Resource Scheduling Optimization Algorithm of Energy Consumption for Cloud Computing Based on Task Tolerance

Liang Hao
Department of Computer Science and Technology, Harbin Institution of Technology, Harbin, China
Email: haoliangoorr@163.com

Gang Cui, Mingcheng Qu, Wende Ke
Department of Computer Science and Technology, Harbin Institution of Technology, Harbin, China
Email: cg@ftcl.hit.edu.cn

***Abstract*—In order to solve that the energy was seriously wasted in cloud computing, an algorithm of energy consumption optimization for cloud computing based on the task tolerance (ECCT) was proposed. The concept of the task tolerance was put forward in the algorithm. The resource utilization of cloud computing node was maximized through increasing the task tolerance. At the same time, the parallelism degree of tasks were increased so that the task waiting time was reduced, which led to a reduction of energy consumption in cloud computing. Parameters were estimated and experiments on actual cloud computing sets were illustrated. The results showed that the total energy consumption and executive time were reduced more in the proposed algorithm than sequential executive algorithm (SEA). In addition, the advantage would be increasingly apparent with the increasing in assignments.**

***Index Terms*—cloud computing, task tolerance, resource utilization, energy consumption optimization**

I. Introduction

With the gradual warming of the global climate, as well as the low-carbon economy and the green energy was proposed, the energy consumption problem in cloud computing has become a key issue [1]. And the energy consumption problem is particularly serious in cloud data centers. According to statistics, there were about 6000 data centers in United States in 2006. These data centers consumed about 610 billion kilowatt hours electrical energy, which was worth up to $4.5 billion and more than the total energy consumption of all color television sets in United States that year. The data from the U.S. Department of Energy showed that the energy consumption of data centers accounted for 1.5% of all energy consumption in United States, and the demand for electricity energy was still rising at a speed of 12% per annum. If the energy continued to rise at the speed to 2011, the data centers would consume 100 billion kilowatt-hours of electrical energy, spending approximately $7.4 billion each year. In addition, the assessment of International Data Corporation market research firm on electrical energy of all the enterprises in the global showed that, they spend about $40 billion on energy consumption each year [2-5]. The reasons resulting that network flow load and the energy consumption in network was gradually increased. But the energy issues of cloud computing are rarely taken into account in resource scheduling algorithm [6-9]. Therefore, how to solve the energy consumption problem of cloud computing in the resources scheduling algorithm has become a research focus gradually.

With the energy consumption issues and the shortage of resources become more and more serious, how to optimize the energy consumption of cloud computing by using the limited resources is particularly critical. In this paper, the resource utilization of data centers, the energy consumption situation, and the task tolerance in cloud computing were studied deeply. An algorithm was designed, in which the energy consumption of cloud computing was optimized through increasing the task tolerance using the limited resources. The simulation results showed that the executive time was reduced and the energy consumption was optimized through taking full advantage of resources.

II. Related Work

There are some distributed resource scheduling algorithms at this stage, such as the article [10], an algorithm was designed about software services of cloud computing to optimize the allocation of cloud computing service resources, which met the service provider at the same time maximized the profits. But the executive time was increased and energy was wasted. In the article [11-14], Rajni proposed to use bacteria foraging optimization technology of grid resource scheduling to design a new bacterial foraging heuristic resource

This work is supported by the National Nature Science Foundation under grant No.61272382 and NO. 61075076, the Key Program of National 863 project under grant 2007AA041603 and 2006AA04Z259
Corresponding author: Liang Hao

doi:10.4304/jsw.9.4.895-901

scheduling algorithm, which could take advantage of the resources to arrange work effectively in a grid environment. The performance of the algorithm was assessed through the toolkit of gridsim in conjunction with the existing scheduling algorithm. The application of user was submitted to the grid by reducing the cost and finished time. But the algorithm lacked of considering the task transmission overhead in the scheduling process, and without considering the dynamic characteristics of the network tasks and the multiple performance indices of tasks. In the article [15], Yifei Tong introduced an allocation algorithm of market mechanism. The resources were reasonably allocated according to the requirements of the market mechanism in the algorithm. And the resources were scheduled according to quality of service as the target. But only the efficiency of resource scheduling was optimized in the algorithm, without considering a wide range of service quality parameters. And the computing capacity and throughput capacity of the proxy node, as well as the transmission overhead required were not considered in the algorithm.

