulation system. So that building the simulation system based on this instance, established by runtime infrastructure can run under different version of RTI and improve the reusability and transplantation of the system.

C. CentralizedApplication Model

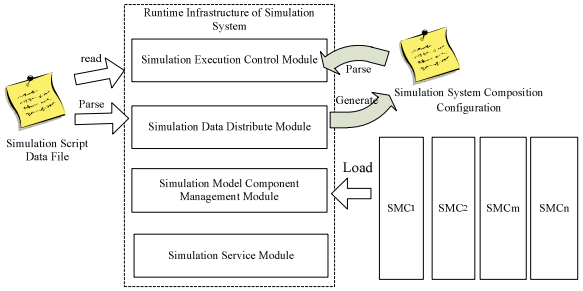


Figure 12. Centralized Running Mode

As shown in Figure 12, under the centralized running mode, the Runtime Infrastructure of Simulation System reads a simulation script data file through human-machine interaction unit provided by simulation execution control module. Firstly, data files are converted into system configuration files by data parse unit of simulation data module and analyzed through configuration parser unit of simulation execution control module. Then the script data is embedded into the system configuration file as an independent information node. Just as the distributed running mode, component management module loads components required by the construction of simulation system according to the system configuration information. While different from distributed running mode, the infrastructure would not need the network service module anymore but as an independent-running simulation system.

VI. SUMMARY

The distributed and centralized-integrative Runtime Infrastructure of Simulation System proposed in this paper could load component-based models according to specific requirements of simulation application system. Through composing model components, it not only improves the developing efficiency of simulation system, but also enhances the reusability of models, thus realizing flexibility of structure and customization of functions for simulation system. Besides, the separation between models and implementation details of specific runtime infrastructure of simulation system helps models to be applied in large-scale distributed simulation system as well as small-scale centralized system, which enhance the adaptability of models.

By using the component technology and data publishing and subscribing mechanism, the coupling degree between models has been reduced, which improves the parallelism [9] of simulation models. Achieving parallelism of model calculation and simulation scheduling is one of the research hotspots in simulation application area, which is also the further work of this paper.

ACKNOWLEDGMENT

This work was supported by the National Nature Science Foundation Grant funded by the People's Republic of China government (Project No.61074108). The authors would like to thank an anonymous referee for their constructive comments, which helped to improve the paper.

REFERENCES

[1] GONG Jian-xing, PENG Yong, HAO Jian-guo etc. .Research on Component-Oriented Methodology for Constructing Simulation Systems[J]. Journal of System Simulation. Vol.22 No.11, Nov., 2010: 2575-2578.

[2] SISO-STD-003.1-2006. Base Object Model (BOM) Template Specification [S]. SISO, 2006.3.

[3] SISO-STD-003.0-2006. Guide for Base Object Model (BOM) Use and Implementation [S]. SISO, 2006.3.

[4] IEEE Std 1516.2-2000 (2010) IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) — Object Model Template (OMT) Specification [S]. IEEE, 8.

[5] HU Rong, HUANG Ping. The Research of Middleware Technology Application Status[J]. Computer Knowledge and Technology, Vol.9, No.22, August 2013:4990-4991.

[6] IEEE Std 1516-2000 (2010) IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) — Frame and Rules [S]. IEEE,8.

[7] Lubachevsky B. D. Efficient Distributed Event-driven Simulations of Multiple-loop Networks [J]. Communications of the ACM 32(1) 1989 : 111-23.

[8] TANG Yan, HAN Ai-qing. Design and Implementation Based on the Bridge Patterns of Multi-platform Shared Memory Management[J]. Computer Knowledge and Technology, Vol.8, No.31, Nov. 2012:75927596.

[9] LI Ting-ting, HAN Liang, WANG Jiang-yun. Research and Application of Time Management in Parallel and Distributed Real-Time Simulations[J]. Journal of System Simulation, Vol.25, No.8 Aug., 2013:1783-1788.

A Multi-Granularity Grooming Scheme for One-to-many Multicast Traffic

Songzhu Zhang, Xingwei Wang, Min Huang
College of Information Science and Engineering
Northeastern University
Shenyang, 110819, P.R. China
wangxw@mail.neu.edu.cn

***Abstract*—With the vigorous development of Digital Broadcasting, the Internet of Things and Cloud Computing, there are more and more multicast applications in the network, leading to network traffic increases exponentially. Although optical network improves the network transmission capacity enormously, the growing network traffic still poses a challenge for the transport mechanism. Therefore, how to transport the communication request as much as possible at the cost of the minimizing energy consumption has become one of the primary problems in the future Internet. With considering network energy consumption, triggering grooming from low granularity channel to high granularity channel and Quality of Service (QoS) demands of the traffic request, this paper proposes a multi-granularity grooming scheme for one-to-many multicast traffic and designs a corresponding traffic grooming algorithm. Simulation results show that the grooming scheme proposed in this paper has better effects in the blocking probability and the energy saving.**

Keywords-multi-granularity; multicast; traffic grooming

I. INTRODUCTION

The emergence of Dense Wavelength Division Multiplexing (DWDM) technology reduces effectively the pressure of network bandwidth that comes from the exponential growth of Internet traffic. At the same time, optical network develops fast and becomes the main way of the Internet backbone network. The fiber can make wavelength division multiplexing with 160 wavelengths, while the transmission rate of the wavelength can reach 100Gbit/s. There is a huge gap between the bandwidth requirements of the business request and the wavelength capacity, which makes the serious waste of bandwidth resources. In order to allocate network resources reasonably and improve resource utilization, the multi-granularity technology is needed for traffic grooming.

With the improvement of the network bandwidth and transmission ability, multicast applications which have strict QoS requirements for network such as IPTV, video conferencing, telemedicine and online multiplayer gaming have becoming more and more popular. Therefore, it is a significant and urgent issue to construct a multi-granularity grooming scheme for multicast traffic.

II. RELATED WORK

At present, it is a research direction to groom a lot of low-speed traffic requests to fewer high-speed traffic channels.

According to whether traffic demands are known in advance or not, traffic grooming can be divided into static traffic grooming and dynamic traffic grooming. Static traffic grooming is to transmit the traffic that is known in advance among various switching nodes in the network with consuming the least amount of network resources. While dynamic traffic grooming is to converge the traffic to reduce the blocking probability under the condition of stochastic dynamic arrival of the traffic. Most early researches about traffic grooming focus on unicast request of ring net or mesh network. Nowadays, as a large number of network applications tend to multicast, multicast traffic grooming has been given much attention.

Multicast traffic can be divided into two types: one-to-many and many-to-many. There is only one source node and a number of destination nodes in one-to-many applications, while in many-to-many applications, each group member is not only the source node but also the destination node.

In the aspect of one-to-many static multicast traffic grooming, in [1], a Mixed Integer Linear Programming (MILP) optimization problem was formulated for multicast traffic grooming to design a light-tree based logical topology with an end-to-end delay bound, incorporating the constraint that the light-trees are given a priori. This reduces the complexity of the problem but may not lead to optimal results. The authors in [2] proposed the problem of grooming of given non-uniform multicast traffic demands on a unidirectional SONET/WDM ring is NP hard, and the goal was to try to minimize the network cost as given by the number of wavelengths required per fiber and the number of electronic Add-Drop Multiplexers (ADMs) required on the ring. The problem with cost function by the number of wavelengths required per fiber could be modelled as a standard circular-arc graph coloring problem. For cost function by the number of electronic ADMs required on the ring， a graph based heuristic algorithm was presented. In [3] a heuristic algorithm was presented to maximize the product of bandwidth-number of destinations according to partial multicast service discipline. In [4], an Integer Linear Programming (ILP) formulation was proposed for optimal assignments of hop constrained light-trees for multicast connections to maximize network throughput. The proposed heuristic algorithm with a polynomial complexity, called Dividable Light-Tree Grooming (DLTG) algorithm, achieves network throughputs which were very close to the ILP formulation results, but with far lower running times. The authors in [5] investigated delay constrained multicast routing for supporting QoS guaranteed point to multi-point communications in IP over WDM networks. The lightpath scheme was adopted to support multicast streams on IP/WDM networks, and hop count constraint was introduced to deal with and queue delay from traffic grooming. However, it is not enough only considering hop count constraint to support QoS guaranteed. The authors in [6] applied the quantum-based Immune Algorithm to minimize the network cost utilization rate and to maximize the QoS degree of the users.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.46

From the above related works, we can see that few studies have investigated the power consumption in static multicast traffic grooming. With considering QoS parameters of the bandwidth, the delay, the jitter, and the error radio, the paper introduces multi-granularity bandwidth traffic such as wavelength, waveband and designs multicast traffic grooming policies and multi-granularity grooming strategies and proposes a multi-granularity grooming scheme for multicast traffic.

III. PROBLEM STATEMENT

A. Network Model

1) Node model

Node model in multi-granularity network has been a hot topic discussed by researchers. To design a node with high efficient and low energy cost has become a new challenge. The node model in this paper consists of the multicast-capable multi-granular optical cross-connects (MCMG-OXC) in optical layer, optical-electronic-optical (OEO) converters between the layers and the core routing module in IP layer which can support multicast, as Fig.1.

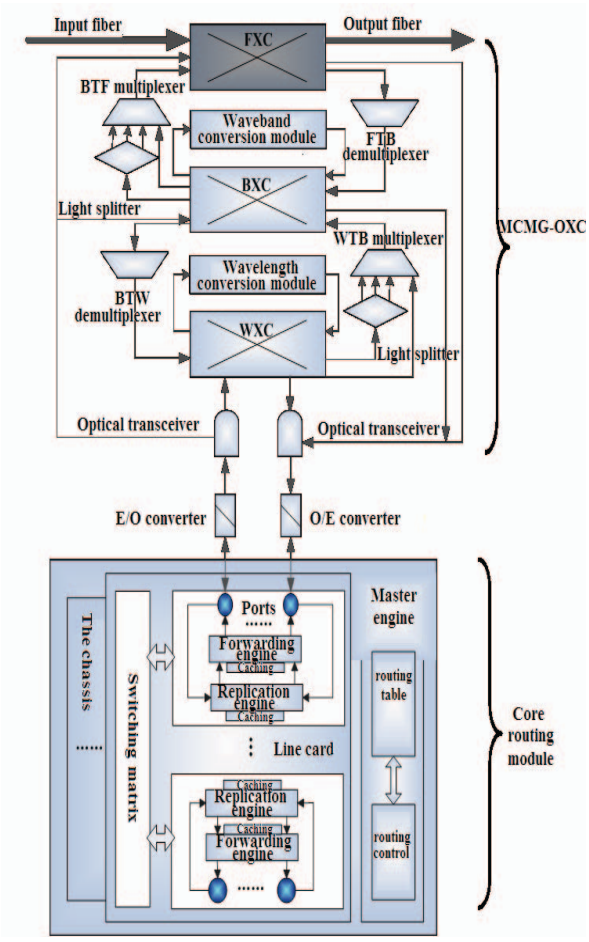


Figure 1. Node Model in Multi-Granularity Transport Network

In the node model of this paper， the core routing module of IP layer adopts modular structure which not only makes the router configuration flexible but also the related devices can be adjusted according to traffic needs to reduce energy consumption. The multi-granularity optical cross connection can not only support multicast traffic, but also can support multi-granularity switching of the wavelength, the waveband and the fiber and the different granularity information can be managed layered so as to reduce the amount of used port to achieve energy saving.

2) Link model

Adjacent nodes are connected by the fiber. In order to prevent the attenuation in the process of optical signal transmission, optical pre-amplifiers, optical inline-amplifiers and optical post-amplifiers are deployed on fiber links. The optical pre-amplifier can improve the transmission power of light; the optical inline-amplifier is used to compensate for loss of energy during the transmission; the optical post-amplifier is used to convert the optical signal into the signal that the optical receiver can accept.

B. The description of multicast traffic

The description of multicast traffic in multi-granularity transport network is the user's demands for network resources and QoS parameters. In this paper, 18 types of requests in ITU-T G.1010 documents are considered as request types of multicast traffic in multi-granularity transport network. Each request type $app_i (1 < i < 18)$ in request set $App = \{app_1, app_2, \ldots\ldots app_{18}\}$ corresponds to a set of QoS parameters. In this paper, 4 kinds of QoS parameters are considered and they are respectively the bandwidth, the delay, the jitter, and the error radio. So, each request type is related to the corresponding the demand of QoS parameters: $QoS_i=(b_i, dl_i, jt_i, l_i)$, where b_i denotes bandwidth of the request, dl_i denotes the delay of the request, jt_i denotes the jitter of the request, and l_i denotes the error radio of the request.

In this paper, the heterogeneous QoS of destination members in multicast requests are considered, i.e., the QoS parameters of each request node are different, which augments the complex of grooming, but it is closing to the real situation [7]. An One-to-many multicast traffic request is defined as $r^{otm}(s, D, App_t)$ ， where $s \in V$ denotes the source node of the request, $D = \{(d_1; b_1, dl_1, jt_1, l_1) \ldots (d_n; b_n, dl_n, jt_n, l_n)\}$ denotes the set of multicast destination members, n denotes the number of multicast members, $d_i \in V$ denotes the destination member and $d_i \neq s$ ， b_i, dl_i, jt_i, l_i respectively denotes the needed bandwidth, the delay, the jitter, and the error radio of the destination member d_i of the request. App_t denotes the application type of the request of one-to-many multicast traffic, where $t \in [1,18]$ and the step length is one.

C. Multicast traffic Grooming Policy

This paper proposes four policies and employs for the convergence of multicast requests according to the condition of multicast, as follows:

Scheme 1 is called single-tree grooming. The multicast request is groomed by an existing light-tree which can overlay source nodes and destination nodes of the multicast request.

Scheme 2 is called multi-tree grooming. The multicast request is groomed by several existing light-trees which can overlay source nodes and destination nodes of the multicast request.

Scheme 3 is called newly-founded tree grooming. Establish the connection for a multicast request on a newly founded light-tree with free wavelength links. This scheme will consume new transceivers at source node and all destination nodes of the newly-founded light-tree.

Scheme 4 is called hybrid-tree grooming. Some destination nodes of the multicast request are groomed by existing light-trees, the others are groomed by newly-founded light-trees. Since it considers Scheme 1, Scheme 2, and Scheme 3 jointly, it is called the hybrid Scheme.

D. Analysis of energy consumption

Energy consumption of the network is divided into two parts: energy consumption of the node and energy consumption of the link. Energy consumption has the following formula (1).

$$P_{total} = \sum_{n \in N} (P_n^w + P_n^b + P_n^f) + \sum_{l \in L} (P_l^w + P_l^b + P_l^f) \qquad (1)$$

Where, N is the set of nodes, L is set of links, P_n^w, P_n^b, P_n^f respectively denotes the energy consumption of any node in wavelength level, waveband level, fiber level. P_l^w, P_l^b, P_l^f respectively denotes the energy consumption of any link in wavelength level, waveband level, fiber level.

E. Construction of the auxiliary graph

The auxiliary graph can be used to analyze multicast traffic grooming. In [8] the authors proposed the related design of the auxiliary graph, which is used by the paper and is expanded to the fiber grooming.

F. Multi-granularity grooming strategies

Each node in multi-granularity transport network is MCMG-OXC and supports multi-granularity switching that not only can improve utilization efficiency of network resources but also provide a differentiated QoS guarantee. In this paper, the multi-granularity means the wavelength, the waveband, and the fiber and different granularity has the different carrying capacity. Multi-granularity grooming scheme is upward integrated, that the wavelength (including sub-wavelength) is groomed to the waveband, the waveband is groomed to the fiber. Therefore, grooming strategy can be divided into the following two:

1) The waveband grooming strategy

The condition of triggering waveband grooming:

a) The physical path from requesting of the source node S to the destination node d_i is more than one hop.

b) There are needed multiple wavelength links, which use the same path, from requesting of the source node S to the destination node d_i to transport.

When the condition of triggering waveband grooming is met, it is needed to deal with according to the following two ways:

a) If there is not a waveband channel in the path, it is needed to establish a new waveband channel.

b) There is at least a waveband channel in the path, the next is to judge whether the remaining waveband channel can meet the demand for bandwidth of the traffic request. If satisfied, the remaining bandwidth will be assigned to the traffic request; otherwise, the new waveband channel will be established.

2) The fiber grooming strategy

The condition of triggering fiber grooming:

a) The physical path from requesting of the source node S to the destination node d_i is more than one hop.

b) There are needed multiple waveband links, which use the same path, from requesting of the source node S to the destination node d_i to transport.

When the condition of triggering fiber grooming is met, it is needed to deal with according to the following two ways:

a) If there is not a fiber channel in the path, it is needed to establish a new fiber channel.

b) There is at least a fiber channel in the path, the next is to judge whether the remaining fiber channel can meet the demand for the bandwidth of the traffic request. If satisfied, the remaining bandwidth will be assigned to the traffic request; otherwise, the new fiber channel will be established.

IV. THE DESIGN OF GROOMING SCHEME

A. The grooming thinking

What is crucial for one-to-many multicast traffic type is to find an existing multicast tree or construct a new tree to meet the demand of the traffic. Firstly, it is needed to merge the similar multicast request together under the constraints of QoS. The similarity refers to: for example, two or more multicast requests have the same source nodes and destination nodes, two or more multicast requests have the same source nodes and part of the same destination nodes, two or more multicast requests have different source nodes and the same destination nodes, two or more multicast requests have different source node and part of the same destination nodes, and so on.

The proposed grooming scheme has two rounds of screening process. First of all, a set of multicast request is divided into three categories according to the destination members. Videlicet, the destination members are totally the same, destination members are part of the same, and destination members are totally different.

Secondly, for one kind of multicast request, because the same destination of different multicast requests have different QoS demands, it is necessary to screen the second round. The member in the same destination member with the demand of QoS that varies within a certain range will be reserved; otherwise, it will be removed. The Specific solution: $R^1(r_1, r_2, ..., r_m)$ is the set of the number of m requests after the first round of screening. v_{QoS} is the set of QoS parameters of the node v_i who has the same destination in each different multicast request, $v_{Qos} = \{r_1^{v_i}(b_1^{v_i}, dl_1^{v_i}, jt_1^{v_i}, l_1^{v_i})...r_m^{v_i}(b_m^{v_i}, dl_m^{v_i}, jt_m^{v_i}, l_m^{v_i})\}$,and $b_m^{v_i}, dl_m^{v_i}, jt_m^{v_i}, l_m^{v_i}$ respectively denotes the bandwidth, the delay, the jitter and the error radio of the node v_i of the destination member of the request m. The different QoS

parameters of the node v_i in different requests can be calculated by formula (2). If the results meet the conditions, the destination members of the request are reserved, otherwise, v_i will be removed.

$$\begin{cases} \partial < \left| dl_i^{v_i} - \dfrac{\sum dl_i^{v_i}}{m} \right| < \beta & i \in (1,2...m) \\ \eta < \left| jt_i^{v_i} - \dfrac{\sum jt_i^{v_i}}{m} \right| < \varepsilon & i \in (1,2...m) \\ \delta < \left| l_i^{v_i} - \dfrac{\sum l_i^{v_i}}{m} \right| < \theta & i \in (1,2...m) \end{cases} \quad (2)$$

Where, (α,β) denotes the fluctuation range of the delay, (η,ε) denotes the fluctuation range of the jitter, (δ,θ) denotes the fluctuation range of the error radio. There the bandwidth is unlimited, because the sum of each request bandwidth can be processed by the corresponding multi-granularity scheme.

The requests that are screened two rounds later can be seen as one request, but the update of related QoS needs to be processed. $R^2(r_1,r_2,...,r_i)$ denotes the set of the request by two rounds of selection, i is the number. $R^2(r_1,r_2,...,r_i)$ can be merged to one request, and QoS parameters of the node of the same destination member in different requests are updated, we can get $R^{new}(S,[d_1(b_1^{update},dl_1^{update},jt_1^{update},l_1^{update})...d_n^{update}(b_n^{update},dl_n^{update},jt_n^{update},l_n^{update})]$, where S is the set of the request source nodes, i.e., all the request source nodes in the set R^2 are added to the set S; d_1 is the first node of the destination member, and $b_1^{update},dl_1^{update},jt_1^{update},l_1^{update}$ respectively denotes the bandwidth, the delay, the jitter and the error radio of the node d_1 of updated destination member. The QoS parameters of the destination members are updated according to the formula (3). For the destination member d_1, the updated bandwidth is the sum of bandwidth of d_1 in different multicast request, the updated delay is the minimum delay of d_1 in different multicast request, the updated jitter is the minimum jitter of d_1 in different multicast request, and the updated error radio is the minimum error radio of d_1 in different multicast request. According to the update rules above, other destination members are updated in turn.

$$\begin{aligned} B_1 &= \sum b_j^1 && ,j \in (1,2...i) \\ DL_1 &= \min dl_j^1 && ,j \in (1,2...i) \\ JT_1 &= \min jt_j^1 && ,j \in (1,2...i) \\ L_1 &= \min l_j^1 && ,j \in (1,2...i) \end{aligned} \quad (3)$$

B. The grooming algorithm

For static multicast traffic requests, generally the integer linear programming is adopted to optimize. Because the static multicast traffic grooming is NP problem, solving the ILP problem may become computationally prohibitive, so the heuristic algorithm is used to solve the problem efficiently. The Static Multicast Traffic Grooming is shown in Algorithm 1.

Algorithm 1 The Static Multicast Traffic Grooming

Input: A set of multicast traffic request
Output: Muticast-trees

```
Join all multicast traffic request into R
// R is the set of the resource on the link
Arrange these requests in descending order
while the first element value of R not is 0, do
  if the first element value of R is 0, then
    end Algorithm
  else
    while the elements of intersection got is K, do
      // K is the number of destination nodes of R1
      Join the request into R1 and remove it from R
      K= K-1
      if K not is 0, then
        Handle these sets: R1, R2,……, Rn
      else
        break
      end if
    end while
    Construct new light-tree for the request of set R
  end if
end while
```

V. SIMULATION AND ANALYSIS

In this paper, the multi-granularity grooming scheme for multicast traffic is implemented by the Java language, running environment is Microsoft Windows 7 SP1 system, and the development tool is MyEclipse 10.

A. The benchmark algorithm

In order to evaluate the performance of the multi-granularity grooming scheme for multicast traffic designed by this paper more objectively and effectively, QoS demands and static traffic grooming are joined in LTD-ANCG algorithm proposed by literature [9] for comparison and analysis.

B. The indexes of performance evaluation

The main purpose of the multi-granularity grooming scheme for multicast traffic is to solve the problem of network block which caused by too much traffic in network and the problem of large energy consumption which caused by wasting of network resources. Therefore, the evaluation indexes of the algorithm in this paper are the blocking probability of traffic and the energy consumption of the network. The energy consumption of network is compared by the different traffic intensity. Traffic intensity is the product of the number of the arrival traffic request in a unit time δ and the average duration time of the traffic request $1/\mu$, and its unit is Erlang. When it is implemented, request arrival rate of the traffic request in the network is Poisson process.

C. The performance evaluation of the grooming algorithm

We have studied the performance of One-to-many Static Multicast Traffic Grooming (OTMSTG) algorithm

and TD-ANCG algorithm in the tested network topology, EON that has 28 nodes and 61 links. We compute the energy consumption and the blocking probability of OTMSTG and TD-ANCG when the number of traffic request is 50, 100, 150, 200, 250, 300, and the results are as shown in Fig.2 and Fig.3.

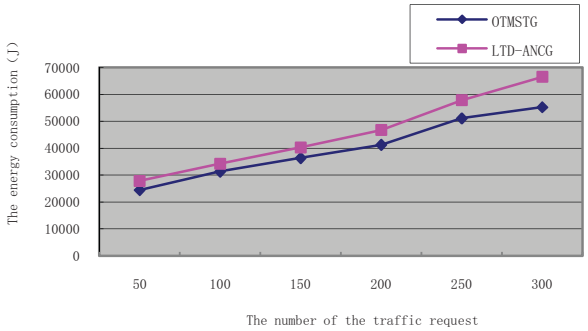


Fig.2 The comparison of energy consumption of multicast grooming algorithms

Fig.2 shows the energy consumption of OTMSTG algorithm is obviously lower than TD-ANCG algorithm, because the multicast light-tree of OTMSTG algorithm can cover more destination nodes of the multicast request. But the light-trees founded by TD-ANCG algorithm are divided into several small child light-trees which are too small to cover multicast requests. So it is needed to establish the new light-tree to groom the multicast request in TD-ANCG algorithm and the energy consumption increases. As the number of communication is increasing, the energy consumption gap between the two algorithms is becoming more obviously. Thanks to the sharing probability of light-tree established by OTMSTG will improve with the increase of the number of communication, the energy consumption will reduce. However, with the increase of the number of communication, more light-tree are needed to establish to meet the multicast request in LTD-ANCG algorithm, which leads to increase the number of optical transmitter, optical receiver, routing ports and so on, so the energy consumption will continue to rise.

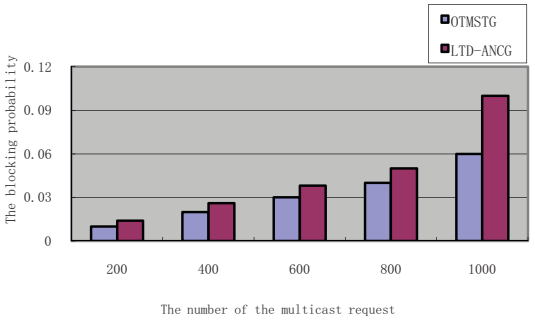


Fig.3 The comparison of blocking probability of multicast grooming algorithms

As shown in Fig.3, the blocking probability of OTMSTG algorithm is lower than LTD-ANCG algorithm, and this advantage will be more obviously with the increase of the number of requests. Because multicast traffic matrix can be known a priori, resources can be allocated reasonably, the whole blocking probability of static multicast is relative low. As the number of communication request increases, the idle network resources will reduce, which leads to the blocking probability rise. Since the multicast tree founded by OTMSTG algorithm can cover more destination members of the multicast request, the blocking probability of OTMSTG algorithm is lower than LTD-ANCG algorithm.

VI. CONCLUSIONS

Simulation results show that the grooming scheme proposed by this paper can reduce the blocking probability and the energy consumption under the condition of meeting user's QoS demands, so it is feasible and effective.

ACKNOWLEDGEMENT

This work is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant No. 61225012 and No. 71325002; the Specialized Research Fund of the Doctoral Program of Higher Education for the Priority Development Areas under Grant No. 20120042130003; the Fundamental Research Funds for the Central Universities under Grant No. N110204003 and No. N120104001.

REFERENCES

[1] De-Nian Yang; Wanjiun Liao, "Design of light-tree based logical topologies for multicast streams in wavelength routed optical networks," INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies , pp.32-41 March-3 April 2003,doi: 10.1109/INFCOM.2003.1208656

[2] Rawat, A; La, R.; Marcus, S.; Shayman, M., "Grooming multicast traffic in unidirectional SONET/WDM rings," Selected Areas in Communications, IEEE Journal on , vol.25, no.6, August 2007,pp.70-83, doi: 10.1109/JSAC-OCN.2007.025806

[3] Turkcu, O.; Somani, AK., "Efficient multicasting approaches using collection-distribution networks," INFOCOM, 2011 Proceedings IEEE , pp.141-145, April 2011,doi: 10.1109/INFCOM.2011.5934928

[4] Rongping Lin, Wen-De Zhong, Sanjay Kumar Bose, Moshe Zukerman, Constrained light-tree design for WDM mesh networks with multicast traffic grooming, Optical Switching and Networking, Volume 10, Issue 3, July 2013, pp. 233-245

[5] Hong-Hsu Yen; Lee, S.S.W.; Mukherjee, B., "Traffic Grooming and Delay Constrained Multicast Routing in IP over WDM Networks," Communications, 2008. ICC '08. IEEE International Conference , pp.5246-5251, May 2008 , doi: 10.1109/ICC.2008.985

[6] Ren Na; Zhang Nan; Wang Hongjiang; Zhang Wenqiang, "Static Traffic Grooming Mechanism Based on Quantum Immune Algorithm and Game Theory," Management and Service Science (MASS), 2011 International Conference , pp.1-3, Aug. 2011, doi: 10.1109/ICMSS.2011.5998139

[7] Xingwei Wang, Hui Cheng, Min Huang, "Multi-robot navigation based QoS routing in self-organizing networks", Engineering Applications of Artificial Intelligence, Vol. 26, Issue 1, January 2013, pp. 262-272,doi:10.1016/j.engappai.2012.01.008.

[8] Xingwei Wang, Hui Cheng, Keqin Li, Jie Li, Jiajia Sun, "A cross-layer optimization based integrated routing and grooming algorithm for green multi-granularity transport networks", Journal of Parallel and Distributed Computing, Vol. 73, Issue 6, June 2013, pp 807-822,doi:10.1016/j.jpdc.2013.02.010.

[9] Rongping Lin; Wen-De Zhong; Bose, S.K.; Zukerman, M., "Dynamic Sub-Light-Tree Based Traffic Grooming for Multicast in WDM Networks," Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE , pp.1-5, Dec. 2010 , doi: 10.1109/GLOCOM.2010.5684181

Study on the Architecture of Intelligent warship's TSCE based on multi-view

He Yelan
College of Computer Science & Technology
Wuhan University of Technology
Wuhan, China
Heyl99@163.com

Chen Hui
Key Laboratory of Marine Power Engineering and Technology of Ministry of Communications
Wuhan University of Technology
Wuhan, China
chenhui_whut@126.com

***Abstract*—Intelligent ship TSCE(Total Ship Computing Environment) is a new generation of shipboard integrated system being developed for current and future combat mission, which leads to changes in design and integration mode of warship system. SOA (Service-Oriented Architecture) is the technology foundation of implementing system integration and the emergence of CPS (Cyber-Physical Systems) provides a new technical idea for TSCE construction. Based on analyses of integration demands for intelligent warship, this paper uses SOA and CPS to design architecture for TSCE from three aspects, which are system view, function view and technology view. And this paper performs comparison and analysis with existing architecture. The result shows that the presented architecture is more comprehensive and advanced, which lays a solid foundation for TSCE research in our country.**

Keywords- TSCE; SOA; CPS; architecture

I. INTRODUCTION

High degree of integration and automation is an irresistible trend of the intelligent ship development. The latest progress of intelligence ship integration issues is TSCE (Total Ship Computing Environment) proposed by US navy. Using network communication and computer control technologies, TSCE integrates all equipments of network, calculation, storage, display and internal communication, to "Sustain all warship system with one public computation and demonstration environment" [1].

From the first proposal of TSCE so far, limited information can be got. The first complete achievement of the TSCE is destroyer DDG-1000 that is still under construction, whose TSCE architecture is also restricted by technological level at the beginning of its design. Domestic related research is still in its infancy. So it is necessary to study on TSCE, so as to improve the integration degree of our country's warship and to meet the requirement of our national defense. In this paper, the real-time SOA (Service-Oriented Architecture) and CPS (Cyber-Physical Systems) technologies will be applied to the design of TSCE, to fulfill the future demand for system integration of intelligent warship, and a new type of architecture for TSCE are proposed from three views, which are system, function and technology.

II. ANALYSES OF INTEGRATION DEMANDS FOR INTELLIGENT WARSHIP

A. *Adapting to Varied Task Environments and Anti-corruption*

Integrated system should support all kinds of computing tasks of warship. Warship task is changeful, therefore, all the compute resources, inner or outer, should be integrated in the state of "plug and play" state, system configuration should be able to adjust with the operational mission changes, in order to guarantee the normal operations to meet the Qos (Quality of Service) requirements. In the case of some computing resources are damaged and fail to work, the implementation of critical tasks should be ensured with resources sharing function of TSCE.

B. *Improving Technology Adaptability*

Modern computers and related technologies are developing rapidly. In contrast, the developing cycle of the warship system is relatively long. So TSCE integrate architecture should have futures of reusability and ability to upgrade of warship hardware and software, to tap into the fullest potentials of the existing warships, and to prolong their service lives.

C. *Support Higher Automatic Level*

Intelligent warship requests a higher and higher automatic level, in order to reduce ship staffing. Therefore, TSCE should have features of multi-system linkage and integration, to eliminate the crystallized pattern that weapons and sensors must be strictly matched in pairs, to cooperate make all kinds of automation systems distributed in different locations of the warship, and to instantly make the optimal decisions according to the whole ship conditions.

III. IDEAS OF TSCE DESIGN BASED ON SOA AND CPS

A. *TSCE Core Idea*

TSCE, composed by meshing computers, is a high-performance distributed real-time computing environment, which solves the integration problems of poor interoperability among warship systems and failed sharing of resources. According to the analyses of integration demands for intelligent warship, TSCE should offer a platform for shipboard systems, which is configurable, scalable, self-organized and high leveled of system integration and automation. The OA (Open Architecture) is the core idea of TSCE, therefore, COTS (commercial off the shelf) products are often put to use in the software and

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.48

hardware system, when middleware and services are used to integrate distributed software system.

DDG-1000 TSCE is using a real-time CORBA middleware technology as the core integration technology. This traditional SOA technology relies on the ORB environment, so it is not easy to expand the system; In addition, CORBA-based application is complicated to develop. Combined with the SOA technology development, improvement of the software integration framework becomes an inevitable tendency of TSCE. Furthermore, the development of CPS technology provides a new idea for shipboard network system integration.

B. *Using SOA to Achieve Seamless Integration of Software Systems*

The SOA architecture is an effective mechanism to deal with the problems of interaction, integration and reuse of subsystems. Based on open standards, SOA achieves its system integration with a rapid assembly and reintegration of the loose coupling and coarse-grained services. On the basis of SOA, virtual resource management architecture controls and schedules resources independently, so as to realize on-demand service. This integrated mode makes it easier to realize interoperability and resource-sharing among different nodes in TSCE, each of which can realize the control of warship's various systems.

The TSCE services have the general characteristics the same as Web services, such as reusable and loosely coupled, but difference is that, TSCE services have the attribute of safe and reliable, real-time and efficient etc. The warship system has distinct domain features and applications, while SOA can be achieved on different underlying technologies. Future trend of TSCE software integration is coexistence of various standards SOA, so as to meet the various levels of shipboard system requirements. For example, detection and weapon system with high real-time requirements can use traditional real-time CORBA technology, and the command control system and other non-real time applications can use Web services based on J2EE. It is necessary to integrate and improve traditional SOA and SOA based on web services according to their characteristics, such as to perform real-time transformation for Web services, or to encapsulate CORBA operation with Web services.

C. *Using CPS Technology to Realize Seamless Integration of Sensor 、 Information and Control Platform*

CPS is a new research field related to interaction and fusion of information systems and physical system. CPS emphasizes the integration of computing, communication, control and physical systems. CPS can adapt to environmental uncertainty changes. It can carry out dynamic self-organizing reconstruction and can implement integration control based on distributed real time network. It also can make the physical system with the computing, communication, control, remote collaborative and autonomous function [3-4].

The basic characteristics of CPS are real-time, distributed, high reliability, strong security, diversity, autonomy [5], which coincided with TSCE demand very well, therefore, it is feasible to use CPS technology to realize organic and depth integration of ship computational resources and physical resources, namely to realize seamless integration of sensors, network, calculation and control units all over the warship. In addition, CPS communication protocol can be used as reference for real-time reconstruction on SOA [6].

IV. TSCE ARCHITECTURE BASED ON THREE VIEW

A. *TSCE structure of software view*

Guided by the principle of open architecture, combined with the development of SOA technology and CPS technology, improved on the basis of DDG-1000 TSCE architecture, the proposed TSCE architecture of software view is proposed shown in Figure 1. The architecture is divided into five layers from bottom to top.

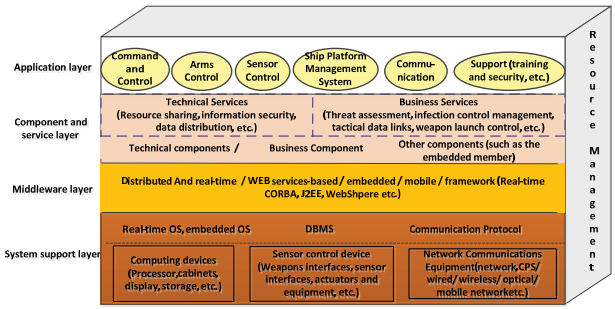


Figure 1. TSCE structure of software view

(1) System support layer. With the application of CPS technology, intelligent processor chips will embed in more shipboard systems. These chips communicate with other computing devices and sensor device to realize perception, execution control and data report. More communication modes are used to satisfy complex network environment of warship.

(2) Core of SOA. Middleware and component & service layer isolates applications from infrastructure. Component library is established on middleware platform, when components and business processes will be packaged as appropriate services and will provide access interfaces. These components can have multiple service interfaces contained in different services. Some embedded components are packaged as web services, for the remote terminal to use. When a change of the system mission happens, TSCE will make application dynamic configuration, to implement discovery, management and assembly of services dynamically, and to deal with the change by Self -reconfiguration.

(3) Resource management layer. The TSCE resource manager can monitor network, middleware and component service resources in a distributed system environment, so that it will realize the function of fault recovery, dynamic resource allocation etc., so as to ensure that the desired QoS can be achieved.

B. *TSCE architecture of functional view*

TSCE is a distributed, real-time and embedded large system. Based on the understanding for working principle of CPS and functional requirements of TSCE, TSCE architecture of functional view is proposed in Figure

2.Layers are divided clearly, and the relationship between boundaries is clear, as TSCE parts are classified accurately according to their functions.

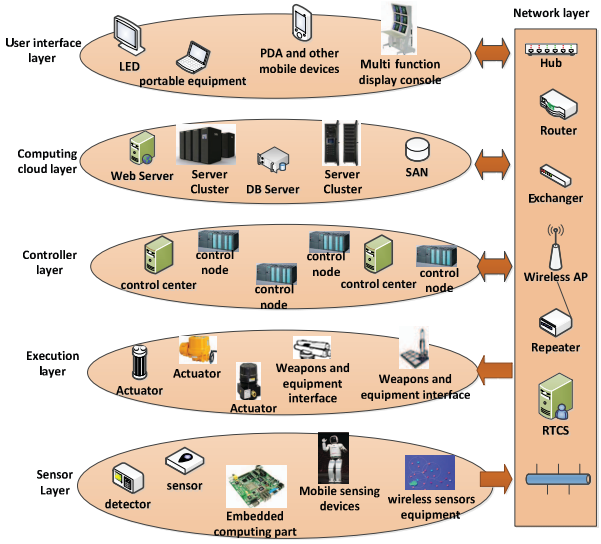


Figure 2. TSCE structure of function view

Sensor layer achieves functions of equipment monitoring, data acquisition, environmental perception, and intelligent data preprocessing. The preprocessed data will be transmitted to control layer and computing cloud layer for global fusion. Execution layer implements interaction with warship physical device according to instructions receive from controller layer.

Controller layer includes IPC (Industrial Personal Computer), PLC (Programmable Logic Controller) and single board computer. Controller layer receives strategy from computing cloud layer and translates it into instructions which will be send to control actuator unit to act appropriately. Computing cloud layer is the data fusion and information processing center of warship systems. Resources in cloud will be allocated dynamically in run time. Devices in these two layers can be placed in any appropriate position of ship.

User interface layer submits various command personnel decisions and combat tasks to computing cloud layer. Separating display and application can flexibly allocate the functions and tasks of operators or stations. Network layer provides low latency, high bandwidth and fast fault recovery capability, as well as reliable, secure network service.

Sufficient devices redundancy is required to guarantee system reliability.

C. TSCE architecture of technology view

There are a series of key techniques to break through when developing TSCE. From the technical point of view, TSCE architecture of technology view is proposed to analyze technical problems which need to be solved in development of TSCE.

(1) Perception recognition technology. Diversified forms of technologies will be used in warship such as intelligent sensor, WSANs, mobile sensor networks because they can get data more flexibly and accurately, and their intelligent processing ability and autonomy directly determine the degree of automation of warship systems.

(2) Network communication technology. As the ship environment is dynamical, solely to enhance the capacity of communication system cannot solve the problem, to which the key solution is to integrate the different communication resources, thus the network fusion technology is important.

(3) Information processing technology. Warship systems have huge amounts of information and complex processing requirements, which requires the cooperation of various levels of technology. Virtualization and cloud computing technologies can solute the problems of resource sharing and information distribution. How to accomplish the task more reliable while saving more energy is a problem should be solved in embedded technology. Information fusion, data mining and huge data processing technology are all the key factors effecting the real time and reliability of system.

(4) Intelligent control technology. Warship system is highly intelligent, while traditional control method is difficult to solve the control problem in TSCE. We will continue to explore new intelligent techniques characterized by self-organize, self-adaptive, self- learning.

(5) System management technology. Dynamic resource management technology on various types of hardware and software resources is the core guarantee for TSCE to maintain efficient operation. Adaptive real-time distribution mechanism as well as information cognition and content based delivery can be referenced to meet the needs of data distribution technology. Service management technology is the problem that SOA architecture must consider.

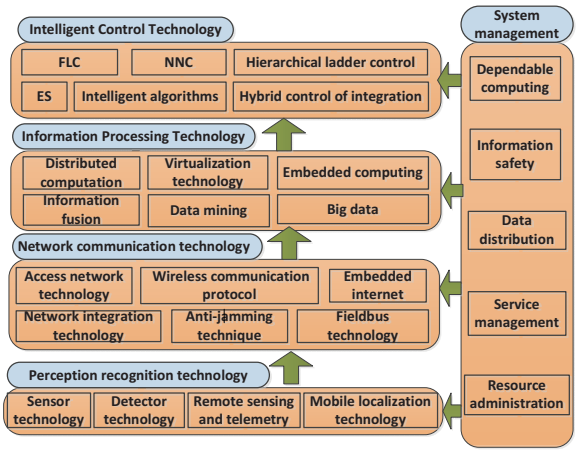


Figure 3. TSCE structure of technology view

V. ANALYSIS AND COMPARISON

TSCE system structure based on three views in this paper is compared with the TSCE architecture design in literature [7][8], the results as shown in table 1.

Relative to the literature [7][8], the three views presented in this paper restrict and connect each other, which completely describes the architecture of TSCE. Relative to the literature [7][8], the three views this paper put forward restrict and connect each other, which

described a complete architecture of TSCE. Real-time SOA and CPS technologies further strengthen seamless integration of the overall sensor, network, calculation and control unit in warship. Services definition in document [7][8] is not clear. Referring to the concept of military information service, this paper clarifies the difference between TSCE services and web services; this services definition is more suitable for the description and design of marine systems. According to CPS and technology of the Internet of things, this paper proposes the application of real-time distribution mechanism, and improvement of non-real-time protocol, making the system more intelligent and more real-time. Therefore, the TSCE architecture proposed in this paper is advanced, which lay the foundation for the further study of TSCE.

TABLE I. COMPARISON OF SEVERAL KINDS OF TSCE ARCHITECTURE

Standard	Architecture		
	DDG-1000[7]	**Document [8]**	**This paper**
frame content	technical level, function level, physical level	technical architecture、integration framework	software view, function view, technical view
design technology	CORBA middleware	SOA	SOA、CPS
Services definition	common services、field service	common services 、field service	TSCE services
data distribution	DDS/J2EE	DDS/ESB	DDS/ Adaptive real-time distribution mechanism
real-time	Considered	Considered	Consider the improvement of the non-real time protocol
technology adaptability	CTOS	CTOS	CTOS、virtualization etc.

VI. CONCLUSIONS

TSCE is a milepost of warship integrated method, which represents the advanced level of warship integration technology. TSCE has been receiving more and more attention of researchers at home and abroad, but few research results are achieved at present. This paper designs architecture for TSCE from three views. SOA, CPS, cloud computing, virtualization and other new technologies can be applied in TSCE to satisfy the demand such as anti-damage, real-time, seamless integration of future warship.

REFERENCES

[1] Huang Yong, “Analysis of foreign new destroyer Combat Systems, ” Ship Electronic Engineering, vol.30, Dec. 2010, pp. 4-8

[2] Xiao Min Yan , Dong Han Quan, “software architecture of the submarine combat system based on SOA”, Fire Control and Command Control , vol.26, Apr. 2011, pp. 76-79.

[3] Wen Jin Rong,Wu Mu Qing,Shu Jing Fang, “Cyber physical system”. Journal of automation, vol.38, Apr. 2012, pp. 507-517.

[4] He Ji Feng. “Cyber Physical Systems”. China Computer Society Newsletter, vol.6,Jan. 2010, pp. 25-29.

[5] [5]He Min，Liang Wen Hui，Chen Guo Hua，Chen Qiu Li，Lian Xiang Lei. “Research on architecture for cyber-physical systems based on multi-view,” Computer Engineering and Applications, ol.49,Dec. 2010, pp. 25-32

[6] Hu Xi Shuang, “Research into Architecture framework of submarine campaign system based on SOA and CPS”, Shipboard Electronic countermeasure, vol.26, May. 2013, pp. 5,1-5+31.

[7] Dong Xiao Ming, Shi Zhao Ming, Huang kun,Wang Yun Feng. “ Analysis on the Architecture of USN DDG-1000 Total Ship Computing Environment”. Chinese Journal of Ship Research. , vol.7, Jun. 2012, pp. 6,7-15.

[8] Dong Xiao Ming, Feng Hao,Shi Zhao Ming, Huang kun,Yao Jin,Zhang Jian. “ Architecture and System Integration Framework of Total Ship Computing Environment,” Chinese Journal of Ship Research. , vol.9, Feb. 2014, pp. 1，8-13,30.

Research on the Cross-System Collaboration Model of E-Government in China: From the Perspective of System Elements

Hu Changping, Chen guo
Center for Studies of Information Resources
Wuhan University
Wuhan, China
Email: delphi1987@qq.com

***Abstract*—in former researches, the constructions of cross-system collaboration model of e-government are mostly limited to some specific technical architecture. To resolve this problem, we constructed a general conceptual model of cross-system collaboration from the perspective of the e-government system elements. Accordingly, the decomposed sub models are discussed, including the resource sharing model, the technology fusion model and the service collaboration model. Each of these sub models are constructed based on the conclusion of general characteristic of former models.**

Keywords- E-government; cross-system; collaboration model; resource sharing; technology fusion; service collaboration

I. INTRODUCTION

The e-government construction has been developed rapidly in China in recent years. It is generally believed that the construction phase of e-government includes the information release platform construction, the online interactive office and the cross-system collaboration of e-government [1]. In the early phase of e-government construction in China, most local governments has built their office systems independently, leading to heterogeneity of data source and incompatible of technology, which has now become the main challenge in e-government construction in China. A main focus of the Chinese e-government construction in the future is to push the cross-system collaboration [2]. Therefore, the cross-system collaboration model of e-government is becoming a new research hotspot. In this paper, we will construct the model from a perspective of system elements of e-government, which is quite a different way comparing with the former researches.

II. THE CONCEPT MODEL FOR CROSS-SYSTEM COLLABORATION OF E-GOVERNMENT

The collaboration of e-government systems is a complex project, so it's necessary to construct the model of it using the top-down approach. Instead of be confined to the information resources sharing, the content of cross-system collaboration focus on the integration of many elements, including its resources, insiders, technologies, infrastructure and so on[3]. Li Manbo believed that the collaboration of e-government contains the personnel collaboration, the resource sharing, the work-flow fusion and the application integration [4]; Zhang Jian divided the collaboration model of e-government into personnel collaboration, information cooperation and work-flow fusion [5]; Lei Yinzhi proposed an integration schema composed of the information sharing, the process integration and the system integration [6]. These researches have all constructed the e-government collaboration model based on the decomposition of system elements. Accordingly, we will construct a new collaboration model based on its elements by abstracting the e-government system as a general information system. As a kind of information system, every e-government system, no matter what the function of it and where it has been applied, contains the elements of resource, technology, service and management. So the cross-system collaboration of e-government can be decomposed into resource sharing, technology fusion, service cooperation and management coordination. The relationship between the same elements from different systems should be clarified in the cross-system collaboration model. Considering the above we can propose a concept model which is briefer and more general than before, as shown in Figure 1.

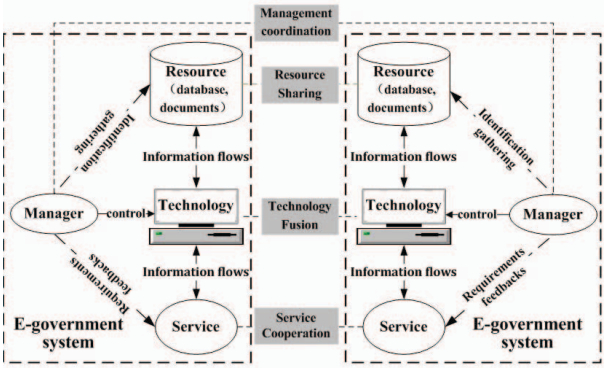


Figure 1. The concept model based on the collaboration of system elements

III. THE RESOURCE SHARING MODEL FOR CROSS-SYSTEM COLLABORATION OF E-GOVERNMENT

Information resource is the most basic element in information systems. All administrative activities in e-government are carried out based on the massive amount of distributed information resource. The resource sharing model aims to build a safe and effective architecture of data exchange among various heterogeneous systems. There are three common requirements of the data sharing model. Firstly, the public information (such as the social insurance information, the education information, the

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.49

employment information and so on) should be integrated either logically or physically. Secondly, a peer-exchange mechanism of data exchange should be built based on uniformed data standard. Thirdly, the data acquisition channels should be developed business-oriented [7]. Considering the differences among e-government services in reality, we can generalize two modes of resource sharing in e-government: the data federation mode and the data integration mode. The detail of them can be shown as Fig.2. These two modes of resource sharing can be used in combination to meet the requirements of complexity, variety and flexibility in e-government transaction processing.

A. *The data federation mode*

The data federation server provides a real-time and integrated view of data, which can translate the distributed heterogeneous data into a logical unit. Data stored in the system databases and other kinds of structured or unstructured data records from different systems are standardized by the data standardized interface. After the universal description, these heterogeneous data will be registered in the resource catalog, so that they can be matched and accessed in real time. In the federation data view, the registered data have been grouped around the same business entity, or been combined within a business processes. So that it can map different data sources into the common virtual data source. In reality, various data requirement are generated from business steps in every e-government affairs, as well as from the webpage, applications or portlets in the e-government portal. The formats of those data requests are quite different, so the unique query interface will translate those requests into queries in the uniform format, which then can be identified by the federation data view. To response those queries, the federation data view will match the queries within the resource catalog and locate the data source, the required data then can be obtained through the generic data access interface. Data requirements in the same business steps will be processed in parallel and the data are responded in real-time. So the data federation mode is applicable to those flexible and real-time e-government affairs.

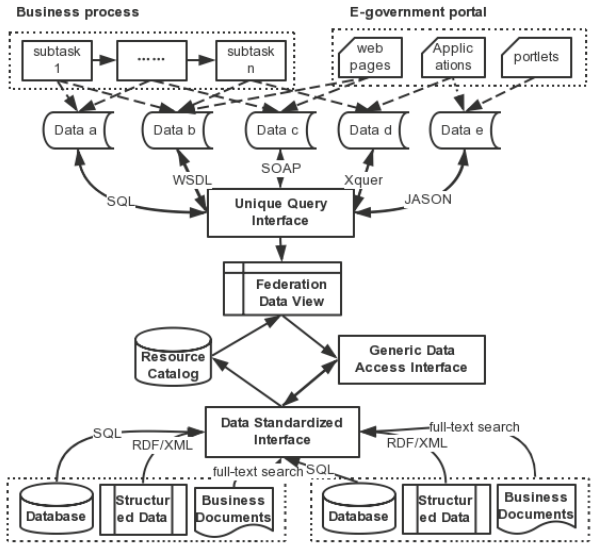


Figure 2. The resource sharing model based on data federation

B. *The data integration mode*

The ETL (Extract, Transform and Load) is utilized to physically integrate distributed data in the form of Data Warehouse. It is distinct from the data federation mode in that data are preprocessed and then restored in a centralized location. The ETL process is the key point of the whole model for there is a huge of ETL mapping rules to be designed and implemented. The mapping rules should be decided by the business logic and the user requirements, which may make up more than a half of workload in the data integration of e-government. Notice that, the data in Data Warehouse are not simply extracted from the original data source, they are processed subject oriented. For example, some important subjects in the e-government include the government departments, the districts and the industries. All data related to different government departments in different systems are extracted and linked based on the department numbers. So the data requirement related to a specific government department can be responded by the Data Warehouse intensively.

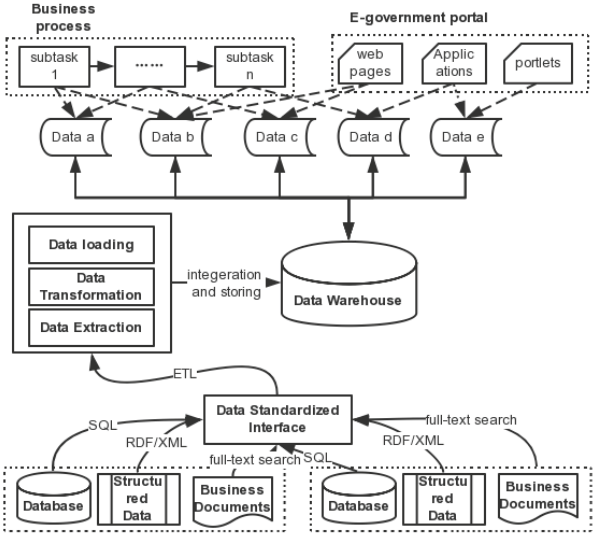


Figure 3. The resource sharing model based on data integration

IV. THE MULTI-TECHNOLOGY FUSION MODEL FOR CROSS-SYSTEM COLLABORATION OF E-GOVERNMENT

Information technology is an essential supportive element in e-government. As there is sharp distinction in the construction model among different areas or departments, the underlying hardware and the software systems are various and the technical compatibility tend to be difficult to manage[8]. So the major difficulty in cross-system collaboration of e-government is the multi-technology fusion, based on which we can take the existing technology into a whole consideration, so the cross-system collaboration system can effectively adapt to the changing business forms of e-government.

The multi-technology fusion of e-government has become a hotspot both in research and in practice. The existing models of it mostly rely on specific technical architecture, for example, the SOA, the Grid and the Cloud Computing. Wang Hongxia proposed an integration model of e-government systems based on the middleware [8]; Ju

Chunhua proposed a collaboration model of e-government technology based on the multi-agent [9]; Lei yinzhi [6] and Xiong shuchu [10] constructed a technology integration model based on the combination of Web Service and work-flow; Lin yinxian depicted the cross-system collaboration model of e-government based on the Cloud Computing architecture [11]. A defect common to all these models is that their universality is not very good. So we hope to conclude some general characteristic of them and focus on a more important question in the multi-technology fusion of e-government, that is, the universal standard for the multi-technology fusion.

The usual way of dealing with a complex system is to decompose it hierarchically into logical layers. In the same way, we can decompose the multi-technology fusion model based on the basis of the function view. The hierarchy of the multi-technology fusion consists of the transport layer, the data layer, the functional layer, the process layer and the presentation layer [4] [12]. The most important work in multi-technology fusion of e-government systems is to establish a uniformly standard. In practice, those commonly accepted technology standards with good scalability should be adopted, so that the heterogeneous e-government systems can be connected and their specific technology advantages can be integrated into the work-flow of e-government [13]. Based on above discussion and analysis, we summed up those technology standards which correlate separately with the 5 different layers of the multi-technology fusion model, as shown in table 1.

TABLE 1 THE MULTI-TECHNOLOGY FUSION LAYERS OF E-GOVERNMENT AND RELATED TECHNOLOGY STANDARDS

Layers	***Function***	***Technology standards***
the presentation layer	To encapsulate the user interface and embed the service functions	PORTLET, JSR, WSRP, OLE, etc.
the process layer	To combine the service functions based on the work-flow of administrative affairs	BPM, WFMS, BPEL, WEB SERVICE, etc.
the functional layer	To support the business logic sharing of e-government systems	EJB, CORBA, DCOM, COM+, JAVA, RMI, etc.
the data layer	To provide the cross-system data access mechanism, including the function of data transformation and data exchange.	DC, Z39.50, OAI, EDI, etc.
the transport layer	To build the network connection and the transport channel for data transfer	HTTP, SOAP, TCP/IP, FTP, RPC, etc.

The transport layer lies in the lowest sub layer in the model. The function of it is to ensure that those heterogeneous e-government systems can exchange their information resource and service functions. Technology standards in this layer stipulate the way of remote access and resource invocation. Some typical standards in the transport layer include the HTTP, TCP/IP, FTP, SOAP (Simple Object Access Protocol), RPC (Remote Procedure Call Protocol), RMI (Remote Method Invocation). The general method in practice is to call API functions of those protocols to realize interconnection between different systems [12].

The data layer focus mainly on fundamental functionalities such as data transformation, data exchange and data integration, so that data in different system can be accessed effectively. The implementation of these functionalities should be achieved from the perspective of data schema and semantic [14]. Some typical standards in the data layer include the DC (Dublin Core Element Set), the Z39.50, the OAI (Open Archives Initiative Protocol for Metadata Harvesting), the EDI (Electronic Data Interchange). The DC is utilized as a standard of cross-domain information resource description; the Z39.50 is utilized to establish universal mapping of user view across different database systems; the OAI is an interoperation protocol utilized to improve the capability of resource sharing as well as to expand the coverage of it; the EDI is utilized to transform data of administrative affairs into a universal format, so as to support the data exchange and the automatic transaction processing between e-government systems.

The functional layer is constructed for the business logical of e-government. The function in it is decided by the business relations and the resource relations between different e-government systems, aiming to solve the problem of user oriented service. Some typical standards in the functional layer include the EJB (Enterprise JavaBean), the CORBA (Common Object Request Broker Architecture), the COM+ (Extended Component Object Model), and the DCOM (Distributed Component Object Model).

The process layer is a medium layer between the functional layer and the presentation layer. It organizes the logical entity in administrative affairs based on the work-flow. The multi-technology fusion in the process layer aims to utilize the underlying protocols and related integration technologies comprehensively so as to achieve the combination of service flow. Some typical technologies in the process layer include the BPM (Business Process Management), the WFMS (Workflow Management System), the BPEL (Business Process Execution Language) and Web Service.

The presentation layer aims to provide a common user interface of service, from which the user can make use of application functions from different e-government system. Therefore, the user interface has to be compatible with multiple applications. Some typical technologies and standards in the presentation layer include the Portlet, the JSR168 (Java Specification Request), the WSRP (Web Service for Remote Portlets), the OLE (Object Linking and Embedding) and so on.

V. THE SERVICE COOPERATION MODEL FOR CROSS-SYSTEM COLLABORATION OF E-GOVERNMENT

An important task currently in practice is to improve the interactive service capability of the e-government systems. Therefore, it is necessary to combine the services in different system and optimize them into new service based on the user requirement and the business logic [15]. In practice, some new service such as “online parallel approval of administrative”, “One-stop service” are supported by the cross-system service cooperation [16]. Some researcher believed that the dynamic cooperation of service is the key point of e-government collaboration at the present stage [15]; because the administrative affairs

are usually consist of different subunits provided by different departments. These sub units can be combined as a "service chain" after the service decomposition and the sub service matching. On the other hand, the service corporation should be flexible and reconfigurable, for the administrative affairs are variety and dynamic. Accordingly, the model should consider the free configuration of the business requirement [17]. A general model of service cooperation in the cross-system collaboration of e-government can be shown as Figure 4.

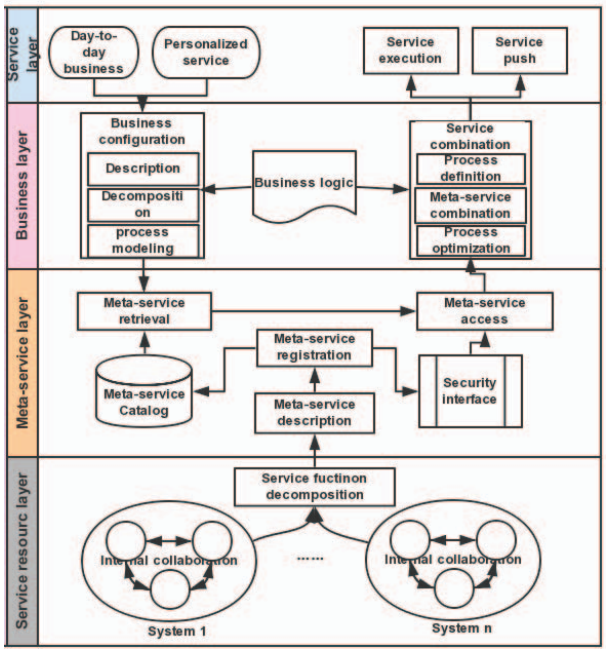


Figure 4. The process-oriented service cooperation model for cross-system collaboration of e-government

The model in Figure 4 consists of four layers including the service resource layer, the meta-service organization layer, the business layer and the service layer. In the service resource layer, the service functions provided by those collaboration systems will be decomposed into independent and fine-grained services, which can be called as "meta-service". In the meta-service organization layer, the meta-service will be described according to a unified standard; the basic information of the description includes its function, constraint condition, input and output parameters. The description will be registered in the meta-service catalog so that the meta-service can be retrieved and accessed.

On the other hand, the demanders submit their configurations throw clients or the e-government portal, and then the system will analyze their requirement of service with reference of the predetermined business logic. In most cases, the requirements will be decomposed into business flow, which contains a sequence of mutually constrained subtasks. Then the subtasks will be matched on the description in the meta-service catalog. The service providers will be chosen after a synthetic evaluation through a combination of factors including the quality, the service capability, the resource, the duty and the response time of them. The security interface will call the meta-service from the decided service provider and then delivered those meta-services to the service assembly module, in which the meta-services will be integrated based on their constraint relationships in business logic. The basic constraint relationships of meta-services contain the sequential, conditional, parallel relationship. The sequential meta-services will be responded in series, while the parallel meta-services will be responded in parallel, so as to enhance the speed of response of collaboration service in e-government [16].

VI. CONCLUSION

From a perspective of system elements, we can propose a more generalized concept model of cross-system collaboration of e-government, and then decompose the concept model into resource sharing model, technology fusion model, service cooperation model and management coordination model. In this paper, we have constructed the resource sharing model, technology fusion model and service cooperation model based on the inductive generalizations of former researches.

Because of limited space, we have not yet touched on the management coordination model, which is also very important in the cross-system collaboration of e-government. One interesting reference is the management coordination model given by W. Moen [18], in which the model is build based on the homogeneity and heterogeneity of organizations. Another model we can reference is the collaboration management model in the practice of e-government in the United Kingdom [19], which consists of 6 layers including the management objectives, the management principles, the policies and regulations, the technical and practical specifications and the development and management of system. In further researches, we will focus on the construction of collaboration model combining the management factor and other factors.

ACKNOWLEDGMENT

This study is supported by the project of National Social Science Foundation of China (11CTQ006).

REFERENCES

[1] Li Manbo, "Collaboration: the futrue of e-government," Software World, Jan. 2005, pp. 44-45.

[2] Tang Zhihao, "Research on the management system and methods of pushing the cross-system collaboration of e-government," E-Government, May. 2014, pp. 80-86.

[3] Hu Changping, Zhang Min and Zhang Liyi, "The Cross-system Collaborative Information Service Organization in the Knowledge Innovation," Library And Information Service, vol. 54, Jun. 2010, pp. 14-17.

[4] Ning Jiajun, "Pay high attention to the collaboration of e-government system," China Information Times, May. 2009, pp. 28-31.

[5] Zhang Jian, "Research on the Mode of Cross-system Collaborative of E-government," Dongyue Tribune, vol. 27, Apr. 2006, pp. 205-206.

[6] Lei Yinzhi, "Research on the Coordinated E-govemment Administration Mode and Its Implementation Mechanism," Information Studies:Theory & Application, vol. 33, Aug. 2010, pp. 52-56.

[7] Gil-Garcia J R, Chun S A and Janssen M. “Government information sharing and integration: Combining the social and the technical,” Information Polity, vol. 14, Jan. 2009, pp. 1-10.

[8] Wang Hongxia, Wu Peng, “The Casestudy of collaboration in E-government: the technology architecture of government service in Singapore,” E-Government, Jan. 2007, pp. 148-156.

[9] Ju Chunhua and Zhang Chaohua, “Research of e-government cooperation work model based on semantics and multi-agent,” Application Research Of Computers, vol. 25, Nov. 2008, pp. 3218-3220.

[10] Xiong Shuchu, Wang Jingtong and Luo Yihui, “Research on Inter-organization of the E-government Collaboration Service Mechanism,” Journal of Hunan University of Commerce, vol. 21, Jan. 2014, pp. 107-112.

[11] Lin Yingxian and Lin Dabing, “Research on Collaborative Information Framework of E-Government Based on Cloud Services,” Journal of Jimei University(Natural Science), vol. 19, Feb. 2014, pp. 152-156.

[12] Zhang Min, “Research on the Cross-system Collaborative Information Service Organization Oriented Knowledge Innovation,” unpublished.

[13] Hu Changping and Qu Chengxiong, “Collaborative Construction of Information Guarantee Platform for National Knowledge Innovation,” Journal of Shanxi University (Philosophy and Social Science Edition), vol. 35, Mar. 2012, pp. 2402-2402.

[14] Xu Gang, Huang Tao, Liu Shaohua and Ye Dan, “Survey on the Core Techniques of Distributed Application Integration,” Chinese Journal Of Computers, vol. 28, Apr. 2005, pp. 433-444.

[15] Xiong Shuchu, Guo Hong and Luo Yihui, “Research and Design of the E-government Collaboration Service Meta-model,” Library And Information Service, vol. 54, Oct. 2010, pp. 119-123.

[16] Yang Xingmei, “Research on the Application of Efficient Collaboration E-government System Across Departemnts,” Information Technology & Informatization, Apr. 2012, pp. 18-20.

[17] Chen Hongjie and Liu Xilin, “Research on Personalized Demand-Oriented E-government Information Structure Model,” Journal Of The China Society For Scientific Andtechnical Information, vol. 26, Mar. 2007, pp. 442-447.

[18] Moen W E. Mapping The Interoperability Landscape For Networked Information Retrieval[C]//Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries. ACM, 2001: 50-51.

[19] He Wei, “The standardization of E-government is a System Engineering,” E-Government, Apr. 2006, pp. 8-11.

Portfolio Pricing Models under Different Collecting Methods in the Green Supply Chain for Home Appliances Industry

Ai Xu
International Business Faculty
Beijing Normal University Zhuhai
Zhuhai, China
e-mail: gdxuai@163.com

Shufeng Gao
College of Computer Science & Technology
Beijing Institute of Technology Zhuhai
Zhuhai, China
e-mail: gsfdl@163.com

***Abstract*—From the perspectives of the operational objectives of the green supply chain management and in consideration of its economic efficiency, social and environmental impact as well as its unique characteristics, this paper examines the pricing issues of the green supply chain for home appliances industry by using game theory. Considering the influences of the effective recycle behavior of the used home appliances on the whole supply chain, the paper proposes three game models for the portfolio pricing for the wholesale, retail and recycle price under three different collection methods respectively. The three collection methods are 1) manufacturer collection (Model M), 2) retailer collection (Model R) and 3) third-party collection (Model 3P). Then the paper analyzes how the wholesale price, the retail price for green home appliances, the recycle price for the used home appliances, and the total channel profits are affected by the choice of the collection methods. The pricing models presented in this paper provide a practical and theoretical guidance for home appliances enterprises in making pricing decisions.**

Keywords-green supply chain; green supply chain management; portfolio pricing decision-making; home appliances industry; game model

I. INTRODUCTION

Nowadays, the urgency and importance of integrating home appliances industry with the green supply chain management has gained more attention all over the world, due to the fact that discarded used or recycled home appliances become hazardous substances and are harmful to the environment if disposing them directly. Most countries are facing a huge pressure of recycling used home appliances, but for China, the problem is even worse. How to deal with these used home appliances and, increasingly, electronic waste is not only an issue of environmental impact but also an issue of improving the healthy development and growth of the green home appliances industry of China. One of ways to influence the behaviors of the consumers is to apply innovative pricing strategies. However, pricing strategies of the green supply chain for home appliances industry are considered to be more complicated due to the intertwined factors among its operational objectives, its economic efficiency, social and environmental impact as well as its unique characteristics. Meanwhile, there are many difficulties about how to make pricing decisions in home appliances industry when considering the influences of the effective recycling behavior of the used home appliances on the whole supply chain.

II. LITERATURE REVIEW

There are a growing number of research papers on green supply chain management that use game theory to model pricing decisions. The present research results include some studies about the pricing problems for the green supply chain management itself and some studies about the pricing problems for the closed-loop supply chain with product remanufacturing focusing on the effective utilization of resources, with the latter as a majority.

Savaskan et al [1] use game theory to present the optimal closed-loop supply chain structures and to study the pricing and recycle strategies based on three reverse channel formats. Ray et al [2] were interested in looking for the optimal pricing and trade-in strategies for durable, recyclable products by focusing on the replacement customers who are only interested in trade-ins. In another study, Gu et al [3]-[5] analyzed the price decision process for reverse supply chain based on game theory. Wang et al [6][7] conducted several studies to systematically examine the pricing strategies of the closed-loop supply chain management and came up with a set of models based on game theory. Ge et al [8], Guo et al [9], Qiu and Huang[10], Sun and Da [11][12], Shi and Chen [13] have studied the pricing and coordination problems of the closed-loop supply chain management base on game theory too. Huang et al [14] and Chen et al [15] have studied the coordination issues about the close-loop supply chain based on third party collection.

Some scholars studied price-making decision and the coordination mechanism in the green supply chain, such as Jiao et al [16], Li [17], Liu and Ma [18], Zhu and Dou [19], etc. Xu and Zhou [20] proposed a portfolio pricing model for the Green Supply Chain of home appliances industry based on retailer collection. Xu and Gao [21] proposed a portfolio pricing model for the Green Supply Chain of home appliances industry based on manufacturer collection. Xu and Zhou [22] proposed a game model and designed a contract for the pricing of recycling used home appliances in accordance with the manufacturer's collecting method.

So far researches about the pricing models of the green supply chain for home appliances industry are still very limit. Therefore, compared with the existing research results, the paper will emphasize the influences of the effective recycle behavior of the used home appliances to the whole supply

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.50

chain and create a new demand function to consider the derivative demand from the recycling quantity. Then the paper will make the price for recycling used home appliances as a variable in the decision-making process and propose game models for the portfolio pricing based on different recycling methods.

III. MODEL NOTATIONS AND ASSUMPTIONS

A. Notations

We use the following notations throughout the paper:

C_m will denote the unit cost of manufacturing a new green home appliance, C_r will denote the unit cost of remanufacturing a returned home appliance into a new one, and C_s will denote the unit cost of selling a new home appliance for the retailer. P is the retail price of the green home appliance and W is the unit wholesale price. P_c will denote the unit recycle price for the used home appliances from the consumer to the collector, and P_r will denote the unit transfer price for the used home appliances from the retailer to the manufacturer. S_b will denote the unit subsidy or penalty that manufacturer obtained from governments. $D(P)$ is the basic demand for the new green home appliance in the market as a function of retail price, and $D'(P_c)$ is the derivative demand for the new green home appliance created by recycling used home appliance as a function of recycle price. $\prod_i^j$ will denote the profits function for channel member i in supply chain model j and $\prod_i^{*j}$ will denote the optimal profit correspondingly. The subscript i will take M, R, 3P and vacancy, which will denote the manufacturer, the retailer, the third-party collector and the centralized manufacturer, respectively. Superscript j will take values M, R, 3P and C, which will denote the Model M, Model R, Model 3P and centrally coordinated model, respectively. P^{*j}, W^{*j}, P_c^{*j} will denote the optimal retail, optimal wholesale price and optimal recycle prices, respectively. P_r^{*R} and P_t^{*3P} will denote the optimal recycle price from the retailer and third-party collector.

B. Assumptions

We consider the following scenario and make the following modeling assumptions.

We assume that the home appliances produced by using used products are the same as a new one by using raw materials in terms of quality and functions, and will be sold at the same wholesale price.

(1) We consider a two-echelon green supply chain and model a bilateral monopoly between a single manufacturer and a single retailer.

(2) While optimizing their objective functions, all supply chain members have access to the same information.

(3) The pricing decisions are considered in a single-period setting.

(4) Producing a new green home appliance by using a used product is less costly than manufacturing a new one, and the cost saving is denoted by Δ, i.e., $\Delta=C_m-C_r$.

(5) r denotes the fraction of the recycled used home appliance that will be put into remanufacturing and the other fraction 1-r will be put into other places, e.g., raw materials regeneration, i.e. $0\le r\le 1$. We assume that the unit residual value of used home appliances is S（$S\le\Delta$）.

From Assumptions (5) and (6), the average unit revenue from recycling can be written as $\Delta'=r\Delta+(1-r)S$.

(6) We assume that the recycling quantity A is only dependent on the recycle price for used home appliances, i.e., $A(P_c)=g+hP_c$, where g and h are parameters and both of them are greater than zero. Parameter g reflects the consumers' awareness of environmental protection and h indicates the level of sensitivity of the consumers to P_c.

(7) We assume that part of recycling quantity will translate into new demand for the green home appliances particularly when taking some means and measures, such as cash incentives from government for older home appliances that are traded in for new green ones. We characterize the conversion rate by τ, i.e., $D'(P_c)=\tau(g+hP_c)$ and $0\le\tau\le 1$. The conversion rate τ can be influenced by appropriate subsidy from the governments to the consumers in practice. $D'(P_c)$ is a derivative demand from the recycle of the used home appliances.

(8) We consider dedicated cost of recycling used home appliances is function of recycling quantity, i.e., $C(P_c)=LA^2(P_c)$ and $L>0$, where L is a parameter of recycling cost.

(9) We assume the basic demand function is $D(P)=\alpha-\beta P$, with α and β being positive parameters.

Here we assume a downward sloping linear demand function.

From Assumptions (7) and (9), the total demand for green home appliances are composed of the basic demand and the derivative demand from the recycle of the used home appliances: $D(P)+D'(P_c)=\alpha-\beta P+\tau(g+hP_c)$.

IV. PORTFOLIO PRICING MODELS UNDER DIFFERENT COLLECTION MODELS

In this section, we will present three portfolio pricing models of the green supply chain for home appliances industry under three different collection methods, i.e., Model M, Model R and Model 3P. As a benchmark case, the Centrally Coordinated Model is analyzed to highlight inefficiencies resulting from decentralization of decision making.

A. Centrally Coordinated Model (Model C)

The centrally coordinated model provides a benchmark scenario to compare the decentralized models with respect to the supply chain profits.

$$\begin{aligned}\prod^C &= [D(P)+D'(P_c)]\cdot(P-C_m-C_s+S_b)+A(P_c)\cdot(\Delta'-P_c)-C \\ &= [\alpha-\beta P+\tau(g+hP_c)](P-C_m-C_s+S_b) \\ &\quad +(g+hP_c)(\Delta'-P_c)-L(g+hP_c)^2\end{aligned} \tag{1}$$

The simultaneous solution of the first-order conditions results and the profits are listed in TABLE Ⅰ. The optimal portfolio pricing strategies here is (P^{*C}, P_c^{*C}).

TABLE I. EQUILIBRIUM RESULTS OF PORTFOLIO PRICING GAME MODELS UNDER MODEL C

	Model C
P^{*j}	$\frac{\alpha+\beta(C_m+C_s-S_b)+\tau A}{2\beta}$
P_c^{*j}	$\frac{2\beta(h\Delta'-g-2Lgh)+\tau h[\alpha-\beta(C_m+C_s-S_b)+\tau g]}{4\beta h(1+Lh)-\tau^2h^2}$
A	$\frac{2\beta(h\Delta'+g)+\tau h[\alpha-\beta(C_m+C_s-S_b)]}{4\beta(1+Lh)-\tau^2h}$
D	$\frac{\alpha-\beta(C_m+C_s-S_b)+\tau A}{2}$
Π^*	$\frac{A(h\Delta'+g)}{2h}+\frac{B^2+\tau AB}{4\beta}$

$B=\alpha-\beta(C_m+C_s-S_b)$

B. Decentralized Pricing Model Based on Manufacturer Collection (Model M)

In this model, the manufacturer is responsible for the promotion and collection of used home appliances. The retailer decides the retail price P and the manufacturer decides the whole sale W for the new green home appliances and the recycle price P_c for the used home appliances.

The profits of the retailer, manufacturer and the total supply chain are given by following equations, respectively.

$$\Pi_R^M=[D(P)+D'(P_c)](P-W-C_s)=[\alpha-\beta P+\tau(g+hP_c)](P-W-C_s) \quad (2)$$

$$\Pi_M^M=[D(P)+D'(P_c)]\cdot(W-C_m+S_b)+A(P_c)\cdot(\Delta'-P_c)-C \quad (3)$$
$$=[\alpha-\beta P+\tau(g+hP_c)](W-C_m+S_b)+(g+hP_c)(\Delta'-P_c)-L(g+hP_c)^2$$

$$\Pi^M=\Pi_R^M+\Pi_M^M=[\alpha-\beta P+\tau(g+hP_c)](P-C_m-C_s+S_b)$$
$$+(g+hP_c)(\Delta'-P_c)-L(g+hP_c)^2 \quad (4)$$

Because the objective function is concave in P, the best responses can be determined from the first-order conditions. And then given P^{*R}, the manufacturer will optimize her profits function. The best responses will be determined and the results are shown in Table Ⅱ. The optimal portfolio pricing strategies is (W^{*M}, P^{*M}, P_c^{*M}).

TABLE II. EQUILIBRIUM RESULTS OF PORTFOLIO PRICING GAME MODELS UNDER MODEL M

	Model M
W^{*j}	$\frac{\alpha+\beta(C_m-C_s-S_b)+\tau A_M}{2\beta}$
P^{*j}	$\frac{3\alpha+\beta(C_m+C_s-S_b)+3\tau A_M}{4\beta}$
P_c^{*j}	$\frac{4\beta(h\Delta'-g-2Lgh)+\tau h[\alpha-\beta(C_m+C_s-S_b)+\tau g]}{8\beta h(1+Lh)-\tau^2h^2}$
A_M	$\frac{4\beta(h\Delta'+g)+\tau h[\alpha-\beta(C_m+C_s-S_b)]}{8\beta(1+Lh)-\tau^2h}$
D	$\frac{\alpha-\beta(C_m+C_s-S_b)+\tau A_M}{4}$
Π_M^*	$\frac{A_M(h\Delta'+g)}{2h}+\frac{B^2+\tau A_MB}{8\beta}$
Π_R^*	$\frac{(B+\tau A_M)^2}{16\beta}$
Π^*	$\frac{A_M(h\Delta'+g)}{2h}+\frac{3B^2+4\tau A_MB+\tau^2A_M^2}{16\beta}$

C. Decentralized pricing model based on retailer collection (Model R)

In this model, the retailer also engages in the promotion and collection of used home appliances in addition to distributing new green home appliances. The manufacturer pays a transfer price P_r per unit returned to her from the retailer. In this model, the retailer decides the retail price P and the recycle price P_c for used home appliances. The manufacturer decides the wholesale W and transfer price P_r.

The profits of the retailer, manufacturer and the total supply chain are given by following equations, respectively.

$$\Pi_R^R=[D(P)+D'(P_c)]\cdot(P-W-C_s)+A(P_c)\cdot(P_r-P_c)-C \quad (5)$$
$$=[\alpha-\beta P+\tau(g+hP_c)](P-W-C_s)+(g+hP_c)(P_r-P_c)-L(g+hP_c)^2$$

$$\Pi_M^R=[D(P)+D'(P_c)](W-C_m+S_b)+A(P_c)(\Delta'-P_r) \quad (6)$$
$$=[\alpha-\beta P+\tau(g+hP_c)](W-C_m+S_b)+(g+hP_c)(\Delta'-P_r)$$

$$\Pi^R=\Pi_R^R+\Pi_M^R=[\alpha-\beta P+\tau(g+hP_c)](P-C_m-C_s+S_b)+(g+hP_c)(\Delta'-P_c) \quad (7)$$
$$-L(g+hP_c)^2$$

Because the objective function is jointly concave in P and P_c, the best responses can be determined from the first-order conditions. And then Given P^{*R} and P_c^{*R}, the manufacturer will optimize the optimal value of her profits function. The results are shown in Table Ⅲ. The optimal portfolio pricing strategies is (W^{*R}, P_r^{*R}, P^{*R}, P_c^{*R}).

TABLE III. EQUILIBRIUM RESULTS OF PORTFOLIO PRICING GAME MODELS UNDER MODEL R

	Model R
W^{*j}	$\frac{\alpha+\beta(C_m-C_s-S_b)}{2\beta}$
P^{*j}	$\frac{3\alpha+\beta(C_m+C_s-S_b)+2\tau A_R}{4\beta}$
P_c^{*j}	$\frac{2\beta(h\Delta'-3g-4Lgh)+\tau h[\alpha-\beta(C_m+C_s-S_b)+2\tau g]}{8\beta h(1+Lh)-2\tau^2h^2}$
P_r^*	$\frac{h\Delta'-g}{2h}$
A_R	$\frac{2\beta(h\Delta'+g)+\tau h[\alpha-\beta(C_m+C_s-S_b)]}{8\beta(1+Lh)-2\tau^2h}$
D	$\frac{\alpha-\beta(C_m+C_s-S_b)+2\tau A_R}{4}$
Π_M^*	$\frac{A_R(h\Delta'+g)}{2h}+\frac{B^2+2\tau A_RB}{8\beta}$
Π_R^*	$\frac{A_R(h\Delta'+g)}{4h}+\frac{B^2+2\tau A_RB}{16\beta}$
Π^*	$\frac{3A_R(h\Delta'+g)}{4h}+\frac{3B^2+6\tau A_RB}{16\beta}$

D. Decentralized pricing model based on third-party collection (Model 3P)

In this model, it is the third-party collector who is responsible for collecting the used home appliances. The manufacturer pays a transfer price P_t per unit returned to her from the third-party collector. In this model, the retailer decides the retail price P and the third-party collector decides the recycle price P_c for used home appliances. The manufacturer decides the wholesale W and transfer price P_t.

The profits of the retailer, manufacturer, third-party collector and the total supply chain are given by following equations, respectively.

$$\Pi_R^{3P} = [D(P) + D'(P_c)] \cdot (P - W - C_s) = [\alpha - \beta P + \tau(g + hP_c)](P - W - C_s) \quad (8)$$

$$\Pi_{3P}^{3P} = A(P_c) \cdot (P_t - P_c) - C = (g + hP_c)(P_t - P_c) - L(g + hP_c)^2 \quad (9)$$

$$\Pi_M^{3P} = [D(P) + D'(P_c)] \cdot (W - C_m + S_b) + A(P_c) \cdot (\Delta' - P_t)$$
$$= [\alpha - \beta P + \tau(g + hP_c)](W - C_m + S_b) + (g + hP_c)(\Delta' - P_t) \quad (10)$$

$$\Pi^{3P} = \Pi_R^{3P} + \Pi_{3P}^{3P} + \Pi_M^{3P} = [\alpha - \beta P + \tau(g + hP_c)](P - C_m - C_s + S_b)$$
$$+ (g + hP_c)(\Delta' - P_c) - L(g + hP_c)^2 \quad (11)$$

Because the objective function is concave in P and P_c, the best responses can be determined from the first-order conditions. And then Given P^{*3P} and P_c^{*3P}, the manufacturer will optimize the optimal value of her profits function.The results are shown in Table Ⅳ. The optimal portfolio pricing strategies is (W^{*3P}, P_t^{*3P}, P^{*3P}, P_c^{*3P}).

TABLE IV. EQUILIBRIUM RESULTS OF PORTFOLIO PRICING GAME MODELS UNDER MODEL 3P

	Model 3P
W^{*j}	$\frac{\alpha + \beta(C_m - C_s - S_b) + \tau A_{3P}}{2\beta}$
P^{*j}	$\frac{3\alpha + \beta(C_m + C_s - S_b) + 3\tau A_{3P}}{4\beta}$
P_c^{*j}	$\frac{4\beta(h\Delta' - 3g - 4Lgh) + \tau h[\alpha - \beta(C_m + C_s - S_b) + \tau g]}{16\beta h(1 + Lh) - \tau^2 h^2}$
P_r^*	$\frac{8\beta(1 + Lh)(h\Delta' - g) + 2\tau h(1 + Lh)[\alpha - \beta(C_m + C_s - S_b)] + \tau^2 gh}{16\beta h(1 + Lh) - \tau^2 h^2}$
A_R	$\frac{4\beta(h\Delta' + g) + \tau h[\alpha - \beta(C_m + C_s - S_b)]}{16\beta(1 + Lh) - \tau^2 h}$
D	$\frac{\alpha - \beta(C_m + C_s - S_b) + \tau A_{3P}}{4}$
Π_M^*	$\frac{A_{3P}(h\Delta' + g)}{2h} + \frac{B^2 + \tau A_{3P} B}{8\beta}$
Π_R^*	$\frac{(B + \tau A_{3P})^2}{16\beta}$
Π_{3P}^*	$\frac{A_{3P}(h\Delta' + g)}{4h} + \frac{\tau A_{3P} B + \tau^2 A_{3P}^2}{16\beta}$
Π^*	$\frac{3A_{3P}(h\Delta' + g)}{4h} + \frac{3B^2 + 5\tau A_{3P} B + 2\tau^2 A_{3P}^2}{16\beta}$

V. COMPARISON OF THE THREE PRICING MODELS

Based on the results summarized in the above tables, some interesting observations can be made on the performance of decentralized portfolio pricing models.

A. Comparison of the optimal quantity of sale

In this paper, the market demand is determined by retail price, recycling quantity and conversion rate since we have assumed that part of recycling quantity will translate into new demand for the green home appliances. It can be seen from Table 1-4 that the optimal quantities of sale are related as $D(P,P_c)^{*C} > D(P,P_c)^{*R} > D(P,P_c)^{*M} > D(P,P_c)^{*3P}$. This shows that the quantity of sale in the Model R is the largest among the three collection methods and the quantity of sale in Model M is larger than the one in Model 3P. It should be noted that the quantities of sale in the three collection methods are all less than the one in the centrally coordinated model. The reason for it is that the supply chain members are closely coordinated with each other and the decisions are fully coordinated in the centrally coordinated model.

B. Comparison of the optimal recycle price

The optimal recycle prices under different collection models are related as follows: $P_c^{*C} > P_c^{*M} > P_c^{*3P}$ and $P_c^{*C} > P_c^{*R} > P_c^{*3P}$. Consequently, $A > A_M > A_{3p}$ and $A > A_R > A_{3p}$. From these relationships, it can be seen that the recycle price in the Model 3P is the lowest, which will affect the consumer's decision to replace his or her used home appliance. While the recycle price, both in Model M and Model R, are higher and thereby can motive consumers to return their used home appliances and recycle quantities will increase. The relationship between the recycle price in Model M and Model R will depend on the specific parameters and no definite relationship. It can be proved that if the expression (12) tenable, expression (13) will be workable, otherwise expression (14) will be workable.

$$2\beta(h\Delta' + g)[8\beta(1 + Lh) - 3\tau^2 h] - \tau^3 h^2[\alpha - \beta(C_m + C_s)] > 0 \quad (12)$$

Then: $P_c^M > P_c^R \qquad A_M > A_R \quad (13)$

Otherwise: $P_c^M \le P_c^R \qquad A_M \le A_R \quad (14)$

It should be noted that the optimal recycle prices in the three collection methods are all lower than the one in the centrally coordinated model. This means the supply chain members can gain the highest recycle price in the centrally coordinated model and thereby gain best recycle quantities. As a result, there will be more fraction of recycling quantity of the used home appliances to the new market demand.

From the perspective of consumer and social welfare, consumers care more about the recycle price of the used home appliances and will be more willing to return their used home appliances to the collector at a higher recycle price than a lower one. The more the recycle quantity of used home appliances is, the more the society will benefit from the recycle of used home appliances. Therefore higher recycle price of the used home appliances will be beneficial for consumers and the whole society.

C. Comparison of the profits

The pricing will affect the profits definitely. Enterprises hope to gain best profits from the portfolio pricing decision. From the comparison, we can see that the total profits of different collection methods are related as: $\Pi^{*C} > \Pi^{*R} > \Pi^{*3P}$ and $\Pi^{*C} > \Pi^{*M} > \Pi^{*3P}$. This shows that the total profits of the whole supply chain, both in Model R and Model M, are larger than the one in Model 3P. While the relationship between the total profits in Model M and Model R will be dependent on the specific parameters and no definite relationship. It can be proved that the following relationship is workable.

$$\text{if } 2A_M > 3A_R \quad \text{then } \Pi^{*M} > \Pi^{*R} \quad (15)$$
$$\text{if } A_R > \frac{13}{18} A_M \quad \text{then } \Pi^{*R} > \Pi^{*M}$$

It should be pointed out that the total profits in the three collection methods are all less than the one in the centrally coordinated model, which shows that there are inefficiencies resulting from decentralization of decision making due to double marginalization in the channel. Therefore, the profits of the total supply chain can be further improved by some

approaches, such as designing appropriate contract to improve the pricing game effects.

It can be also found that the manufacturer's profits in Model 3P are the lowest. In some certain conditions, e.g., $A_R > A_M$, the manufacturer's profits in Model R are the largest.

VI. CONCLUSIONS

In consideration of the impact of the recycle quantity of the used home appliances on the demand for the green home appliances, a new demand function is created. Portfolio pricing models of the green supply chain for home appliances industry are presented under three collection methods, i.e., manufacturer collection (Model M), retailer collection (Model R) and third-party collection (Model 3P), which are mainly about the decision-making of the portfolio pricing for the wholesale, retail price of the green home appliances and recycle price for used home appliances. As a benchmark case, the Centrally Coordinated Model is also analyzed to highlight the inefficiencies resulting from decentralization. The analysis shows that there exist double marginalization and different collection methods will affect the pricing decisions, the profits of supply chain members and the total profits of the whole supply chain. The portfolio pricing models should be further improved by designing some supply chain contracts and make the total profits of the supply chain reach the level of centralized system. This could become the tasks of the future research.

Connecting the recycling of the used home appliances with the sale of the green home appliances and making portfolio pricing strategies from the perspective of the whole supply chain, will be helpful to recycle the used home appliances effectively. At the same time, this can induce consumers to choose green home appliances when they return their used home appliances back, and reduce the bad influences of the home appliances to the environments during the usage to improve the environmental benefits.

This paper proposes alternative models to solve the pricing problems of the complicated supply chain operations, especially the green supply chain management. The pricing models presented in this paper for the green supply chain of home appliances industry provides a practical and theoretical guidance for home appliances enterprises in making pricing decisions. It is also of significance in improving the effectiveness and efficiency of the whole supply chain.

REFERENCES

[1] R. C. Savaskan, S. Bhattacharya, and L. N. VAN Wassenhove, "Closed-loop supply chain models with product remanufacturing," Management Science, vol. 50, Feb. 2004, pp. 239-252.

[2] S. RAY, T. BOYACI and N. ARAS, "Optimal prices and trade-in rebates for durable, remanufacturable products," Manufacturing Service Operations Management, vol. 7, Mar. 2005, pp. 208-228.

[3] Q. L. Gu, T. G. Gao and L. S. Shi, "Price decision analysis for reverse supply chain based on game theory," Systems Engineering-theory & Practice, Mar.2005, pp. 20-25.

[4] Q. L. Gu, J. H. Ji and T. G. Gao, "Research on price decision for reverse supply chain based on fixed lowest quantitative demand," Computer Integrated Manufacturing Systems, vol. 11, Dec. 2005, pp. 1751-1757.

[5] Q. L. Gu and J. H. Ji, "Price decision for reverse supply chain based on fuzzy recycling price," Information and Control, vol. 35, Apr. 2006, pp. 417-422.

[6] Y. Y. Wang, B. Y. Li and F. F. Le, "The Research on two price decision models of the closed-loop supply chain," Forecasting, vol. 25, Jun. 2006, pp. 70-73.

[7] Y. Y. Wang, B. Y. Li and L. Shen, "The price decision model for the system of supply chain and reverse supply chain," Chinese Journal of Management Science, vol. 14, Apr. 2006, pp. 40-45.

[8] J. Y. Ge, P. Q. Huang and Z. P. Wang, "Closed-loop supply chain coordination research based on game theory," Journal of Systems & Management, vol. 16, May. 2007, pp. 549-552.