Some studies about cloud energy consumption were proposed recently, such as the article [16-17], Jayant carried out a detailed measurements and statistics on the energy consumption of transfer and conversion in the public cloud and private cloud, the energy consumption of three services of cloud computing that software as a service, storage as a service, platform as a service were analyzed, the energy consumption of computers, routers, data centers were listed in detail and the energy consumption situation of the three services were compared. However, no specific algorithms to reduce the energy consumption was generated in the cloud computing. A simulation environment of cloud computing with a plurality of data centers was designed, the energy consumption of the transmission, conversion, connection between multiple data centers were designed through the actual distribution of the workload, the impact that power control such as voltage, frequency, dynamic switching was applied to the computing and networking was demonstrated [18-20]. In the contributed volume [21-22], a detailed investigation about the hardware, operating system, virtual machine, and data centers was made, the relationship between high-power and the energy consumption of cloud computing was analyzed in detail and the energy efficiency in the future was discussed deeply. In the contributed volume [23], Berl reviewed the usage method and techniques on the energy consumption operation of computer hardware and network equipment, conducted a survey of methods in this field, distinguished the energy consumption technology used in the cloud computing. But there was no specific energy consumption algorithm expression, only elaborated on the future energy consumption model.

There never was the cloud energy research in the perspective of resource scheduling in existing cloud energy consumption model. In view of this situation as well as deficiencies in the existing studies, resource scheduling optimization algorithm of energy consumption for cloud computing based on the task tolerance was proposed in this paper, which reduced the energy consumption of implementation and optimized the executive time by improving the parallelism degree of tasks.

III. PROBLEM MODEL

The executive energy consumption of cloud computing was mainly focused in this paper, so all cloud data centers ran at full capacity was assumed. In the task scheduling model of cloud computing, computing resources, storage resources, network resources, etc. existed in the form of services. The task scheduling model of cloud computing is shown in figure 1.

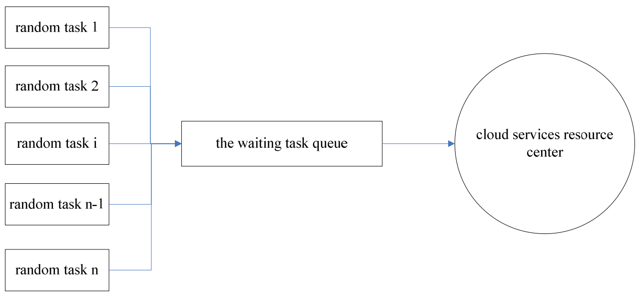

Figure 1 the task scheduling model of cloud computing

In figure 1, a set of random tasks is scheduled to cloud service resource nodes for executing through the tasks waiting queue. The arrival time of random tasks is uncertain, ranging from resource requirements. The study assumed that the tasks waiting queue was long enough, and cloud service resource center was responsible for scheduling task to execute reasonably. To further illustrate the energy consumption of resource scheduling algorithm, we give some definition of the resource model in the following.

Definition 1, the set of random tasks is defined as $R(T, E, A)$, T is a set containing n tasks, $0 \le i \le n$. If i is not equal to j, T_i is not equal T_j. $E(i, j)$ is the set of directed edges between the tasks, while $i < j$, the task T_i will be executed after T_j is finished. A is the set of tasks' workload requirements.

Definition 2, the cloud service model is defined as (C, B, U, P_1, P_2), where C is the set of computing nodes. B is the matrix where random tasks corresponding to computing nodes. B_{ij} represents the task t_i performing on computing node C_j. U is the set of resource utilization for each computing node, P_1 is the peak power of computing nodes, and P_2 is the idle power of computing nodes.

Definition 3, the task tolerance is defined as the ratio of the longest time for the task can be executed and the actual required time, the formal expression is $\tau = \frac{T}{t} \times 100\%$, where T is the longest time for the task can be executed, t is the actual required time.