[9] Y. J. Guo, S. J. Li and L. Q. Zhao, "One coordination research for closed-loop supply chain based on the third part," Industrial Engineering and Managemen, May. 2007, pp. 18-22.

[10] R. Z. Qiu and X. Y. Huang, "Coordination model for closed-loop supply chain with product recycling," Journal of Northeastern University (Natural Science), vol. 28, Jun. 2007, pp. 883-886.

[11] H. S and Q. L. Da, "Pricing and coordination for the reverse supply chain with random collection quantity and capacity constraints," Journal of Systems Engineering, vol.23, Jun. 2008, pp. 720-726.

[12] H. S and Q. L. Da, "Pricing and coordination of remanufacturing closed-loop supply chain based on product differentiation," Chinese Journal of Management, May.2010, pp. 733-738.

[13] C. D. Shi and J. H. Chen, "Study on coordination in the closed loop supply chain with production remanufacturing," Soft Science, vol.23, Jun. 2009, pp. 60-62.

[14] Z. Q. Huang, R. H. Yi and Q. L. Da, "Study on the efficinecy of the closed-loop supply chain with remanufacturer based on third-party collecting," Chinese Journal of Management Science, vol. 16, Mar. 2008, pp. 73-77.

[15] J. H. Chen, C. D. Shi and F. L. Guo, "Contract design on closed-loop supply chain with product remanufacturing based on third-party collecting," Industrial Engineering and Management, vol. 15, Feb. 2010, pp. 17-21.

[16] X. P. Jiao, J. P. Xu and J. S. Hu, "Research of price-making decision and coordination mechanism in a green supply chain," Journal of Qingdao University (E & T), vol. 21, Jan. 2006, pp. 86-91.

[17] W.N.Li, "Research on the coordinate mechanisms of node enterprises in green supply chain," Guilin University of Electronic Technology, 2008.

[18] Q. Liu and J. R. Ma, "Research on the price coordination of supply chain under information sharing," Logistics Technology, vol. 27, Dec. 2008, pp. 97-101.

[19] Q. H. Zhu and Y. J. Dou, "A game model for green supply chain management based on government subsidies," Journal of Management Sciences in China, vol. 14, Jun. 2011, pp. 86-95.

[20] A. Xu and Z. Q. Zhou, "A portfolio pricing model and contract design of the green supply chain for home appliances industry based on retailer collecting," The Twelfth Wuhan International Conference on E-Business. New York: Alfred University Press, May, 2013, pp. 822-832.

[21] A. Xu and S. F. Gao, "A Portfolio Pricing Model and Contract Design of the Green Supply Chain for Home Appliances Industry Based on Manufacturer Collecting," 12th International Symposium on Distributed Computing and Applications to Business, Engineering & Science. Guilin: IEEE Computer Society Conference Publishing Services (CPS), Nov, 2013, pp. 482-485.

[22] A. Xu and Z. Q. Zhou, "A Game Model and Contract Design for the Pricing of Recycling Used Home Appliances in Accordance with the Manufacturer's Collecting Method," The Thirteenth Wuhan International Conference on E-Business. New York: Alfred University Press, May, 2013, pp.735-743.

Collaborative Filtering Recommendation Model Based on User's Credibility Clustering

Zhao Xu *(Lecturer)*
Tianjin Sino-German Vocational Technical College
Tianjin, China
aslanzala@126.com

Qiao Fuqiang *(Associate Professor)*
Tianjin Sino-German Vocational Technical College
Tianjin, China
qiaofq@126.com

Abstract—**Aiming at the long response time, inaccurate recommendation and cold-start problems that faced by present recommendation algorithm, this paper, taking movie recommendation system as an example ,proposes a collaborative filtering recommendation model based on user's credibility clustering. This model divides recommendation process into offline and online phases. Offline, it uses the result of user's credibility for clustering and then writes the clustered information into a table in database.Online,finds the cluster that target user belongs to and then gives recommendation. As a whole, the model reduces the response time, improves the accuracy of the recommendation rate, and solves the new user's cold-start problem.**

Keywords- Collaborative Filtering; User's Credibility; Dynamic Clustering

I. INTRODUCTION

With the fast development of information processing and storage technology, the digital resources that user can visit becomes more and more abundant. More and users are confused in finding the most satisfied ones in the vast resources.

Collaborative filtering recommendation algorithm is to recommend according to the relativity between target user and other users. When the system finds one or a group users has/have the same consumption preferences with the target user, it will predict the target user' consumption behavior based on these user's consumption behavior. This paper, tasking movie recommendation system as an example, predicts and provides the suitable information for target users based on the movie rating record and audience's needs.

Collaborative filtering recommendation technology is the most successful personalization recommendation technology which is applied in many fields. Its outstanding advantage is that the decision is made on the basis of "users", not "the analysis of content". It can filter any form of content and process very complex and difficult concepts to give a surprising conclusion[1]. While the time is so long that the user's satisfaction degree decreases greatly. And there is no historical record for new user's initial login to create any recommendation.

This paper, taking move recommendation system as an example, proposes a collaborative filtering recommendation model based on user's credibility clustering. This model divides recommendation process into offline and online phase to solve the problems of long response time and inaccurate recommendation. And the new user will be classified into a category with users that have same status. The resources category will be divide to cluster in this scope to provide suitable resources for new user.

II. COLLABORATIVE FILTERING TECHNOLOGY

2.1 Description of traditional collaborative filtering algorithm

The detailed description of traditional collaborative filtering algorithm are as follows:

Input: given user set U={u1,u2,……,um}
Resource set={m1,m2,……,mn}
Rating matrix Rm*n={Rui,Mj},
Rui, Mj are the score given by user ui to resource Mj.

Output: Predict value Pui of resource Mx given by target user Ua

Calculate each user's (ui) similarity (sim ua,ui) of ua and U according to formula (1)(Adjusted cosine similarity).

$$\mathrm{sim}(ua,ui)=\frac{\sum_{j\in M}(Rua,j-\overline{Rj})(Rui,j-\overline{Rj})}{\sqrt{\sum_{j\in M}(Rua,j-\overline{Rj})^2}\sqrt{\sum_{j\in M}(Rui,j-\overline{Rj})^2}} \quad (1)$$

$$Pua,mx=\overline{Rua}+\frac{\sum_{n\in U'}\mathrm{sim}(ua,n)(Rn,j-\overline{Rn})}{\sum_{n\in U'}\mathrm{sim}(ua,n)} \quad (2)$$

In the formula,sim(ua,ui) shows the similarity between user ua and user ui; M is the resource number;Rua,j are the values given by user ua to resource j; $\overline{Rj}$ shows the average values given by all users to resource j; j is the common resource evaluated by user ua and user ui.

Calculate user ua prediction value(Pua,mx)for resource j,the first n nearest neighbors that has the top similarity should be chosen to calculate.U' shows the nearest neighbor user set of user ua according to formula(2).

2.2 Shortage of traditional algorithm

The traditional collaborative filtering algorithm[2] uses user-resource rating matrix to recommend resource for users, The premise for the effectiveness of this algorithm is

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.51

that all the users are trustworthy and all the rating values in the matrix are reliable, which often do not exist For this or that reasons, for example, low user scores, or malicious scores(too mean or great disparity in scores),certain noise data in the rating matrix will reduce the accuracy of prediction greatly[3-5] and impact the function of the recommendation system.

As the resource numbers that users have scored are far less than the total amount of resources, there is data sparsity[6], at the same time, the total number of users is too large which leads to the time of online scanning for target user's nearest neighbor is so long that affects user's effect when the target user is newly registered who does not have any rating records, in turn, couldn't produce recommendation purely by collaborative filtering algorithm ,which presents the cold start problem.

This paper proposes a collaborative filtering recommendation model based on user's credibility clustering, considering the above three problems. By measuring the user's rating credibility to guarantee that the ratings in the prediction are trustworthy; By dividing recommendation process into offline and online and online phases, reduce the online recommendation time greatly; By using clustering algorithm, reduce the target user's neighbor number; By classifying user and resource according to category to solve the cold-start problem of new users.

III. COLLABORATIVE FILTERING RECOMMENDATION MODEL BASED ON USER'S CREDIBILITY CLUSTERING

3.1 Basic Structure of Collaborative Filtering Recommendation Model

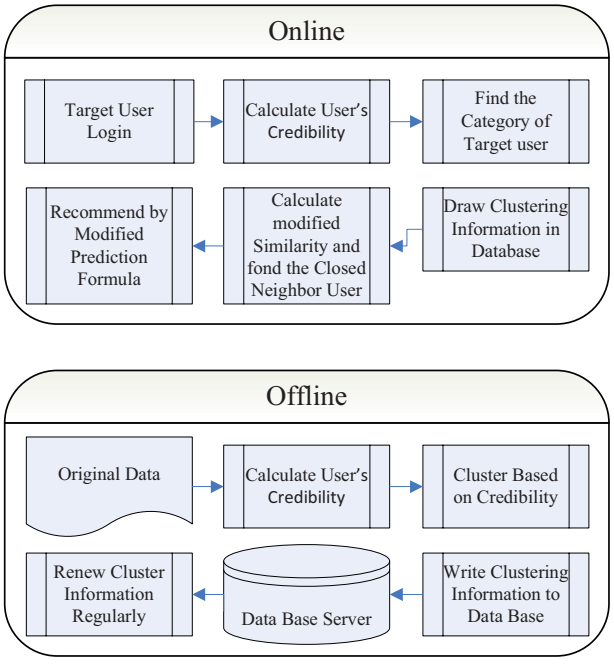


Figure 1 Basic Structure of Movie Recommendation System

This paper proposes a collaborative filtering recommendation model based on user's credibility clustering .This model divides recommendation process into offline and online phases. offline, it calculates users credibility first and takes the few users that has high credibility as the clustering center to cluster other users and records the clustered information. Online, the system finds the cluster that target users belong to, get the clustering information and then gives recommendation. The user clustering numbers are far less than the user number offline, so only the similarity between target user and few clustering centers needs to be calculated online. Prediction value formula of credibility is used that the accuracy of recommendation will be increased greatly, time needed will be reduced .Basic structure is shown in figure1.

3.2 Definition

This paper introduces user's credibility to evaluate user's rating which will obtained by user's counting on the evaluated resource set. Taking movie recommendation system as an example, user's activity, watching rate, rating impartiality are considered mainly.

Definition 1—User's activity: refers to user's activity of resource rating. The more resources user rates, the more contribution they make, the more active. This paper uses user's resource rating numbers as an indicator for evaluating user's activity, which is shown by Act(u), the formula is as follows:

$$\mathrm{Act}(u)=\mathrm{Count}(x) \quad (3)$$

Count(x) is the accumulated number of rating resources.

Definition 2—User's watching rate : refers to the proportion of the movie resources users have watched out of the movie resources users have evaluated. The higher the user's watching rate, the more effective the rating because that means the rating was given rationally by the user after watching movie. User's watching rate is shown by Wat(u), the formula is as follows:

$$\mathrm{Wat}(u)=\frac{\mathrm{Count}(y)}{\mathrm{Count}(x)} \quad (4)$$

Count(x) is the resource number that users(u) have rated, Count(y) is the resource number that users(u) have watched.

Definition 3 — User's Impartiality: refers to the impartiality of user's rating to resources. The impartiality of rating data is reducing because of the malicious rating, the accuracy of target user's prediction will be reduced finally. This paper adopts the method of calculating user's resources mean square deviation to evaluate the impartiality of users. The smaller of the mean square deviation value, the more impartial of users(single and malicious rating will be reduces).User's impartiality is shown as Imp(u), the formula is as follows:

$$\mathrm{Imp}(u)=\sqrt{\frac{\sum_{j\in M}\left(\mathrm{Ruj}-\overline{\mathrm{Rj}}\right)^2}{\mathrm{Count}(x)}} \quad (5)$$

Count(x) is the resource number that user(u) have rated,

RJ shows the average value given by all users to resource.

Corresponding treatment will be given to the three above values for the sake of data accuracy and unity. The dimensionless methods include the following three ways: extremum regularization method, standardization method, equalization method. This paper will use extremum regularization method to make thee indexes being dimensionless after comparison[7], the data is mapping between[0,1], the rules are as follows:

$$X' = (X - X_{min}) / (X_{max} - X_{min}) \quad (6)$$

Xmax is the maximum data in the original data; while X min is the minimum one of the original data.

Three indexes get from regularization according to formula(6): Act'(u), Wat'(u) and Imp'(u).Definition 4 User's credibility is the credibility of single user's rating composed of user's activity, watching rate and impartiality of user's rating. Credibility is shown as Cre(u), the formula is as follows:

$$Cre(u) = a * Act'(u) + b * Wat'(u) + c * Imp'(u) \quad (7)$$

a , b, and c are the weight value of three indexes. The improved Delphi method will be adopted in this paper to give weight value for . Matrix Cm*3 is the rating matrix (a+b+c=1) for three weight values by m experts. Vector R = R1,R2……,Rm} T is the self-evaluation for self-authorization by m experts. Taking movie recommendation system as an example, detailed analysis will be given to specific questions in this paper. Thereinto, Ri=Hi/∑Hi、Hi= d+e+f+g+h (defgh is shown in the table 1).So,

$$[a,b,c] = R^T * C \quad (8)$$

Correspondingly, the formula to modify similarity sim'(ua,ui) and prediction value P'ua, mx are as follows:

$$Sim'(ua,ui) = sim(ua,ui) * Cre(ui) \quad (9)$$

$$P'ua,mx = \overline{Rua} + \frac{\sum_{n \in U'} sim(ua,n)\, Cre(n)\, (Rn,j - \overline{Rn})}{\sum_{n \in U'} sim(ua,n)\, Cre(n)} \quad (10)$$

TABLE 1 EXPERT AUTHORITATIVE SELF-EVALUATION STANDARD

	Index	Rating Standards	Corresponding Value
d	Occupation	Professional Movie Review, Movie Related, Movie Unrelated	10, 8, 6
e	Judgement Basis	Real Experience, Reference, Subjective	10, 8, 6
f	Confidence Degree for Rating	Very Confident, Confident, Common	10, 8, 6
g	Enthusiam for Movie	Quite Enthusiastic, Enthusiastic, Common	10, 8, 6
h	Rating Frequency	Frequent, Often, Few	10,8,6

3.3 Offline User's Credibility Clustering

The calculation for user's credibility online is so much that impact the speed of real-time recommendation seriously and will delay user's waiting time. The satisfaction of user toward the recommendation result will be reduced and result in the lost of client. So this paper proposes the idea of offline user's clustering, and store the clustering information in the data base.

The specific measures are as follows: set up clustering information table(Cluster-table, including five fields: Similar,User1,User2.Cluster and Credit to show similarity value, user 1, user 2, cluster and user's credibility value. This table is used for store the clustering information of user.

The K-mean clustering method[8] combined with calculation of user's credibility are adopted to cluster all users in this paper, the detailed calculation is as follows:

Input: user rating matrix and clustering quantity k

Output: k clustering

(1) Each user's credibility is calculated according to formula(7) to find out k users with the best credibility as the center of clustering, marked as{W1,W2,……, Wk };
(2) Other user and each clustering center's similarity is calculated according to similarity formula(1) to distribute to the cluster with the closest similarity;
(3) Write the clustering information into the data base;
(4) Ranking the user's credibility regularly, re-clustering, renew the data base information, keep the accuracy of the data.

3.4 Online recommendation

Based on the result of offline data, recommend online which will reduce waiting time for users greatly, at the same time increases the accuracy of recommendation through the user's credibility clustering method. The user's satisfaction will be enhanced all-round. The detailed steps are as follows:

(1) Calculate the similarity of target user ua with each cluster center Wk according to similarity formula(1),distribute the target user to the cluster with the closest similarity;
(2) Take all user's information from the cluster ua belongs to in data base;
(3) Calculate the modified similarity sim '(ua,ui) of target user ua with each user in this cluster according to formula(9), choose the the N neighbors that have the closest modifying similarity as the closest neighbor set U';
(4) Calculate the Prediction rating P'ua,mx given by target user us to all projects that haven't been rated according to formula(10), choose the few resources that have the highest rating as recommendation.

3.5 Solve the problem of new user's cold start

When the newly register users log in, recommendation will not be given since no rating record exists, which is the cold start problem. Through observation, it is easy to find out that the favored resources will be close for users that have similar interests. This paper holds that new users can be classified into users that have the same attributes like interests, sex, age, etc. Resources can be classified correspondingly too to form the correspondence of user and resource. Cluster and give recommendation in this range. It is more pertinent to give recommendation to new users in the range of specific user's cluster and resource cluster. It is shown as figure 2 (Favored movie types is the classifying attribute in this paper):

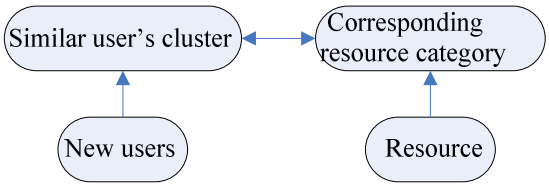


Figure 2 User –Resource model

This paper adopts online method to solve the problem of cold start, the detailed steps are as follows:

(1) Classify users that have the same category according to new users' favorite movie types to form user's cluster; At the same time, divides corresponding resource category Ms according to this category;

(2) Calculate each user's credibility in Us according to formula(7) to find out k users that have the most credibility as the clustering center;

(3) Calculate the similarity of each other user with each clustering center in Us according to similarity formula(1) and distribute to the cluster that has the closest similarity;

(4) Calculate the modified similarity sim' (ua,ui) of clustering center with each user in each clustering center according to formula(9) and choose N neighbor users that have the closest modifying similarity to be the closest neighbor set U';

(5) Calculate the predict rating given by each clustering center for all resources in Ms and choose N recommendations in each clustering center as final resource to be recommended to new users.

IV. CASE ANALYSES

The model will be tested by the data set provided by Movielens station in this paper. Movielens is a research recommendation system developed by Grouplens project group based on web. Movielens is used to receive the use's rating for movie and provides corresponding movie recommendation list. Its data set consists 100000 rating data for 1682 movies given by 943 users, among which each user rates at least 20 movies. This paper divides the data into training set and test set in a proportion of 4 to 1.

This paper adopts mean absolute error MAE, which is easy to understand and calculate in statistical accuracy measurement methods, as the standard of recommendation accuracy. The recommendation quality is measured by calculating the predict and real user's rating deviation. The smaller the MAE, the more accurate the recommendation is. Set the prediction rating set of target user as {p1,p2,…… pn}, corresponding real rating set as{q1,q2,……qn}, the MAE is:

$$MAE = \frac{\sum_{i=1}^{N} | pi - qi |}{N} \quad (11)$$

In the experiment, assume there are 5 rating experts, the vector of rating matrix and expert's authoritative self-evaluation omitted, then conclude:[a, b, c]=[0.4509,0.3123,0.2368]

In the test, comparison was made on different clustering numbers. This paper takes the clustering number is 40 as an example to compare the MAE value of model and traditional algorithm. The testing result is shown is figure 3.

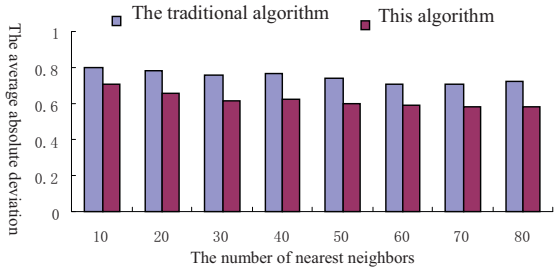


Figure 3 Comparison of Mean Absolute Error

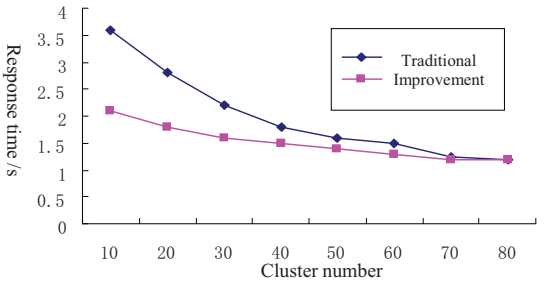


Figure 4 Comparison of Responding Time

The closest neighbor number increases from 10 to 80,the interval is 10.From figure 3, when the clustering number is 40,no matter how man the closest neighbor are, the model proposed by this paper has the least MAE value. Then, we can conclude that the model proposed by this paper can improve the recommendation accuracy greatly.

To test the algorithm real-time, this paper compares offline algorithm with traditional online algorithm, the responding time is shown in figure 4.

The lateral axis shows the number of clustering , the vertical axis shows the responding time. When credibility clustering is offline and stores related clustering information, the original data will be drawn for recommendation. Since the clustering number is far less than the user's number, the recommendation time needed is clearly less than the recommendation time by traditional online method, which is shown in figure 4. There is almost one time difference in effectiveness of this two methods especially when the clustering number. The neighbors that need to calculate similarity online will be reduced greatly, the time difference to calculate online or offline is not large , but the effectiveness of the model proposed by this paper is superior than that of the former one before improvement.

V. CONCLUSION

With the widen of e-commerce application fields, recommendation technology is applied broadly which will draw people's more and more attention to the problems of cold start, recommendation accuracy and responding time that exist in collaborative filtering algorithm. This paper proposes a collaborative filtering recommendation model based on user's credibility clustering. This model divides

recommendation process into offline phases. The result of experiment proves that this model can solve the above mentioned problems to recommend suitable resources for users.

REFERENCES

[1] Chen Mengjian. Research on Collaborative Filtering Recommendation Algorithm in E-commerce[J]. E-commerce, 2008(539):137-139.

[2] SARWAR B, KARYPIS G, KONGSTAN J. Item-based collaborative filtering recommendation algorithms[C]//Proc of 10th International Wide Web Conference. New York: Springer, 2001:285-295.

[3] Cao bo, Su Yidan, etc. Top-N Recommendation System Based on Ants Clustering [J].Micro Computer Information. 2009, 3-3:p225-226.

[4] Li Tao, Wang Jiandong, Ye Feiyue. A New Similarity Algorithm in Recommendation System[J].Computer Science,2007,34(8):187-189.

[5] Guo yanhong, Deng Guishi. Research on a Personalized Recommendation Algorithm of Collaborative Filtering[J]. Computer Application Research, 2008,25(1):39-41.

[6] Wu Yan, Shen Jie. Solving Data Sparse Problem Exists in Collaborative Filtering recommendation System[J]. Computer Application Research,2007, 24(6):94-97.

[7] Han Shengjuan. Research on Data Dimensionless Method in SPSS Clustering Analyses[J]. Scientific plaza, 2008 (3): 229-231.

[8] Ning ZhengYuan, Wang Lijin. Common Algorithm And Its Realization for Statistics And Decision Making [M]. Beijing: Tsinghua University Press, 2009:210-213.

The Influencing Factors of Knowledge Sharing Behavior on College Students in Virtual Communities

Hu Changping
School of information management
Wuhan University
China, Wuhan
e-mail: hcpwhu@163.com

Wan LI
School of information management
Wuhan University
China, Wuhan
e-mail:towanli@126.com

Abstract—**The paper builds the influential factor model of virtual community knowledge sharing behavior from the four dimensions.By using structural equation model, the paper makes an empirical research on the key factors of promoting the college students' knowledge sharing behavior in the virtual community and the impact of knowledge sharing behavior of virtual community members on the community loyalty.**

Keywords-knowledge sharing; influential factors; virtual community; user relationship;structural equation model

I. INTRODUCTION

Virtual community is the important platform of information and knowledge exchange, virtual community bring together like-minded people to form a network for knowledge exchange and sharing. Knowledge sharing is Knowledge sharing is the behavior of the organization members will spread the knowledge they have acquired to other members of the organization[1], it's the key way to meet the information demand of virtual community's members. The biggest challenge in virtual community is the supply of knowledge from members. It's important for us to know why virtual community's members choose to share their knowledge when they do a choice and the influence to community loyalty.We makes an empirical research, data collected from 226 members of professional virtual community,and the respondents are college students.

II. THEORETICAL BACKGROUND AND HYPOTHESIS

To explore the knowledge sharing behavior in virtual community,we draw on the social cognitive theory[1-3] to conceptualize an integrated model of this study (see Fig. 1). In social cognitive theory Individual factor, Contextual factors and Members behavior act as interacting determinants.We developed an integrated model of virtual community knowledge sharing behavior from the four dimensions, individual factors (self-efficacy, outcome expectations), contextual factors (reciprocal norms, trust), knowledge factors (knowledge quality, knowledge growth), members' behavior (knowledge sharing behavior, community loyalty).

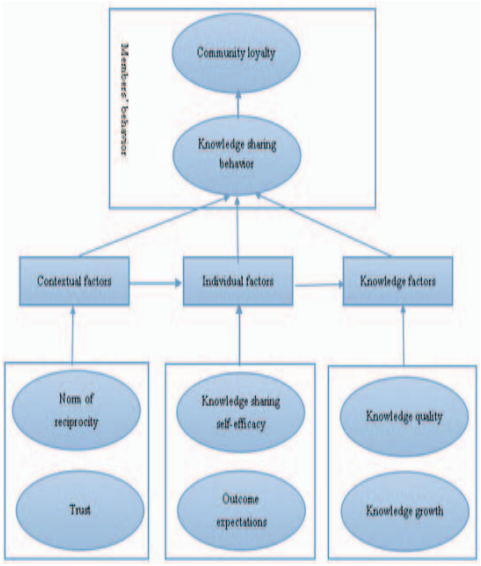


Figure 1. Research model.

A. Contextual factors

Based on social cognitive theory, we may reasonably assume that norm of reciprocity and trust should have influence on individual's behavior. The norm of reciprocity refers to knowledge exchange and sharing by the virtual community members as obligatory and fair. Recently, research shows that the norm of reciprocity do significant influence on members' knowledge sharing behavior[4-5]. Trust treated to be a key element of contextual factor that is crucial to influence on members' knowledge sharing behavior[6-7].

In virtual community,members based on norm of reciprocity to establish trust [8]. And studies have shown that trust can significantly affect self-efficacy [9-10].Thus,the following hypothesis is proposed:

H1:The norm of reciprocity is positive related to the knowledge sharing behavior of members in VC.

H2:The trust is positive related to the knowledge sharing behavior of members in VC.

H3:The norm of reciprocity is positive related to the trust in VC.

H4::The trust is positive related to the knowledge sharing self-self-efficacy of members in VC.

978-1-4799-4169-8 2014

DOI 10.1109/DCABES.2014.52

B. Individual factors

Knowledge sharing define as a process of participants involving the provision and acquisition of knowledge. outcome expectations is the expected consequence of one's own behavior,outcome expectations can be divided into two dimension: personal outcome expectations and community-related outcome expectations. Researchers have found various factors that affect individual's willing to knowledge sharing.The knowledge sharing self-efficacy, outcome expectations are treated as two major Individual factors to influencing individual's knowledge sharing behavior[11-13]. According to [14], outcome expectations impact of knowledge quality in virtual community. Thus,the following hypothesis is proposed:

H5:The knowledge sharing self-efficacy is positive related to the knowledge sharing behavior of members in VC.

H6:The community-related outcome expectations is positive related to the personal outcome expectations of members in VC.

H7:The personal outcome expectations is positive related to the knowledge sharing behavior of members in VC.

H8:The personal outcome expectations is positive related to the knowledge quality in VC.

H9:The community-related outcome expectations is positive related to the knowledge quality in VC.

C. Knowledge factors

This article refer to the knowledge quality is the knowledge quality that internal circulation of virtual community. The higher the virtual community of knowledge sharing knowledge quality, the more a member of his own tendency to acquire knowledge to help solve the problem becomes. According to [15], idea of knowledge growth,knowledge of quality learning in a virtual community, enabling members of their own knowledge growth. Learning objectives are associated with motives of the growth of knowledge[16-17]. Thus,the following hypothesis is proposed:

H10:The knowledge quality is positive related to the knowledge sharing behavior of members in VC.

H11:The knowledge growth is positive related to the knowledge sharing behavior of members in VC.

H12:The knowledge quality is positive related to knowledge growth of members in VC.

D. Members' behavior

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

Virtual community of knowledge sharing is the main source of knowledge of the virtual community, members involved are free to obtain such knowledge. Butler (2001) indicated that extensive knowledge postings/viewings or frequent on-line interactions all have the potential to support[18]. The more members take part in knowledge sharing activities in VC,the more they likely to positively promote VC.Thus,the following hypothesis is proposed:

H13:The knowledge sharing behavior is positive related to community loyalty of members in VC.

III. RESEARCH METHODOLOGY

A. Respondents and procedure

The web-based survey has some advantages than traditional paper-based survey including lower costs,faster responses, higher responses rate.And traditional paper-based survey has shown that respondents provide clearer,more patiently responses. So Web-based and traditional paper-based survey were conducted by this study. By the survey time closed, 226 visitors participate in the survey, of which 173 complete and valid questionnaires were analyzed.The study centered on college students members of several professional virtual community, included pinggu.org, emuch.net/bbs, CSDN, tianya.cn, DXY.CN, zhidao.baidu.com. Table 1 presents the sample demographics.

TABLE I. SAMPLE DEMOGRAPHICS OF THIS STUDY

Sample demographics		***Number***	***Percentage (%)***
Gender	Female	84	48.55%
	Male	89	51.44%
Age	15-20 years old	29	16.76%
	21-25 years old	118	68.20%
	26-30 years old	20	11.56%
	31-40years old	6	3.46%
Education	College and below	19	10.98%
	Bachelor	126	72.83%
	Master	11	6.35%
	Doctor	17	9.82%
Community history	3 months below	39	22.54%
	4-12 months	17	9.82%
	1-2 years	36	20.80%
	2-3 years	35	20.23%
	Over 3 years	46	26.58%

B. Data analysis and results

We should assess the reliability and validity of measures before their use in the model. In order to ensure the validity of the evaluation model fit, reliability test must be carried out first. As shown in Table 2, it can be seen that the relationship expected between measured items and their respective constructs were highly consistent. The composite reliability of constructs range from 0.7020 to 0.8808. The average variance extracted range from 0.5250 to 0.6964. The cronbach α range from 0.733 to 0.895. Hence, all conditions for convergent validity were meet.

TABLE II. RELIABILITY OF MEASURES

Construct and indicators	*Items*	*Composite reliability (CR)*	*Average variance extracted (AVE)*	*Cronbach α*
NR	3	0.9013	0.6964	0.895
TR	3	0.7677	0.5250	0.733
KSSE	3	0.8209	0.6052	0.802
CROE	3	0.8172	0.5987	0.818
POE	3	0.7746	0.5348	0.791
KG	3	0.7779	0.5394	0.779
KQ	2	0.7020	0.5415	0.877
KSB	4	0.8208	0.5347	0.820
CL	4	0.8808	0.6493	0.881

The second step we should test validity of the measure, including the content validity and convergent validity. Content validity of the measurements mainly from subjective judgments, the author of the scale, the reference target to improve binding studies abroad on the basis of the relative maturity of the scale, high content validity. Convergent validity through factor analysis to verify the author, first observed KMO and Bartlett test. Common criteria KMO factor analysis is to be at least 0.6 or more, KMO is 0.871 here in, suitable for factor analysis.

Then, check the rotation factor loading table. The un rotated factor loadings are generally not explicitly represent the true meaning of each factor, hence the need for factor rotation. The author uses orthogonal rotation varimax, get rotated factor loadings table, factor KQ3 load factor of less than 0.5 are deleted. As shown in table 4: The other factor loading volume greater than 0.5. The greater the absolute value of the load factor, the greater role in interpreting the factor matrix, the general and words, it has utility greater than 0.5. Visible, questionnaire set more reasonable, does not require adjustment.

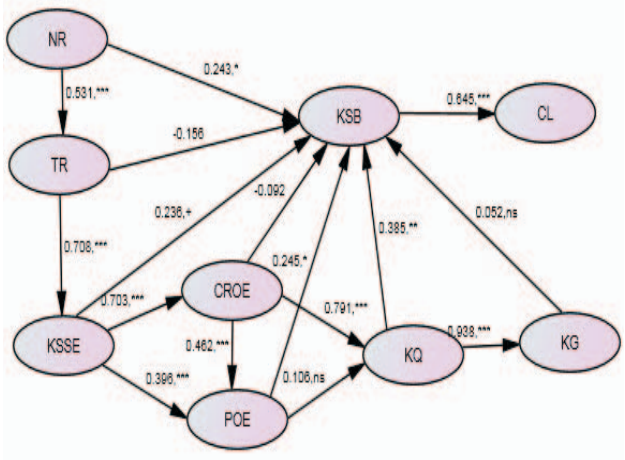
TABLE III. ROTATED FACTOR TABLE

	Factor						
	1	*2*	*3*	*4*	*5*	*6*	*7*
NR1	-.047	.212	**.839**	.164	.050	.103	.173
NR2	-.019	.161	**.848**	.177	.092	.121	.164
NR3	.013	.135	**.859**	.038	.090	.156	.102
TR1	.182	.188	.280	-.102	.185	.215	**.666**
TR2	.176	.156	.240	.002	.371	.092	**.619**
TR3	.019	.123	.101	.238	.162	.070	**.749**
KSSE1	.021	.131	.009	.301	**.737**	.243	.185
KSSE2	.195	.177	.038	.084	**.714**	.216	.171
KSSE3	.159	.116	.169	.054	**.805**	.054	.156
KSB1	.175	.155	.295	**.624**	.363	.019	.142
KSB2	.224	.210	.229	**.719**	.139	.046	.066
KSB3	.244	.238	.146	**.718**	.088	.097	.056
KSB4	.318	.126	-.076	**.712**	.015	.153	.014
KG1	.072	**.769**	.213	.172	.198	-.008	.076
KG2	.223	**.709**	.238	.017	.257	.069	-.065
KG3	.317	**.610**	.035	.183	.051	.083	.294
KQ1	**.642**	.176	-.033	.143	.189	.090	.151
KQ2	**.653**	.150	-.020	-.016	.035	.112	.253
CROE1	.140	**.665**	.134	.200	-.067	.245	.384
CROE2	.166	**.675**	.034	.174	.117	.254	.173
CROE3	.106	**.633**	.176	.206	.063	.404	.010
POE1	.125	.165	-.023	.227	.112	**.764**	.199
POE2	.085	.230	.231	.065	.142	**.764**	.054
POE3	.205	.126	.217	-.023	.220	**.730**	.090
CP1	**.772**	.108	.076	.246	.099	.087	.075
CP2	**.806**	.052	-.004	.283	.053	.062	.041
CP3	**.734**	.094	-.022	.215	.239	.198	-.097
CP4	**.829**	.176	.025	.090	-.024	.001	-.010

IV. MEASUREMENT MODEL

A. Test of the structural equation model

The hypotheses,the paths between the items and the latent construct are examined with the structural model. Most of the model-fit indicates of the structural model exceeded their respective common acceptance levers: χ^2 =435.813,CMIN/DF=1.329,GFI=0.837,IFI=0.952,CFI=0.943, TLI=0.951, RMSEA=0.047. As shown in Fig. 2.,the norm of reciprocity, knowledge sharing self-efficacy, personal outcome expectations, knowledge growth, knowledge quality are strong positively related to knowledge sharing behavior.



(+ p<0.1, * p<0.05, ** p<0.01, *** p<0.001)

Figure 2. Results of structural equation model analysis

B. Empirical Analysis

1) Contextual factors

As for college students, norms of reciprocity (β=0.243, P<0.05) is significant factors that influence their knowledge sharing behavior in virtual community. Surprisingly ,the trust (β=-0.156) did not show a significant influence on college students' knowledge sharing behavior in virtual community. This may be due to their risk perception for the network, rights protection, they do not fully trust this virtual community for knowledge sharing platform. Norms of reciprocity can actually enhance interpersonal trust (β=0.531, P<0.001), trust is positively and significantly related to knowledge sharing self-efficacy (β=0.708, P<0.001), thereby indirectly affecting college students' knowledge sharing behavior in virtual community.

2) Individual factors

Knowledge sharing self-efficacy (β=0.236, P<0.1) and personal outcome expectations (β=0.245, P<0.05) plays a vital role the knowledge sharing behavior. The Results of structural equation model analysis show that the community-related outcome expectations (β=-0.092) did not show a significant influence. This may be due to the knowledge of the virtual community management deficiencies, the need to strengthen construct the articles of

association, organizational norms, organizational rewards, virtual community interests and the interests of members so closely linked. As expected, knowledge sharing self-efficacy was positively related to personal outcome expectations (β=0.396, P<0.001) and community-related outcome expectations (β=0.703, P<0.001), the personal outcome expectations (β=0.106, ns) and community-related outcome expectations (β=0.791, P<0.001) were positively related to knowledge quality. The results also revealed that community-related outcome expectations was positively related to personal outcome expectations (β=0.462, P<0.001).

3) Knowledge factors

The results show that the knowledge quality (β=0.385, P<0.01) is significantly affect knowledge sharing behavior. Knowledge growth (β=0.052, ns) has little effect on knowledge-sharing behavior, the results also showed the knowledge quality is strongly positive related to the knowledge growth (β=0.938, P<0.001).

4) Members' behavior

Our results suggested that the knowledge sharing behavior (β=0.645, P<0.001) was significantly positive related to community loyalty. The effective and beneficial knowledge sharing activity may motivate members to invite new members to join the virtual community. This do a favor to the develop of virtual community.

V. DISCUSSION AND CONCLUSION

Previous research has shown that management of knowledge sharing by its members is very important for the development of virtual communities. Our study suggested that contextual factors, individual factors, knowledge factors should be appropriate for selecting the knowledge sharing activities and knowledge sharing behavior is is positive related to community loyalty. The result also suggested that norms of reciprocity, knowledge sharing self-efficacy, knowledge quality, personal outcome expectations play an important role in knowledge sharing activities by college students. Surprisedly, for college students, trust, community-related outcome expectations, knowledge growth are not the main factor affected in their knowledge sharing behavior. Management of virtual communities should take some measures and provide some facilitates such as guidelines, support mechanisms, incentive rules to increase college students' trust and community-related outcome expectations so that they are likely to participate in knowledge sharing activities more often.

VI. LIMITATION AND FUTURE RESEARCH

First, the results should be interpreted as only explaining knowledge sharing of college students, the research model should be tested for anther sample. Second, based on a sample of 173 respondents from virtual communities, it's not sure whether our result and model could be generalized to all type of virtual communities. Further studies, we will examine the knowledge sharing behavior of different types of knowledge sharers.

REFERENCES

[1] Ryu S, Ho S H, Han I. Knowledge sharing behavior of physicians in hospitals[J]. Expert Systems with Applications, 2003,25(1):113-122.

[2] Bandura A. Social foundations of thought and action[M]. Prentice Hall.: Englewood Cliffs, NJ, 1986.

[3] Bandura A. Social cognitive theory of personality[J]. Handbook of personality, 1999, 2: 154-196.

[4] Bandura A. Guide for constructing self-efficacy scales[J]. Self-efficacy beliefs of adolescents, 2006, 5(307-337).

[5] Wasko M M L, Faraj S. Why should I share? Examining social capital and knowledge contribution in electronic networks of practice[J]. MIS quarterly, 2005,29(1):35-57.

[6] Howard Rheingold.The virtual community: Homesteading on the electronic frontier[M]. MIT press,2000.

[7] Ridings C M, Gefen D, Arinze B. Some antecedents and effects of trust in virtual communities[J].The Journal of Strategic Information Systems, 2002,11(3): 271-295.

[8] Lucas L M. The impact of trust and reputation on the transfer of best practices[J]. Journal of Knowledge Management, 2005,9(4): 87-101.

[9] Aselage J, Eisenberger R. Perceived organizational support and psychological contracts: A theoretical integration[J]. Journal of Organizational Behavior, 2003, 24(5): 491-509.

[10] Hsu M H, Ju T L, Yen C H, et al. Knowledge sharing behavior in virtual communities: The relationship between trust, self-efficacy, and outcome expectations[J]. International Journal of Human-Computer Studies, 2007, 65(2): 153-169.

[11] Usoro A, Sharratt M W, Tsui E, et al. Trust as an antecedent to knowledge sharing in virtual communities of practice[J]. Knowledge Management Research & Practice, 2007,5(3):199-212.

[12] Bock G W, Kim Y G. Breaking the myths of rewards: An exploratory study of attitudes about knowledge sharing [J]. Information Resources Management Journal (IRMJ), 2002,15(2): 14-21.

[13] Kankanhalli A, Tan B C Y, Wei K K. Contributing knowledge to electronic knowledge repositories: an empirical investigation[J]. Mis Quarterly,2005,29(1): 113-143.

[14] Marakas G M, Johnson R D, Clay P F. The Evolving Nature of the Computer Self-Efficacy Construct: An Empirical Investigation of Measurement Construction, Validity, Reliability and Stability Over Time[J]. Journal of the Association for Information Systems, 2007, 8(1):15-46.

[15] Chiu C M, Hsu M H, Wang E T G. Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories[J]. Decision support systems,2006,42(3): 1872-1888.

[16] Cabrera A, Collins W C, Salgado J F. Determinants of individual engagement in knowledge sharing[J]. The International Journal of Human Resource Management, 2006, 17(2): 245-264.

[17] McLure Wasko M, Faraj S. "It is what one does": Why people participate and help others in electronic communities of practice[J]. The Journal of Strategic Information Systems, 2000, 9(2): 155-173.

[18] Gray P H, Durcikova A. The role of knowledge repositories in technical support environments: Speed versus learning in user performance[J]. Journal of Management Information Systems, 2006, 22(3): 159-190.

[19] Butler B S. Membership size, communication activity, and sustainability: A resource-based model of online social structures[J]. Information systems research, 2001,12(4): 346-362.

A User Classification Solution Based on Users' Reviews

Feifei Zhao
Wuhan University of Technology, Wuhan,China
642670122@qq.com

Qizhi Qiu
Wuhan University of Technology, Wuhan,China
736019272@qq.com

Wenyan Zhou
Wuhan University of Technology, Wuhan,China
240350123@qq.com

***Abstract*—**With the continuous development of the Internet technology, nowadays personalized service and recommendation technology have been paid more attentions. The paper aims at accurate user classification for tag application systems and proposes the feasible solution which can mine users' intention in reviews and extend the tag semantics by open knowledge platform. Experiments validate the proposed solution. The research can not only be used as the basis of the users' interests and preference research, but also can be employed in non-Tag application.

Keywords
Tags Extraction,Semantic Extension,Tags Clustering,User Classification

I. INTRODUCTION

With the development of Web2.0 ,it has brought a novelty idea about personalization and free information for the Internet users[1],opened the new Internet era that is User-centered.Therefore,the personalized service and recommendation technology has developed rapidly. As a result, there have been more and more new sharing community websites where tagging technique [2-3]has been widely used to help users annotate resources. The research on user tags can achieve the user classification , explore users' interests and preference so as to enhance the quality of services.

II. Related work

In recent years,the related research for user classification and tagging technique is paid attentions.

Nancy Montanez[4] did a quantitative analysis for blog search engine Technorita and drew the conclusion that tag can help user classification.SuYang[5] proposed a method of building a ring of synonym by the relation of tags ,which can be the basis of user classification.Wang Cuiying[6] proposed a model about the relationship between tags and user preference,including single-interest user and multiple-interest user ,suggested to consider the timeliness of subject in the user preference research.Shi Hao's[7]paper quantitatively solved the problem of user classification by judging whether users use the same tags.Zhang Yantao[8] proposed a model of user's interests that based on the degree of annotation consistency.Zhan Yunzhong[9] applied FCA theory to tag classification, mining user preference and analyzing the transfer of user preference.Tang Lijuan[10] extracted and clustered the tags in various types of blog text by considering four aspects include tag generation,tag clustering,tag behavior analysis and tag semantic studying.

In summary, the most of researches above from domestic and abroad references,related to the user classification and the transfer of user's interest in tag application system,but rarely pay attention to subject timeliness and process of Chinese tags.So this paper employs the reviews with timeliness from *Douban* Website as the data source to develop the study on user classification.

III. User Classification Based on Users' Reviews

The lack participation of users in practical application of tag system results in that systems often collect few meaningful tags. On the other hand, the user reviews are relatively easy to collect and reveal the users' truly preference. If these implicit information can be excavated ,the users can be more accurately classified.In this paper,we use the method above,select user reviews with timeliness as research subject to achieve user classification.Specific process as shown in Figure 1:

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.53

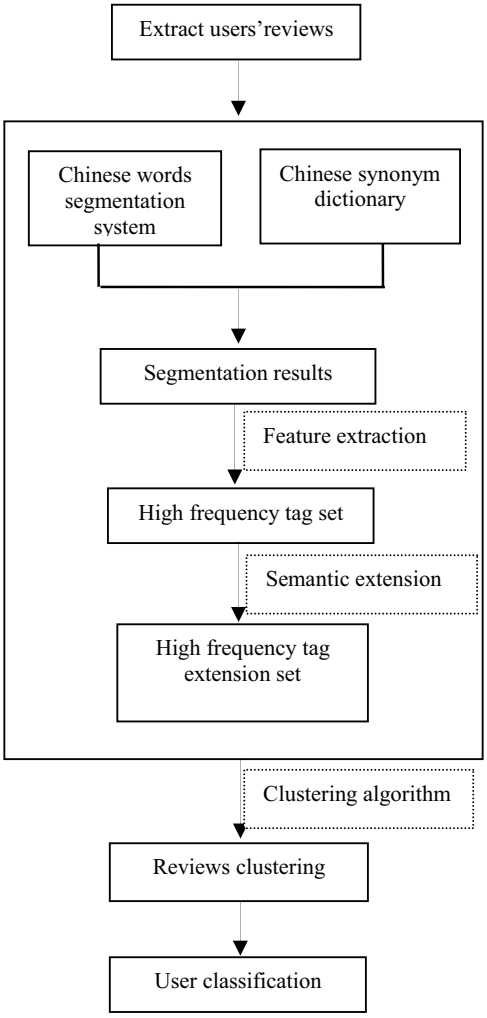


Figure 1. The flow chart of user classification based on reviews

Our solution selects users' long reviews as research subject. Chinese words segmentation techniques is used to process these reviews, we employ content-based method $TF \times IDF$ to extract high frequency tags. The solution extends high frequency tags by using semantic tags extension based on open knowledge platform.The main idea about semantic tags extension is described in detail in section *Tag Semantic Extension Based on Open Knowledge Platform*. Clustering algorithm uses tag-based semantic algorithm, specific algorithm described in section *Tag-based Semantic Clustering Algorithm* .