Definition 4, the total energy consumption of cloud services is defined as E, f is CPU frequency of cloud computing nodes, c is the utilization of computing nodes,

τ' is defined as the actual tolerance when a task is executing. When the task entered the cloud resources service center from the waiting queue, the task was scheduled to the current relatively idle node for executing. Resource utilization within the unit time was increased by improving real-time tasks tolerance. The problem model is as follows:

$$\tau'_k \leq \tau_k \quad (1 \leq k \leq n) \tag{1}$$

$$\mathrm{s.t.}\ \sum_{k=1}^{n} x_{ik} = 1, \quad \forall i \in V \tag{2}$$

$$x_{ik} \in \{0,1\} \tag{3}$$

$$z_i \ll z_j - \sum_{k=1}^{n} t_{ik} x_{ik} \tag{4}$$

$$\mathrm{Max}\, U_i \qquad (1 \leq i \leq m) \tag{5}$$

In the above formula, x_{ik} is a Boolean variable, which means that the task can only select a service to execute. z_i represents the end time of each task. (1) represents the task tolerance should be less than or equal to the agreed tolerance of itself. (2) represents the task can only select a service to execute every time. (3) represents x_{ik} is a Boolean variable (4) represents the task should meet partial order (5) represents the resource utilization of each node was maximized.

IV. Iterative Inspired Algorithm

A. Algorithm Description

In resource scheduling optimization algorithm of energy consumption for cloud computing based on the task tolerance (ECCT), the total number of nodes of cloud computing, the number of tasks and the initial task tolerance were given. The cloud system checked the resource utilization of each cloud computing node firstly when there was the task in the task waiting queue. If the resource utilization of any cloud computing node was not 100%, tasks were scheduled to the cloud node for executing. When the task tolerance was less than or equal to the agreed tolerance of itself, the task was scheduled into the cloud computing node, and the node of relatively low utilization was selected to schedule the task for executing. At the same time the resource utilization of node was real-time collected. Otherwise, next task of the quest chain queue was scheduled for executing. The iterative calculation was done, and the total energy consumption was calculated when the resource utilization of every computing node was 100%. Algorithm is as follows:

```
For all computing codes and incoming tasks
Generate  τ', τ, U_i, R
End for
   For all computing codes
   Compute U_1, U_2, ... U_m
         For (i=1; i<m; i++)
             For(j=i+1; j≤ m; j++)
                  If U_i > U_j
                  U=U_i ; U_i=U_j ; U_jU ;
                  End if
             End for
         End for
      End for
While R > 0
     For (i=1; i≤m; i++)
         While U_i < 100%
             For (k=1; k<n; k++)
                If   τ' ≤ τ
                Scheduling T_k to computing code C_i
                Compute U_i
                End if
             End for
         End while
     End for
End while
Renturn E
```

B. Algorithm Analysis

In ECCT algorithm, the utilization of energy resources in the unit time was increased by increasing the parallelism degree of tasks, which was increased by improving the task tolerance, and the energy consumption was optimized finally. Optimization rules: First of all, tasks were scheduled to the lowest resource utilization of cloud computing nodes for executing, followed by tasks were scheduled successively to the maximum resource utilization of the computing node until the resource utilization of all cloud computing nodes is 100%, the difference calculation of the energy consumption as follows:

$$\Delta E = \sum_{i=1}^{m} \int_0^t f_i(t) c_i(t) - m P_1 t' \tag{6}$$

As can be seen from equation (6), when the cloud services ran at full capacity, each node power of the data center was the peak power P_1, the average executive power was increased a little, the executive time was shortened for a lot, wherein the total executive time is calculated as follows:

$$T = \sum_{i=1}^{m} \frac{\sum_{k=1}^{n} w_k}{f_i c_i} \tag{7}$$

As can be seen from equation (7), with the improvement of the CPU utilization of computing nodes, the executive time was gradually shortened. Thereby the total energy consumption was reduced. The resource-time graph can be used to represent services in cloud computing, ECCT algorithm's resources-time graph is shown in figure 2 and figure 3.