1.Tag Semantic Extension Based on Open Knowledge Platform

The popular method of words semantic extension is using synonym dictionary.However,the synonym dictionary is not suitable to process these kinds of words, such as names,professional terms,buzz words and other emerging Internet catchwords. After the deep study,this paper selects an open knowledge platform as a way to extend tag semantic.Considering the subject of study are Chinese tags, *Baidu* Encyclopedia is used to achieve tags semantic extension.

Baidu Encyclopedia is an open-content, web free encyclopedia ,which is an outstanding fruit embodying the intention of the *Baidu* Encyclopedia users.For each word there are tags at the bottom of Baidu Encyclopedia web page which are tagged by the users.All the tags generally have relevant information with the word which is very suitable for tag semantic extension.For example,the word "dynamic programming" is a branch of operations research,"dynamic programming" is relevant with the tags such as "algorithm", "computer terminology","computer technology" which are shown at the bottom of the web page. Apparently these semantic extension tags can't be achieved in the normal synonym dictionary.Therefore,this paper using the open knowledge platform to extract extension tags is much more feasible and effective.

2.Tag-based Semantic Clustering Algorithm

After tags semantic extension,this paper puts forward a semantic clustering algorithm based on tags.The algorithm based on classical clustering algorithm K-Means,and avoiding the limitation of difficult to determine the effective K value. First, we compare the similarity between two extended tag set to determine whether two tags are similar . Second,based on the similarity of each tag to determine the similarity among the reviews, we achieve the purpose of clustering.The basic idea of tag-based semantic clustering algorithm as follows:

Step1, Preprocess the data set. The Chinese lexical analysis system and $tfidf$ algorithm are used to extract 10 tags for each review.

Step2, Let the 10 tags of first review act as the cluster center of first category.

Step3, Extract next review r in order, compute the similarity between the review r and each existing cluster centers by the following cosine similarity formula:

$$\cos(a,b) = \frac{a_i \times b_j \times sim(a_i, b_i)}{\sqrt{\sum_i a_i^2 \times \sum_j b_j^2}}$$

$$(1 \le i \le 10,\ 1 \le j \le 10) \qquad (1)$$

where : a is a vector consist of 10 tags belong to review r; b is a vector consist of 10 tags belong to each existing cluster centers; a_i is the ith $tfidf$ value of a ; b_j is the jth $tfidf$ value of b ; $sim(a_i, b_j)$ represent the degree of similarity between the semantic extension tag set a and b ,this paper argues that if the semantic extension tag set of tag a_i have four or more tags consistent with the semantic extension tag set of tag b_j , $sim(a_i, b_j) = 1$; otherwise, $sim(a_i, b_j) = 0$.

Step4,Build an array of similarity values which consist by review r comparing with all existing cluster centers. The array is $M=[m_1, m_2 \ldots, m_k]$ (m_i is the cosine similarity calculated from formula(1)) . If all the value less than 0.01 from m_1 to m_k , set review *r* as the (*k+1*) category,skip to step 6,otherwise,go to step 5.

Step5,Select the maximum value in array $M=[m_1, m_2 \ldots, m_k]$,merge review *r* to the cluster center which is the maximum value m_i belong .The principle is to merge the 10 tags of review r and 10 tags of original cluster center,if they have the same tag, recalculated the $tfidf$ value of this tag as

$$0.8 \times (tfidf_i + tfidf_j)$$

where, $tfidf_i$ is the $tfidf$ value of original cluster center, $tfidf_j$ is the $tfidf$ value of review *r*.

Otherwise, directly add the 10 tags of review *r* to the original cluster center,order the 20 tags by $tfidf$ value and select the the 10 largest $tfidf$ value as the new cluster center. Append the user which the review r belong to the category user list.

Step6,If all the reviews have been similarity determination,output clustering results.

The above steps get the classification results of all the reviews,that means,we can get the user classification results.

IV. VALIDATION

In this paper,data source is obtaining from users of Douban website,we crawl 50 reviews (totally 51966 words) from 5 users by Douban API.After the preprocessing,the 500 tags extracted from 50 reviews are regarded as the experiment data set.

Experiment 1: the tag-based clustering algorithm without tag semantic extension. “semantic extension” step in Figure 1 is not performed,so the parameter $sim(a_i, b_j)$ in formula (1) is always equals 1.The results are shown in Table 1,which shows in the user list are users name,the following number in brackets are review number belong to the user.

Table 1. Tag-based clustering results

Category	Cluster center	User list
1	National,examination,knowledge, country,culture,examine,history, Nihilism,colony,outline...	cold755(1,5) etone(3)
2	Social Sciences,Natural Sciences, influence,truth,pursue,Soros, psychological, Psychoanalysis, aspect,emotion...	cold755(2,7)
3	System,speculate,forecast,market, shock,plan,casino,psychological, idea,Lost Wulin...	cold755(3,4)
4	Lu Xun,Rou Shi,Bannong Liu, Qiejie Pavillion,article,essay, philosophy,concession,harbor,Ism ...	cold755(6) etone(1,4)
5	Theory,confidence,concept, Drucker,nature,staff,policy,leader, management,liberalism...	cold755(8, 10)
6	Botton,religion,Judd,future, salmon,Jones,God,May all your wish come true,doctrine,expect...	cold755(9) Ciyunw(9, 10)
7	Nursery rhyme, Rowling, Harry, Holmes,witness,adults,detective, Humanity,The Casual Vacancy, Hogg Watts...	Ciyunw(1,6)
8	Li Bai,poetry,Dachun Zhang, Great Tang Dynasty,poet,poetry, poetic genius,music,article...	Ciyunw(2)
9	Rose,Chinese rose, rosebush, plant,bush,The Name of the Rose, Sinojackia,Austen,flowers and plants,lily...	Ciyunw(3)
10	science fiction, moonflower, winter,poetry of the Tang Dynasty,poetry of the Song Dynasty,three character primer,the Classic of Mountains and Rivers, novel,mute,literal...	Ciyunw(4) fionapan(10) etone(2)
11	Wilde,Bilbo,The Lord of the Rings,information,Hobbits, Maybell,Frodo,drama,marriage, Rubert...	Ciyunw(5,8)
12	Alev,friend,Melua,love,travel, Corn poppy, lovers, coffee, classmate,virgin...	Ciyunw(7) fionapan(3,8 ,9)
13	Chiang Mai,market,Yamamoto Fumio,museum,marriage,fish, Slight Cold,travel,history,Life...	ycll0009(1,2 ,3,4)
14	Murakami Haruki,lonely,pain, rhythm, puberty,feelings,soul, write,proposition ...	ycll0009(5)
15	Li Juan,Aletai,Corner of Aletai, Winter pasture, bookmark,four seasons,capacity,back cover, winter,designer...	ycll0009(6)
16	Copycat,case,family,couples, Ogitani Miko,Miyabe Miyuki, machine,surrender,Dementia,soul. ..	ycll0009(7, 10)
17	Father,Edward,smile,curiosity, virtue,life,girl,magic,joke,fairy tale...	ycll0009(8)
18	Kindaichi,The Inugamis, inference,chrysanthemum,uncle, cute,model,Japan,pine nut,	ycll0009(9)

	bamboo...	
19	Boss,basement,assistant,writing, Beijing,bookstore,DouBan,tower of ivory,Psychic Consultant, goddess...	Fionapan(1)
20	Ginkgo,prescription,vessel,calf, blood,heart,acne,traditional Chinese physician,Korea,body...	Fionapan(2, 6)
21	Monnai,painter,Jawei Zhang, Paris,garden,Picasso, Impressionism,Raphael,Chardin, Michelangelo...	Fionapan(4)
22	Ang Lee,National,Eileen Chang, Wei Tang,woman,Zhenni Lu, Shuying Shang,Sicheng Liang, Jiazhi Wang,film...	Fionapan(5) etone(7)
23	Bamboo rice,delicacy,bean curd, Taro dumplings, bamboo, Beancurd balls, wushu, food, lard,pork...	Fionapan(7)
24	Algorithm,Theory of Computation, Concrete mathematics,computer science, compute,mathematics,structure, Automata,computation complexity, combinatorial mathematics...	etone(5,6,8, 9,10)

Experiment 1 clustered 50 reviews to 24 categories, the clustering effect is not satisfied ,because each category is too small to meet the reality. For example, the 10 reviews from user ycll0009 have been divided into six categories, not reach an effective classification purposes.

Experiment 2: the tag-based clustering algorithm with tag semantic extension .Using the method of which is mentioned in the section T*ag-based Semantic Clustering Algorithm*.The clustering results are shown in Table 2.

Table 2.Results of Tag-based semantic clustering algorithm

Category	Cluster center	User list
1	Algorithm,Theory of Computation,Concrete mathematics,National, computer science,compute,Social Sciences,Natural Sciences,examin-ation,mathematics...	cold755(1,2, 7,8) etone(3,5,6, 8,9,10)
2	System,speculate,forecast,Copyca-t,family,philosophy,couples, market,shock,plan...	cold755(3,4, 5,6) ycll0009(10)
3	Father,Lu Xun,Murakami Haruki, Li Bai,Rou Shi,Botton,Bamboo ri-ce,Ang Lee,delicacy,religion...	cold755(9, 10) Ciyunw(1,2, 6,9,10) ycll0009(5,7 ,8) fionapan(7) etone(1,4,7)
4	Rose,Chinese rose,rosebush,plant, bush,The Name of the Rose,Sinoj-ackia,Austen,flowers and plants, lily...	Ciyunw (3)
5	science fiction,Bilbo,moonflower, winter,The Lord of the Rings, information,Hobbits,poetry of the Tang Dynasty,poetry of the Song Dynasty,three character primer...	Ciyunw (4,5) fionapan (10) etone(2)
6	Wilde,Melua,corn poppy,boss,Ma-ybell,basement,assistant,writing, Beijing,drama...	Ciyunw(7,8) fionapan(1)
7	Chiang Mai,market,Li Juan,Aleta-i,Alev,Corner of Aletai,Yamamot-o Fumio,friend,National,travel...	ycll0009(1,2 ,3,4,6) fionapan(3,5 ,8,9)
8	Kindaichi,The Inugamis,inferenc-e,chrysanthemum,uncle,cute,mode-l,Japan,pine nut,bamboo...	ycll0009(9)
9	Ginkgo,prescription,vessel,calf, blood,heart,acne,traditional Chine-se physician,Korea,body...	fionapan(2,6)
10	Monnai,painter,Jawei Zhang, Paris,garden,Picasso,Impressionis-m,Raphael,Chardin,Michelangel-o...	fionapan(4)

The results of Tag-based semantic clustering algorithm clustered the 50 reviews into 10 categories, three categories only include one member,the size of others are similar. The result avoids the case of data too discrete in table 1,also take ycll0009 as example,ycll0009 only be divided into three categories. Observed clustering center of each category,which can be inferred about the user's classification and preferences ,we can concluded that the clustering effect is good.In a word the proper classification purposes is achieved.

In short, similarity matrix in Experiment 1 is too sparse,the clustering result is not satisfied,user classification effect not good; The method of Experiment 2 improves clustering effect by semantic extension,achieved better user classification results.

CONCLUSION

This paper selects users' reviews as research subject rather than users' tags in order to acquire the users' intention and timeliness. By making use of lexical analysis system and *tfidf* algorithm ,the paper extracts the tags for each review.The proposed Tag-based semantic extension clustering algorithm is to achieve the accurate user classification. This paper chooses Douban Website as empirical research,bringing in open knowledge platform as a way to tag semantic extension and the experimental results demonstrate the feasibility and effectiveness of the algorithm.

The proposed solution can also be applied to carry out dynamic user classification in non-tag application systems by analyzing the users' behavior with timeliness which can indicate users' interests and preference. The users' behavior

can be analyzed by users' reviews,blog posts,etc.The output of the proposed solution is the basis of research which indicates user's interests and preference transferring.

The above research done in this paper is only a preliminary study,the further work includes :(1)Obtaining high quality corpus and tags. It plays an important role in the accuracy of users' classification.(2) Creating a powerful semantic extension of dictionary is a direction of the future research.(3) Improving and optimizing the tag clustering algorithm.

REFERENCES

[1] Su Shanjia. Study of Semantic Retrieval Systems Model Based on Folksonomy in P2P Networks[D]. Xi'an: Management of the Xi'an Elextronic and Science University, 2010.

[2] Conklin, H.C. Folk classification: a topically arranged bibliography of contemporary and background reference through 1971[D].New Haven, Connecticut: Department of Anthropology,Yale University, 1972.

[3] Mathes A. Folksonomies-cooperative classification and communication through shared metadata[J].Computer Mediated Communication, 2004, 47(10): 1-13.

[4] BrooksCH, MontanezN. Ananalysis of the effectiveness of tagging in blog.[2010-03-10]. http://www.aaai.org/Papers/Symposia/Spring/2006/SS－06－03/SS06-03-002.pdf

[5] Su Yang, Shi Hao, Lai Wen, Zhao Ying. Using Synonym Rings to Improve the User Classification According to Folksonomy Tags.2011,4(5):58-61.

[6] Wang Cuiying. Study on Tags Clustering Analysis[J]. New Technology Of Library And Information Service, 2008, 5: 67-71.

[7] Shi Hao, Li Hongjuan, Lai Wen. Study of Users Classification Based on Folksonomy Tags[J].Library And Information Service, 2011, 5（2）: 117-120.

[8] Zhang Yantao, Wang Guoyin, Yu Hong. A users' interest similarity calculating method in Folksonomy[J]. Journal Of NanJing University(Natural Sciences), 2013, 5: 588-595.

[9] Zhang Yunzhong, Yang Meng, Xu Baoxiang. Research on FCA-based User Profile Mining for Folksonomy[J]. New Technology Of Library And Information Service, 2011, 6: 72-78.

[10] Tang Lijuan. Multilingual Tags Clustering and Its Application[D]. Nanjing: Nanjing University of Science & Technology, 2013.

Information Service Mashup for Industrial Knowledge Innovation Cluster under the Social Network Environment

Weiwei Yan
Center for Studies of Information Resources,
Wuhan University
Wuhan, P.R. China
anteryww@163.com

Peng Cao
Center for Studies of Information Resources,
Wuhan University
Wuhan, P.R. China
caopeng0703@126.com

***Abstract*—Information service mashup is essential to the industrial knowledge innovation cluster, since it can integrate the distributed innovation resources and web services. Under the social network environments, for advancing the collaboration and innovation resources sharing among the innovators in industrial knowledge innovation cluster, the cross-system information service mashup should choose appropriate mashup modes. After illustrating the three typically used modes, which are information mashup, process mashup and website mashup, this paper proposes a mixture mashup mode and explains the architecture when utilized in industrial knowledge innovation cluster. The pipeline framework is also constructed to show how the resources extracted from enterprises, scientific research institutions and universities, intermediary service institutions and governments processed and integrated to provide functional widgets on customized portal for innovators in industrial knowledge innovation cluster.**

Keywords-information service mashup; industrial knowledge innovation cluster; widget; social network

I. INTRODUCTION

As an important part of national innovation system, the innovation of cluster is the key to promote the capability of regional innovation and the level of economic development. Under the Social network environment, since the national development pattern transforms from the closed innovation, which is implemented by independent organization to the open innovation, which is leaded by the collaboration among industries, universities and research institutes [1], the knowledge innovation oriented industrial cluster gradually formed. In general, this industrial knowledge innovation cluster includes regional multi-innovators, which are not only libraries and scientific research institutions, but also enterprises, universities, intermediary service institutions and governments. They join up to share innovation resources and innovation abilities, and coordinated work for related industrial innovation objects. Hence, information service for industrial knowledge innovation cluster should meet the increasingly complex innovation needs, avoid the duplication of innovation resources, reduce the work of access distributed innovation resources, and advance the development of cross-system collaborative innovation. However, as a matter of fact, distributed innovation resources still lack of effective integrating and organizing in industrial cluster.

Information service mashup is a lightweight platform architecture, which supports rapid development, easy deployment and personalized settings. In addition, as a type of integration, information service mashup could access open APIs and data sources to produce results beyond the predictions of the data owners. It could integrate the resources and services of the innovators, and enhance their cross-system collaborations. Hence, the innovators of industrial knowledge innovation cluster should follow the idea of collaborative innovation development, co-construct and share innovation resources, and utilize cross-system information mashup to fully invoke the innovation resources in cluster. Carrying out cross-system information service mashup for collaborative innovation in cluster is not only an urgent requirement for advancing innovation efficiency, but also an inevitable choice for realizing regional innovation as well as the strategy of innovation-oriented country.

This study aims to firstly discuss the different information service mashup modes based on cross-system collaboration under the social network environment. Then, this study points out the appropriate one to integrate the information service resources of innovators in industrial knowledge innovation cluster and other web services. Finally, this study explains the structure of the pipeline framework, and constructs the architecture of cross-system information service mashup in industrial knowledge innovation cluster.

II. INFORMATION SERVICE MASHUP MODES BASED ON CROSS-SYSTEM COLLABORATION

According to the selected mashup elements which are organized for the innovators of industrial knowledge innovation cluster, information service mashup modes can be classified into information mashup, process mashup, website mashup and mixture mashup from the perspective of mashup development based on cross-system collaboration [2].

A. Information mashup

Social network environment advances the big data trend. In this case, the user's demand of information reveals the

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.54

characteristics of dynamic and comprehensive, which means the single data source cannot adapt to the user's complex demand for information acquisition in general [3]. Information mashup focuses on the processing of data and information, and work for retrieval and obtain related data from two or more open data sources, which are in local or on the web. After combining the retrieval data into new data object, it would be posted by feeds or other widgets. The representative development environment of information mashup is Microsoft Popfly. It can be regarded as a visual dataflow language, which supports connect various data sources through wiring the 3D blocks. All the data blocks are reusable, and they also can be configured flexibly according to the mashup requirements. Information mashup can be further divided into simple combination and analysis aggregation according to the data process ways [4].

1) Simple combination: Simple combination is used for reorganizing the data resources according to simple attributes, such as subject, location, time, etc. For instance, iSpecies.org, which is a species information search engine, repectively queries the genome sequence from the GenBank database of the national center for biotechnology information (NCBI), the distribution information from the global biodiversity information agency (GBIF), the image resources from Yahoo image search engine, and the related researches from Google Scholar at the same time. Then the search results would be simply combined as a comprehensive output [5]. In industrial knowledge innovation cluster, simple combination could be used to combine the industrial dynamic information, which means the diverse and dynamic innovation resources of innovators could be combined and showed together. User could use one platform to follow the news of partners and the development of industry.

2) Analysis aggregation: Different from simple combination, analysis aggregation not only obtains various resources, but also extracts needed information from the obtained resources by using specific technology, and creates new objects based on the analysis and processing. For example, HealthMap.org, which is a disease monitoring and early warning platform, utilize the informal web resources to realize real-time monitoring outbreak situation and emerging threat of public health. It helps in providing a wide range of emerging infectious diseases information for the government, the local health department, the library, the international travelers, etc. It mixed the various information sources such as news aggregator, expert discussion, eyewitness reports and official verification report. It also uses Google Map for multiple views and unified global infectious disease status display. Through the systematic and automatic processes of monitoring, organizing, integrating and filtering, the disease status could be analyzed and displayed according to disease onset time and region, disease category and other attributes. Then the results would be translated into nine languages with the automatic translation techniques, which would help the world users timely tracking global public health threat [6]. In industrial knowledge innovation cluster, analysis aggregation could be used to realize the comprehensive cousluting of innovative technology and its application market. The experts' viewpoints would be aggregated with the results of sci-tech novelty retrieval to provide the statistic data and analysis report.

B. Process mashup

The mashup mode, which is oriented by information, would not consume processing resources if it's not used by client browser. However, the mashup mode, which is oriented by process, is instantiated conduct even without the client control [7]. Hence, the process-oriented mashup application should provide control flow, which is independent of client browser. Rather than information mashup, process mashup have much more control in data, behavior and other status information about process, such as time, location and way, etc. It emphasizes on the automatic mashup by coordinating and organizing of services, forms and other resources in work flow. When user needs to perform a task, the automatic manipulation, which include information obtaining, aggregation, filtering and assembly, would be loop execution until the process is ended. Serena Business Mashups is one of the process mashup environments. It advances the creation of collaborative business application, which is used for realizing office automation. The most important part of Serena Business Mashups is the definition of workflow. It includes an editor for workflow definition, and a graphic interface similar with the one that integrated development environment provides. Textual languages with domain specific extensions are provided to realize the reasonable implementation of work flow logic.

In order to keep the agility of mashup service, process mashup needs to accelerate the construction of system, and to shorten the development period at the same time. It requires to choose specific implement functions from the various web services, and to organize them according to workflow. Take the workflow of weather forecasting service Weather Bonk as an example. Firstly, the user's location would be obtained by retrieving the IP address on HostIP.info. Then, the adjacent city's information would be showed by utilizing Google Map. Thirdly, the city's information would be used to query for weather information on the weather forecast web site Weather.com, and for real-time weather images on the visualized weather application, respectively. Finally, all these information would be labeled on Google Map. In this case, user could visualize weather search based on map [8]. Process mashup could be organized for industrial knowledge innovation cluster according to the innovation activity phases. The innovation procedural order of knowledge innovation, technology innovation, innovation diffusion and innovation application, which is the structure of the innovation value chain, would be implemented with the mashups of existing innovation service functions and the third party software functions based on the phased innovation objectives and requirements.

C. Website mashup

Website mashup is the direct way to improve website services by changing the interface of website, adding extra functional unit, or simplify other function elements. It is the most simple mashup mode, since it provides one-stop service by means of choosing view objects and assembling them

together. For instance, Inter MashMaker allows user taking programming instance methods for extracting data from website, cleaning the tables and trees of the data, and pulling them into webpage through additional widgets.

The objects of website mashup are all presentation layer components. They are usually HTML code snippets, which could be reused by pasting them into third-party HTML webpage without compiling. The mashup process only needs to define the display attributes of the browser components, and the input and output attributes of the contents. The former is emphasis on the settings of the interface layout, while the latter can be further divided into data type and operation type. The data source assigned by data type components could be RSS feeds, binary data files, or REST request which can return structure data, etc. Meanwhile, this kind of component also needs to define display modes which are adapting to different output component types, such as data grid and map, statistical graph and timeline component. However, operation type component generally connects interfaces by adopting embedded scripting language. It needs to define web service call interface, operation methods, etc. Representation layer component can also be defined into XML object. Through registering on the third-party mashup application server, the component can be embedded on the website for implementing website mashup.

Netvibes is a typical website mashup application, which references to the portal and portal block ideas. It divides the portal into several portal blocks which customized by user. Each portal block is an individual functional component. Users can choose the needed functional blocks and design the layout according to their own preferences. Moreover, in order to integrate library resources, National science library of Chinese academy of science adopts website mashup to construct iLibrary. It realizes the service access and management based on repository, and displays the comprehensive service by portal. It adapts the requirements of retrieval and utilizing variety of literature resources [9]. For the industrial knowledge innovation cluster users, website mashup can help in providing existing customized components such as market news, real-time supply and demand information from the innovators or the extensive websites, and simply combining them together in one portal to rich user experience.

D. Mixture mashup

Mixture mashup aims to take advantage of information mashup, process mashup and website mashup, and realize the comprehensive service mashup from information to procedure. Then multi-functional widgets are encapsulated on customized platform for individual layout and display by using website mashup. In industrial knowledge innovation cluster, mixture mashup can better meet the industrial cluster innovation and development requirements of diversity, complexity and phase-ordering of the service, which is based on innovators resources and web resources. The architecture of mixture mashup is shown in *Fig. 1*.

From the perspective of data sources, the mashup service for industrial knowledge innovation cluster aims to meet diverse innovation needs of the innovators in cluster. For example, enterprises focus on the competitive intelligence service. They not only need to obtain market status, but also need to promote the ability of technology innovation and the transformation of innovative achievements. Scientific research institutions and universities concentrate on knowledge innovation. Hence, they have higher requirements on the sci-tech novelty retrieval and literature analysis. Meanwhile, they need convenient ways for knowledge sharing and dissemination. For intermediary service institutions aim to support innovation and innovation interaction, they need to gather market information about the specific industry. While governments' responsibility is to supervise and advance the industry development, so they need to know the industry development and innovative technology level by taking market analysis. Therefore, the mashup data sources cover widespread web sources besides the self-construct database, which are from the official websites or portals of the innovators, and the third-party socialized websites.

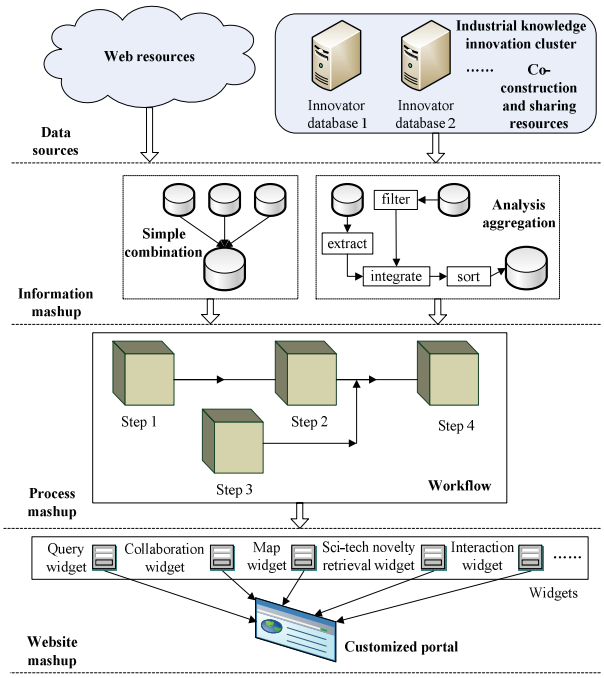


Fig. 1. The architecture of mixture mashup for industrial cluster.

Based on the diverse resources, information mashup could implement with simple combination or analysis aggregation. For example, the mashup of industrial governmental information can divided into national macro policy, regional development policy, incentive policy and other security mechanisms. The information can be obtained from the national, provincial and regional levels of government portals. After labeling the information referred to the target area of the content, they are simple combined according to the labels and provided for supporting comprehensive policy information. However, for the bid information of the industrial knowledge innovation cluster, it not only needs gathering the information

from intermediary service institutions, but also needs fully obtaining the supply and demand information of the innovators in the cluster. These bid information could be further refined after filtering and ordering. Considering the supply and demand information released ways are different, processes such as extracting and filtering are also required to accurately locate related information and aggregate them to prepare for publish. Furthermore, the supply and demand information can be sorted according to urgency degree.

Since the information sources have been preliminary treatment, it would be guaranteed in the phase of process mashup. According to the daily operation procedure and the innovation workflow of innovators, the process mashup could provide a full range of information service support based on workflow. Because the process mashup usually utilize module structure for implementation, the functional work modules could be created and configure related mashup information. For instance, sci-tech novelty retrieval is the key step for confirming innovation trends and innovation objects of cluster. When designing the sci-tech novelty retrieval module, it needs the comprehensive knowledge resources which focus on the innovation direction for support, and provides the quantitative results, which discovered after statistical analysis, for the latter innovation phases.

All the functional modules that created in process mashup would be assembled in an easy to use form, and be provided on the portal for user choosing through the website mashup. Since innovators' innovation tasks are generally different, the procedures of them are also distinguished from each other. Innovators could choose the useful ones according to their actual innovation needs. In this case, the individual portal would be created for innovators in the industrial knowledge innovation cluster.

III. THE PIPELINE FRAMEWORK OF INFORMATION SERVICE MASHUP IN INDUSTRIAL CLUSTER

Based on the mixture mashup mode, the information service mashup for industrial knowledge innovation cluster should be implemented under the pipeline framework, which is typically used for mashup, to keep the mashup working automatically and seamlessly. The pipeline framework is composed by a series of functional components. Each component executes specific operation such as access data, filtering data and combing data. The output of each step would be the next step's input [10].

Pipeline framework usually contains the extracting, filtering and processing component to provide input data. After obtaining the input data, other functional components in the pipeline framework would implement their missions, and the process procedure would execute along the pipeline. The outstanding characteristic is the convenient construction of pipeline framework [11]. The procedure of mixture mahsup is data-driven, and every pipe in it is made a clear definition. Data resources could be processed for creating new mashup applications. Moreover, pipeline framework is conductive to maintaining, which means the functional components could be updated according to the variations of user's needs. Filter is the core component of pipeline framework. It can be further decomposed into sub-pipeline, which includes a data loader, several data filter, and a format converter (as shown in *Fig. 2.*). Among them, data loader is used for loading data, data filter is used to delete irrelevant data or duplication ones, while format converter is used for transforming the data format into standard one.

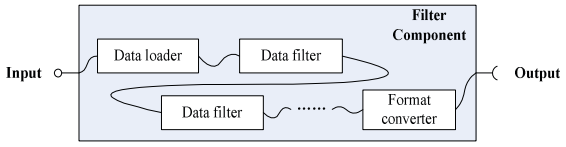


Fig. 2. The pipeline framework of filter componet.

Another component that contains sub-pipeline is generator which is corresponding to the filter. According to the data presentation goals, it transforms data into several expression forms and scales. For example, the sub-component of generator could be data loader which loads the formatted data transferred from filter component, graphic converter which visually process data into several types, space converter which further realizes the point, area and 3D space display (as shown in *Fig. 3.*).

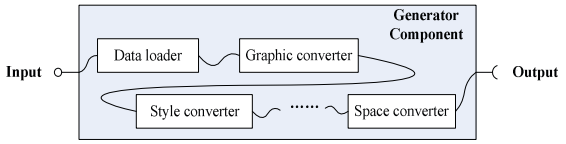


Fig. 3. The pipeline framework of generator componet.

Information service mashup based on pipeline framework is the orderly integrating of functional services. It uses Representational State Transfer (REST) protocol, which is a simpler alternative to Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) based Web services, to access resources and transfer data. It uses HTTP methods explicitly, abstracts all things on network to be resources with, and gives them unique resource identification. The operation on resources through the universal connector interface is stateless [12]. According to the information service mashup goals, the pipeline framework could be divided into several data process pipes to realize the service function reorganizing. Each data process pipes contains variety of adapters which is used for functional process of data, such as content aggregation, data format converting, resources filtering, resources sorting, etc. Each data process pipe could be assembly into individual widgets which are provided to users [13]. Widget is a small and independent application which can run on user client desktops (e.g. Yahoo Widgets) as well as be personalized organized on browsers (e.g. iGoogle). The structure of widget is metadata-driven and open-standard which meet the needs of reuse and extensibility. User could choose and manipulate them according to their preference through the visual interface. Meanwhile, among the different widgets, data could be transferred by connecting them. Hence, cross-system data resources extract from enterprises, scientific research institutions and universities, intermediary service institutions, governments, and other extensive third-party web resources could be transferred, processed, integrated and

shown in different visual forms on client browsers, as shown in *Fig. 4*.

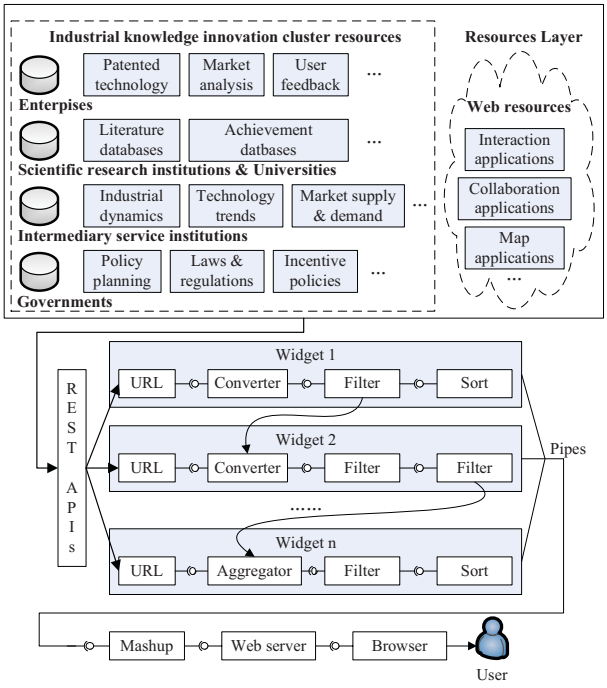


Fig. 4. The framework of cross-system information service mashup in industrial cluster.

IV. CONCLUSION

Information service mashup provides a lightweight and convenient way for integrate innovators' service resources and widespread functional web services. Under the social network environment, mixture mashup, which takes advantages of information mashup, process mashup and website mashup, is the applicable mode for the cross-system information service mashup in industrial knowledge innovation cluster. It not only aggregates cross-system resources, but also encapsulates various functional widgets to meet the variety of user needs. Moreover, widgets can be well embedded in the innovation activities with the workflow reengineering and the pipeline framework designing. Hence, automatic information service mashup could be implemented to promote innovation efficiency, and the functional components also could be updated in time. However, in order to improve the user experience of information mashup, the mashup visualization should be further enhanced. For example, the information mashup for industrial knowledge innovation cluster could put the time and location attributes into the mashup process, and provide the extend services with the visual timeline and map. Meanwhile, the mashup also could be further classified by industrial technology, and providing more specific theme related mashup resources.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (Grant No. 71273197).

References

[1] H. Chesbrougy, W. Vanhaverbeke and J. West, Open Innovation: Researching a new paradigm, Oxford: Oxford University, 2006.

[2] L. Grammel, and M. A. Storey, The Smart Internet: A Survey of Mashup Development Environments, Berlin: Springer Berlin Heidelberg, 2010.

[3] H. M. Rahimi, "Social Network Improvement Utilizing Information Fusion," Switzerland Research Park Journal, vol. 103, January 2014, pp.751-770.

[4] C. W. Li, "Study on Library Mashups," New Technology of Library and Information Service, December 2009, pp. 1-6.

[5] R. D. M. Page, "Biodiversity Informatics: The Challenge of Linking Data and The Role of Shared Identifiers," Briefings in Bioinformatics, vol. 9, May 2008, pp. 345-354.

[6] J. S. Brownstein, and C. C. Freifeld, "HealthMap: The Development of Automated Real-time Internet Surveillance for Epidemic Intelligence," Retrieved on June 2, 2014, from http://www.eurosurveillance.org/ViewArticle.aspx?ArticleId=3322.

[7] P. de Vrieze, L. Xu, and A. Bouguettaya, et al., "Building Enterprise Mashups," Future Generation Computer Systems, vol. 27, May 2011, pp. 637-642.

[8] K. Kim, W. R. Lee, and J. Altmann, "Patterns of Innovation in SaaS Networks: Trend Analysis of Node Centralities," Retrieved on June 3, 2014, from ftp://147.46.237.98/DP-104.pdf.

[9] K. Wang, S. S. Ji, and F. Liu, et al., "iLibrary: Presentation Layer Mashup Service and System Implementation," New Technology of Library and Information Service, November 2010, pp. 30-36.

[10] Y. Liu, X. Liang, and L. Xu, et al., "Composing Enterprise Mashup Components and Services Using Architecture Integration Patterns," Journal of Systems and Software, vol. 84, September 2011, pp. 1436-1446.

[11] J. López, F. Bellas, and A. Pan, et al., "A Component-based Approach for Engineering Enterprise Mashups," 9th International Conference on Web Engineering (ICWE), 2009, pp. 30-44.

[12] A. Rodriguez, "RESTful Web Services: The Basics," Retrieved on June 15, 2014, from http://www.ibm.com/developerworks/webservices/library/ws-restful/.

[13] V. Hoyer, and M. Fischer, "Market Overview of Enterprise Mashup Tools," The International Conference on Service Oriented Computing (ICSOC), 2008, pp. 708-721.

A Novel Anomaly Detection Method for Worms

Xiaojun Tong
computer science and technology
harbin institute of technology
Weihai, China
e-mail: tong_xiaojun@163.com
Zhu Wang
information science and technology
harbin institute of technology
Weihai, China

Miao Zhang
harbin institute of technology
Weihai, China
Yang Liu
harbin institute of technology
Weihai, China
Hui Xu
habin institute of technology
Weihai, China

***Abstract*—This paper proposes a novel anomaly detection method of network worms. The algorithm detects unknown worms by multidimensional worm abnormal detection technology, extracts its feature string via analyzing worm data with leap-style and creates new rules to detect the corresponding worm in case that the unknown worm attacks again. The paper has realized the automatic detection of unknown worms. Experiment data has showed that the method has high success detection rate and low false alarm rate.**

Keywords-Network worms; Automatic detection of worms; Anomaly detection; Feature extraction

I. INTRODUCTION

There are many existing worm detection methods [1-3]. The [4] uses honey pot with low speed to detect worms, but when they found the worms, worms had broken out. The [5] uses the anomaly of host connection request. The [6-7] detect worms by collecting ICMP Type-3 (destination unreachable) message from remote or a threshold which is difficult to choose. The [8] uses multiple routers to monitor the network, but it cannot be used in local area networks and the requirements of detection environment are high. The [9] uses traffic self-similarity whereas its calculations are too big to be applied to real-time network. The [10] compares worm behavior of two operating systems such as the system call sequence similarity, but it can only be applied to a single host and cannot detect worms in the whole network.

This paper proposes a detection method based on both feature detection and abnormal detection. It fuses the two detection methods by a distributed and centralized control detection system which can convert unknown worms to known worms successfully. In this model, the feature detection module detects known worms and the abnormal detection module discovers unknown worms and sends them to the worm features extraction module. The feature extraction module creates new rules for later feature detection.

II. ANALYSIS ON WORM DETECTION TECHNOLOGY

A. analysis on worms' traffic characteristic

Network worm is generally a standalone program which runs without any user intervention. When a host gets infected, it turns to be a source of infection and infects other hosts. The paper calls the progress of target host turning to the source of infection (DS) link communication mode.

As showed in Fig. 1, host A, B, C and D formed a link communication mode, link communication is A->B->C->D.

B. Worm propagation topology

The transport layer communication among network hosts can be represented by a weighted directed graph. Because the worm flow is mixed with normal network flow, this paper uses topology graph to get the substantive characteristics of worm propagation. It can avoid considering each worm infection event isolated and integrate the large scale and each worm infection event which seems like independent together according to the internal relations.

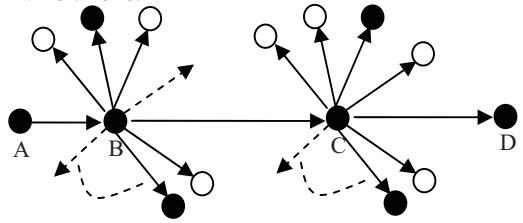


Figure 1. The traffic mode of worm propagation

Fig. 2 shows the worm propagation topology graph.

In the graph (Fig. 2), nodes represent hosts of network and the lines represent communication between hosts and the dotted rectangle represents source of the infection and its infection conduct. Properties of communication are mapped as the line of graph. The directed graph weighed is network communication topology.

III. RESEARCH ON MULTI-DIMENSIONAL WORM ANOMALY DETECTION ALGORITHM AND SKIPPING MULTI-FEATURE EXTRACTION ALGORITHM

A. Multi-dimensional worm anomaly detection algorithm

Feature extraction in this paper is the key to realize the automation of worm detection. In this paper, the extraction algorithm possesses the faculty of extracting substrings and counting substrings.

1) Algorithm description

s: It represents the application layer data of currently being handled packet

p: It represents the start position of substring being processed in s

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.55

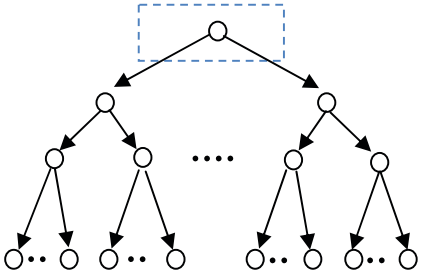


Figure 2. Worm propagation topology

q: It represents the end position of substring being processed in s

s <p, q>: represents a substring from the p to q in s

Y: It is used to store a collection of characters in substring extracted from the class of unknown worms

ML: The lower limit of extracted string length

2) Jump more feature extraction algorithm string

The algorithm extracts substrings from the detected unknown worm data. Each substring has the max depth location information and max frequency information.

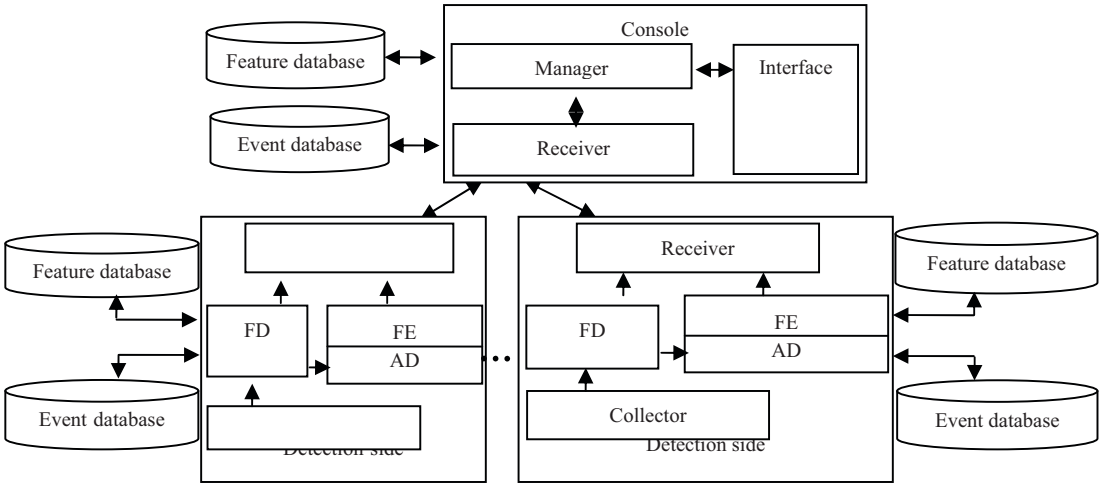


Figure 3. The system structure of unknown worm automatic detection system

Step by step, but jumps forward to improve efficiency. Jump multi-feature string extraction algorithm is as follows:

a) Take a pending data packet s of unknown worm.

b) If the length of s is less than ML, then switch to a, otherwise go to c.

c) p= 1, q = ML-1.

d) q = q +1.

e) If s <p, q>exists in Y already, and q is less than or equal to the length of s, then switch to d, otherwise go to f.

f) If the value of q is greater than s, then switch to k, otherwise switch to g.

g) Join s <p, q> to Y, set the occurrence frequency of s <p, q> as 1, set depth to q.

h) Determine whether the length of the substring s <p, q-1> is less than ML, if so, then switch to j, otherwise switch to i.

i) s<p, q-1> frequency pluses 1, if the depth of s<p, q-1> is less than q-1, update it as q-1.

j) j= q-ML +2, go to d.

k) If the length of the substring s<p, q-1> is less than ML, then switch to n, else go to m.

l) s<p,q-1> frequency pluses 1. If the depth of s<p, q-1> is less than q-1, update it as q-1.

m) If this is the last packet of the unknown worm, then go to r, otherwise go to a.

n) Generate feature detection rules for unknown worm, update feature database, and the algorithm ends.

The features of network worm can be extracted and put into the feature set whether they are in a continuous fragment or distributed in several fragments at different locations. When a new feature string is generated, the algorithm will skip forward. The skipping distance is related to the sum of ML and the length of new feature. The sum is greater, the distance is longer, and the processing speed is faster. The sum is less, the distance is shorter, and the processing speed is slower. Each feature generated by extraction algorithm can give max depth information and can be utilized to improve the efficiency of feature detection.

IV. DESIGN OF THE AUTOMATIC DETECTION SYSTEM OF UNKNOWN WORM

A. Structure of unknown worm automatic detection systesm

The system's structure is concluded as follows. This system is compose of two parts-Server and Client. The server is the center of the entire distributed system. It logs each clients' work state and manages all alarms of the system. Also, it matains and keeps the lasted version of the feature database. The console detects data packets and alarm by feature detection based on content and abnormal detection based on worms flow. Also, it extracts features for abnormal data, update local rules database with the newly created features and send the new rules to every client.

It is displayed as in Fig. 3. In Fig. 3,FD: Feature detection; FE: Feature extraction; AD: Anomaly detection.

B. The realization of the feature detection and rules creation methods

1) Sift feature codes

The process of sifting mainly aimed at character string set created by feature extraction module. Because every worm data packet contains worm feature in the process of propagation, the corresponding character string appears more frequently than other character strings in the process of feature extraction. In that case, sift feature strings based on the ratio of frequency of the substring's appearance and the total flow.

If ((substring. length - the smallest feature string. length)/total flow >= feature sift threshold value)

{

This substring is the worm feature string;

}

Judging every substring of character string set extracted by feature extraction module, one or more feature strings which represent the worm could be got.

2) The creation of the detection rules

Feature database is the most important part of the feature detection method. The qualities of the feature database have a direct impact on the feature detection system's accuracy and effectiveness. Qualities of the created rules represent qualities of the feature database. The paper creates feature detection rules according to following rule.

Action protocol source IP source port ->destination IP destination port(msg:content:; depth:; classtype:; sid:; rev:;)

V. THE ANALYSIS OF EXPERIMENTS

A. Experiment data

In the procedure, real-time detection function is used to analyze and log real time network traffic. The hosts A, B, C and D of LANs are running normal internet applications. Homemade sending packets programs are running on host C and host A. They send UDP packets whose destination ports are 1434 to any other IP addresses. At last, the Slammer worm data are combined with homemade program data and the traffic of normal network applications.

TABLE I. ADJUSTMENT EFFECTS OF SEVERAL THRESHOLDS (√：DETECT ONE WORM，×：DO NOT DETECT WORM)

	N	MLN	WORMCONF	Slammer	Witty	Normal application
10s	30	3	1	√	√	√
10s	30	3	2	√	×	×
20s	30	3	2	√	√	×
30s	30	3	2	√	√	×
30s	30	4	3	√	√	×

To Witty data set, this article writes a part of data to snort log database and updates records in table recording application layer data by programming. In the updating procedure, there is one record without Witty feature in every ten packets. The generation procedure of Witty packets is to generate worm packets with suitable size.

B. Analysis of unknown worm detection results

1) Set the initial value of thresholds related to anomaly detection

△t: △t is a period of time in which the network traffic is handled. It is set to 5s.

N: N represents the out-degree lower limit of bouquet-type node. It is set to 5.

MLN: Lower limit of communication chain length, namely, a communication chain includes at least MLN nodes, it is set to 2.

WORMCONF: It represents the lower limit of worm recognized degree.

2) Analysis of experiment results

This article processes worm traffic by anomaly detection method after setting the initial value of relevant thresholds. The infected hosts' out-degree and ratio of out-degree are larger and communication link comes into being.

Table1 shows that with the growth of △t, there is more and more information in network flow. By adjusting the threshold values according to the above, Slammer and Witty worms are successfully detected. False alarms are avoided which caused by homemade program, port scanning and P2P applications. The result of local detection is shown as Fig. 4. The adjustment of the thresholds has significant influence to the system, as showed in Table 1.

Experiments show that the system proposed in this paper can discover worms in the early time. When $\triangle t$ is between 20s and 30s, N equals 30, and MLN is 3, WORMCONF is 2, it can discover worms effectively with low false alarm rate. If the value is too big, although the system can discover unknown worms, the time delay is extreme and cannot make real-time detection.

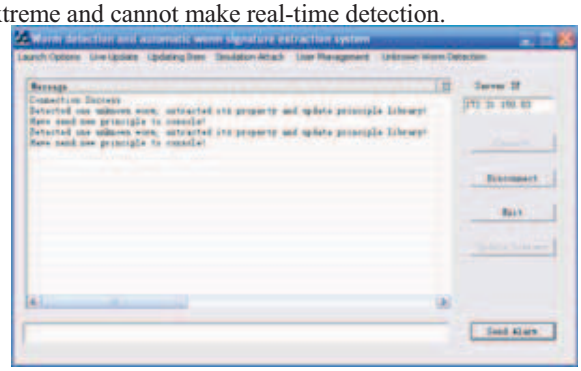


Figure 4. Local side detects two unknown worms

C. *Analysis of results of worm feature extraction and rule generation worms*

a) Setting related threshold

ML: It is the lower limit of extracted string length, which is set to 5

DP: It is the lower limit which is the selected character substrings with the total flow in the proportion when substring is filtered, be set to 0.55.

TABLE III. COMPARISON OF RESOURCE CONSUMPTION

Items	A system[20]	Multi-feature extraction algorithm
Time consumption	50S	40S
Memory consumption	27M	25M

D. *Experiment results*

Extract features of Slammer and Witty after they are discovered. And then sift the character strings in order to create detection rules for feature detection and update local feature database. At the same time, send rules to the console and the console will update other clients' feature database. The clients will create worm rules themselves as showed in Fig. 5. Console receives rules as showed in Fig. 5.

TABLE IV. PERFORMANCE TEST

Types of	Data	Alert	Detectio	Omissio
Witty	48600	41056	84.5%	15.5%
Slammer	49400	48521	98.2%	11.8%
Ramen	50000	42953	85.9%	14.1%
Total	148000	132530	89.5%	10.5%

Compare the rules created in our system with rules of Snort as showed in Table 2. The rules extracted by the system are more refined than automatically created rules. According to realization method of the feature extraction module, the rule generated automatically is longer. They are marked in Table 2 with bold ("Sock" corresponding: "73 6F 63 6B" and "send" corresponding: "73 65 6E 64.")

In this paper, the Witty worms' packets which are different with each other expect worm features of random position are constructed. The system can extract worm features correctly and create feature detection rules. Those rules are the same with Snort's except feature position. The biggest position is 238.

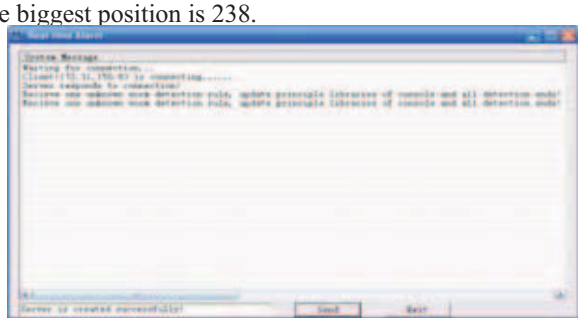


Figure 5. Console receives unknown worm detection rules from detection

Although there are noise data in Slammer and Witty network flow, the system can extract the two worms' feature correctly via adjusting the value of DP. It represents that system has a certain capacity processing noise.

The paper sets the smallest value of character strings between 5 and 10.

In order to compare the efficiency of self-growth Chinese character feature extraction, the paper used 25s's Slammer data. Both of the methods had extracted the same feature strings, but there is no position information of feature strings from system A. The paper gives all the resource consumption as following in Table 2.

As showed in Table 3, the speed of system based on multi-feature extraction algorithm has improved comparing to the system based on self-growth achieving Chinese character combination model. The resource consumption reduces.

a) Use the rules generated above to do feature detection

The local detection side can detect Slammer worms and Witty worms, which are alarmed and are sent the alarms to the console, as showed in Fig. 6 and Fig. 7. In this procedure, there has been no false positive.

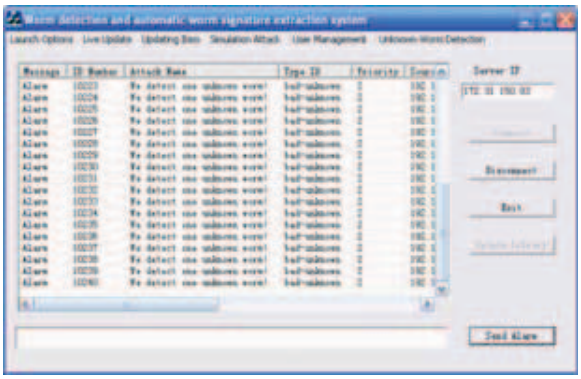


Figure 6. Alarm information of local detection side

The experiment results show that the automation of worm detection is realized. The skipping multi-feature extraction algorithm can extract worm feature set effectively for unknown worm traffic, and the system generates feature detection rules seccessfully after filtering the feature set.

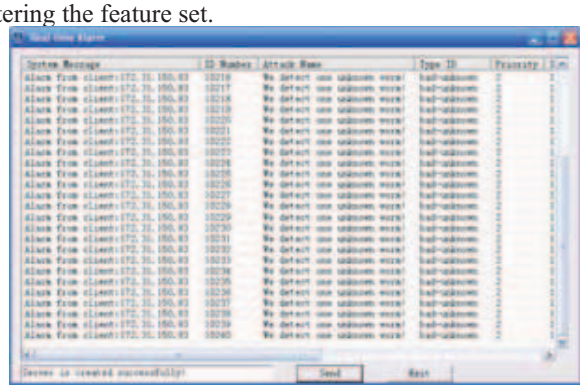


Figure 7. Console receives alarm information of worms from detection side

b) The analysis of experiment results

See Table 4, the performance analysis of experiment results. Experiments show that this system can detect worm event on the large-scale network monitoring, has low false alarm rate, low omission rate and high detection rate.

TABLE II. THE COMPARISON OF SYSTEM-GENERATED RULES AND SNORT RULES

<table>
<tr><th>worm</th><th>rules</th><th>Detection rules</th></tr>
<tr><td rowspan="2">Slammer</td><td>system-generated rules</td><td>alert UDP any any -> any 1434 (msg:"We detect one unknown worm!"; content:"|04 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 DC C9 B0 42 EB 0E 01 01 01 01 01 01 01 70 AE 42 01 70 AE 42 90 90 90 90 90 90 90 90 68 DC C9 B0 42 B8 01 01 01 01 31 C9 B1 18 50 E2 FD 35 01 01 01 05 50 89 E5 51 68 2E 64 6C 6C 68 65 6C 33 32 68 6B 65 72 6E 51 68 6F 75 6E 74 68 69 63 6B 43 68 47 65 74 54 66 B9 6C 6C 51 68 33 32 2E 64 68 77 73 32 5F 66 B9 65 74 51 68 73 6F 63 6B 66 B9 74 6F 51 68 73 65 6E 64 BE 18 10 AE 42 8D 45 D4 50 FF 16 50 8D 45 E0 50 8D 45 F0 50 FF 16 50 BE 10 10 AE 42 8B 1E 8B 03 3D 55 8B EC 51 74 05 BE 1C 10 AE 42 FF 16 FF D0 31 C9 51 51 50 81 F1 03 01 04 9B 81 F1 01 01 01 01 51 8D 45 CC 50 8B 45 C0 50 FF 16 6A 11 6A 02 6A 02 FF D0 50 8D 45 C4 50 8B 45 C0 50 FF 16 89 C6 09 DB 81 F3 3C 61 D9 FF 8B 45 B4 8D 0C 40 8D 14 88 C1 E2 04 01 C2 C1 E2 08 29 C2 8D 04 90 01 D8 89 45 B4 6A 10 8D 45 B0 50 31 C9 51 66 81 F1 78 01 51 8D 45 03 50 8B 45 AC 50 FF D6 EB|"; depth:376; class type: bad-unknown; sid:1000001; rev:5;)</td></tr>
<tr><td>Snort</td><td>alert UDP $HOME_NET any -> $EXTERNAL_NET 1434 (msg:"MS-SQL Worm propagation attempt OUTBOUND"; content:"|04|"; content:"|81 F1 03 01 04 9B 81 F1|"; content: "sock"; content: "send"; class type: misc-attack; sid:2004; rev:7;)</td></tr>
<tr><td rowspan="2">Witty</td><td>generated rules</td><td>alert UDP any 4000 -> any any (msg:"We detect one unknown worm!"; content:"|65 74 51 68 73 6F 63 6B 54 53|"; depth:238; class type: bad-unknown; sid:1000003; rev:5;)</td></tr>
<tr><td>Snort</td><td>alert UDP any 4000 -> any any (msg: "ISSPAM/Witty Worm Shellcode"; content:"|65 74 51 68 73 6f 63 6b 54 53|"; depth:246; class type: misc-attack; sid:1000078; rev:1;)</td></tr>
</table>

VI. CONCLUSIONS

The existing worm detection technology could not distinguish worms from normal network applications. In order to solve this problem, automatic detection technology for unknown worms is proposed in this paper. It uses anomaly detection technology to discover unknown worms, get the feature string sets and create new rules to detect worm feature. The experiments have proved that the detection method can discover new worms and has the ability to detect unknown worms automatically. Furthermore, the algorithm has the low rate of false alarm and gains a high detection rate.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (60973162), the Natural Science Foundation of Shandong Province of China (ZR2009GM037), Science and Technology of Shandong Province, China (2013GGX10129, 2010GGX10132, 2012GGX10110), the Soft Science of Shandong Province, China (2012RKA10009), the National Cryptology Development Foundation of China (No. MMJJ201301006) and the engineering technology and research center of Weihai information security.

REFERENCES

[1] Lianmin H u. A Novel Method of Network Traffic Anomaly Detection. Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on, 2011, 9:4757 – 4759J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] W. Yu, X. Wang, P. Calyam, D. Xuan, and W. Zhao. Modeling and Detection of Camouflaging Worm. Dependable and Secure Computing, IEEE Transactions on, 2011, 8(3):377-390.

[3] C. V. Zhou, C. Leckie, S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. Computers and Security, 2011, 29(1): 124-140.

[4] X. Y. Zhang, S.H. Qing, Q. Li. A Collaborative Detection Method Based on the Local Network Worms. Journal of Software. 2007, 18(2): 412-421.

[5] T. Jamie, M. W.. Matthew. Implementing and Testing A Virus Throttle. Proc of the 12th USENIX Security Symposium. 2003.285-294.

[6] S. Robertson, M. Miller, et al. Surveillance Detection in High Bandwidth Environments. Proc of DARPA DISCEX III Conference, 2003.130-139:

[7] B. George, H. B. Vincent. Early Detection of Internet Worm Activity by Metering ICMP Destination Unreachable Messages. Proc of SPIE - The International Society for Optical Engineering, 2002. 33-42.

[8] G. S. Zhao, T. Zhang. Design of Worm Detection System Based on Worm Propagation. Computer Security. 2009. 114-118.

[9] ZT Xiang, YF Chen, YB Dong, DM Lu. The Research Progresses of Worm Detection Technology. Computer Engineering and Design. 2009, 30(5): 1060-1064

[10] DJ Malan, MD Smith. Host-Based Detection of Worms through Peer-to-Peer Cooperation. In: Keromytis AD, ed. Proc. of the 2005 ACM Workshop on Rapid Malcode. Fairfax: ACM Press, 2005.72-8

Study and Design of Enterprise Public Security Platform based on PKI

Xiao Yingbin
China Tobacco Jiangsu Industrial Co., Ltd.
Nanjing, Jiangsu,China, 210011
e-mail: cngallop@aliyun.com

Zhao Yuanyuan
China Tobacco Jiangsu Industrial Co., Ltd.
Nanjing, Jiangsu, China, 210011
e-mail: zhaoyy@jszygs.com

***Abstract*—In order to solve the application security problems in the business operation system in a intensive, uniform and regular manner, a public fundamental platform model based on PKI technology is proposed, and some advanced design concepts like SCA, BPEL, etc. are introduced for enhancing the reconstructing and reusing of the system. According to the characteristics of PKI system structure, it is expanded as the foundation and realizes the support for WPKI technology. The technical realization of unified user management, unified authority management and unified application security management, as well as the approaches of integrating the peripheral business system are given by analyzing the requirements of general security management of the informatization system. Furthermore, we also proposed methods to further maintain the security and stability of platform operation with technologies like queues, connection pool, etc. Those methods provide a much more uniform solution to the application security problems. The feasibility and effectiveness of the platform in improving the application security problems have been discussed by the illustrations and analysis of the core transaction processing flows of this platform.**

Keywords-PKI; WPKI; LDAP; user management; authority management; application security

I. INTRODUCTION

With the constant development of enterprise informatization construction, the informatization level, as well as people's understanding about the informatization has been improved steadily. To satisfy the requirements of expanding informatization management, enterprises have constructed various types of information systems. At the same time, after years of informatization construction, the accumulated informatization assets, such as business data, business flows, etc. become be more and more significant for the development of enterprise. Prior to the increasingly urgent of security management requirements, problem of how to manage the normal operation of the information system uniformly, safely and effectively, to satisfy the demands of legal visit, and to reduce the system maintenance cost, is a realistic problem facing the enterprise administrator.

However, at present, the business system of enterprise adopts decentralized management, and it is also developed and implemented by different manufacturers. The generality management, including the user account management, user authority management and application security management of each system is scattered in individual business systems, which is independent from each other. On the other hand, they are still sharing the same contents, flows, methods and concepts of management. The repeated construction of these functions increase the informatization cost. To manage these information systems in a much more effectively and safely way, an integrated, unified and public foundation platform is required, for strengthening the centralized management and administration.

II. PUBLIC SECURITY PLATFORM MODEL DESIGN BASED ON PKI

A. *Introduction to the public foundation platform*

PKI (Public Key Infrastructure) technology is a well-recognized system structure for solving the internet security problems, which is widely applied in e-commerce [1]. Such security platform is mainly built on the PKI technology and integrates the WPKI (wireless Public Key Infrastructure) technology to unify user management and application security. The PKI-based public security platform shall integrate the sub-systems of business to unify management of users, authority and application security. The integration of platform and business system shall not only guarantee the instantaneity, reliability and security, but also be characterized by perfect openness, expandability and flexibility. Meanwhile, since the business systems are distinct in the hardware platform, operating system and application platform, it may be distributed in different geographic locations. Consequently, the PKI-based public security platform requires an appropriate integration framework as the support.

SCA (Service Component Architecture) is a brand new programming model proposed by the OpenSOA organization, as well as a set of service system construction framework agreement proposed by targeting at the SOA. It does not only integrate the IOC (Inversion of Control) thought, but also improves the target-oriented reuse to business-module reuse from the code reuse. Meanwhile, it realizes the deployment for service interface, which is completely separated from the calls, and it is assembled and bonded in flexible configuration mode. Therefore, the application of SCA normative design realizes a simple and systematic integration plan that is completely in accordance with the SOA thought, and it can better fulfill the system integration requirements.

B. *Design of public foundation platform model*

The design of OKI-based public security platform model is shown in Fig.1. The platform is deployed behind the firewall of the enterprise, between the user terminal

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.56

and enterprise application system. It mainly performs the unified account management and unified authority management, and provides unified application security guarantee measured between the user and application system. PKI-based public security platform consists of the secure connection service, task queue, service registration center, system platform, service construction layer, service bus layer, flow service layer and system application layer.

PKI-based public security platform will be the integrated management pivot of the enterprise, and it may confront substantial access requests. In order to ensure the stability, security and effectiveness of these requests in the platform processing, the platform employs secure connection service, task queue and task handling mechanism of the thread pool, which provide the ability to deal with substantial concurrent accesses. The secure connection service establishes SSL-based (Secure Sockets Layer) secure communication between the client and server. Meanwhile, it manages the access request connection to guarantee that the connecting request will not lose. In the task queue technology, the platform monitors the access request from the clients through the secure connection service. After receiving the access request, instead of dealing with it directly, the platform places it in the task queue. Tasks in the task queue will be processed according to different business operations by the back-stage processors. By applying these technologies, it can alleviate the pressure of concurrent accesses, improve the system resource utilization rate and guarantee reliable processing of substantial accesses, eventually reaching 100% success rate. Furthermore, it will not lose tasks during to the substantial accesses of users.

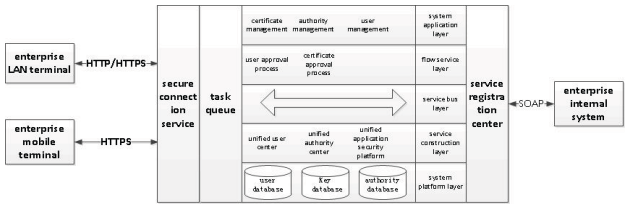


Fig. 1 design of PKI-based public security platform

The system platform mainly consists of the fundamental application software required by the operation of platforms, such as the user database, authority database, certificate database, key database, etc. The service building layer is designed according to the functions of PKI-based public security platform. The functions of the platform consist of many service components of fine granularity, and it is packaged into a series of coarse-grained service by SCA standards, namely Composite. These services are independent from each other, self-contained and can be reused. PKI-based public security platform includes three service modules: the unified user center，unified authority center and unified application security platform. The service bus layer and service registration center will accomplish the calls of service modules in the platform, as well as the integration between the platform and external system. With combining the BPFL (Business Process Execution Language) and BPMN (Business Process Model and Notation) standard, the flow service layer mainly recombines and arranges the service modules in the platform, so as to realize the modeling and implementation of business flows of the platform. The system application layer is the human-machine interaction interface, which provides the operation and system management for the platform.

III. CORE FUNCTION DESIGN OF THE PLATFORM

A. *Unified user center design*

The design of unified user center is shown in Fig.2, and it aims to establish standard, unique and complete user information resource center for the enterprise, which provides data sharing and service for all business systems.

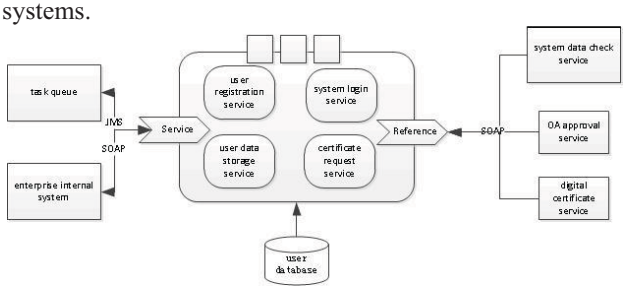


Fig.2 design of unified user center

Unified user center, as a composite, consists of several components, including service, reference, wire and property. Unified user center mainly provides services, such as the user registration initiated by the clients, user log-in service, user information data sharing service, etc. Meanwhile, it shall also quote the services provided by other systems to accomplish its own functions. For instance, user identification verification service provided by the HR system, application for approval service provided by OA system, digital certificate generating service provided by unified application security platform, etc.

In order to accomplish a series of operations, such as the registration/verification of user information and issuance of digital certificate, the unified user center shall support several specific business processes. To achieving the reconstruction and reuse of unified user center in the process flow, BPEL and BPMN standards is introduced into the design of platform. The development or management personnel of the platform shall model the business flows with the BPEL/BPMN visualization modeling tool. Such elements involved in the flows as the service component and business targets are represented by visual modeling symbols for selection and application. The configured and defined flow template shall be instantiated by the BPEL workflow engines for realizing the systematic function. BPEL provides advanced conceptual mechanism, such as the abnormity, business and compensation, and applies the partner links for abstracting the services as the process nodes. The specific binding agreement and service terminal are generated at the bottom BPEL container during the deployment.

The unified user center will process the inquiry and verification of user information frequently. For the purpose of improving the user experience and system

operation efficiency, it will select an appropriate storage way for storing the user data. LDAP (Lightweight Directory Access Protocol) can carry out special optimization for intensive operation, and it will be an order of magnitudes faster than the OLTP (On-Line Transaction Processing) in reading data from the relative database. Meanwhile, it is also a good choice to stores user information in the LDAP server due to its cross-platform, low cost and simple operation.

B. Design of unified authority center

Unified authority center takes the RBAC model as the basic design of the authority management. RBAC model introduces the concept of roles, and realizes the separation of user from the authority with roles, as well as the authority management with user-role and role-authority assignment [4]. In the authority management system based on RBAC model, the application system administer endows the user with roles, while the user takes the role assigned and obtains the corresponding permissions of their roles. Then the user is allowed to carry out corresponding operations. Such access control prevents the direct connection between the user and authority, further separates the authorization process and specific application, finally decreases the complexity of the authority management and improves the flexibility of authority management.

The unified authority center is in charge of the authority management based on RBAC model by taking LDAP as the technology of information organization and storage. Since LDAP is characterized by high query efficiency, dendroid information management model, distributed deployment frame, as well as flexible and fine access control, it is widely applied in the fundamental and critical information management [5]. The design of directory tree for unified authority center is shown in Fig.3. There are three types of sub-items in the directory tree of LDAP, including the digital certificate item, user item and application system item [6]. The digital certificate in the LDAP catalogue is applied for storing the PKI-based digital certificate (regular digital certificate) and WPKI-based digital certificate (wireless application certificate). The user information item is applied for storing the unified user log-in account information. The application system item is applied for storing the roles, authority and operating information specially set for specific enterprise application system. In the enterprise, each application system only has sub-directory trees representing the application system in LDAP tree in the PKI-based public security platform [7]. As shown in Fig. 3, in the sub-directory tree of user information, the DN (distinguished name) of the role is appointed in the role property of the user, which directly stores the binding information of role and user into the corresponding property of the user item. In the sub-directory tree of the role, the role item makes the roles and authority [8]. The binding of role/authority is mainly realized by directly storing the authority in the role node, in which the authority is mainly realized by appointing resource DN and carry out DN based on such resource [9]. Such design of directory tree is explicit in function, which makes the user, role, resource and operating information independent from each other, and makes it convenient for integrated management.

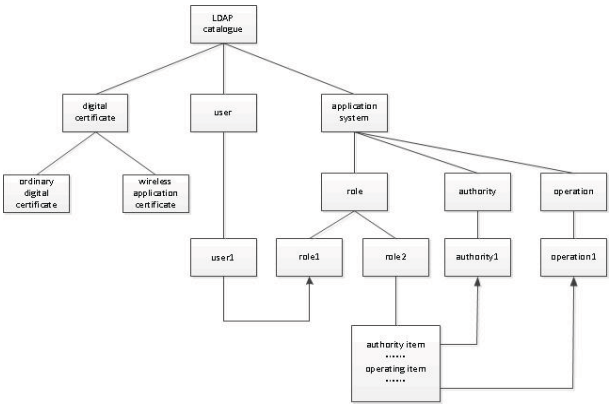


Fig.3 Design of directory tree structure of unified authority center

In the authority management center, the existing application system of the enterprise can issue authority to the authority management center through Web service, as well as corresponding operating service information. The newly constructed application system can construct authority management model in the authority management center directly, and keep a close relation with each business system. In this way, each business system will form a star-shaped intersecting structure logically by surrounding the authority management center.

C. Design of unified application security platform

The PKI (Public Key Infrastructure) is corresponding to the construction of infrastructures, such as the identity authentication, data encryption, digital signature, etc. with the public key theory and technology on the basis of unified security standards and authentication standards. by combining the symmetric cryptography and abstract algorithms based on the public key cryptographic algorithm, the PKI guarantees the confidentiality, completeness, non-repudiation and identity authentication of the data transmission with technologies like the digital signature, digital certificate, etc. [10]. WPKI is a set of key and certificate management system that introduces the PKI technology into wireless network with PKI successful experience as its foundation. It manages the secret key and digital certificate applied in the wireless network environment, and establishes a secure and reliable wireless network environment. WPKI simplifies and optimizes the traditional PKI in certificate format, secret key, decipherment algorithm, etc. for adapting to the finiteness of computer resources of mobile terminal device.

The design of unified secure platform follows the technology and management mechanism of ‘double certificates’ and ‘double centers’ employed in the digital certificate authentication system. The platform mainly consists of secret key management center (KMC), digital certificate authentication center (CA), audit registration center (RA), OCSP service, certificate issuing service, as well as application system interface. The design of unified application secure platform is shown in Fig. 4.

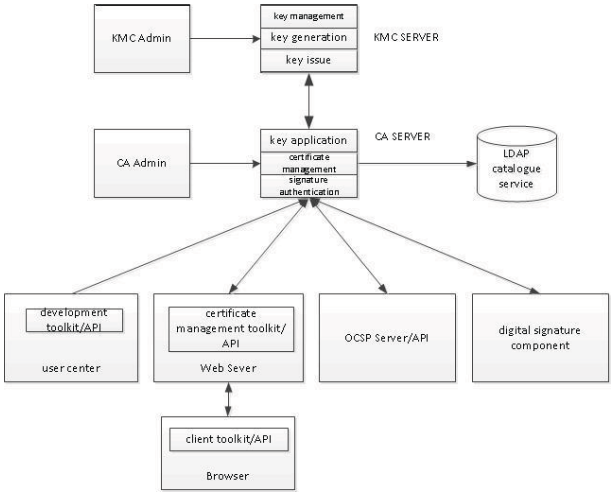


Fig.4 design of unified applied security platform

Combining the requirements of current mobile application security, KMC SERVER supports the Elliptic Curves Cryptography in design for satisfying the secure demands of wireless mobile application. CA SERVER supports the X.509 digital certificate of PKI technology and the X.509 digital certificate of WPKI. The unified user center is the registration center of the unified application security platform in design.

IV. MAIN OPERATION SERVICE PROCESS OF THE PLATFORM

A. User and digital certificate management process

Each business application system realizes the close integration with platform by taking advantage of the service provided by the public security platform. The user can realize the registration of new user in the unified user center of the platform when logs in the business application system for the first time. The system account information registered by the user will be valid in each business application system. The unified user center will undertake the RA function of the unified application secure platform, and submit the user information required by the generation of digital certificate to CA SERVER. The unified application security platform will generate digital certificate to realize the generation and issuance of digital certificate. Such certificate can be sent to the client terminal through API interface, or generate USB-form digital certificate.

B. Authority management process

System administrator accomplishes the assignment in the unified authority center according to the log-in authority management center, as well as the authority demand information acquired, and authority assignment information. The authority assignment information will be stored in the LDAP server. When the user logs in the application system, the access request inquires the authority assignment information stored in the LDAP server. The local client may set or store the authority information according to the initialization system of the achieved authority information. The above mentioned procedures make it convenient for access control.

C. Security management process

The identity authentication process: when the user logs in the business application system, the system will load digital certificate or remind the user to insert in the USB-form digital certificate, and the identity authentication request will submit the client digital certificate to the OCSP SERVER of the unified application security platform with the application support center. OCSP SERVER will send feedbacks about the effective information of user identity. Then the identity authentication is accomplished.

Digital signature process: the business application system client will carry out the digital signature for the submitted business data by calling the digital signature component or API of unified application security platform. After reaching the digital signature of the application system server, the digital signature information will be submitted to the unified application security platform verify the signature information. After confirmation of the digital signature and verifying the digital signature feedbacks, the application system server will continue the follow-u business processing. The work flow of identity verification and digital signature is shown in Fig.5.

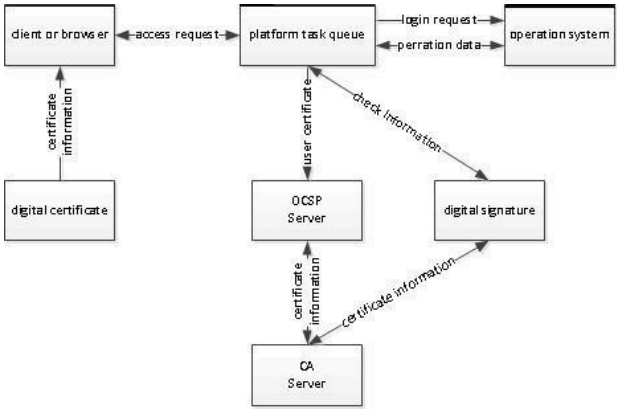


Fig.5 Work flow chart of authentication and digital signature

V. CONCLUSION

In this paper, a PKI-based public foundation platform model is proposed by studying the unified user management, authority management and application security management. Such proposal will satisfy the current business application system, as well as the requirements of user account authority and security management in wireless application. It also satisfies the management requirements of the application system on user account, authority and security. In order to strengthen the integrated control strength of the system and enforce the standardized control, a new realization thought and method is provided. Meanwhile, it can also decrease the development difficulty of each business application system, reduce the investment cost, and enhance the system maintenance efficiency.

REFERENCES

[1] Fu Jiandan, Xiong Xuandong, Fan Yan, Zhang Liangzhong, Wang Songfeng, A PKI System Application Integration Plan based on the Message-oriented Middleware [J]. Computer Applications and Software, 2012, 29 (11): 208-213.

[2] Wang Tingting, Research on Mobile Electronic Commerce Application Based on WAP an WPKI Protocol [D]. Shandong: Shandong University, 2010: 32-39.

[3] Huang Mu, Design of Authority Management System based on RBAC Model [D]. Jilin: Jilin University, 2013: 2-9.

[4] Wu Guangcheng, Shi Yunfeng, Implementation of Privilege Management System based on RBAC [J]. Electronic Test, 2008, 5: 87-91.

[5] Tan Shenglan, The Design and Implementation of the Campus Network Unified Identity System Based on LDAP [J]. Journal of Dongguan University of Technology, 2009, 3 (16): 82-86.

[6] Wang Zi, Wen Qiaoyan, Wireless Application Security Platform Based On WPKI And PMI [J]. Control & Automation, 2007, 23 (1-3): 56-60.

[7] Zhu Yabin, Design and Implementation of Uniform Authorization Management System Based on LDAP [D]. Beijing: Beijing University of Technology, 2006: 43-59.

[8] Xu Xiaohong, Shi Yongge, Zheng Kaixin, Research and Implementation of Unification User Management System Based on LDAP [J]. Computer and Modernization, 2008, 5: 114-120.

[9] Zhu Shaomin, Liu Jianming, Wei Xiaojing, Design and Realization of Enterprise Unified User Identity Management System Based On Lightweight Directory Access Protocol [J]. Nuclear Electronics & Detection Technology, 2008, 3 (28): 662-666.

[10] Wang Yue, Yang Pingli, Gong Dianqing, Design and Realization of Information Security Access Control System Based on PKI Technology [J]. Computer Engineering and Design, 2011, 32 (4): 1249-1253.

Memory Integrity Protection Method based on Asymmetric Hash Tree

Ma Haifeng[1,2], Chengjie[1], Gao Zhenguo[2]

[1]Computer and Information Engineering Institute
Heilongjiang University of Science and Technology
Harbin, China
e-mail: mhf_2000@163.com, borlandsteven@163.com

[2]Institute of automation
Harbin Engineering University
Harbin, China
e-mail: gag@hrbeu.edu.cn

***Abstract*—Focus on this question of high verification overhead of hash tree memory protection scheme, this paper proposed a hash tree optimize method called AH-Tree. Utilizing the locality character of memory accessing, the memory is divided into two parts and an asymmetric hash tree (AH-Tree) is constructed. The advantage of AH-Tree is low average operation overhead. The analysis and experiment results proved that the proposed scheme has shorter average verification path and better verify efficiency than hash tree scheme. AH-Tree is a feasible memory integrity protection scheme.**

Keywords—memory; integrity; replay attacks; hash tree

I. INTRODUCTION

Recently, the attacks focus on computer storage system are emerging everywhere. The attacks of storage system are divided into software attack and hardware attack. Software attack is to attack the system by software means such as malicious software and virus. Hardware attack is break security measures using customized spoofing hardware or device to destroy or replay the data got from the bus [1].

Software attack can be protected by regular safety technology such as antivirus software, but most software or even light weight hardware based protections cannot resist this kind of hardware attacks [2]. Data integrity and confidentiality can prevent this kind of attack. Integrity is to prevent data illegally tampered with or tampered data can be detected. Confidentiality is to limit the sensitive data access. The attacker can't understand the meaning even get the data. This paper emphasis is data integrity protection.

For data integrity protection, many work presented new schemes for uniprocessor [3][4][5] and Multi-processors platform[6][7][8]. This paper focused on the integrity protection scheme of uniprocessor. Authentication in XOM (eXecute Only Memory) cannot detect replay attacks [1]. The Merkle tree scheme used in Gassend et al. [9] causes severe performance degradation on runtime. The CHTree scheme [9] improves runtime performance at the cost of cache space. M-TREE [10] reduced to 32-bit MAC length by 256 bit and the computational overhead, but safety remains to be proven. The Authenticated Speculative Execution proposed by Shi et al.[11] employs timely authentication, but requires extensive modifications to the memory controller and on-chip caches. The TEC-Tree [12] reduces computational overhead, the Bonsai Merkle Tree (BMT) [13] is a novel Merkle tree-based memory integrity verification technique, to eliminate these system and performance issues associated with prior counter-mode memory encryption and Merkle tree integrity verification schemes. TEC-Tree and BMT both need to maintain a hash tree. Although many security architectures have been proposed, these schemes either have inadequate security protection or have high performance overheads.

Focus on the problem of hash tree, this paper proposes a hash tree optimized scheme. It can take more efficient integrity verification. The security model was presented in Section 2. The architecture of the scheme was described in Section 3. The overhead and performance analysis were discussed in section 4. The scheme was evaluated on the simulator in section 5 and the paper was concluded in Section 6.

II. HARDWARE ATTACKS AND SECURE COMPUTING MODEL

Broadly, attacks can be classified into software and physical (hardware) attack, and physical attacks can be classified into two main categories: passive attacks and active attacks [14]. Passive attacks are the ones the adversary simply observes the data going to and from the processor chip in a non-intrusive way. Active attacks are the ones where the adversary corrupts the data residing in memory or transiting over the bus. The most common active attacks are discussed below:

1) Spoofing attacks: the attacker tries to change the data value at a memory location and tries to pass it off as valid data.

*2) Splicing attacks:*involve taking valid data values and duplicating them or replacing them with values at other locations.

*3) Replay attacks:*the attacker records old values of data blocks and replays them at a later point of time.

Secure Computing Model of this paper is considered built around a single processor with external memory and peripherals. It can resist active attacks. The model comprises a tamper-resistant processor (TCB [1], trusted computing base), external memory and peripherals. The TCB consists of the processor core, on-chip cache, encryption and integrity verification mechanism.

The trusted components of security model is the processor's core, which means that the processor is invulnerable to physical attacks and its internal state can't be tampered or observed. The untrusted components are off-chip memory, the system bus and peripheral devices. Their states can be observed and tampered by an adversary. The target of an adversary is to tamper with the contents of external memory in such a way that the system produces an incorrect result while looks correct to the

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.57

system user. Once a program executes some special instructions to enter the security executing environment, TCB is responsible for the protection of the program. The TCB or the processor needs to detect that whether memory operations are normally executed or not.

To make it simple, the untrusted memory in this paper only means the RAM, although in fact, the scheme presented here can be applied to other data storage devices such as hard disks only with a little change.

III. PRINCIPLE OF INTEGRITY PROTECTION APPROACH

A. *The Principle of AH-Tree*

By the principle of locality, each area of the memory within a time period isn't visited equal probability to access. It has different access frequent. We can set lower access overhead for frequent access area and set higher access overhead for infrequent access area. From this, the asymmetric hash tree is put forward. It basic idea is the protect memory area is divided into hot area and cold area. The area with high access frequent is hot area, its proportion is small (10%-20%), and the area with low access frequent is cold area, its proportion is big (80%-90%).

Take memory block as leaf node, hash trees are constructed for hot area and cold area respectively. The tree build by Hot area is lower and build by cold area is higher, and then two hash trees are connected to build an asymmetric hash tree. The root node of hash tree is saved in TCB.

Compared with hash tree, the advantage of AH-Tree is lower average operation overhead. The reason is the verification path of Hot area is shorter; the operation overhead is lower, and number of operations is higher, while the verification path of cold area is longer; the operation overhead is higher, and number of operations is lower. In general, the average operation overhead is lower.

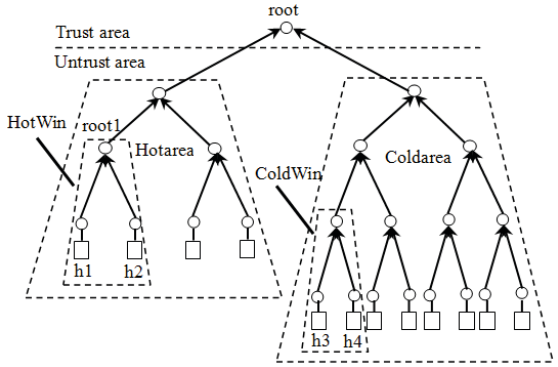


Figure 1. AH-Tree structure

AH-Tree is binary tree. Its structure is shown in figure 1. The area in the left dotted line is Hot area, the area in the right dotted line is cold area (Only to state the question, Hot area and cold area may be not only one). The root node of hash tree is saved in cache, can't tampered with.

About security, AH-Tree is the improvement of hash tree, so its verification pattern is the same as hash tree. They both verified from leaf node to root node of security area. Therefore, the data integrity protection ability is also same as the hash tree, and they both can prevent hardware and software attack behaviors including replay attack.

B. *The Regulations and algorithms of AH-Tree*

To make AH-Tree work efficient, we define some operations and algorithms. For the convenience to description, we use the concept of access window. Smaller access area in Hot area called HotWin, which made of by some memory blocks. In figure 1, h1 and h2 made of a HotWin in a certain period of time.

The root1 is the root of HotWin of hash tree. A Hot area has one or more HotWin. Smaller access area in cold area called ColdWin, h3 and h4 made of a ColdWin. A Hot area also has one or more ColdWin. The HotWin is same size as ColdWin, its size is smaller than the whole memory. For the convenience to description, HotWin and ColdWin both called access window.

For the size of access window, suitable width should be set. When access window is smaller, for a bit of and discrete distribute memory access, the cover of access area is more reasonable and performance is better. When access window is bigger, for massive and continuous memory access, less access windows are needed. There are less Hot area movements and performance is better.

How to specific confirm the windows size depend on simulation or mathematical calculations.

1) AH-Tree regulations: Specific rules of AH-Tree are as follows:

- The access window width is fixed, and the whole memory space to be protected is the integral multiple of window width.
- Access window can move with access area at horizontal direction according to windows width
- There are multiple access window, they can continuous distribution to cover a bigger access cluster area. They can also discrete distribution corresponding many discontinuity access cluster area.
- Memory block size of access window is same.

*2) The Process of AH-Tree:*Under the condition that conforms to the above rules, we define processes as followed:

*a) Split Hot Area:*Use counter method to calculate hot area and cold area, and then hot area and cold area are divided into many equal access window.

*b) Initialize Hash Tree:*For convenience of description, Hot area and cold area are collectively known as access area, the detail steps are followed:

- Calculate the hash value of each block in access area to get leaf nodes.
- Connect the two adjacent leaf node values of access area and commutate it hash value.
- And so on, until hash sub-tree root node of access area.
- Build higher level rest hash tree based all hash sub-tree nodes.
- The Root node of hash tree is stored in trusted storage.

c) Verify a Node of Access Window.

- Read the node and its sibling node.

- Connecting their data and proceed hash computation.
- The result is compared with parent node to check match or not.
- If two node match, continue verify until root node.

d) Update Access window node

- Update memory data as new data.
- Connect the node and sibling node data and compute hash value.
- Update parent node with the hash result
- Repeat the process until the root node of hash tree.

e) The Translation of Hot area.

- Hot area translates to other position according to integral multiple of access window.
- Execute the process Split Hot Area
- Execute Initialize Hash Tree.

C. Overhead and Performance Analysis of AH-Tree

In this part the overhead and performance of AH-Tree is analyzed, the comparison object is hash tree. For convenient to describe, tree structure is binary tree.

1) The operation overhead of AH-Tree: To calculate the operation overhead of AH-Tree, we need calculate leaf node overhead, internal nodes overhead and average operation overhead of hot area and cold area.

a) The leaf node operation overhead: The overhead is shown in (1), α is byte access time, s is memory block size, k is the time required for each hash computation and T is overhead function.

$$E = T(\alpha \times s + k \times s) \tag{1}$$

Equation (1) can be measured $Cost = s(\alpha + k)$. For α and k are constant value, then the operation overhead of leaf nodes is proportional to the block size.

b) Operation overhead of access internal node: Each internal node involves two child nodes, and the internal node size depends on the hash function used. And then the operation overhead of access internal node is shown in (2).

$$E_i = T(2\alpha \times i + k \times i) \tag{2}$$

An algorithm (MD5 or SHA-1.etc) is chosen as hash function, therefore, the operation overhead of hot area and cold area are the same.

c) The relationship between verify depth and block size. Hash verify depth of hot area is d_h, and cold area is d_c. The hot area size is *H* and the cold area size is *C*. The relationship between verify depth and block size of hot area and cold area is show in (3) and (4):

$$d_h = \log_2(H / s) \tag{3}$$

$$d_c = \log_2(C / s) \tag{4}$$

d) One time verify overhead: One time verify overhead include leaf node overhead and verify path overhead. The latter is the product of verify overhead of each node with verify deep. One time operation overhead of hot area is show in (5):

$$E_h' = E + E_i \times d_h \tag{5}$$

If only consider the latency of node access, (1), (2) substitute into (5) we can get $E_h' = T(s + 2i \times d_h)$, it can be expressed:

$$Cost_h = s + 2i \times d_h \tag{6}$$

In similar way, one time operation overhead of cold area can be expressed:

$$Cost_c = s + 2i \times d_c \tag{7}$$

e) The average operation overhead: It associated with access frequency (probability). We set the probability of access hot area is *p*, the probability of access cold area is 1-*p*. Then the expectation of one time operation overhead are:

$$\overline{E} = p \times E_h' + (1-p) \times E_c' \tag{8}$$

Equation (6), (7) substitute into (8). We can get minimum cost of memory block size when parameters are confirmed (include α, k, i). The average operation overhead is:

$$\overline{Cost} = p \times Cost_h + (1-p) \times Cost_c \tag{9}$$

2) The operation overhead of Hash-Tree: We set A is the memory space need protected, s is memory block size. Then the verify depth of hash tree is:

$$d = \log_2(A / s) \tag{10}$$

We set hash tree and AH-Tree based on same hash function. Then one time operation overhead of a data block is $E' = E + E_i \times d$, (1) and (2) substitute into it, the overhead can be expressed:

$$Cost = s + 2i \times d \tag{11}$$

3) Performance gain

The AH-Tree performance gain compared with hash tree is:

$$k = (Cost - \overline{Cost}) / Cost \tag{12}$$

Equation (9), (11) substitute into (12) and simplify. The memory space to be protected is 1GB. Inner node size is 512B and s is 1KB. The Hot area (hot) proportion is set 10%, 20%, 30% respectively. Then we obtained performance gain of AH-Tree compared with hash tree, the results are shown in Figure 2. The horizontal coordinate *p* is the access probability of Hot area, and the vertical coordinate *k* is the performance gain rate.

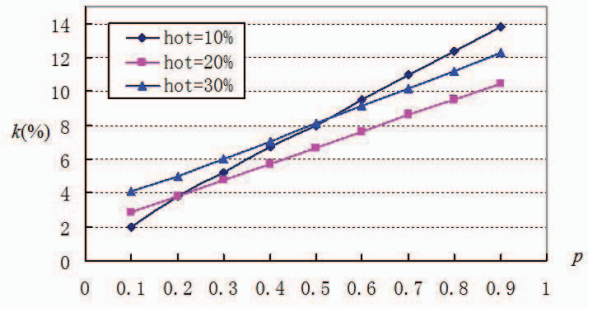


Figure 2. AH-Tree gain rate compared with hash tree

As shown in Figure 2, when p is in the range of 0.1-0.9, gain rate is 2%-12%. Access probability of Hot area is proportional to the gain rate. The reason is the access overhead of Hot area is lower, therefore, the higher the

Hot area visits, the lower the average cost. When access probability is fixed, the greater proportion, the slower of performance promote. The reason is the higher Hot area proportion, the taller hash tree of Hot area, the higher average overhead, the smaller performance advantage.

IV. EXPERIMENT AND SIMULATION

A. Simulation Framework

The simulation work is evaluated by SimpleScalar tool set [15], it runs binary instructions set. We evaluated AH-Tree and hash tree. We modified sim-outorder simulator to supports AH-Tree and hash tree. The simulation benchmark is "Base". The term "Base" refers to a standard processor without integrity verification or encryption.

The main architectural parameters of simulator is: L1 cache is 64KB (2-way, 128B line), L2 cache is 1MB (4-way, 128B line), memory latency is 80 cycles, hash latency is 160 cycles, hash length is 128bits and hash throughput is 3.2GB/s. We set the memory space needs protected is 1GB, memory block size is 1K and the ratio of Hot area is 10%.

For all the simulations in this section, eight SPEC2000 benchmarks [16] are used as representative applications are vortex, vpr, art, parser, gzip, mcf, equake and mesa. To capture the characteristics in the middle of computation, each benchmark is simulated for 100 million instructions after skipping the first 1.5 billion instructions.

B. IPC Performance Evaluation

We first investigate the IPC Performance of AH-Tree and hash tree (HashTree). IPC is an important index of evaluate system performance. Its results is shown in Figure 3, all IPC values are normalized with Base. As shown in the Figure, AH-Tree or HashTree has some impacts on performance, but the degree of performance degradation is different.