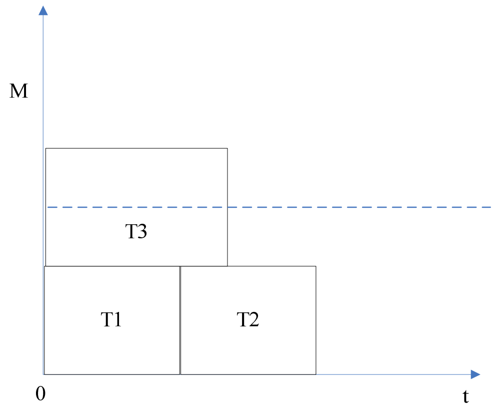

Figure 2 resource-time without tasks tolerance

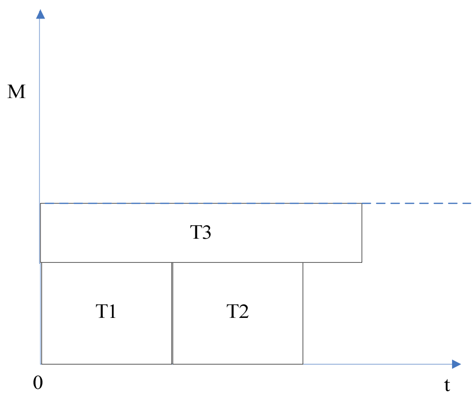

Figure 3 resource-time with tasks tolerance

In figure 2, the abscissa t represented the task executive time, the longitudinal axis M indicated the service resources of cloud computing node, and the dotted line was the actual resource capacity of the cloud node. When the task T_1 and T_2 was running, some part of cloud resources were idle. When the task T_3 arrived, the existing resources couldn't meet the resource requirements of T_3, so T_3 could be executed until the end of the task T_1 and T_2. If so, it was a waste of resources. In figure 3, the task tolerance was increased by utilizing the function of the CPU executive time division multiplexing. T_3 used a small amount of resources to complete the scheduled execution, the executive time of the task was increased, but the overall task executive time $t_1 + t_2 + t_3$ was shortened. The area of task T_3 in figure 3 was the same size as figure 2 by calculating.

C. The Complexity of ECCT Algorithm

The complexity of ECCT algorithm depends on the gap of the resource utilization between the initial resource usage and full load operation. Because the iterative number of the algorithm was determined by the gap. The maximum resource utilization was got close gradually after each Iterative calculation of the algorithm, which was the peak power of the node. So the algorithm is convergent. The complexity of the algorithm depends on the number of cloud nodes in data center, which is related with its iterative. When resource utilization is close to 100%, the tasks are calculated iteratively. So the maximum complexity of the algorithm is $|mN|$, m is the selectable cloud service nodes, N is the total number of tasks.

V. Experimental Analysis

In order to test the performance and operating efficiency of ECCT algorithms in different modes, the experimental environment was composed of 12 PC whose systems was Windows 7 kernel, CPU was Inter Core (TM) 2 Duo 2.20GHz, RAM was 4G, hard disk was 320GB, the network bandwidth was 100M, and the algorithm was programmed using the Java programming language.

In order to compare the performance of ECCT algorithm, the sequential executed algorithm (SEA) was proposed in the study. The SEA algorithm's basic idea was: when there are the tasks in the task waiting queue, the tasks were scheduled serially to cloud computing nodes for executing, the task tolerance didn't need to be considered. The task was scheduled directly to the next node for executing when the resource requirements of task can't be met. The SEA algorithm would be end until there was no idle service node in the data centers. The three parameters of the scale of the task, the number of nodes of the data center, the task tolerance were selected in the experiment. The number of the experimental task nodes are set to 8, 16, 32, 64 nodes, the tasks were simulated to {20, 40, …, 100} several sizes, the tasks tolerance was simulated to 100%, 140% ... 400%.

The experimental results were taken from the average of different sizes. The average energy consumption, the average executive time and other aspects were selected in the algorithm to compare comprehensively the relationship between the two algorithms. If the average value of the same data set in the two algorithms were in comparison, such as the comparison of the average energy consumption, improvement rate of the algorithm' performance is calculated as: $\frac{E_B - E_A}{E_A} \times 100\%$.