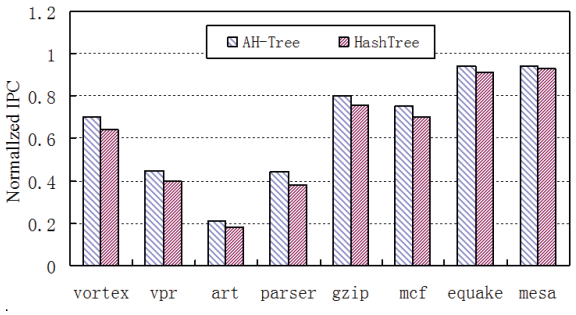


Figure 3. AH-Tree performance compared with hash tree

For AH-Tree performance decline degree, minimum is 6% (mesa), maximum is 79% (art) and average is 34%. For HashTree performance decline, minimum is 7% (mesa), maximum is 82% (art) and average is 39%. Compared with HashTree, AH-Tree has 5% performance improvement in average.

The reason is for hot area of AH-Tree, tree height and once operation overhead is lower, and operation number is higher. For cold area of AH-Tree, tree height and once operation overhead is higher, and operation number is lower.

C. Cache Miss-rate Evaluation

Cache miss rate is another important index of system performance. We evaluate the L2 cache miss-rate of AH-Tree. The miss-rate is shown in Figure 4. When cache is 256KB, miss rate is high. The reason is AH-Tree node contend with programs run on the processor. The L2 cache is too small to enough space to load most data.

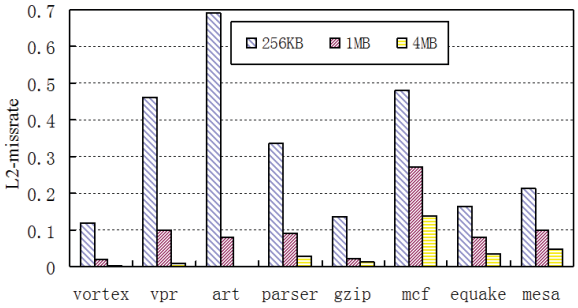


Figure 4. AH-Tree L2-missrate comparision with different cache size

Therefore, the processor often read data from the memory, which causes high L2-cache miss rate and decrease the system performance. With the L2 cache size increase, the miss rate decrease quickly. When cache size increases to 4MB, the miss rate is decrease to a low range. The reason is with the L2 cache increase, the number of processor read data from the memory is reduces gradually. The cache contention problem also relieve gradually. When cache size increases to certain degree, most data that needed can be load. Therefore, the number of cache miss rate is less.

It should be pointed out that many factors such as tree architecture parameters, test sample and select index will affect the experimental results. The result isn't very accurate, but combined with above analysis, can prove the scheme we propose is a effective method of storage integrity checking.

V. CONCLUSION

This paper proposed an improvement method of hash tree verification. It can detect most hardware attack behaviors. The performance analysis and simulation results have proved that the scheme we proposed has better efficiency (2%-12% improvement) than hash tree. Ongoing work is further improve the scheme the security and performance, and how to apply to cloud storage system.

ACKNOWLEDGMENT

This work is supported by Scientific Research Project of education department of Heilongjiang Province under Grant No. 12533052; Natural Science Foundation of China under Grant No. 61073047.

REFERENCES

[1] Suh, D. Clarke, B. Gassend, M. van Dijk, and S. Devadas. Efficient memory integrity verification and encryption for secure processors [C]. The 36th International Symposium on Microarchitecture, 2003: 339-350.

[2] Abhishek Das, Gokhan Memik, Joseph Zambreno. Detecting/preventing information leakage on the memory bus due to malicious hardware [C]. The Conference on Design, Automation and Test in Europe. 2010: 861−866.

[3] W. Shi and H.-H. Lee. Authentication Control Point and Its Implications for Secure Processor Design [C]. In Proc. of the 39th Annual International Symposium on Microarchitecture, 2006.

[4] C. Yan, B. Rogers, D. Englender, Y. Solihin, and M. Prvulovic. Improving cost, performance, and security of memory encryption and authentication [C]. In Proc. of the International Symposium on Computer Architecture, 2006.

[5] S. Nimgaonkar, M. Gomathisankaran, Energy efficient memory authentication mechanism in embedded systems [C]. In Proc of the 2011 International Symposium on Electronic System Design (ISED), 2011: 248–253.

[6] [10] Brian Rogers, Chenyu Yan, Siddhartha Chhabra, Single-Level Integrity and Confidentiality Protection for Distributed Shared Memory Multiprocessors [C]. In Proc.of the International Symposium on Computer Architecture, 2008.

[7] P. Cotret, J. Crenne, G. Goniat, J.-P. Diguet, L. Gaspar, and G. Duc. Distributed security for communications and memories in a multiprocessor architecture. Workshop RAW, 2011, 5: 326–329.

[8] Y. Zhang, L. Gao, J. Yang, X. Zhang, and R. Gupta. SENSS: Security Enhancement to Symmetric Shared Memory Multiprocessors [C]. In International Symposium on High-Performance Computer Architecture, 2005.

[9] B. Gassend, G. Suh, D. Clarke, M. Dijk, and S. Devadas. Caches and hash Trees for Efficient Memory Integrity Verification [C]. In Proc of the 9th International Symposium on High Performance Computer Architecture (HPCA-9), 2003.

[10] C. Lu, T. Zhang, W. Shi, and H.-H. S. Lee. M-TREE: A High Efficiency Security Architecture for Protecting Integrity and Privacy of Software [J]. Journal of Parallel and Distributed Computing, 2006, 66(9): 1116-1128.

[11] W. Shi, H.-H. Lee, M. Ghosh, and C. Lu. Architectural Support for High Speed Protection of Memory Integrity and Confidentiality in Multiprocessor Systems. In Proceedings of the International Conference on Parallel Architectures and Compilation Techniques, pages 123–134, September 2004.

[12] Elbaz, R., Champagne, D., Lee, R.B., Torres, L., Sassatelli, G., Guillemin, P. TEC-Tree: A Low Cost and Parallelizable Tree for Efficient Defense against Memory Replay Attacks. In: Cryptographic Hardware and embedded systems (CHES), pages. 289–302,2007.

[13] Rogers, B., Chhabra S., Solihin. Using address independent seed encryption and bonsai merkle trees to make secure processors os- and performance-friendly [C]. In Proceedings of the 40th International Symposium on Microarchitecture. IEEE Computer Society, Los Alamitos, CA, 183–196.

[14] Austin Rogers, Aleksandar Milenković. Security extensions for integrity and confidentiality in embedded processors. Microprocessors & Microsystems. Volume 33, Issue 5-6, August 2009. Pages: 398-414.

[15] Douglas C. Burger and Todd M. Austin. The SimpleScalar Tool Set, Version2.0. UW Madison Computer Sciences Technical Report, 1997,6.

[16] H. Al-Zoubi, A. Milenkovic, M.Milenkovic. Performance Evaluation of cache Replacement Policies for the SPEC CPU2000 Benchmark Suite [C]. In Proc. of the 42nd ACM Southeast Conf, 2004.

Parking Guidance System Based on ZigBee and Geomagnetic Sensor Technology

Fengli Zhou
Faculty of Information Engineering
City College Wuhan University of Science and Technology
Wuhan 430083，China
thinkview@163.com

Qing Li
Faculty of Information Engineering
City College Wuhan University of Science and Technology
Wuhan 430083，China
57024241@qq.com

***Abstract*—Concerning the phenomenon that common parking service could not satisfy the increasing demand of the private vehicle owners, an intelligent parking guidance system based on ZigBee network and geomagnetic sensors was designed. Real-time vehicle position and related traffic information were collected by geomagnetic sensors around parking lots and updated to center server via ZigBee network. On the other hand, out-door Liquid Crystal Display screens controlled by center server can display information of available parking places. In this paper, guidance strategy was divided into four levels, which could provide clear and effective information to drivers. The experimental results prove that the distance detection accuracy of geomagnetic sensors was within 0.4m, and the lowest package loss rate of the wireless network in the range of 150m is 0%. This system can provide solution for better parking service in intelligent cities.**

Keywords- ZigBee; Internet of Things(IoT); geomagnetic sensor technology; intelligent parking

I. INTRODUCTION

Car ownership in China is more than 120 million at the end of 2013 and increases 63.92 million over the end of 2008, it also continued to speed growth. Sharp increase in car ownership has caused urban traffic congestion and parking difficulties, these serious impact on people's life quality and restrict the development of urban transportation, automotive industry and economy[1-4]. With the acceleration of transportation construction pace, intelligence transportation has become the development direction of urban traffic management, it is an important way to ease traffic congestion and improve the efficiency of transportation and management. Current road traffic information collection technologies mainly include that GPRS wireless communication technology, infrared detection technology and radio frequency identification (RFID) technology. GPRS communication technology must pay extra fee for network using to operators and will easily lead to communication delay when the network is busy; infrared detection communication distance is shorter and will be affected by car's thermal radiation; although passive RFID technology can achieve accurate vehicles' position and anti-jamming performance is ideal, it only supports star network, the amount of transferred data and communication distance will be very limited, so it is not easy to mass deployment and difficult to promote.

In order to solve the questions of parking difficulties and poor urban parking information, and improve urban traffic intelligence, an intelligent parking guidance system based on ZigBee network and geomagnetic sensors is designed[5]. It has combined GPRS, infrared detection and RFID technology, the advantages of this combined program are as follows:

- Flexible networking(It can support star, tree, mesh and topology).
- It can carry a large amount of data and is easy to large-scale deployment.
- Geomagnetic sensor has stronger anti-interference ability than infrared detector.

The system based on these advantages will achieve transparent system operation and precise parking information. It can enhance the signal immunity and solve the questions of network coverage range, high system scalability and non-automotive etc. Building intelligent parking information network platform can improve the efficiency of car park, reduce vehicle detour distance, allow drivers to find car parking spaces quickly and establish good urban traffic environment, so development of parking guidance system which is integrated of a variety of advanced technologies has a very important practical significance.

II. OVERALL PROGRAM DESIGN

Parking guidance system has consisted of two modules that are called parking information collection and parking information distribution[6-7]. The first module can detect in & out vehicles' status by using geomagnetic sensor which is installed on the city parking spaces in real time, and access parking information for each car park, finally collect and upload the data. The second module will transport the collected information through ad hoc network built by Zigbee, ZigBee coordinator can receive parking information for each car park nodes and release it through a variety of terminal ways, such as website inquiries, phone inquiries, traffic guidance screen etc. The overall structure of system is shown in Figure 1.

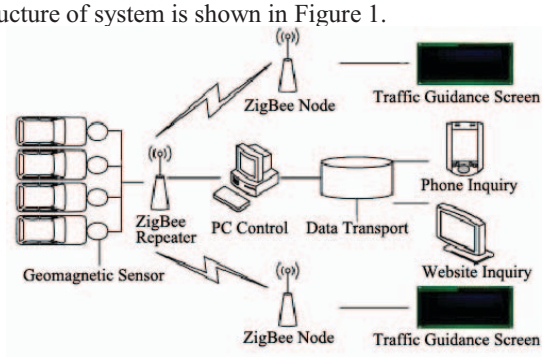


Figure 1. Overall structure of parking guidance system.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.58

III. SYSTEM FEATURES MODULAR DESIGN

A. Hardware Design

1) Parking Information Collection System Design

Geomagnetic sensor HMC5883 has collected parking information, then send the data to ZigBee processor CC2530 for appropriate treatment by standard Inter-Integrated Circuit (I^2C) bus protocol, finally transmit processing results to the traffic monitoring PC by serial communication interface RS232, and send them to ZigBee router nodes by ZigBee wirelss communication, processing result will be displayed on the liquid crystal display(LCD). The overall structure of system hardware design is shown in Figure 2.

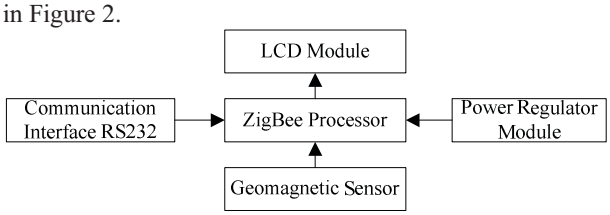


Figure 2. Overall system hardware structure design.

Serial physical interface communication protocol that the system chosen is RS232 standard, which developed by U.S. EIA(Electronics Industry Association) and BELL companies and announced in 1969. It is suitable for the communication that data transfer rate range is between 0 to 20kb/s. The level conversion chip of serial interface unit chip the system adopted is single power level converter chip MAX232 and its single power supply is +5v. The chip can satisfy all technical standards of all the RS-232C, and is integrated of two RS-232C drivers in its internal, and has typically low supply current of 5mA. The serial communication interface unit circuit of RS232 is shown in Figure 3.

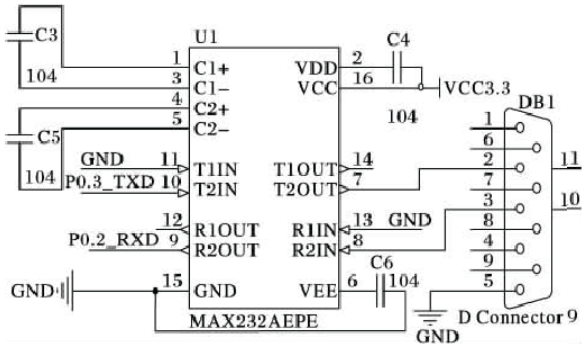


Figure 3. Serial communication interface unit circuit of RS232.

When a vehicle passes, disturbances will be generated on the earth's magnetic field around it, then magnetoresistive sensor HM5883 can detect the presence of vehicles and analyze the state of vehicles that is passing or parking[8]. Geomagnetic sensor HMC5883 and ZigBee terminal CC2530 communicate by I^2C communication protocol, and standard mode of data transfer rate is 100kb/s or 400kb/s. Geomagnetic sensor detection process is shown in Figure 4. If geomagnetic sensor detects a vehicle entering, number of vacant spaces subtracts one. And when a vehicle has exited, number of vacant spaces adds one.

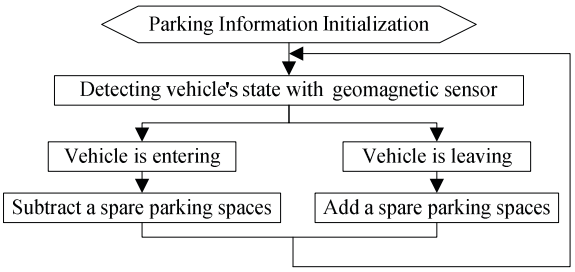


Figure 4. Geomagnetic sensor detection process.

2) Parking Information Distribution System Design

The main roads of city can be divided into three kinds: roads, secondary roads and branch roads, parking areas are distributed in cities randomly. The system uses four-level guidance system for parking vehicles and realizes intelligent parking guidance. The overall block diagram of four-level parking guidance is shown in Figure 5. We can achieve urban intelligent parking through four-level guidance: First-level display screen is set on the urban roads and used to publish name, location, prediction of parking status for multiple car parks. Second-level display screen is also set on the urban roads and used to publish name, driving directions, and prediction of parking status information for car park. Third-level display screen is set near the parking lot entrance and used to release name, actual parking status, and other requirement information for car park. Forth-level display screen is set in the parking inside and used to publish the distribution of parking spaces, empty or full status for car park, this can help drivers to find parking quickly and park accurately.

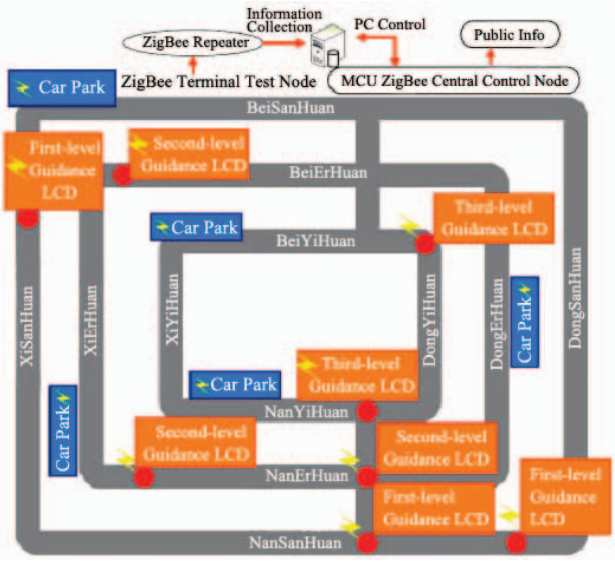


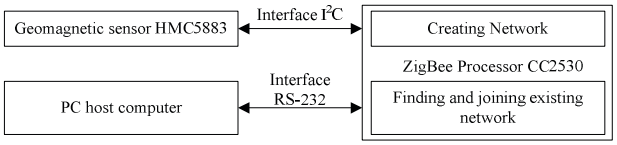
Figure 5. Overall block diagram of four-level parking guidance.

B. System Software Design

1) Overall Design of Information Distribution System

The block diagram of system's overall software design consists of two parts: software design for sensor nodes and PC, both of which communicate by common computer serial port RS232. In the software design of sensor node, chip CC2530 based on ZigBee protocol initializes and sets up ZigBee network firstly, then finds router nods and

terminal nodes by a certain networking communication protocol for ZigBee, adds them to the existing network and forms a specific network topology. At the same time, geomagnetic sensor HMC5883 begins to collect traffic data after system initialization task is completed, then delivers the information to chip CC2530 for data analysis and processing via I2C communication protocol. Processor CC2530 can store the post-processing results or sent them to ZigBee's central monitoring node. Information distribution system is shown in Figure 6.



Information distribution system design diagram.

2) Realization of Parking Information Query Interface

Development platform of parking information query interface is Visual Studio 2010 and SQL Server 2008. Serial receiving procedures is done on the platform of Visual Studio 2010, and the establishment of a database is completed based on SQL Server 2008. Serial program is a winform program by using C# language, it uses serialport control to read data in the cache by asynchronous thread and delegate, then separates data by split function, finally updates the database by processed data. Data receiving port interface is shown in Figure 7.

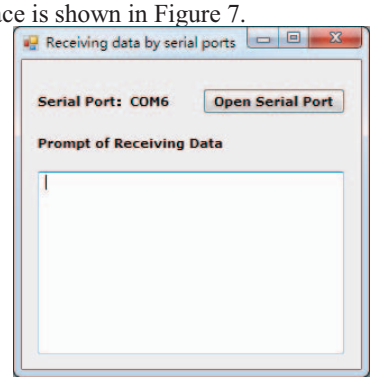


Figure 6. Data receiving port interface.

The software platform can receive data packets from ZigBee coordinator by serial port when working properly, then analyze them and display the relevant information to the service platform. The flow chart of system program is shown in Figure 8.

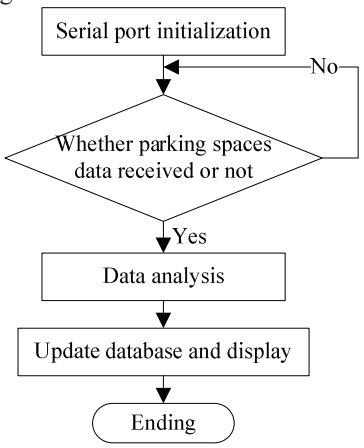


Figure 7. Parking information services platform process.

Serial port receiver can get real-time parking information data of car park, and then upload it. Parking information services system has updated and displayed data of each car park, realized immediate releasing for parking number and parking spaces. Users can choose effective terminal to access free parking spaces' number and location information of car park by logging system, then quickly select and park, shorten parking time, improve parking efficiency and reduce traffic congestion.

IV. SYSTEM EXPERIMENTS AND RESULTS ANALYSIS

A. Geomagnetic Sensor Test

Testing change of magnetic field strength while distance between sensor and vehicle is in the range of [0.2,1.4], From the experimental data it can be seen the relationship of magnetic field strength and sensor distance, and the curve is shown in Figure 9.

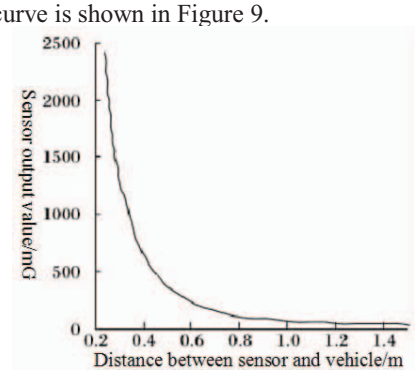


Figure 8. Curve of magnetic field strength.

Fitting experimental data by using Matlab, curve equation obtained is: $G = 57.96d^{-2.593} + 22.43$, $R^2 = 0.9987$. Where G is sensor output value(unit is mG), d presents the distance between sensor and vehicle(unit is m). Test results show that geomagnetic sensor HMC5883 can effectively detect vehicles approaching by magnetic field changes in the range of 0.4m.

B. ZigBee Network Testing

Package loss rate of wireless transmission has reflected communication performance of ZigBee network. System testing has carried out data collection and analysis of system package loss rate, the purpose is to review work and practical application performance of system.

The experiment is conducted without routing node firstly, and can obtain a calculated value of package loss rate by point detection method. Then moving node, judging whether the correct information is received by the data on the LCD screen, finally under conditions of increasing routers, obtaining test data of package loss rate by point detection method and experimental data of package loss rate under no router nodes. Test number is 50 times and results are shown in Table1.

TABLE I. SUBSTITUTION RATE OF NO ROUTING NODE

Coordinator	Terminal	Distance /m	Package loss rate /%
A1	A2	80	0
B1	B2	>80	1.5

The experimental data of substitution rate by increasing router nodes is shown in Table2.

TABLE II. SUBSTITUTION RATE OF INCREASING ROUTING NODE

Coordinator	Router	Terminal	Distance /m	Package loss rate /%
A1	A2	A3	150	0
B1	B2	B3	>150	5

Experimental results show that effective transmission range between ZigBee coordinator and terminal is 10~100m, the transmission distance and network range can be extended by relaying between routers and nodes. The ZigBee ad hoc network built in this paper can implement normal information collection work of each urban parking area and be applied to ordinary city.

V. CONCLUSION

The system uses ZigBee wireless RF processor CC2530 to build hardware platform of sensor node, completes the design of ZigBee wireless RF unit and HMC5883 magnetic sensor unit, and applies software engineering concepts to the entire design and implementation of system software, completes ZigBee protocol stack and geomagnetic sensor module.

System features are as follows:

- Traditional induction coil is replaced by new geomagnetic sensor, vehicle detectors will have more advantages such as small size, high sensitivity, small construction volume, long life, small damage on the road, etc.
- Adopting embedded ZigBee wireless communication technology which includes hardware of printed circuit board and embedded software, it will have some features such as ad hoc networks, automatic-restoration, automatic search for best route, etc.
- Realizing integration of wireless traffic monitoring system according to the concept of IoT integration, it includes that integration of hardware's chips, design of embedded software, development of monitoring center, and then we can upgrade existing intelligent transport system from network, intelligence and integration all-round.
- Design four-level information release strategy in this traffic guidance system, propose design concept and development plan of wireless traffic guidance system. It integrates of traffic test and wireless communication technology from hardware and software, and uses four-level guidance LCD to realize intelligent parking guidance. Thus providing an effective solution for the realization of a modern intelligent transport system.
- Construction of online information service platform, drivers can check traffic condition through logging into this platform by using computers and mobile phones. Realizing real-time convenient, fast and smooth information sharing in the Internet era.

ACKNOWLEDGMENT

The work in this paper is in part supported by the Education Department Foundation of Hubei Province of China under Grant No. B2013262 & No. B2013259.

REFERENCES

[1] Yang C, Li Y. Reasearch on traffic congestion of big cities in out country[J]. Journal of Beijing University of Civil Engineering and architecture, 2007, 23(4): 62-64

[2] Gu Z, Tao H. Decision-making research on city traffic management planning[J]. Hebei Jiaotong Science and Technology, 2005, 2(1): 1-5

[3] Theodore T, Nikolas G. City size, network structure and traffic congestion[J]. Journal of Urban Economics, 2013, 76(3): 1-14

[4] Jiang Y. Research on causes and contermeasures to urban traffic congestion based on game thory[J]. Journal of Lanzhou Jiaotong University, 2007, 26(1): 112-114

[5] Yang X, Xue K, Bai Y. A study of the structure of parking guidance information system[J]. Journal of Tranportation System Engineering and Information Thchnology, 2004, 4(1): 93-96

[6] Jie J. The research on the urban parking guidance system based on the technology of "The Internet of Things"[D]. Beijing: Beijing University of Posts and Telecommunications, 2011.

[7] Wang Z, Fan Y, Mao E. Study on parking guidance system of city[J]. Computer Engineering and Design, 2006, 27(2): 188-194

[8] Chen G C, Lee S Y. Evaluation of distributed and replicated HLR for location management in PCS network[J]. Journal of information Science and Engineering, 2003, 19(1): 85-101

Distributing monitor system based on WIFI and GSM supporting SCPI

Xingyue HAN
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha, P. R. China
e-mail: 18684919453@163.com

Chunhui ZHAO
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha, P. R. China
e-mail: 412755830@163.com

***Abstract*—Internet of things (IOT) has become one of the most promising technology of the 21th century. In this paper, we combine the WIFI and global system for mobile communication (GSM) and build a distributing monitor system supporting standard commands for programmable instruments (SCPI), in which the WIFI is used to construct a local communicating network; the GSM is used for long distance communication, and the command of the system is based on the SCPI. The presented system can be widely used, such as in domain of smart home and environment monitor. It also provides a reference for relative design.**

Keywords- internet of things; state monitor; WIFI; GSM; SCPI

I. INTRODUCTION

Are you still worried about the safety of your house when you go out for a visit? Do you still frequently go to your newly decorated house to test the toxic air pollution? Now, we provide a solution to liberate you from them. In the paper, we combine the WIFI technology, GSM and the SCPI, designing a distributing monitoring system, which can be used to monitor, alarm and etc. For example, is the windows being opened, is there a moving object in the room. If one of these happens, the system can send you a message as an alarm. When the index of air pollution decreases to a predefined level, the system may also give you a cozy tip: dear, you may move into your new house now.

II. SYSTEM DESIGN

WIFI [1] is a wireless connection technology which can easily construct a fare free network in a relatively small area. Based on GSM [2, 3], the global system for mobile communication, we can easily set up the communication in a long distance. SCPI [4，5], the standard commands for programmable instruments, is a set of standard commands based on the IEEE488.1 and IEEE488.2, has been widely used with the characteristics of simple rules, easily to be remembered and used etc. The presented system is based on the three technologies. The basic designing idea of the system goes as follows.

The system is constituted of some blocks. In each block, a build-in processor is the center of the mini system, and its IOs are used for general programmable inputs and outputs (GPIOs). Analog-to-digital converter (ADC) is used for sampling the analog voltages, the output of the sensors. WIFI is used to build a local network, connecting the blocks. The GSM is used for data (sending the state of the system, sending the sampling datum, receiving the commands and etc.) exchange, and the data (command) is based on the SCPI.

A. SYSTEM ARCHITECTURE

The system is comprised of several blocks, as shown in figure 1. It contains four blocks: block 0, block 1, block 2 and block n. However, it may not include just four blocks, the number of the blocks can be changed as needed. There are two control platforms connected to the system. Each of the two control platforms can communicate with the system. The number of the control platforms is not two only. Only platform supporting GSM and SCPI, such as mobile phone, a personal computer (PC) is possible.

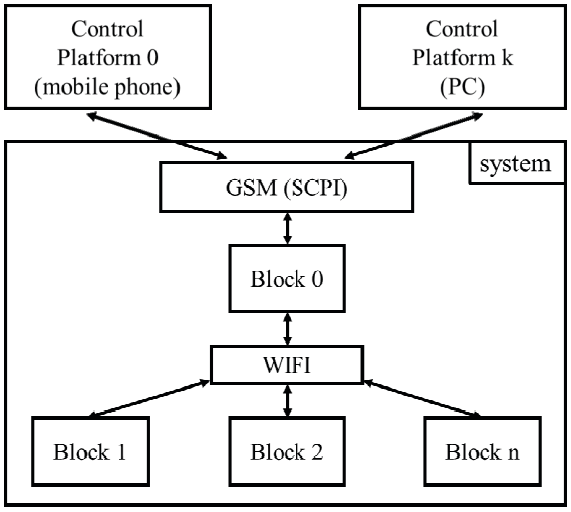


Figure 1. Architecture of the distributing monitor system.

The blocks are connected via the WIFI, and all the sampled datum are sent to the block 0 based on the WIFI. The block 0 receives the commands from the control platform via GSM and decodes the commands (SCPI), and then send the data to the outside. If the designed rules are satisfied, a message can also be sent automatically to the platforms via the GSM. The GSM data transmission is based on the message. The system can be controlled by a mobile phone, a PC or something else, as long as it can support the GSM message and strings satisfying the SCPI.

B. BLOCK STRUCTURE

The structure of the block is shown as figure 2. The block's core is a processor, which may be a digital signal processor (DSP), a single chip microprocessor or a field

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.59

programmable gate array (FPGA). An ADC is used for data sampling. The processor's IOs are used as GPIO. The input of ADC and GPIO is combined, forming the sensor interfaces, which is used to connect the sensors. We have to declare that we don't specifically define the sensors' interface, so it can be defined by users, as needed. Each block contains a WIFI sub-block, used for connecting with other blocks and data exchange. A GSM sub-block may also be included if needed, used for message receiving and sending. A UART is designed for each block, used for block debugging and data transmission to a local computer. The SCPI command is decoded in the processor. Generally, each block, to meet the minimum need, LEDs, keyboard and switches are usually required.

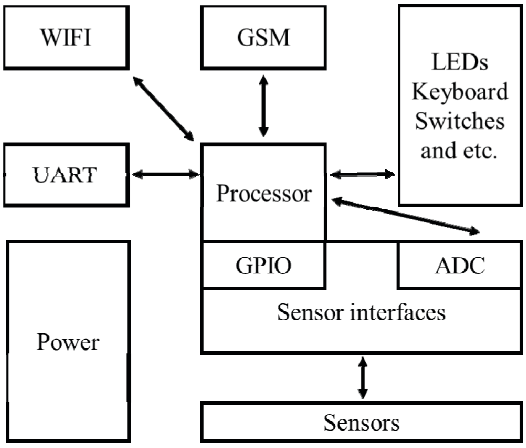


Figure 2. Structure of a block.

C. SOFTWARE ARCHITURE

The main function of the embedded software is to sample the signals, which are the output of sensors. And then, transmit the data to the processor to compute and analyze. Eventually it will display the data on screen and send a message to control platforms to notify or alert the users. The structure of the embedded software is shown as figure 3.

In operation, all of first, the interrupts will be disabled and the system will be initialized, including all the registers, RAMs, timer, setting of the interrupt etc. After the initialization, the processor is waiting the digital signal which is from sensors and digitalized by ADC. And then, the processor will compute and analyze the data. Finally the system will notify or alert the users of the value of the measured quantity.

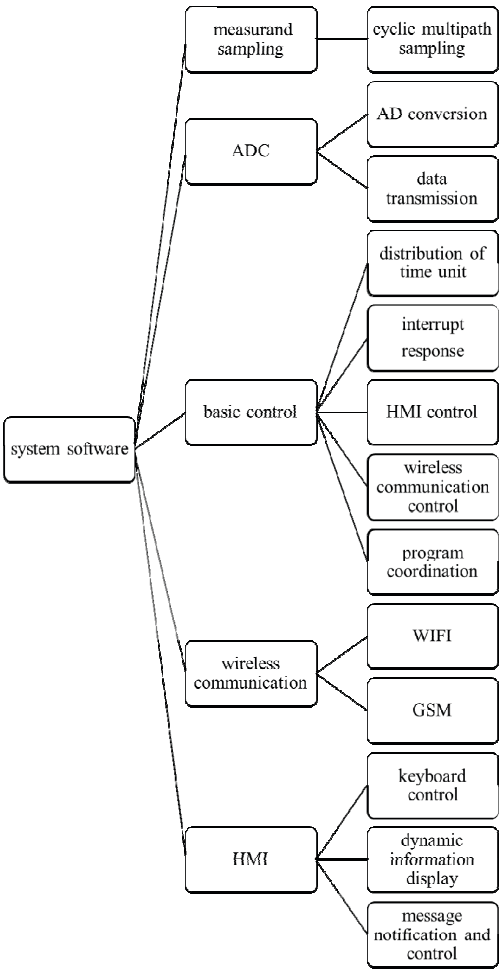


Figure 3. Architecture of the embedded system software

III. EXPERIMENT AND RESULTS

Based on the system architecture, software and block structure, we construct a test system as figure 4. In the system, there are two blocks and two sensors. One block is in the areas B2 and B3. It connects a sensor in area C3. It also connects a GSM sub-block at the lower-left corner of the area B2. A SIM card is inserted into the GSM sub-block. The WIFI sub-block is located at the upper-right corner of area B2. The other block is in the area B1, and it connects a sensor on area C1.

Actually, the circuits of each block are nearly identical. Hence, we utilize one of the blocks, known as the master block, to develop the wireless network (WIFI), and other blocks, known as the slaves, are required to join the WIFI network developed by the master block.

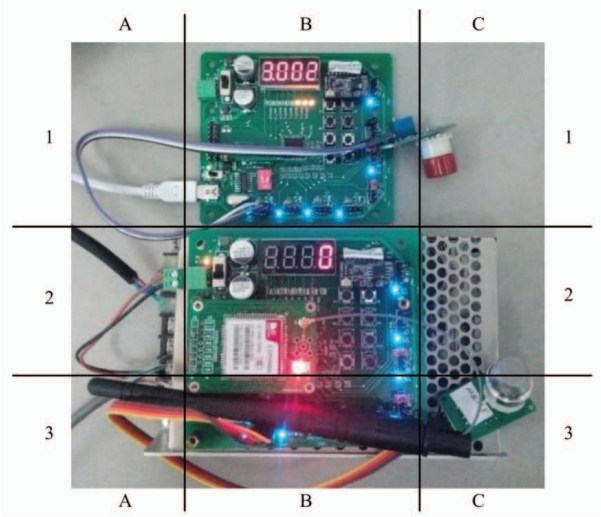


Figure 4. The designed experiment system.

Before the operation, we define the keys as key1, key2…key8 from left to right, respectively. The following is the function of each key. (You can redefine the function as needed). And we set the AD value from 0 to 5V.

Key1: reset the system
Key2: GSM power bottom (generally, it is on)
Key3: WIFI sync signal (manual data sync)
Key4: delay counter reset
Key5: send a message to users instantly
Key6: change the display form on screen
Key7: display the value of the AD value on screen
Key8: undefined

When you turn on the system, it will experience a self-checking period (about 5 seconds), during which time, we can see “8.8.8.8” is displayed on the screen. And under the screen are LEDs, which are flashing periodically, from 1 to 8 in form of the binary code. After that, all the settings are initialized. Hence all the datum will be transmitted automatically.

In operation, if a key is pressed, the number of the key will be displayed on screen for just a few seconds, and a corresponding function will be achieved. If the number don’t disappear, which means there is something wrong with it, press Key1 to reset the system.

If there is a number in form of “1xxx”, it denotes an alert message has been sent out just now and the new alert message can’t be sent out until the delay counter decreases progressively to zero, to avoid the wrong operation. Surely, the delay time can be changed as needed.

When certain measured value exceed the threshold, the system will automatically send you an alert message in form of “V1 is too high, the AD value V1=1.234V”. If we send a SCPI based message “read i” to GSM sub-block, we can also receive a message like “AD value Vi=0.02V”. Also, you can send a message “reset” to the GSM sub-block to reset the system.

Several tests are executed. We find, in ideal condition, the valid range of WIFI network net is about 50m.

The experiment results show that the system can fulfill our predefined function, so the scheme of our distributing monitor system based on WIFI and GSM supporting SCPI is feasible.

IV. CONCLUSION

In this paper, we have presented a distributing monitor system which can be controlled by a message and auto report the alarm. The system is also based on the international standard SCPI which owns good compatibility and interchangeability. Thus, the system can be used in not only the smart home, but also the detections of landslide and bridge safety and etc. which shows promising application prospect.

REFERENCES

[1] A. Zafft and E. Agu, "Malicious WiFi Networks: A First Look," presented at the 7th IEEE Workshop on Security in Communication Networks 2012, 2012.

[2] C. D. Oancea, "GSM infrastructure used for data transmission," presented at the 2011 7th International Symposium on Advanced Topics in Electrical Engineering(ATEE 2011), 2011.

[3] H. F. Qi, X. H. Yang, R. Jiang, B. Liang, and S. J. Zhou, "Novel End-to-End Voice Encryption Method in GSM System," presented at the Proceedings of 2008 IEEE International Conference on Networking, Sensing and Control (ICNSC 2008), 2008.

[4] F. Bode, "Introduction & UPDATE to an open standard for instrument control: SCPI standard commands for programmable instruments," presented at the Wescon '92, Anaheim, CA, USA, 1992.

[5] Z. Zhu, H. Zhao, and L. Shen, "The application of structure arrays and files in the SCPI parsing system," presented at the 2010 International Conference on Intelligent Computation Technology and Automation (ICICTA 2010), 2010.

Research on Personalized Indoor Routing Algorithm

Weijun Bian, Yucheng Guo*, Qizhi Qiu
School of Computer Science and Technology
Wuhan University of Technology, Wuhan, China
bwj1990032368@gmail.com; ycheng.g@gmail.com; 736019272@qq.com

Abstract—As the mature GPS positioning technology cannot work well in indoor environment, there is no mature indoor navigation system for civil use. Besides indoor maps and indoor positioning technology, indoor routing algorithm is an important part of indoor navigation technology. This paper first discusses the problem of cross-storey in buildings and gives a solution. Unlike outdoor routing, shortest path is usually not the best path indoors, a personalized path considering user preference and interest can be better. To achieve personalized routing, this paper comes up with a way to model and acquire user preference. On this basis, this paper improves traditional A* algorithm by considering user preference and then gives an example to show the advantage of the personalized A* algorithm. The results show that the improved algorithm can find a better path by considering path length and user preference synthetically.

Keywords—indoor navigation; routing algorithm; user preference; personalized.

I. Introduction

With the development of the society, increasingly, there are more and more buildings with large scale and complex internal structure and the demands for the indoor navigation service are also increasingly urgent. Although in the past few years, electronic maps and navigation software had made great processes, the research on indoor navigation still progressed at a slow pace. The mature GPS positioning navigation system can not work indoors because it can not receive signals. Because there are bottlenecks in the indoor positioning technology, most studies on indoor navigation are focusing on the robotic navigation and there are no mature civil indoor navigation products by far[1-5]. Meanwhile, in the era of network, information and digital things, personalized technologies have been used in various fields. Because there are bottlenecks in the indoor positioning technology, this technology has not been well used in the indoor navigation field. However, it will be meaningful to use personalized indoor navigation according to users' interests in today's high-paced life. Personalized indoor navigation service should not only be based on the technologies of indoor maps and indoor navigation, but also be supported by the personalized indoor routing algorithm. This is also the focus of this paper.

II. Cross-storey problem

The problem that the routing algorithm needs to solve is to find the path between the two points in the graph and this path is usually the shortest path between these two points. The common algorithms are Dijkstra algorithm, Floyd algorithm, Bellman-Ford algorithm, A* algorithm and so on. These algorithms can be well used in the outdoor routing. However, these algorithms cannot be directly used in the indoor routing because there are so many storeys in the building[6].

a. First of all, we should consider the circumstance that we compose all the needed graphs to make a big connected graph and at that time the topological structure of the graph need to keep changing. By doing this will spend much more time in changing the topological structure which will make the operational efficiency become low and the coding will be very difficult, too.

b. We can connect all the storeys through the stairs to make a bigger connected graph. In this connected graph we can inquire any shortest path between two points without changing topological structure of the graph. However, when using the blind search algorithms such as the Dijkstra algorithm, the execution times of the algorithm will be greatly increased.

c. When considering the stairs as one side of the connected graph, the weight will not be well determined. When the weight is too small, while the heuristic algorithms such as the A* algorithm are searching paths, they will take the first path it finds as the path without considering other paths which is obviously unreasonable. On the other hand, when the weight is too big, the algorithms will not to choose the stairs, which will make it finding no paths.

We've considered that when users are going to different floors, they will usually use the same stair, especially when users are using shuttle elevators. Therefore, this is our solution to the cross-storey problem.

(a) To find the path between the position of the user and one of the stair in the building.

(b) To find the path between the corresponding stair in the target floor and the target position of the user.

(c) Take the paths (a) and (b) have found as a feasible path.

(d) Repeat (a) to (c) and find the feasible paths for every stair.

(e) Compared all the feasible paths, which have been found before and then choose the best one.

Therefore, the cross-storey problems have become the routing problems between two points. After getting all the paths when users choose every stair, we can get a best path by comparing these.

III. Personalized routing algorithm

Traditional routing algorithm is used to find the shortest path between two points and it is fully applicable for the outdoor navigation routing, which can arrive at the destination as quickly as it can. For some indoor places such as the airports, the train stations, the traditional shortest routing can be also very effectual. However, for the more common

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.60

scenarios in people's daily life, the shortest path doesn't mean the best path. For example, when users are in a big shopping mall, if there are much more places that the users interested in on one path, he can pass through these places when he choose this path, sometimes he can even find other more places that he interests in. This will not only save users' time in another way but also enhance the user experience of the indoor navigation. Even though this path is not the shortest one, it is a better one[7]. Therefore, it will be meaningful to use personalized indoor navigation to find a best path for users according to users' interests.

A. Preference modeling

Preference modeling is the process that generalizes the measurable modeling from the information about users' preference and behaviors. The accurate descriptions about users' preference cannot be modeling. Preference modeling is not the common description about users, but a kind of users' formalized description, which faces to algorithms and have specific data structure [8].

In this paper, we choose to use the notation, which is based on the vector space model. It is a way, which uses the vector in the key word vector space to express the user models, and it depends on the particular cases. The main basis is to be determined by the places in the building that the users are interested in. if users are in a shopping mall, the subject terms of their interests may be determined as entertainment, food, sport and so on. So users' interests can be expressed as a vector of subject terms: $\mathbf{U}=\{k_1, k_2, k_3,...k_n\}$, k_n means the weight of the nth subject term. The weight will choose a number between 0 and 1 according to users' preference. On the other hand, the contents on each position are expressed by an n-dimensional vector to express the content features of the position. By calculating the similarities of these two vectors, we can know the preference of users. The calculation formula of the similarity is as follow, among them, **m** expresses the preference feature vector of users and **n** expresses the content feature vector of the position:

$$sim = \frac{\vec{m} \bullet \vec{n}}{|\vec{m}||\vec{n}|} \tag{1}$$

B. Preference acquisition

The acquisition of users' preference is based on the communication between system and users. There are two kinds of personalized feedbacks: the implicit feedback and the explicit feedback. The explicit feedback needs users to take part in it directly which means that users should provide his information by themselves and give evaluations to the present program and system in order to get the preference. The implicit feedback doesn't require users to provide their information and all the tracks are done automatically by the system. Implicit feedback can be got by tracking users' browsing histories and checking the users' behavior log.

This paper uses the notation of the vector space model to express users' preference. We use user action logs to inspect the positions that users used to take as the destinations because the influences on the present position information are not so big. We can suppose that the destinations are $\mathbf{l_1}, \mathbf{l_2}$ ..., $\mathbf{l_n}$,the times that take the places as destinations are $\mathbf{t_1}, \mathbf{t_2}$ ..., $\mathbf{t_n}$, we can get users' preference vector by using the weighted sum of the destination vector. If $\mathbf{T}=\mathbf{t_1}+\mathbf{t_2}+...+\mathbf{t_n}$, then the calculation formula of users' preference is as follow:

$$\vec{m} = \sum_{i=1}^{n} \left(\frac{t_n}{T} \vec{l}_n \right) \tag{2}$$

When users use the system at the first time, they cannot get the preference vector because they don't take any places as the destination. At this moment users can get the preference vector by answering the multiple-choice questions that offered by the system. For example, if the choices of one-subject terms are hate, fair, like, enjoy, we can ensure that the weight of this subject term is 0, 0.4, 0.6, 0.8 according to users' choices. Users will get a preliminary user preference vector **s** after answering every question. As users' behaviors come into being, that is to say, to navigate by taking some positions as destinations, we can get a preference vector **m'** by the using the ways explained before according to these positions. Then we can get the present preference vector by using the weighted sum of the two vectors, which is $\mathbf{m}=\mathbf{w_1 s}+\mathbf{w_2 m'}$. Weight $\mathbf{w_1}$ and $\mathbf{w_2}$ will be determined by the specific circumstances. Because the preliminary user preference vector **s** can not reflect the users' preference and interests well, when the times of navigation reach some scale, we can totally use **m'** to determine the user preference and at this time **m**=**m'**.

C. The personalized A* algorithm

This paper uses the improved A* algorithm to solve the routing problems between two points. The A* algorithm has defined the evaluation function $\mathbf{F}=\mathbf{G}+\mathbf{H}$ in order to find the shortest path. Among which **G** expresses the consumptions which have been produced between the start point and the right now point, and H expresses the forecast consumption which will be produced between the right now point to the final point [9]. H is a forecast value and has many ways to be defined. This paper chooses to calculate the distance from the present coordinate (**x**, **y**) to the target location coordinate ($\mathbf{x_1}$,$\mathbf{y_1}$). The calculation formula is as follow:

$$H = \sqrt{(x - x_1)^2 + (y - y_1)^2} \tag{3}$$

In order to find a personalized routing, this paper has improved the evaluation function. The value of **G** has changed from the consumptions which have been produced between the start point and the right now point to the consumptions which is related to users' preference value, that is :

$$G = \sum \left((1 - sim_i) D_i \right) \tag{4}$$

$\mathbf{D_i}$ expresses the consumption weight of this path while $\mathbf{sim_i}$ expresses the cosine similarity between the content feature vectors of this path's destination and users' preference features vector. When the destination of this path is just at the target position, **sim** is 0, which means that there's no need to consider the similarity between destination and users' preference. To begin the algorithm, two arrays should be built. The CLOSED array saves the nodes, which have been inspected; The OPEN array

saves the nodes, which have not been inspected. The pseudo code of the algorithm is as follows:

Algorithm 1 Personalized A* algorithm

```
while(OPEN!=NULL)
//when OPEN is empty,the path to target place doesn't exist
{
n=node which has the lowest F in OPEN;
if(n==TargetNode)//find the path to the target place
break;
for(each adjacent node x of n){
if(x is in OPEN){
compute F' and G' of x //compute new G and F of node x
if(G' < G){//this path is better
parent node of x=n;//set n to the parent node of x
G=G';F=F';//refresh values of G and F for node x
}//end if
}//end if
if(x is in CLOSED)
continue;
if(x is not in both){
parent node of x=n;//set n to the parent node of x
compute G and F of x;
Insert x into OPEN;
}//end if
}//end for
remove n from OPEN;
insert n into CLOSED;
}//end while
```

D. Evaluation of the algorithm

In the connected graph showed as Fig 1, we can assume that the estimate value H between $\mathbf{L}_i$(i=0,1...,6) and the end point is respectively 7、6、5、4、3、2、1 and the similarity $\mathbf{sim}_i$(i=1,2...,6) between destination and users' preference is respectively 0.6、0.4、0.4、0.7、0.4、0.5. The weight of the path length has been marked on the figure.

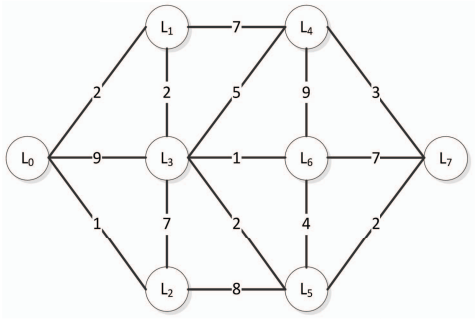


Fig. 1. Connected Graph

When using the traditional A* algorithm, the path we can find is L0->L2->L5->L7, this path is the shortest path and the sum of the length weight is 11. While using the personalized A* algorithm, the sum of the length weight is 12 which is greater than the shortest path. However, because during the way users will pass through the L1 and L4 which have high similarities to user preference, if considering the path length with weights of user preference we can get the sum as 5.9 which is less than 7.4, the sum of the path which traditional A* algorithm finds. Therefore, the path found by the personalized A* algorithm is considered as a more coincident path to users' preference.

This kind of algorithm has considered both path length and users' preference. The algorithm will search the target position because the **H** in the estimating function is the distance between the user's position and the target position. We need to consider two extremes. When their cosine similarities are similar to each other, the algorithm will choose the same path which the traditional A* algorithm will choose. If the differences of the consumption value were little and the differences of the sim value were big, the algorithm will choose the path, which is more suitable for users' preference. For other situations, the algorithm will consider both the path length and the preference in order to choose the best path.

IV. Conclusion

This paper has studied the personalized indoor routing algorithm. First of all, the paper puts forward the cross-storey problem in the indoor routing and provides the solution to the problem. By doing this, the paper has turned the cross-storey problems into routing problems between two points. Second, the paper models the users' preference and uses space vector to express users' preference and position's content. On this basis, the paper improves the A* algorithm in order to make the routing more personalized and finally reach the aim of using personalized indoor routing.

REFERENCES

[1]Lei Cao. Study of the Location and Navigation Services in Complex Indoor Scenes Based on Android Mobile Computing Platform[D]. Wuhan University of Technology. 2013

[2]YuCheng Guo, Lei Cao. Study of the Location and Navigati on Services in Complex Indoor Scenes Based on Android Mob ile Computing Platform[C]. DCABES 2012,Guilin,2012:91-93.

[3]Han Changyong, JAEGEOL Y, GYEYOUNG I. Implement ation of Web Services for ILBS[J]. Communication in Comput er and Information Science, 2011(261):415-422.

[4]HUANG Haosheng, GARTNER G. A Survey of Mobile ind oor navigation systems[J]. Lecture Notes in Geoinformation an d Cartography, 2013(3):305-319.

[5]Jain, M. ,Bangalore, Rahul ,Tolety, S. . A study on Indoor n avigation techniques using smartphones[J]. Advances in Comp ting, Communications and Informatics(ICACCI), 2013 Interati onal Conference on 22-25 Aug. 2013:1113-1118.

[6]Shaohua, Luo. Research on the Navigation Algorithm[D].B eijing University of Posts and Telecommunicaions.2011.

[7]Dudas, P. M. ,Ghafourian, M. ,Karimi, H. A. .ONALIN: On tology and Algorithm for Indoor Routing[J]. Mobile Data Man agement:Systems,Services and Middleware, 2009.MDM '09. T enth International Conference on 18-20 May 2009:720-725.

[8]]Li Chun, Zhu Jianmin, Yejian, Zhou Jiayin. Survey on rese arch inpersonalization service[J]. Application Research of Co mputers, 2009,26(11):4001-4009.

[9]Liu Hao, Bao Yuanlu. A* Algorithm for Finding the Optim al Path on Vector Maps[J].Computer Simulation,2008(4):253-257.

* Yucheng Guo, Corresponding Author

Making a Virtual Sand Table Based on Unity 3D Technique

Liu xiaofeng
Dept of Computer Application
WuHan University of Technology
WuHan, China
xiaofengtop@126.com

Chen Tianhuang
Dept of Computer Application
WuHan University of Technology
WuHan, China
thchen57@126.com

Qin tingchao
Dept of Computer Application
WuHan University of Technology
WuHan, China
429509506@qq.com

***Abstract*—**The 3D virtual sand table is a kind of relatively complicated technology with the very important point that the map datum should generate rapidly. Based on DEM datum , This article is devoted to come up with a systematic method to structure a 3D virtual sand table and identify its feasibility : a grayscale-map generates by means of linear interpolation in a short time and a vivid 3D-topographic map is constructed in Unity3D;the feasibility of the method is proved by a experimental way .

Keywords- Unity 3D; virtual sand; GIS

I. INTRODUCTION

With the prominent specialty in virtually displaying 3D spatial structure， ZHANG Dian — hua，CHEN Yi — min [1] , Zhu Huijuan [2] has applied Unity3d to the virtual campus areas and made a detailed study. Zhanggao Wei, Wang Tingting, Zhang Fuqiang [3] have built three-dimensional GIS model based on Unity3D environment, focusing on the attribute data and spatial data organization and management, virtual sand-table is widely applied in estate, military field, water conservancy and transportation, Compared with the traditional sand table， virtual sand table has the advantage in manipuility maintainability and it's also easy to be produced. With the development of the technologies in the virtual reality virtual sand table is going to be applied in more and more fields.

Unity3D is a game engine developed by the Unity Company of Denmark, which is more prominent in the three-dimensional rendering capabilities, detail and strong performance (see Figure 1), supports 3DMAX, MAYA, Sketcheup, CAD, and other three-dimensional model, and integrates audio, network, physical processing module. Based on the scripting language of C # and Java, Unity 3D can create executable program facing on Windows Andriod, Web, Flash, IOS and so on [4]. Due to its powerful graphics processing, scalability and ease of development, etc., is widely used in the development of games and virtual reality.

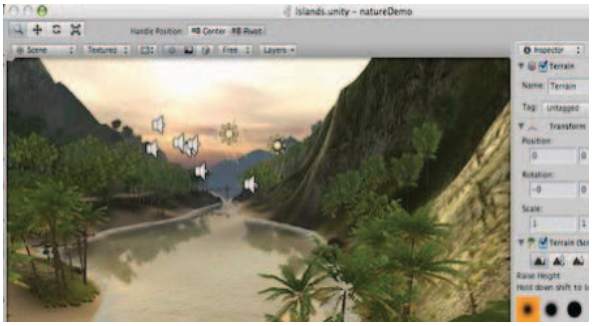


Figure 1, Unity3D examples of virtual reality development

II. OBJECT RECOGNITION METHOD BASED ON FUSING MULTI-FEATURE

A. The analysis on Unity3D terrain data forma

Three-dimensional surface model is a true three-dimensional simulation of the actual situation of the surface, truly reflecting undulating surface and so on, and also known as data set of three-dimensional surface or three-dimensional topography [5]. Open Unity3D, use brush pen tool to create the mountain [6], shown in Figure 2, and save the topographic maps.

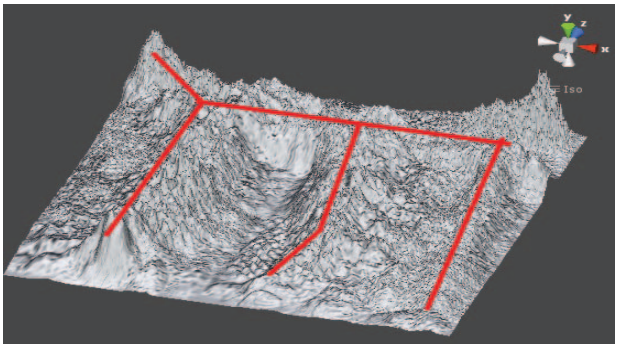


Figure 2, Constructing the mountain inUnity3D (Redlining the ridge direction)

This topographic elevation data is stored in an image file, the filename extension is RAW [7]. Open the RAW file in Photoshop, Figure 3, comparatively analyze the width and

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.61

height, definite the format by 2^n+1pixels. Analyze the gray value of the file in Photoshop , The maximum limit number of independent model triangle ,which Unity3D software can carry, is 65535. [8] The surface of the mountain was white, the whiter, and the higher.

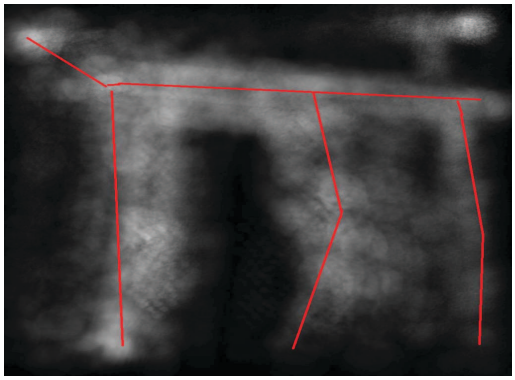


Figure 3, The grayscale identified in Photoshop (Redlining the ridge)

B. The analysis on Elevation raster data

"Three-dimensional spatial data" is the human visual geometric description of the objective world, the virtual expression of objective world: spatial distribution, location, and geometry. It reflects real-world phenomena and changes, as well as describes the space properties and time properties on various entities and phenomena on the earth [9]. The virtual sand-table technology uses the geographical area elevation data as the underlying data of a three-dimensional map . The Analysis is mainly used in Global Mapper as the representative of the terrain format. Since Unity3D just accepts the RAW date , we firstly need format the format of common date to RAW .The following is the example of Global Mappper for exporting the ASC-format elevation grid graphics.

Open the ASC in text format, and analyze its file data structure shown in Figure 4

```
ncols         987
nrows         865
xllcorner     111.4858506598
yllcorner     28.2218404170
cellsize      0.00083333333333334
nodata_value  -9999
49.088 45.843 42.867 40.299 38.48 38.216 38.503 39.169 :
40.953 42.181 43.888 43.971 43.444 43.801 43.328 41.451
33.997 31 31 37.243 50.675 67.941 72.743 67.746 71.839 :
108.891 113.801 110.515 107.886 112.322 117.354 113.979
```

Figure 4, ASC opened in text format file

The first line "ncols" is the number of data of rows; the second row "nrows" is the number of data of columns. The third and fourth lines are the latitude and longitude of the data points of upper left corner , the fifth line indicates a single point range; The sixth line is no-data-value-spot defined as -9999 . Treat this definition above as terrain gist. the seventh line to the last indicate the altitude date.

C. Implementing a virtual sand table instances by writting C# language codes

Write codes in C # program , read the ASC file and generate 16-bit grayscale RAW format files of the width and height of 2^n+1.

1) To read the file

The main code is as follows:

```
string filename;//filename is asc file name to open.
openFileDialog1.Filter="asc|*.asc";
if (openFileDialog1.ShowDialog()==DilalogResult.OK)
{filename= openFileDialog1.FileName;
}// Read the file into ASC file.
Streamreader ascfile=new streamreader(filename);
string line;//Read the entire row of data.
line=ascfile.readline();
//Decompose an entire row of data into a single string.
string[] ascstring=line.Split （ new char[ ] {} ,stringSplitOptiongs.RemoveEmptyEntries）;
```

2) The calculation of the interpolation data

In Unity3D ,1 unit indicates 1 m. Remove the single data element individually, and calculate the gray value of all the points in the 2^n+1 square by method of interpolation .

The expression of the formula for gray value is $n=\frac{t}{m}\times 65535$, n is a gray value, t is the height value extracted, m is the height value of the highest point (maximum elevation value in the figure or the maximum altitude value 8848 of surface of the earth can be taken as m), 65535 is the maximum unsigned 16-bit representing value , so the altitude value can be translated into hexadecimal grayscale in 0-65535.

Calculate the gray values of all points in 2^n+1 square by methods of interpolation . The linear interpolation formula is:

$$p(x)=f(x_0)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0) \qquad (1)$$

About the calculation of the linear interpolation error ,SONG Shi-cang ,CHEN Shao-chun , SUN Hui-xia [10] have made a thorough study. According to the error range of Rolle's Theorem $|R^r| \le \frac{(x_1 - x_0)}{8} \max_{x_0 \le x \le x1} |f''(x)|$,then the greater the curvature between two close points of the mountain body ,the greater the error of the interpolation between the two points.

The below is the Gray value algorithm code :

int nrows, ncoles ;/ / define the the number of data of rows and columns.

coles = Convert.ToInt32 (lines [0,1]);

nrows = Convert.ToInt32 (lines [1,1]) ;/ / the ASC file number of rows and columns is taken out as a cycle measurement.

ArrayList grey = new ArrayList () ;/ / generic grey is used to store the gray values.

for (int i = 0; i <nrows)

{for (int j = 0; j <ncoles; j + +)

{double t ;/ / define the altitude value stored.

int n ;/ / define the gray value corresponding to t .

/ / altitudestring is the altitude strings stored.

t = Convert.ToDouble (altitudestring [i, j]);

n = (UInt16) ((t / m) * 65535);

grey.add (n);

}

}

Extends the grayscale with the width w and the length h to the 2^n +1 square matrix figure, the value for n should satisfy $w \le 2^n + 1 < 2 \times w$, and h 2^n +1 <2 × h. Firstly define the interpolation points. The 2^n +1 code is:

int resolution = 2;

while ((resolution <w) | | (resolution <h))

{resolution = resolution * 2;

}

resolution + = 1;

Calculate the interpolation in the original two corresponding point . Suppose interpolation point you want is x, it corresponds to the original position x' which should be between x_0 and x_1 .

$x' = (x / \text{resolution}) \times w$. $x_0 = x'$ Rounded, $x_1 = x_0 + 1$. Then substituted into the interpolation formula

$$p(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$ calculated p(x) that is $f(x')$. The main code is as follows:

/ / the length is "resolution" , do the extended interpolation along the x-axis direction.

int x0, x1;

UInt16 [] p = new UInt16 [resolution];

for (int i = 0; i <resolution; i + +)

{ x0 = (int) (i / resolution * w);

x1 = x0 +1; / / f [] array is the value the original grayscale stored (one-way).

p [i] = Convert.ToUInt16 (f [x0] + (f [x1]-f [x0]) * (i / resolution * w-x0));

}

/ / Do "resolution" cycle interpolations along the x-axis direction , then the needed gray values of the matrix diagram can be generated .

3) Generating grayscale

Store the gray values calculated of all interpolation points into a file,save it as RAW format file. The main code is as follows:

int resolution ;/ / the value of "resolution" is 2^n +1 which can be understood as the resolution value.

int resolutionsquare = Math.Pow (resolution, 2);

FileStream rawfile = new FileStream ("D: \ \ test.raw" , FileMode.Creat, FileAccess.Write);

BinaryWriter writer = new BinaryWriter (rawfile);

byte [] b = new byte [resolutionsquare * 2] / / store a gray value by two bytes.

```
for (int i = 0; i <resolutionsquare; i + +)
  {/ /  store the gray value in the array grey [i] .
  byte [] bs = ToByte (grey [i], sizeof (ushot));
/ / operate by bytes , the storage order can be checked whether normal or not in debugging .
  b [i * 2] = bs [0];
  b [i * 2 +1] = bs [1];
  }
writer.Write (b, 0, b.length);
rawfile.Colse ();
```

Open the resulting grayscale in Photoshop to check whether it is normal or not, shown in Figure 4.

Figure 4, Using Photoshop to check the Generated grayscale whether correct or not

4) Importing the map

Import the generated RAW terrain-map into Unity3D, shown in Figure 5.

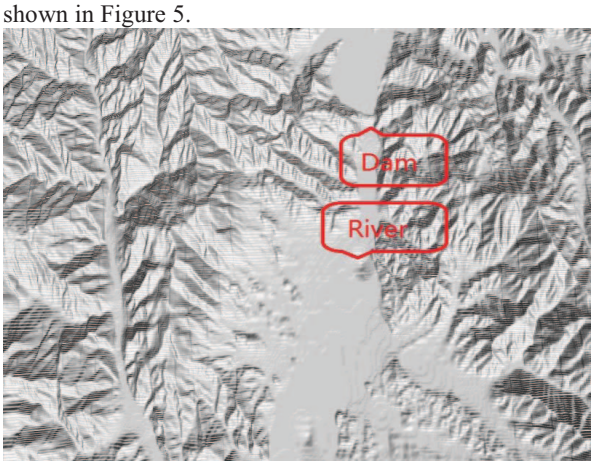


Figure 5, To import RAW format map in Unity3D

III. CONCLUSION

Unity3D engine promotion in our country is very fast. With its easy to use, multi-platform and rich physics rendering, it has won many programmers' love. From a technical point of view the development of such a virtual-reality system, the methods of using geographic raster data conversion in this paper will greatly improve the efficiency of development. With the advances in computer technology and virtual-reality technology, the establishment of a reality-based virtual sand table will come true, and it's not in a long time that the sand table would finally be remotely available by network computing and Internet technologies ultimately.

REFERENCES

[1] ZHANG Dian－hua, CHEN Yi－min.Design and Implementation of Multi － platform Virtual Campus Based on Unity3D, Computer Technology and Development, February.2014,pp.127-130.

[2] ZHU Hui-Juan. Virtual Roaming System Based on Unity3D, Computer Systems & Applications,vol 21, February.2012,pp.36-39.

[3] ZHANG Gao-wei1, WANG Ting-ting, ZHANG Fu-qiang. A New A pproach of Realize 3D G IS in Testing G round, SOFTWARE, August.2012,pp. 44-47.

[4] Unity3d, Unity3d Software, [http://unity3d.com/]. Accessed,25.06. 2013.

[5] Arens Calin,Stoter Jantien,van Oosterom Peter, Modelling 3D spatial objects in a geo-DBMS using a 3D primitive.Coputers and Geosciences, vol. 294,2005:165-177.

[6]Wang Hai. Creating a simple terrain, [http://blog.csdn.net/ pleasecallmewhy/article/details/8519577].Accessed,25.05.2014.

[7]MA Gong-li, YANG Ming, ZHI Xiong-fei, ZHOU Peng, MA Xiu-shui. The Modeling of 3D Seabed Terrain Based on Unity3D. Journal of Anhui Vocational College of Electronics & Information Technology, June.2013,pp. 24-27.

[8] HUANG Yang, WANG Yang,etcThe Study of Virtual R eality Scene Making in Digital Station Management Application System Based on Unity3D. Geomatics & Spatial Information on Technology, June.2013,pp. 50-51.

[9] K.Hinrichs,J.Nievergelt,The grid file:a data structure to support proximity queries on spatial objects.Proc.of the WG'83(Intern.Workshop on Graph Theoretic Concepts in computer Science), 1983,pp.100-113.

[10] SONGShi-cang ,CHEN Shao-chun ,SUN Hui-xia. Some Intensive Estimations for Linear Interpolation and Application in Finite Element Method.Mathematica Applicat,January.2003,pp.92-9

A novel solution: Building Real Time Monitoring System on Oceanographic Research Vessel in Distant-Water based on Beidou and Digital Ocean

Wei wang
Institute of Remote Sensing and Digital Earth
Chinese Academy of Sciences
Beijing, China
wangw@irsa.ac.cn
National Marine Data & Information Service
State Oceanic Administration
Tianjin, China

Wenfang Chen
Polar Research Institute of China
State Oceanic Administration
Shanghai, China
chengwenfang@pric.gov.cn

***Abstract*—It is very necessary to develop a monitoring system on oceanographic research vessel with the rapid development of marine resource exploitation. While problems on how to realize data communication in far sea, save large amount of coordinates, extract data and display real-timely and so on are the keys to construct such a system successfully. A novel solution which employs Beidou SMS as the data communication components and utilizes China Digital Ocean Framework System to save or visualize data real-timely proposed in this paper. The success of the deployment of the Real Time Monitoring System of Oceanographic Surveying, developed by the State Ocean Administration, shows that the solution discussed in this paper is feasible.**

Keywords-Beidou; SMS; Digital Ocean; Monitoring System;

I. INTRODUCTION

With the land resources depletion resulting from human's devastating exploitation to nature, people look to the sea which amounts to 3/7 of the earth surface and development of marine resources have become one of the important subject of modern science and technology. While, to exploit marine resources, marine survey must be carried out firstly. The main oceanic countries all over the world, such as America, Japan, Britain etc., pay more and more attention to marine oceanographic investigation. Times of special or normalized investigation, carried out by State Oceanic Administration of China (SOA), show an overall upward trend year after year. At the same time, the scope of the surveying activities has been expanded from the near sea to the far ocean. More and more research ships sail to the center of the Pacific, the Atlantic, even the Pole where it is far from the shore and it is very insecurity because of special environment of ocean and bad climate. In order to strengthen the regulation and monitoring on Oceanographic research vessel, it is very necessary to construct a vessel monitoring system based on the technology of Geographic Information System (GIS) which can tell the regulator where the vessel is located in time and make the regulator and shipmen keep in touch whenever necessary (Berzins, 1989). Furthermore, real time data, obtained by the surveying, can be transmitted to the command center by this system without expensive communication cost.

For reaching these management targets, SOA puts forward a program to build a Real Time Monitoring System on all their Oceanographic Research Vessels to realize the efficiency, accuracy and security of oceanographic investigation. While, the key to construct such a monitoring system is how to solve the problems such as data communication link, saving of large amount of coordinates, data extracting and visually display and so on. Several technical solutions in related fields, using the Automatic Identification System (AIS), have been put forward at present. Torkild Eriksen proposed a solution to monitor Maritime traffic using a space-based AIS receiver (Torkild, 2006). Detsis also proposed a solution to solve the problem of combating illegal, unreported and unregulated fishing by constructing a vessel monitoring system (Detsis, 2012)(Knoska, 2008). However, the lack of Wireless Internet Service such as Global System for Mobile Communication Systems (GSM) make it is impossible to construct a monitor system in the Distant-Water like near the shore. Factors, such as expensive costs charged by AIS and fears over safety, make the scientific research institution that conducted the oceanographic survey would not adopt AIS as data link to implement data exchange between command center and vessels.

Optionally, the development and policy of Chinese GNSS System, which is known as BeiDou (or COMPASS) similar in principle to GPS, make it is possible to construct a monitoring system by a novel manner (Chengzhi, 2013)(Yu Longyang, 2012). At the end of 2012, the Beidou navigation satellite system consisted of 14 satellites, including five geostationary Earth orbit (GEO) satellites, five inclined geosynchronous orbit (IGSO) satellites (two in-orbit spares) and four medium Earth orbit (MEO) satellites. The current service covers China and part of the Asia-Pacific region with positioning accuracy of better than 10 m, velocity accuracy of better than 0.2 m/s and timing accuracy of 50 ns. The Beidou navigation satellite system with global coverage will be completely established by 2020, which will be a constellation of 35 satellites, including five GEO satellites and 30 MEO satellites. The main function of Beidou is the positioning, velocity measurement, one-way and two-way timing and short message communications (SMS) (Jin S,

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.62

2013). The specific characters of global coverage and SMS make it possible to build a data link real-timely.

At the same time, China Digital Ocean Framework System (CDOFS), also called China Digital Ocean Prototype System (CDOPS), has been constructed successfully just at last year. This system consists of three distinct layers, from the bottom up, these include the data, function, and application layers respectively (Zhang X, 2011). Through this system, ship's real time coordinates can be saved, extract and display in a virtual globe. Based on Beidou and CDOFS, a novel solution has been proposed in this paper.

II. REQUIREMENTS

The requirements for an effective vessel monitoring system that would benefit Oceanographic surveying activities would need to have the characteristics described in Table 1. Firstly, continuous, global coverage is necessary and will become absolutely essential in the future in Oceanographic surveying monitoring because of the trends that the position of investigation is farther and farther from the command center which is located in the shore.

Secondly, it is very valuable to the regulators of the investigation to implement textual data transmission real-timely no matter where the vessel is located in the globe (CHENG, F. L., 2009). On the one hand, Command Center can receive the status data of vessel such as position, speed, course etc. or even parts of investigation data real-timely. The vessel which is operation in the sea can be draw in the screen by the monitor system and varies data about the vessel can be queried by the regulators freely. On the other hand, people who worked on the vessel can also receive or send instant message conveniently. Specific location-related service information can be provided by the Command Center to effectively guarantee the safety of the investigation.

Moreover, the security of data transmission is also important. The investigation is confidential in most cases. Information such as the worked scope, course, and contents of received or sent message etc. must be encrypted before sent by the system.

The last but not the least, the cost of data transmission must be very cheap. Extra expenses should not be charged by the communication service provider.

TABLE I. REQUIREMENTS FOR OCEANOGRAPHIC SURVEYING MONITOR SYSTEM

Requirement	Purpose
1. Continuous, global coverage	Effective monitoring of Distant-Water and Polar regions
2. Real time data transition	Sending or receiving varies information by the regulators or shipmen
3. Secure data transition	Encrypted data can only be used by users of command center and authorized shipman
4. Low cost of data transition	Deployed without any other capital investment

III. SYSTEM COMPONENTS

As early as 1960's, the latest achievements of science and technology were applied to marine investigation (Olchi-Oglu, 1963). In order to realize the efficiency, accuracy and security, several latest devices and technology have been applied to the construction of monitoring system on Oceanographic Research Vessel. From physical point of view, just as depicted by Figure 1, this monitor system can be composed by three sections.

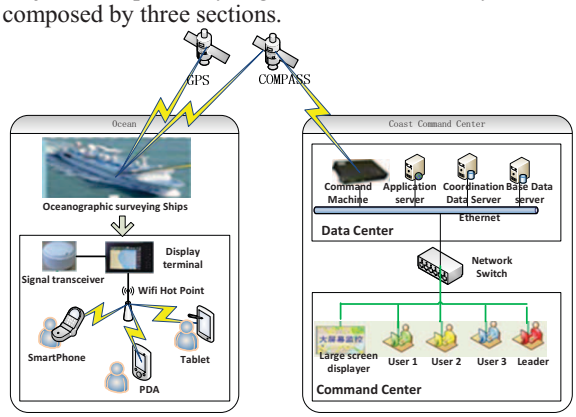


Figure 1. System components and interrelations

The first section is data communication link (DCL) which connects vessels surveying in the ocean and Coast Command Center (CCC) together. Real time data, such as position coordinates, speed, course which were told by GPS or COPASS satellite to the vessel, was transmitted to the CCC by the DCL right now. At the same time, textual data coming from CCC can be transferred to the vessel through DCL. Consideration of the characteristics of SMS for civil which include unreliable channel, service with 60 seconds interval and limitation of 106 bytes one time, a data transmission protocol, which can implement data transmission securely, efficiency and creditably, must be developed and installed into DCL.

The second section is ship borne software system which hosted in an embedded device named Display terminal. The software system is an Embedded Geographic Information System. Electronic chart, which contains high precision of basic geographic features like disputed sea area, baseline of territorial sea, island etc., can be displayed in the terminal. Functions such as vessel location, course querying, receiving or sending instant short message from or to CCC have been integrated into this system.

The last section, Monitoring software system hosted in CCC which consists two main parts of data center and command center, is the most important part in this monitor system. More than 100 vessels' real time data can be received or sent out through a critical component which is called by Command Machine. Also, application server, coordination data server and base data server are indispensable parts of data center. After processed by the software hosts in the application server, real time data coming from command machine can be saved in the real time data server. In the command center, varies users can look through any information about every ship surveying on the ocean through a large screen displayer or send service information to any ships.

A. Data Communication Link（DCL）

DCL is composed by five core components, Beidou Satellite, Signal transceiver, WIFI hot point on ship, Command machine and Digital Ocean private network. The ability of positioning, velocity measurement, one-way and two-way timing and short message communications of Beidou is critical to DCL. Data can be exchanged through Short Message Service (SMS) provided by Beidou between CCC and vessels.

Signal transceiver is a device that can receive position, velocity and course coming from Beidou. It is fixed on the ship and connected to display terminal through a cable. Formatted, encrypted, compressed text data processed by the terminal return to signal transceiver. Then, it sent this data to CCC by BeiDou' SMS.

WiFi is a popular technology that allows an electronic device to exchange data or connect to the internet wirelessly using radio waves. A WiFi device has been installed into the display terminal. Varies mobile devices, such as smart phone, PDA, tablet etc. used by shipman, connect to the terminal. Link channel between shipman and CCC is established.

Command machine is deployed in the CCC and connected to the digital ocean private network. It can receive text data coming from amount to 100 surveying ships at the same time through Beidou' SMS. It can send point to point message or broadcast service message to the ships that are managed by it.

Digital Ocean private network is one of the major elements of Digital Ocean Project undertaken by SOA of china. This network connected the coastal provinces' Information Center, Institute of Marine Science, SOA of china and other government departments together. Almost all ocean workers can access this network which makes the most monitors directly apply their requirements through this network in the procedure of Oceanographic surveying.

The data flow in Data Communication Link can be depicted by Figure 2.

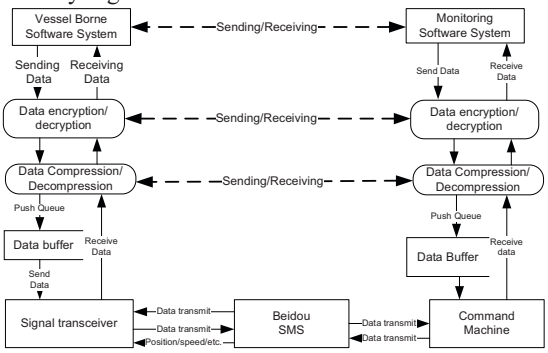


Figure 2. Data Flow Diagram

B. Ship Borne Software System (SBSS)

SBSS host in the display terminal which connects to the signal transceiver through a cable and connects to many mobile devices hold by ship crew through WiFi hot point. Logically, SBSS can be divided into four layers: data layer, components layer and functions layer.

Firstly, Signal transceiver, WiFi hot point and base data which is composed by map file data and ocean floor elevation data, make up of data layer. Data coming from Beidou can be read and write through computer serial port communications. Data which is relative to crew was accessed by a WiFi enabled interface. Similarly, base data can be accessed by a common data interface.

Secondly, two core components, named message manager and map render, make up of components layer. Functions such as data encryption, decryption, compression, decompression etc. are realized in the component of message manager. A sending queue which charges of sending task scheduler is created and maintained by massage manager. With regard to the map render, textual coordinate data can be rendered in the screen which makes it very easy to locate ship's position and display ship's course.

Lastly, two kinds of module, map and message relative functions were developed in functions layer. The map module includes functions such as map browsing, vessel positioning, course tracing and so on. The message module includes functions such as message receiving, message sending, sending interval setting and message sending and receiving logging etc. Users on the ship can use these functions directly through a graphical user interface.

The logical architecture can be depicted by Figure 3.

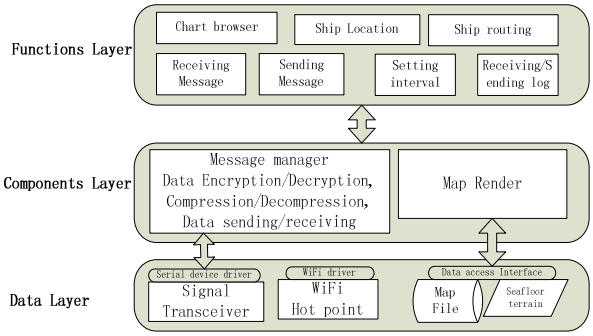


Figure 3. Logical architecture of SBSS

C. Monitoring software system in CCC

Monitoring software system is constructed based on DOFS which include data layer, function layer and application layer respectively from the bottom to up (Zhang X, 2011). Function modules which made up of the monitoring software system had been developed and insert into corresponding layer as an add-in.

The data service bus, constructed in DOFS, had implemented the movement of data between applications and the transformation of data from specific source application format to the common canonical model format and to the format of the target systems. The data service bus supports the interaction patterns of "publish and subscribe" and "request and reply". Using this data service bus, a local utility is developed to handle data movement coming from SBSS. Functions like reading text message from Command

Machine, encryption message, decryption message, compressing message, decompressing message and writing message to database are developed in this module.

Functions like ship positioning, course tracing etc. were added to the function layer of DOFS. On the one hand, the public service modules in this layer had been extended which make it is possible to access data coming from SBSS through data layer. On the other hand, application function module in this layer had been extended also. Functions like sending or receiving short message, ship positioning, course tracing etc. had been added to this module.

As for application layer, driven by users' requirements, factors like easy to use, fast to access, accuracy to analyze are concerned. Take advantage of custom development ability of DOFS, functions like ship positioning, course tracing etc. had integrated into application layer. Users can fetch varies real-time information about surveying ship in Distant-Water include ship's position, course, speed etc. and send or receive short message to or from surveying ship.

Figure 4 depicts the relation between monitoring software system in CCC and DOFS.

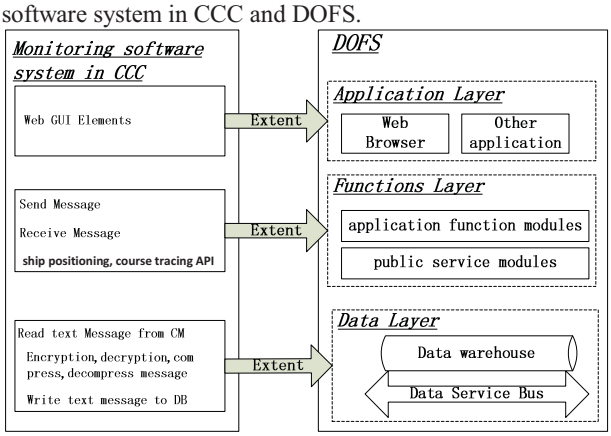


Figure 4. monitoring software system based DOFS

IV. System Implements and Evaluation

A Real Time Monitoring System of Oceanographic Surveying (RTMSOS) has been developed based on the solution designed in this paper by State Oceanic Administration of China (SOA). Functions such as real-time ship's location and course tracing, depicted in Figure 4, are implemented. Ship's status information such as coordinates, speed, course etc. is sent to the command center in real time via BeiDou's SMS. Instant messaging between seaman and the command center is also implemented without expensive communication cost charged by INMARST-B satellite communication system.

Through the encryption algorithm used in this system, data can be transmitted safely via by Beidou's SMS and through the data compression algorithm, more data can be sent in one interval time. This system has been deployed not only to the command unit of SOA, but to the units, the ships' owner, which can get ship's real-time position conveniently and remain contact at any time. At present, system has achieved the aim to monitoring the ships in China Marine Reach Vessels real-timely. Figure 5 and Figure 6 are the screen shot of RTMSOS.

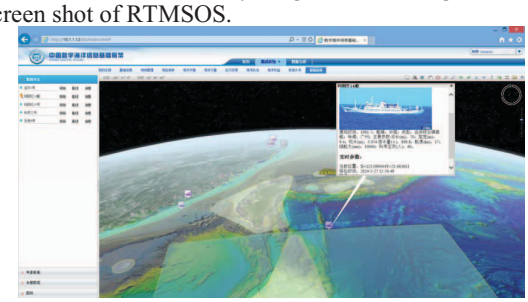


Figure 5. The ship's real-time location

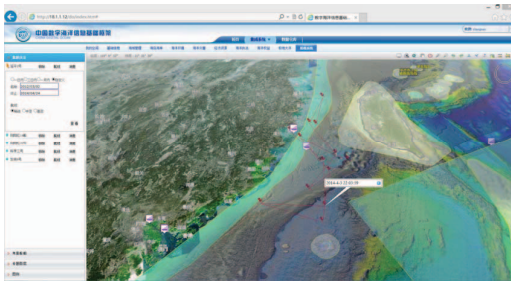


Figure 6. The ship course tracing

V. Conclusions

The solution discussed in this paper can realize real-time monitoring on Oceanographic Research Vessels with low cost and rich functions. Beidou's SMS, used in the solution and developed by China, make it possible to transmit varies data between Command Center and vessels conveniently. Through the application of compression algorithm and encryption algorithm, relatively more data can be transmitted in one interval and the security of information transmitted in the channel can be protected effectively. In addition, functions of monitoring system can be implemented easily by the form of extending the API provided by DOFS. The successful construction of RTMSOS indicated that the solution provide a new way to solve the problems of the development of varies monitoring system.

Acknowledgment

The authors thank Jiye Jin, Director of Key Laboratory of Digital Ocean, who provided an initial idea on developing a monitoring system which can make the command center on the shore keep in touch with the marine science research vessel operating in the far sea in real time without paying expensive cost of communication. The first author also thanks Xuemin Gao, an section chief of State Oceanic Administration, who substantial support the deployment of experiment device on the vessel which make the test work smoothly. Lastly, we thank Chaosheng Feng at Sichuan Normal University who provided professional proofreading for this paper.

REFERENCES

[1] Eriksen, T., Høye, G., Narheim, B., & Meland, B. J. (2006). Maritime traffic monitoring using a space-based AIS receiver. Acta Astronautica, 58(10), 537-549.

[2] Detsis, E., Brodsky, Y., Knudtson, P., Cuba, M., Fuqua, H., & Szalai, B. (2012). Project Catch: A space based solution to combat illegal, unreported and unregulated fishing: Part I: Vessel monitoring system. Acta Astronautica, 80, 114-123.

[3] Knoska, J. J., Dalrymple, M. L., & Williams, W. D. (2004). U.S. Patent No. 6,816,088. Washington, DC: U.S. Patent and Trademark Office.

[4] Chengzhi, L. (2013). The Chinese GNSS—System development and policy analysis. Space Policy, 29(1), 9-19.

[5] Yu Longyang, Wang Xin & Li Shujian. (2012). Positioning data compression and reliable transmission based on beidou short message. Communication and Network,*38*(11), 108-111.

[6] Jin, S. (2013). Recent progresses on Beidou/COMPASS and other global navigation satellite systems (GNSS)–I. Advances in Space Research, 51(6), 941.

[7] Zhang, X., Dong, W., Li, S., Luo, J., & Chi, T. (2011). China digital ocean prototype system. International Journal of Digital Earth, 4(3), 211-222.

[8] Olchi-Oglu, N. I. (1963, April). A description of some foreign research vessels and their equipment. In Deep Sea Research and Oceanographic Abstracts(Vol. 10, No. 1, pp. 87-91). Elsevier.

[9] Berzins, G., Phillips, R., Singh, J., & Wood, P. (1989, October). Inmarsat-Worldwide mobile satellite services on seas, in air and on land. In *Malaga International Astronautical Federation Congress* (Vol. 1).

[10] CHENG, F. L., ZHANG, Y. F., & LIU, J. J. (2008). Long Message Communication Protocol Based on the" Beidou" Satellite Navigation System.*Ocean Technology*, *1*, 009.

A Simulation Model of Boarding Process for Narrow-body Aircraft

Yue-ya SHI
Air Traffic Management College
Civil Aviation Flight University of China
Deyang, China
shiyueya@126.com

Qi-feng MOU
Air Traffic Management College
Civil Aviation Flight University of China
Deyang, China
mouqifeng@sina.com

***Abstract*—In this paper, we proposed a simulated boarding process for narrow-body passenger aircraft with consideration of passengers' individual boarding time. The individual boarding time, consisting of walking time in aisle, time to store luggage and seating time, were analyzed at first. With characteristics of seat layout in narrow-body aircraft, a simulation model of boarding process for narrow-body aircraft was established by Monte Carlo stochastic modeling method. Then the model was applied to three typical aircrafts under five different aircraft boarding strategies including Back to Front, Random, Reverse Pyramid, Wilma Outside in and Wilma Block boarding strategies. Our numerical results of the total passenger boarding time were presented in three distributing graph forms. Finally, the results indicated that Reverse Pyramid and Wilma Block are more effective than the other three boarding strategies. The model parameters and boarding sequences also can be adjusted to adapt to different narrow-body aircrafts and various boarding patterns.**

Keywords- Boarding process;Individual boarding time; Narrow-body Aircraft;Computerized simulation; Monte Carlo

I. INTRODUCTION

Recently, airlines have paid a great deal of attention to boarding time because they believe it affects the overall success of an airline. The standard practice is to allow first and business class passengers, passengers needing special assistance, as well as members of airlines' frequent flyer programs to board first. For the remaining passengers, random boarding strategy is the most commonly adopted boarding style. Block boarding strategy is usually adopted in some American airlines. In this way passengers board in groups from back to front, or from the outside in, that is, window seats first, middle seats second, and aisle seats last, or combine the two together. In some airlines in Hong Kong, Australia and Japan, the way to get people on an airplane would be to board them individually from back to front, or the outside in by calling each one of the passengers individually to board the aircraft. Some other airlines would arrange boarding order according to the passengers' conditions, such as passengers without baggage board preferentially (e.g. U.S. Frontier airlines) or the first comer, the first board (e.g. U.S. Southwest Airlines) and so on.

Many experts and scholars, both at home and abroad have been carrying out different researches on the subject from 1960s. Van Landeghem and Beuselinck conducted a simulation-based study on airplane boarding which showed that one pattern that seemed practical and efficient was boarding passengers by half-row, that is, by splitting each row into a starboard-side group and a port-side group and then boarding the half-rows one by one [1]. Liu Shan et al. adopted Buffering Dual- Stack Pattern (BDSP) to model and analyze the boarding process, using airport channel as the temporary buffer and arranging two groups to board at a time. Comparison of the results with current excellent project shows a 34.76% degree of result in boarding time reduction [2]. Zhang Yuxiang et al. analyzed and simulated an airplane boarding model with "block" as a basic object to found that the boarding time can be shortest when the block size equal to 1 at the Back-to- Front with Window-to-Aisle method and Reverse-Pyramid method [3]. Tang Tieqiao et al. proposed a new aircraft boarding model with consideration of passengers' individual properties. The model was applied to explore the dynamic properties of passengers' motions under three different aircraft boarding strategies including the random boarding strategy, the boarding strategy based on passenger's seat serial number and individual properties. The numerical results illustrate that this boarding strategy is more effective than the other two boarding strategies [4]. Soolaki Majid et al. examined the different kinds of passenger boarding strategies and boarding interferences in a single aisle aircraft, and offered a new integer linear programming approach to reduce the passenger boarding time. A genetic algorithm was used to solve this problem [5]. Iyigunlu. Serter et al. implemented six different boarding strategies (Wilma, Steffen, Reverse Pyramid, Random, Blocks and By letter) for Boeing 777 and Airbus 380 aircrafts by using Agent-based modelling approach. Results from the simulation demonstrates Reverse Pyramid method is the best boarding method for Boeing 777 and Steffen method is the best boarding method for Airbus 380 [6].

II. RULES

A. In this model, we focus on narrow-body passenger airplanes, such as the Airbus A320 and the Boeing 737, which have a central aisle and rows of three seats on both sides of the aisle.

B. We only explore the aircraft boarding in the economy class and don't take special passengers into consideration.

C. With 100% seat occupancy, the aircraft door is front-cabin door.

D. Each passenger's carried luggage has the same attributions (e.g., the size and weight of the carried

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.63

luggage, the portability of the carried luggage, etc.). Every passenger stows their carried luggage in overhead bins before he is seated.

E. *The physical condition and action ability of every passenger is the same.*

F. *There is no unexpected case during boarding such as being late, taking wrong seat, swapping seats and etc..*

G. *Passengers enter the plane one by one.*

H. *When a passenger blocking another passenger's access who is in the same row to his seat, he must get up to the aisle of the aircraft. The aisle space in a row is able to hold only one people.*

I. *The behavior of one passenger can only influence the followed one's boarding and have no subsequent impact on other passengers[2].*

III. BASIC TIMES

The seat layout of a narrow-body passenger aircraft are presented in Fig. 1. The interior configuration is divided into cells along the aisle and each cell's length or width is the distance between two adjacent seats. The number shown in Fig. 1 represents the seat layout: 0 represents aisle, 1 represents aisle seats, 2 represents middle seats and 3 represents window seats.

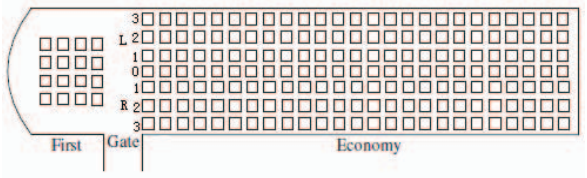


Figure 1. The seat layout of a narrow-body passenger aircraft

The boarding process is a typical service operation in which the customer participates. This means that any procedure to improve the system will also have to pay attention to customer comfort and determine the total airplane turnaround time [7].

The boarding time of we modeled starts when the first passenger enters the plane and ends when the last passenger is seated in his assigned seat. The boarding time consists of the time for passengers to find his seat, store their luggage and be seated. The three basic times are listed in the following section.

A. *Walking time in aisle* t_b

Walking time in aisle means the time a passenger walking without blocking from the front door to his row , which relates to passenger's row number and walking speed. Due to the rule that every passenger's physical condition and action ability is the same, let t_0 be a basic time variable represents passenger passing one cell, so walking time in aisle is as the following expression.

$$t_a = x \cdot t_0$$

where x is row number of a passenger.

B. *Time to store luggage* t_a

According to the rule that each passenger's carried luggage has the same attributions (e.g., the size and weight of the carried luggage, the portability of the carried luggage, etc.), every passenger stows their carried luggage in overhead bins before he is seated. Let t_a be the time to store luggage represented by the basic time t_0 as the following expression.

$$t_a = \delta \cdot t_0$$

where δ is a coefficient of average time to store luggage.

C. *Seating time* t_c

Seating time starts when a passenger begins to walk from his row and ends when he installs themselves in their assigned seats, consisting of walking time and seat interference time.

Seat interferences occur when passengers seated close to the aisle block other passengers seated in the same row. Consider, for example, an aircraft with rows progressively numbered from front to back and seats of each side labeled 1 to 3 from middle to window as shown in Fig. 1. A passenger sitting in seat 71 (the aisle seat in row 7) could block the passenger seeking seat 73 (the window seat) and will have to stand in the aisle for the passenger in 73 to be seated. The interference is even worse when passenger 71 arrives and passengers 72 and 73 are seated.

Seating time t_c is represented by three variables as the following expression.

$$t_c = t_1 + t_2 + t_3$$

Where t1 t2 t3 are respectively the time for outside-seat passenger to exit from seat into aisle, the time for inside-seat passenger install in seat and the time for outside-seat passenger return to his seat.

We indicate the types of seat interferences by the boarding order of the passengers as shown in table 1. As seen in the table, boarding window seats before middle seats and middle seats before aisle seats reduces the number of expected seating time significantly.

TABLE I. SEAT INTERFERENCES AND SEATING TIME（UNIT: t_0 ）

	Seat interferences	T_1	T_2	T_3	T
1	[1]	0	1	0	1
2	[2]	0	2	0	2
3	[3]	0	3	0	3
4	[1]→[2]	1	2	1	4
5	[1]→[3]	1	3	1	5
6	[2]→[3]	2	3	2	7
7	[3]→[1]	0	1	0	1
8	[3]→[2]	0	2	0	2

9	[2]→[1]	0	1	0	1
10	[2] [3]→[1]	0	1	0	1
11	[1] [3]→[2]	1	2	1	4
12	[1] [2]→[3]	3	3	3	9

D. *Individual boarding time T*

Individual boarding time is determined by walking time in aisle, time to store luggage and seating time, which can be obtained by using (1).

$$T=x \cdot t_0 + \delta \cdot t_0 + t_c \quad (1)$$

We established simulation model of passenger boarding process for individual boarding time, and then the total boarding time can be obtained.

IV. SIMULATION MODEL ESTABLISHMENT

A. Simulation model of passenger boarding process

Different boarding process which is related to boarding strategy and seat layout of aircrafts results in different values of boarding time.

After determining the algorithm of individual boarding time, we established a simulation model of passenger boarding process, which simulates each boarding strategy. Then we judged each boarding strategy and proposed the optimal one for narrow-body aircrafts by the simulation-based studies and comparative analysis of the results. The process of simulation analysis is shown in the following section.

1) Seat number

Seats are progressively numbered by row from front to back, labeled 1 to 3 from middle to window in each side and labeled 0 to 1 from left-side to right-side. Let a three-dimensional coordinates(x,y,z) be the seat in row x, column y and side z (mapped from 0-1, 0 represents left-side seat and 1 represents right-side). Every seat number is unique.

2) Initializing the variables of seat occupancy

A passenger is considered as an independent object, who is randomly assigned to the single seat according to some boarding strategy. Let J_{xyz} be seat occupancy and its initial value is 0. If a seat (x,y,z) is distributed to a passenger, then $J_{xyz}=1$.

3) Total boarding time

We can conclude the boarding time by analyzing the various orders in which people enter the airplane. In a boarding process, each passenger regard as an individual object gets to his seat one by one. So we analyzed respectively every passenger's individual boarding time and obtained the following results.

Check seat occupancy in the same row and side firstly. For example, if x=a, y=b, z=c, there are two conditions on J_{ayc} (y∈1-3 but y≠b) in the following section.

$$\sum_{y=1,y\neq b}^{3} J_{ayc} = 0$$

a) If $\sum_{y=1,y\neq b}^{3} J_{ayc} = 0$ *, there is no seat interference, which belong to the case 1-3 in table 1. The individual boarding time is shown as the following expression.*

$$T=\delta \cdot t_0 + a \cdot t_a + b \cdot t_0$$

$$\sum_{y=1,y\neq b}^{3} J_{ayc} = 1$$

b) If $\sum_{y=1,y\neq b}^{3} J_{ayc} = 1$ *, there is one seat occupied in the same row and side, which belong to the case 4 in table 1.Then we analyzed whether there is seat interference on the occupied seat. Suppose in row a side c, the column number of the occupied seat (a,y,c) is* $\alpha_{J_{ayc}=1}$*. Compared* $\alpha_{J_{ayc}=1}$ *with b value, if* $\alpha_{J_{ayc}=1}>b$*, there is no seat interference. So the individual boarding time can be obtained by using the following expression.*

$$T=\delta \cdot t_0 + a \cdot t_0 + b \cdot t_0$$

If $\alpha_{J_{ayc}=1}<b$*, the individual boarding time is shon as the following expression.*

$$T=\delta \cdot t_0 + a \cdot t_0 + (2Y + b) \cdot t_0$$

$$\sum_{y=1,y\neq b}^{3} J_{ayc} = 2$$

c) If $\sum_{y=1,y\neq b}^{3} J_{ayc} = 2$ *, then there is two seats occupied in the same row and side, which belong to the case 10-12 in table 1.*

Then we analyzed whether there is seat interference on the occupied seat. Suppose in row a side c, the column numbers of the occupied seat (a,y,c) are $\alpha^1_{J_{ayc}=1}, \alpha^2_{J_{ayc}=1}$. We compared them with b in the following section.

a) If $\alpha^1_{J_{ayc}=1}>b$*,* $\alpha^2_{J_{ayc}=1}>b$*, there is no seat interference, which belong to the case 10 in table 1. The individual boarding time is shown as the following expression.*

$$T=\delta \cdot t_0 + a \cdot t_0 + b \cdot t_0;$$

b) If $\alpha^1_{J_{ayc}=1}>b$*,* $\alpha^2_{J_{ayc}=1}<b$ *or* $\alpha^1_{J_{ayc}=1}<b$*,* $\alpha^2_{J_{ayc}=1}>b$*, which belong to the case 11 in table 1, then there is one seat interference. The individual boarding time is shown as the following expressions.*

$$T = \delta \cdot t_0 + a \cdot t_0 + (2Y_2 + b) \cdot t_0$$

or

$$T = \delta \cdot t_0 + a \cdot t_0 + (2Y_1 + b) \cdot t_0$$

c) If $\alpha^1_{J_{ayc}=1}<b$*,* $\alpha^2_{J_{ayc}=1}<b$*, which belong to the case 12 in table 1, then there is two seat interferences. The individual boarding time is shown as the following expressions.*

$$T=\delta \cdot t_0 + a \cdot t_0 + (2Y_1 + 2Y_2 + b) \cdot t_0$$

4) If seat number (a,b,c) was occupied, then $J_{abc} = 1$.

5) At late, all the passenger's individual boarding time should be added to the total boarding time.

B. Monte Carlo stochastic model

In fact, no matter what boarding strategies are, boarding is a random process, which is reflected in the boarding sequences. On the other hand, it is also reflected in the randomness and uncertainty of each passenger's behavior due to individual difference. According to the rule that each passenger arrives at a constant rate and costs a constant time, In this case, the only thing needed to be simulated is the boarding sequences.

In our model passengers are randomly assigned, so a random boarding sequence based on Monte Carlo stochastic modeling method will be generated by computer. After that, the passenger boarding process will be simulated by using the simulation model, and then a total boarding time is obtained, as shown in Fig. 2. We can respectively computerize simulate every boarding strategy a certain number of times on a typical aircraft, and obtain mathematical expectation of the total boarding time of every strategy on a given aircraft.

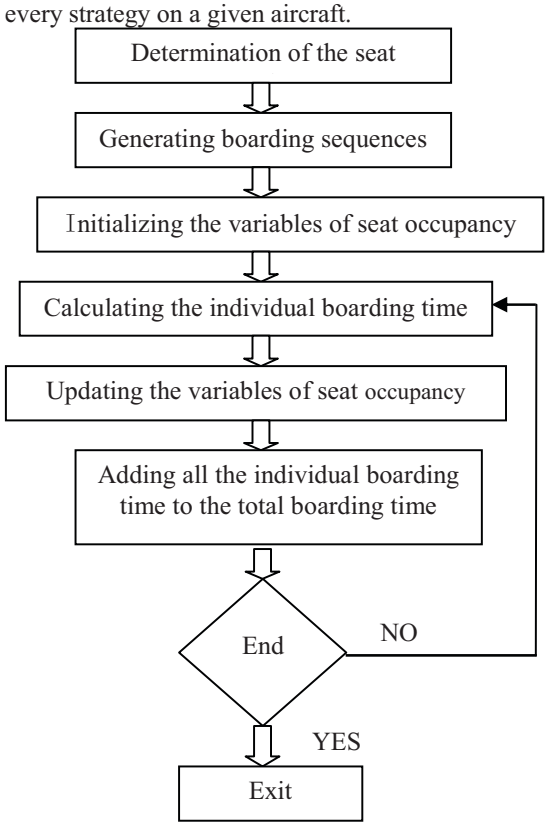


Figure 2. Boarding process based on Monte Carlo stochastic modeling method

V. CALCULATION

A. The aircraft types to be simulated

Our research focus on Boeing 737-300, Airbus A320 and Boeing 757, which reflected typical cabin arrangements as shown in table 2.

TABLE II. THE NUMBER OF SEATS OF SEVERAL TYPICAL AIRCRAFTS

Types	Number of seats in the economy cabin
B737-300	136
A320	144
B757	184

B. Boarding patterns

The five boarding patterns including Back to Front, Random, Reverse Pyramid, Wilma Outside in and Wilma Block boarding patterns are presented in Fig. 3. Each schematic shows the seat layout of a narrow-body airplane. The colors shown on each seat are the boarding sequences in which the seats are assigned. The lighter the color is, the earlier passengers enter the airplane.

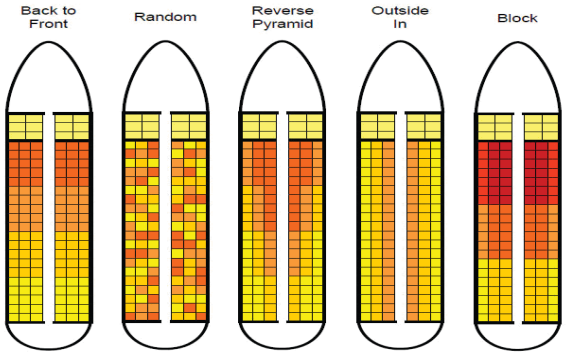


Figure 3. Five boarding patterns

C. Simulation process

The values of the parameters in expression (1) turn to various results, Let $t_0 = 1s$, $\delta = 8$ [8]. The model can also adjust related parameters to adapt to the different boarding situations. The model parameters can be adjusted to adapt to different aircrafts and various boarding situations.

Using the simulation model, we respectively simulated each of the boarding strategies 30 times on the three typical aircrafts. Then, we obtained enough data for all the strategies in the form of distributing graph, as shown in Fig. 4-6.

1) *Simulation of Airbus A320*

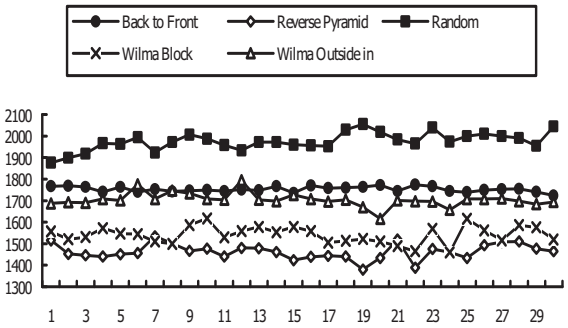


Figure 4. Simulation of Airbus A320

2) Simulation of Airbus Boeing 737-300

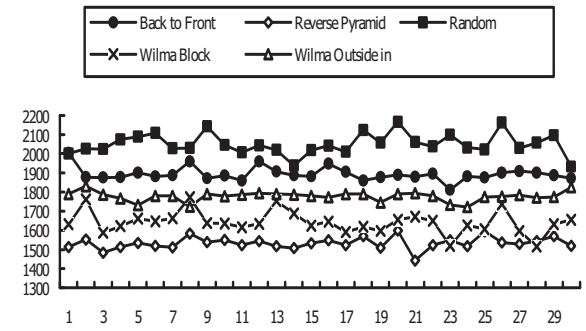


Figure 5. Simulation of Boeing 737-300

3) *Simulation of Airbus Boeing 757*

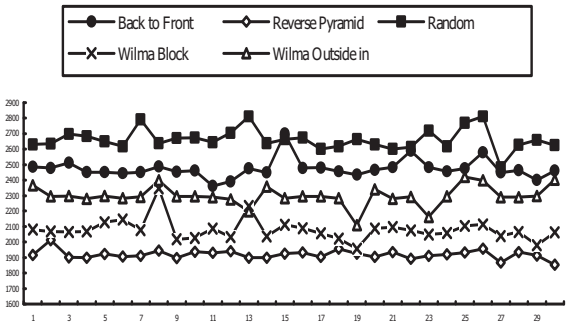


Figure 6. Simulation of Boeing 757

D. Simulation results

From Fig. 4-6, we can conclude the following findings: for narrow-body aircraft, Reverse Pyramid and Wilma Block are more effective than the other boarding strategies, following by Wilma Outside In, Back to Front and Random, which are less competitive comparing with other three schemes.

VI. CONCLUSION

Aiming at the total passenger boarding time, on the premise that passenger entering the plane one by one, based on the individual boarding time, the paper established a simulation model with Monte Carlo stochastic modeling method, which is feasible and achievable.

There are certain flaws in the accuracy and the validity of the analyzing result because of the rule that every passenger's physical condition and action ability is the same. Besides, one of the assumption is passenger getting into cabin one by one in the boarding process so that if one passenger haven't been seated, the next one would not enter the plane. Therefore, the total time obtained is longer than actual situation, especially for the strategy "back to front".

ACKNOWLEDGMENTS

The present research was supported by Youth Science Foundation and General Program of Civil Aviation Flight University of China. (Project No.: Q2012-072、J2010-30).

REFERENCES

[1] Van Landeghem H, Beuselinck A. "Reducing passenger boarding time in airplanes: a simulation based approach," European Journal of Operational Research, vol.142, pp.294−308, 2002.

[2] Liu Shan, Li Tianshun, Yu Ligeng, Jia Lei, and Du Yu, "New optimal model for boarding time," Journal of Civil Aviation University of China. Tianjin, vol.26, pp.50-52, 2008.

[3] Zhang Yuxiang, Xiao Chunjing, "An airplane boarding model and its optimization," Computer Simulation, Beijing, vol. 26, pp.243-246, 312, 2009.

[4] Tang Tieqiao, Wu Yonghong, Huang HaiJun, Lou Caccetta, "An aircraft boarding model accounting for passengers' individual properties," Transportation Research Part C: Emerging Technologies, vol.22, pp.1-16, 2012.

[5] Soolaki. Majid, Mahdavi. Iraj, Mahdavi-Amiri. Nezam, Hassanzadeh. Reza, Aghajani. Aydin. "A new linear programming approach and genetic algorithm for solving airline boarding problem". Applied Mathematical Modelling. New York, vol.36, pp.4060- 4072, 2012.

[6] Iyigunlu. Serter, Yarlagadda. Prasad, and Fookes. Clinton, "Agent-based application on different boarding strategies," Applied Mechanics and Materials. Switzerland, vol. 568-570, pp.1893-1897, 2014.

[7] Vanden Briel M H L, Villalobos J R, Hogg G L, "The aircraft boarding problem," Proceedings of the 12th Industrial Engineering Research Conference (IERC-2003)(CD-ROM), Portland, No. 2153.,2003

[8] Liu Yang, Liu Zhenzhao, Jia Limin. "Adaptive approach to aircraft boarding strategy," Journal of Transportation Systems Engineering and Information Technology. Beijing, vol.8, pp. 118-123, 2008.

The Face Recognition Algorithm Based On Double Coding Local Binary Pattern

Gao Ye, Gao Kao
School of Computer Science and Technology, Xi`an university of Science and Technology
Xi`an, 710054, China
E-mail: gaoye@xust.edu.cn，642854472@qq.com

***Abstract*—This paper proposes a new Double Coding Local Binary Pattern algorithm (d-LBP) to improve the weakness of traditional LBP algorithm, such as, incompletely features extraction, too much sample points, low computational efficiency and so forth. Firstly, it defines two thresholds: the amplitude threshold and the difference threshold, which succeed in taking full consideration of the relationship among pixel gray values and reducing sampling points. Secondly, the paper uses the d-LBP algorithm to extract statistical characteristics in each small block of the original face image. Finally, it fulfills the face recognition by using K Nearest Neighbor algorithm.**

Keywords—face recognition, LBP, d-LBP, recognition rate, histogram Introduction

I. INTRODUCTION

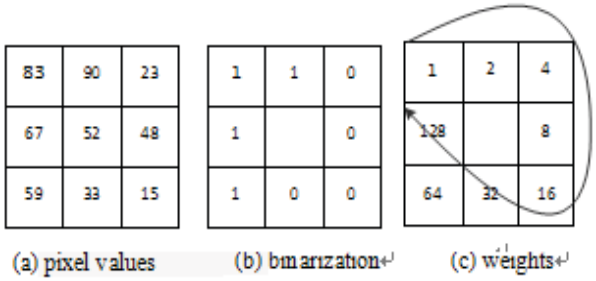
Face recognition is a computer technology for identity authentication by comparing the information of human visual features [1]. Currently，it is a hot topic in pattern recognition and artificial intelligence, and widely used in the identification, video surveillance and other aspects. The process of recognition mainly contains the image capture, the face positioning, the image preprocessing, and the face recognition [2]. And the face recognition is the most important stage in the process of recognition. Among various kinds of face recognition algorithm, Local binary pattern (LBP) face recognition algorithm has been widely concerned.

LBP is used to describe the texture information and advanced by Ojala as early as 1996 [3]. The characteristics of LBP are very simple and effective to describe the texture. In 2002, Ojala proposed the uniform mode of local binary pattern to improve the LBP [4]. The improved LBP can make the texture information of local neighborhood in the gray image to adapt to the different rotation and illumination. In 2006, Based on block image of local binary pattern had been proposed by Ahonen etc [5]. This algorithm is more effective to collect the texture information of face image and effective to improve the recognition rate. Now LBP algorithm has been very good use in image retrieval, texture segmentation and classification, face recognition and other fields..

II. BASIC LBP

LBP operator is derived from a local neighbor texture definition and is a kind of texture measure within the scope of gray level. The core idea is to get the binary code of neighborhood pixels by comparing the center pixel values with neighborhood pixel values [6]. First, if the neighborhood pixel value is greater than the center pixel value, then the corresponding binary code is 1, else instead of 0. Then, it is concluded that the binary numbers which are strung together according to clockwise as a new center pixel values. Finally, the local binary pattern of center pixel is calculated by converting binary numbers to decimal number.

The basic LBP operator is defined in a 3 * 3 neighborhood. In the 3 * 3 neighborhood, the center pixel grey value compared with the other 8 pixels grey values which are in the 3 * 3 neighborhood. If the 8 pixels grey values are larger, then the corresponding binary code is marked as 1, otherwise 0. So you can get an 8 bit binary numbers and convert the 8 bit binary numbers to a decimal integer. That is the value of the neighborhood of LBP. As shown in figure 1.



The binary number of LBP = 11000011,
The decimal number of LBP=128+64+0+0+0+0+2+1= 195.

Figure 1 basic LBP operator

In order to extract and adapt to the texture feature of different scales, and achieve the objectives of the rotation invariant and gray level unchanged, Pietikainen improved the basic LBP operator to expand the original 3 * 3 neighborhood to any neighborhood with a circular field instead of the original square neighborhood. The improved LBP was allowed to have any sampling points in the circular neighborhood with radius R [7]. As shown in figure 2, the basic LBP operator is defined as shown in formula [4] (1).

$$LBP(R,\ P) = \sum_{p=0}^{p-1} 2^p S(i_p - i_c),$$

$$S(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \qquad (1)$$

Where: i_c is the gray value of center pixel in the local area; i_p shows the p-th grey value of sampling point in the center pixel neighborhood area; p is the sampling point, the radius of circle is R.

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.64

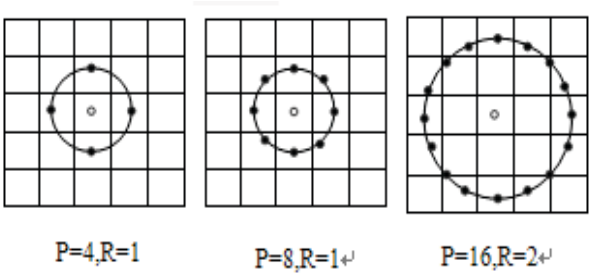


Figure 2 several different LBP operator.

III. DOUBLE CODING LBP OPERATOR

Although the basic LBP algorithm has ability to describe the texture information itself, further research can find that in a certain local area, basic LBP operator only considers the differences of gray values between center pixel and neighborhood pixels, but the amplitude relationship between center pixel gray value and the neighborhood pixel gray values are ignored. Because every pixel gray value cannot be made full use of, it could lead to a drop in the final recognition rate when the face texture features are extracted.

Moreover, too much sampling points will make the algorithm more complicated, which may result in the decrease of the rate of recognition. Due to the above problems, this paper presents a double coding local binary pattern (d-LBP).

Firstly, reducing the complexity of the algorithm, the sampling points of d-LBP operator is reduced to 4 from 8 of the basic LBP operator. As shown in figure 3.

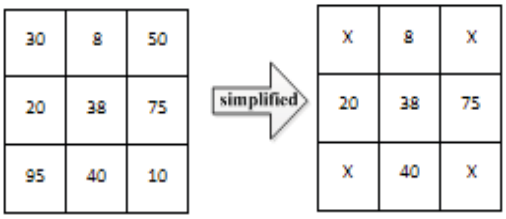


Figure 3 simplify the sampling points

Then, making full use of the relationship among the gray values of each pixel within a certain local area. So θ is defined as the amplitude threshold and ε is defined as the difference threshold. As shown in formula (2), (3).

$$\theta = \frac{1}{p}\sum_{k=0}^{p-1}|i_k - i_c|, \quad p = 4 \qquad (2)$$

Where: i_c is the gray value of the center pixel in the local area; i_k shows the k-th grey value of sampling point in the center pixel neighborhood area; p is the number of sampling points.

$$\varepsilon = \frac{1}{p}\sum_{k=0}^{p-1} i_k - i_c, \quad p = 4 \qquad (3)$$

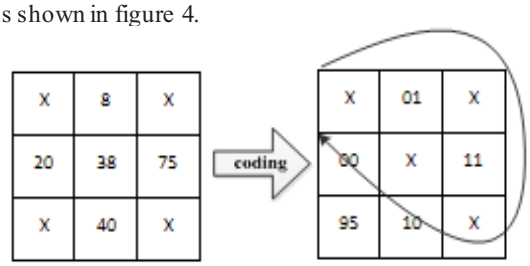
Where: i_c is the gray value of the center pixel in the local area; i_k shows the k-th grey value of sampling point in the center pixel neighborhood area; p is the number of sampling points.

Finally, in order to describe the facial texture information in detail, the basic LBP operator with one binary coding is replaced by the improved LBP operator with two binary encoding. The first binary code is related with the difference between the neighborhood pixels gray values and the center pixel gray value. Compared with difference threshold (ε), if it is larger, the binary code is 1, on the other hand, is marked as 0. The second binary code is related with amplitude between neighborhood pixels gray values and the center pixel gray value. Compared with amplitude threshold (θ), if it is larger, the binary code is marked as 1, otherwise 0. As shown in formula (4).

$$dLBP = \sum_{k=0}^{p-1} 2^{2k} S(i_k, i_c) \ ,$$

$$s(i_k, i_c) = \begin{cases} 00, & i_k - i_c < \varepsilon \text{ 且 } |i_k - i_c| < \theta \\ 01, & i_k - i_c < \varepsilon \text{ 且 } |i_k - i_c| \geq \theta \\ 10, & i_k - i_c \geq \varepsilon \text{ 且 } |i_k - i_c| < \theta \\ 11, & i_k - i_c \geq \varepsilon \text{ 且 } |i_k - i_c| \geq \theta \end{cases}, \quad p = 4 \qquad (4)$$

Where: $\theta = \frac{1}{p}\sum_{k=0}^{p-1}|i_k - i_c|$ is amplitude threshold; $\varepsilon = \frac{1}{p}\sum_{k=0}^{p-1} i_k - i_c$ is difference threshold; i_c is the gray value of the center pixel in the local area; i_k shows the k-th sampling point grey value in the center pixel neighborhood area; p is the number of sampling points. As shown in figure 4.



θ =1/4(0+37+2+18) =22;
ε =1/4(8+75+40+20) -38 = -2;
d-LBP = 01111000 = 120

Figure 4 d-LBP operator.

IV. LBP FEATURE EXTRACTION AND MATCHING

In order to fully improve the effectiveness of the d-LBP operator and make better use of d-LBP to describe face image, this paper proposes three steps new algorithm to fulfill it. First of all, the d-LBP map of face image should be blocked; then, the images of each block are calculated to get the histogram of d-LBP respectively; finally, each partitioned histograms is connected according to a certain order to get a composite feature vector, that is, the d-LBP histogram of overall face image. The process of d-LBP feature extraction is shown in figure 5.

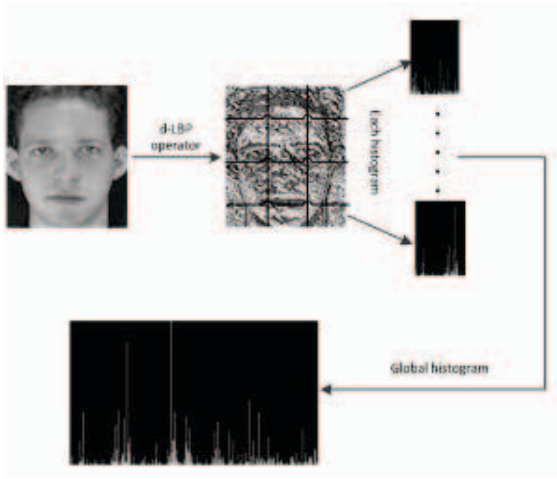


Figure 5 d-LBP feature extraction

Nonparametric statistical method is used to determine the histogram similarity between samples after getting a d-LBP histogram. As shown in formula (5).

$$\varphi^2(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)} \quad (5)$$

Where: $H_1(i)$ is the training sample; $H_2(i)$ is the sample to be classified; Similarity is measured by the distance of φ^2, the smaller the distance, the more similar the two faces [8].

V. EXPERIMENTAL RESULTS AND ANALYSIS

The experiment selects the ORL face database to verify and analysis the validity of the d-LBP algorithm. ORL face database with different illumination, facial expression and hairstyle was made by Olivetti research laboratory of the University of Cambridge, and has a certain number of rotation angles. The experiment of face recognition would to be done by selecting different training samples in ORL face database. Image One, Image three and Image five from each sample are selected as training samples, and the rest are as the test samples. The face image would be blocked into 4*4 blocks to get the d-LBP histogram. Recognition rate is shown in table 1, the average training time of each images is shown in table 2.

Table 1 face optimal recognition rate comparing in ORL

	1 Sample	3 Samples	5 Samples
$LBP_{(1,4)}$	71.39 %	86.79 %	95.00 %
$LBP_{(1,8)}$	73.89 %	88.21 %	96.00 %
d-LBP	74.72 %	89.29 %	98.50 %

Table 2 face unit training time comparing in ORL

	$LBP_{(1,4)}$	$LBP_{(1,8)}$	*d-LBP*
Unit Training Time(ms)	3.25	3.75	3.35

The experimental results show that: The more numbers of training samples, the better conducive classified information will be learn, thus the higher recognition rate will be acquired; d-LBP operator extracting facial features has better efficiency of identification than basic LBP operator, and improves a certain recognition rate; d-LBP operator has less sampling points, reduces the algorithm complexity effectively and speeds up the training speed, so as to improve the recognition speed.

VI. CONCLUSIONS

Double coding local binary pattern (d-LBP) algorithm of face recognition is developed based on the basic LBP algorithm. Because of the shortcomings of the basic LBP algorithm, the d-LBP algorithm fully considers the relationship between the center pixel gray value and neighborhood pixels grey values, and reduces the number of sampling points. Experimental data shows that the d-LBP algorithm can be more comprehensive to extract the partial features of the face image, and more efficient, accurate and quick to get the local texture feature information. In addition, higher training speed and recognition rate have achieved on the ORL face database..

ACKNOWLEDGMENT

Humanity and Social Science foundation of Ministry of Education of China (Grant No.12YJAZH022), and the Fundamental Research Funds for the Central Universities (WUT: 2014-Ia-039)

REFERENCES

[1] Shang-Hung Lin. An introduction to face recognition technology[J]. Informing Science The International Journal of An Emerging Transdiscipline, 2000, 3(01): 01-02

[2] Zheng C W. The summary of human face recognition methods[J]. Shanxi Electronic Technology, 2013, (04):0095-0096

[3] Ojala T, Pietikainen M, Harwood D. A comparative study of texture measures with classification based on feature distributions[J]. Pattern Recognition, 1996, 29(01): 51-59

[4] [Ojala T, Pietikainen M, Maenpaa T. multiresolution gray-scale and rotation invariant texture classification with local binary patterns[J]. Transaction on Pattern Analysis and Machine Intelligence, IEEE, 2002, 24(07):971-987

[5] Ahonen T, Pietikainen M, Hadid A. Face Description with Local Binary Patterns Application to Face Recognition.[J]. Transactions on Pattern Analysis and Machine Intelligence, IEEE, 2006, 28(12):2037-2041

[6] Jiang R, Xu J L, Zhang A P. The facial expression recognition based on improved local binary pattern[J]. Journal of Zhejiang Sci-tech University, 2013, 30(04):546-547

[7] Zhao G, Pietikainen M. Dynamic texture recognition using local binary patterns with an application to facial expressions[J]. Transactions on Pattern Analysis and Machine Intelligence, IEEE, 2007, 29(06):915-928.

[8] [Yuan B H, Wang H, Ren M W. Face recognition based on completed local binary pattern[J]. Application Research of Computers, 2012, 29(04):1557-1559

Application of Machine Vision in Defects Inspection And Character Recognition of Nameplate Surface

zhangwg@xust.edu.cn

Li Jing-mei
Collede of Computer Science and Technology
Xi'an University of Science and Technology
Xi'an, China
1021833489@qq.com

Zhang Wei-guo
Collede of Computer Science and Technology
Xi'an University of Science and Technology
Xi'an, China

Han Rui-li
College of Science
Xi'an University of Science and Technology
Xi'an, China
778955994@qq.com

Abstract--Aiming at a lot of weaknesses in manual inspection for defects inspection and characters recognition of nameplate, the paper proposed an application about inspecting and recognition of nameplate based on machine vision. This article firstly utilizes seed algorithm to process image, and remove incomplete nameplate image caused by shooting in order to retain nameplate's integrity. Secondly, it uses BLOB analysis algorithm to achieve precise detection for printing defect. Lastly, it adopts character recognize algorithm to divide region of character and count the ratio of white pixel to black pixel, accordingly attaining effective recognition of characters. Experiments show that the method of machine vision in defects inspection and character recognition of nameplate surface has faster inspection speed and higher precise, also commendably meets the requests of actual production.

Keywords--Machine vision; Defects inspection; Character recognition; Image processing

I. INTRODUCTION

Nameplate is a print which fixed on production, provides manufacturers of brand recognition, products parameters and avoids damaging device. The detection of nameplate defect can ensure accurate information, and the recognition of nameplate character is propitious to information query.

At present, the method for defect detection and character recognition nameplate mainly depends on manual inspection, Manual inspection relies on experience and subjective judgment of personnel, who are easily influenced by subjective and objective factors, such as susceptible mood and environment[1], resulting in some hidden risks to the detection and identification of product, and lower detection efficiency. Therefore, this paper imports the machine vision system into defect detection and character recognition. Machine vision system is through machine vision products (that is image capture devices, including CMOS and CCD), converting target into image signal, and sending it to a dedicated image processing software; then according to image feature, converting it into a digital signal; finally image system diversely makes operations on these signals to extract features of target, and determines control the equipment movement by the outcome result. The proposed machine vision system can improve flexibility of detection and automation degree, and make rapid, objective and stable detection of nameplate defect and character identification, greatly improving efficiency and accuracy of detection and recognize characters of nameplate[2,3]

II. INSPECTING SYSTEM

The proposed nameplate defect detection and character recognition system consists of image acquisition, image processing software and motion control devices[4-5]. The designed light source system of image acquisition, which can be supply of inadequate ambient light, transfers capturing image into industrial computer's memory and store, collaborating with an industrial camera which captures nameplate image. Image processing software extracts computer memory's data for defect detection and character recognition, and saves the processing result to the report, then transmits corresponding signal to the motion control device[6]. The results show that the machine vision system is superior to the manual inspection. Figure 1 is a nameplate defect detection and character recognition system based on machine vision.

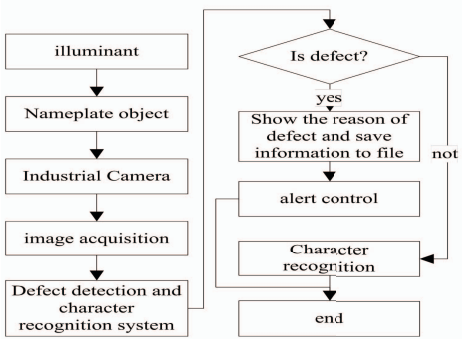


Figure 1. Machine vision system

III. INSPECTIN PROCESS

Image processing software which researches and develops in the VC++6.0 developing environment, has

The National Natural Science Foundation of China (coal joint fund) (No.U126114); Shan xi Province scientific research and development program of China (No. 2012JM8029); the Provincial Department of education science research project of Shan xi (No. 12JK0929)

978-1-4799-4169-8/14 $31.00 © 2014 IEEE
DOI 10.1109/DCABES.2014.74

abilities of intelligent analysis and character recognition, the detection process is shown in figure 2:

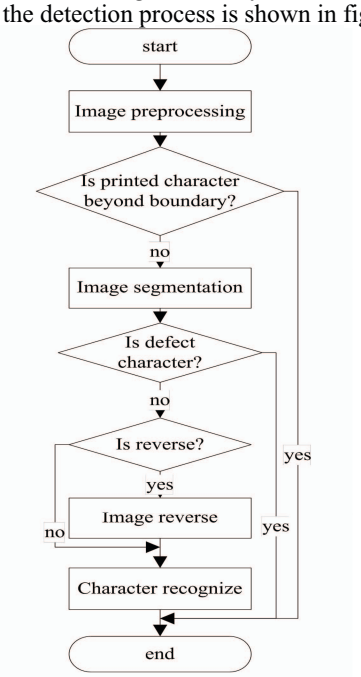


Figure 2. Image detection flow chart

Because of the industrial camera view is larger, and the label intervals relatively compact, acquired image gets part of upper and last. Therefore, this paper adopts seed algorithm to process the image, maintaining the integrity of nameplate image. Then, employ the BLOB analysis to obtain large connected regions of image, then judge if there has character beyond silver bar. If not, the silver bar code region form nameplate for identifying the characters need to be segmented, then estimate whether there is bias print defects of the recognized character. If there is not, judge whether the recognized character is reverse. If it is reverse, reversal operation is carried on the segmentation region; if it is not reverse, the character recognition algorithm would recognize printed characters.

A. Image preprocessing algorithm

The function of image preprocessing algorithm is to remove the incomplete nameplate image. Seed algorithm can effectively remove the part of nameplate from the nameplate image.

1) Convert the image into binary image. In this way, the image pixel value has forms of 0 and 1. Sign the start position of edge pixel named mark pixel, and establish a data structure queue that is empty.

2) Get next pixel and judge whether is coincident with the mark pixel. If not, put it as seed pixel; if is, end this algorithm.

3) Judge whether seed pixel is true or not. If false, go to step (2); if true, made its value into zero and put it into queue.

4) Judge whether the queue is empty or not. If empty, go to step (2); otherwise, obtain the four neighborhood pixels of head pixel of the queue successively. If the four neighborhood pixel values are true, made them false and put them into the queue; if not, ignore it.

Algorithm1 Image Processing

Input: gray Image
Turn into binary image
Read edge pixel and signed flag
Get next pixel, pixel = flag?
If not, pixel = seed pixel
If is, end this algorithm
seed pixel = true?
If is, seed pixel = false,
Queue num = num+1
If not, read next pixel
Queue is Empty?
If is, four neibourhood pixel process,
If not, read next pixel
Output: Preprocess Image

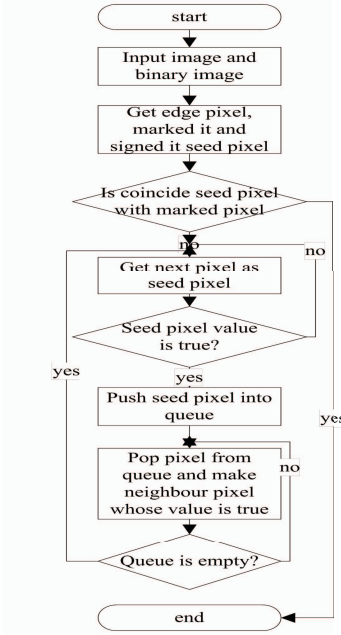


Figure 3. Flow chart of image preprocess algorithm

Seed algorithm has advantages and disadvantages. The seed algorithm is complex, and has longer running time, also robust. This algorithm is almost suitable for complex image.

B. Across boundary detection algorithm for defect

The main defect of image is that whether print character is out of bar bounds. Adopt BLOB analysis technique for detection. The theory is as follows: firstly, make the image into binary image, and then conduct BLOB analysis operation on binary image, getting the number of larger connected region. If the number of connected regions is less than a specified proportion, it is showed that printing character in the silver bar code has defects; otherwise, there is not.

Algorithm2 Across boundary detection

Input: gray Image
Turn into binary image
BLOB analysis

Remove smaller region
Output: retain larger region

C. *Image segmentation algorithm*

Recognize the character which is print in the silver bar code and reduce the interference to the character recognition, make the operation of segmentation on nameplate image's silver bar code region, the theory of algorithm is as below: firstly, preprocessing image to binary image and BLOB, eliminate smaller connected region and retain the larger. The function cvBoundingRect can easily get four coordinates of larger connected region and extract the reference image region.

Algorithm3 Image segmentation

Input: gray Image
Turn into binary image
BLOB analysis
Remove smaller region
retain larger region
cvBoundingRect get four coordinates
extract the image region
Output: segmentation image

D. *Character recognition algorithm*

For image without defects, the main steps of recognizing the printed character are as follows:

1) The template character is made manual separated, and BLOB analysis for each template character. The template character region is divided partition by7*5. Calculate the black and white pixels scale of each partition, and save them to the file.

2) Judge printed characters whether it is reversed. If not, go to steps (3); if is, reverse image.

3) Access the file data, make the segmentation gray level image into binary, and make BLOB analysis, then extract regional characteristics in specified size, and remove smaller connected regions[7-8].

4) Get the characters position of BLOB characters[9], and sort, ensuring the order of BLOB character sequence and actual.

5) Successively divide characters into 7*5 region, calculate the black and white pixels scale of each partition, and compute Euclidean distance for template character as shown in formula (1), obtaining the minimum distance value. In this way, consider that the recognized character is consistent with the template character, then make end for the character recognition.

$$Dist = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (1)$$

Algorithm4 Character recognition

Input: gray Image
Turn into binary image
Manual separate template character;
BLOB template character
Template character divided 7*5
Calculate black and white pixels scale of each partition and save it to file
Sort character which is recognized;
Read file data and BLOB template character;
Template character divided 7*5
Calculate black and white pixels scale of each partition
$Dist = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$
Calculate the distance by formula and get Min one, then think it is character
Output: Character recognition

For this algorithm, it mainly depends on how many the region is divided. Divided region is less, run time is shorter, but the recognition of precision is lower; with regions numbers increasing, run time is longer, accuracy is greatly improved. When the division region is up to a certain extent, speed will be slow and accuracy will not be improved. Therefore, thinking about precision and time of recognition, the final choice is dividing Blob character into 7*5 region.

IV. THE RESULTS OF EXPERIMENT

A. *Image Preprocess*

This paper adopts seed algorithm to retain a complete nameplate, removing the incomplete nameplate which place in up and bottom.

Figure 4 is the original image gathered by industrial camera which on the up is not complete and includes printed characters. There are some interferences for process. At the same time, retain the integrity of the nameplate and eliminate unnecessary interference factors, what's more, reduce the image processing region, and greatly improve the image processing speed. The experimental results of seed algorithm is shown in figure 5: the algorithm's time complexity is O(m*n), the running time is1.90s.

Figure 4. Nameplate image acquisition and image

Figure 5. Seed algorithm

B. *Character boundary detection*

The accurate detection of image preprocessing algorithm adopts BLOB analysis technology which can judge the printed character whether across boundary or not. As shown in Figure 6, the printed character is cross-border, such as the elliptical section shown in this picture, the printed character is beyond the silver bar region. By the BLOB analysis technology, this paper employs a rectangular to enclose connected image region, suggesting that the silver code is a connected region, then deduces that printed the characters have already cross-border, which is considered as unqualified products, makes direct alarm or removes it. Figure 7 is

shown that the printed character is not cross-border. Printed character is full displayed in silver bar code. After the BLOB analysis, two rectangles enclose gray image region, showing that there is not defect between printed character and silver bar code, it is qualified products and next following character identification procedures can be done.

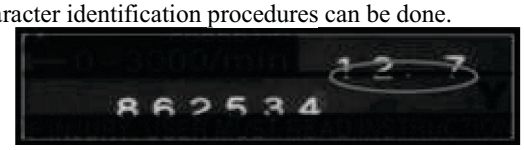


Figure 6. Characters printed cross-border effect chart

Figure 7. Characters printed not overflow effect chart

C. Image segmentation and character recognition

Figure 8 is segmentation image about the recognized character region. Adopting image segmentation algorithm can extract complete character recognition in the silver bar. Thereby, it can remove the unnecessary image, and greatly reduce processing data for the algorithm and the processing time. Figure 9 is shown a recognized character. Through a series of processing for character recognition, it achieves effects of accurate recognition of character. The gray image in the figure shows that the recognized character is "1260517", and the white part of image shown "1260517" is recognized character. Experiment shows that, the character recognition algorithm is feasible and effective, and the running time of this algorithm is 1.86s.

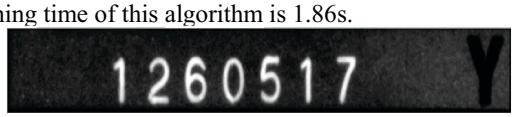


Figure 8. Recognize the character image segmentation

Figure 9. character recognition effect chart

V. KNOT THEORY

The research method of this paper for plate defect detection and character recognition is based on machine vision. This method employs image processing technology to the nameplate and judges printed character whether beyond the silver bar code; and then separates the specific region, and utilizes the character recognition algorithm to identify character; finally, the image processing software saves the results to the report file, so that the quality inspection personnel can quickly search report information, conveniently search results, also pass it to the motion control part for the unqualified products processing. Experiments show that: the nameplate defect detection and character recognition system based on machine vision, can be highly efficient, and has higher quality detection and character recognition for the nameplate defects.

REFERENCES

[1] Zhao Weijie, Gao Yong. The application of machine vision in the detection ofdefects in optical fiber[J].modernelectronictechnique, 2011,34 (19):136-137

[2] Feng Pei, Liu Jiaqiang, Qiu Da Hong, Yang Chongchang, Gan Xuehui. Machine vision system for spinneret microhole automatic detection and control based on[J]. equipment, 2012,35 (1):64-66

[3] Shi Bingxia [D]. of machine vision in real-time character recognition technology. Hebei: Hebei University of Technology, 2010

[4] Yang Xue. Algorithm of image detection in machine vision research andapplication [D]. Jiangsu: Jiangnan University, 2013

[5] Shuxia Li, Duoyu Gu, Hongxing Chang. Bearing defect inspection based on machine vision[J]. Measurement, 2012,45(4):719-733

[6] Majid Dowlati, Miguel dela Guardia.Application of machine-vision techniques to fish-quality assessment[J].TrAC Trends in Analytical Chemistry, 2012,40:168-179

[7] Xu Min, Tang Wan, Ma Qianli, Hao Jianqiang. Research on [J]. Packaging engineering online printing defect detection based on Blob algorithm, 2011,32(9):20-23

[8] Zheng Chengyong, Li Hong. The license plate character analysis of the overallcharacteristics and blob character segmentation based on [J]. Journal of Huazhong University of Science and Technology (NATURAL SCIENCE EDITION),2010,38 (3):88-91

[9] Zhang Dongjuan, Tang Wan. Research on [J]. Packaging Engineering Online Defect stamping detection based on Blob algorithm, 2013,34 (17):16-19

Author Index

Author Index

IEEE Computer Society
Technical & Conference
Activities Board

IEEE Computer Society Publications

The world-renowned IEEE Computer Society publishes, promotes, and distributes a wide variety of authoritative computer science and engineering texts. These books are available from most retail outlets. Visit the CS Store at *http://www.computer.org/portal/site/store/index.jsp* for a list of products.

IEEE Computer Society *Conference Publishing Services* (CPS)

The IEEE Computer Society produces conference publications for more than 300 acclaimed international conferences each year in a variety of formats, including books, CD-ROMs, USB Drives, and on-line publications. For information about the IEEE Computer Society's *Conference Publishing Services* (CPS), please e-mail: cps@computer.org or telephone +1-714-821-8380. Fax +1-714-761-1784. Additional information about *Conference Publishing Services* (CPS) can be accessed from our web site at: *http://www.computer.org/cps*

Revised: 18 January 2012

CPS Conference Publishing Services — CPS Online™ IEEE Online Collaborative Publishing Environment

CPS Online is our innovative online collaborative conference publishing system designed to speed the delivery of price quotations and provide conferences with real-time access to all of a project's publication materials during production, including the final papers. The ***CPS Online*** workspace gives a conference the opportunity to upload files through any Web browser, check status and scheduling on their project, make changes to the Table of Contents and Front Matter, approve editorial changes and proofs, and communicate with their CPS editor through discussion forums, chat tools, commenting tools and e-mail.

The following is the URL link to the ***CPS Online*** Publishing Inquiry Form:
http://www.computer.org/portal/web/cscps/quote

Published by the IEEE Computer Society
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

IEEE Computer Society Order Number E5396
ISBN 978-1-4799-4170-4
BMS Part Number CFP1420K-CDR