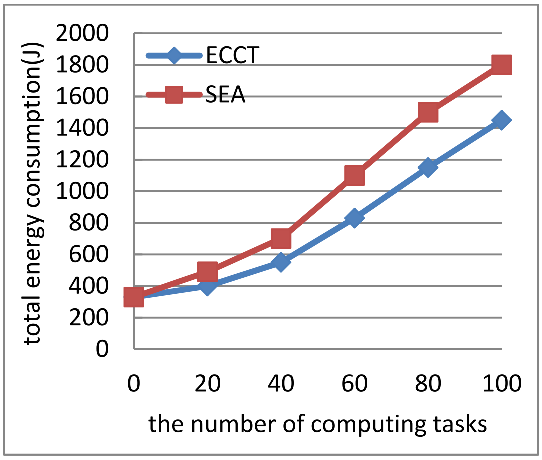

Figure 4 the number of tasks with energy diagram

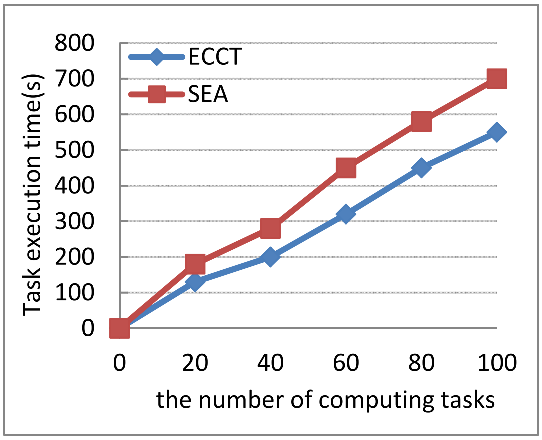

Figure 5 the number of tasks with time diagram

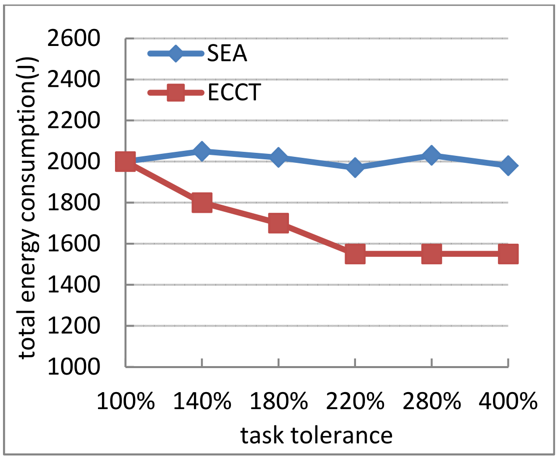

Figure 6 the task tolerance and energy diagram

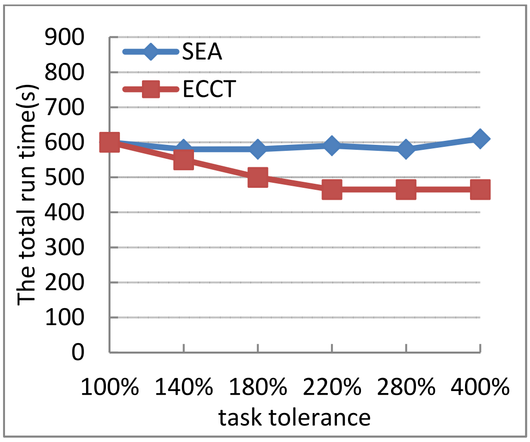

Figure 7 the task tolerance and time diagram

As can be seen from figure 4, with the increase in the amount of tasks, the energy gap of the two algorithms was gradually increased, which was mainly due to the task tolerance was improved in ECCT algorithm, the system resources of cloud computing are fully utilized to increase the parallelism degree, and reduce the relatively idle time of resources. As can be obtained by the average value of the different scale of the tasks, the average energy consumption of ECCT algorithm was improved by 12.64% compared to SEA algorithm. The advantage would continue to increase with the increase in the amount of tasks. Similarly, the relationship of number of tasks and time was shown in figure 5, the gap in the total run time was increasingly obvious with the increase in the amount of system tasks. Due to the increase of task tolerance, the full utilization of resources, and a relative reduction of a large number of tasks waiting time, so the advantage of time would be increasingly apparent with the number of tasks gradually was increased.

As can be seen from figure 6, SEA algorithm was irrelevant to the task tolerance, so as to the increasing of the task tolerance, the total energy consumption of SEA algorithm was approximated as a straight line. The total energy consumption of ECCT algorithm decreased gradually with the increasing of the task tolerance. The energy consumption was not changed when the task tolerance reached a certain value. The main reason for this phenomenon was: Even if more tasks were executed with the function of time-multiplexed when the resource utilization reached 100%, the total executive time was not changed, while the system was always maintained at the peak power, so the energy consumption was not changed. Similarly, the relationship of the task tolerance and executive time was shown in figure 7. The total executive time was not changed with the tolerance was gradually increased in SEA algorithm. However, the total executive time of ECCT algorithm was decreased gradually. It would not continue to be changed when the task tolerance reached a certain value.

VI. Conclusion

Resource scheduling optimization algorithm of energy consumption for cloud computing based on the task tolerance was proposed by the study of the energy consumption in cloud computing. The task waiting time was reduced by improving the task tolerance, through which the service resources of cloud computing were fully utilized. At the same time the parallelism degree of tasks was improved, and the energy consumption of cloud computing was reduced. As can be seen from the large number of experimental results, the total energy consumption of the cloud system and the total executive time would be reduced with the task tolerance was gradually increased, and the advantage of ECCT algorithm in saving energy would become increasingly obvious with the amount of tasks was gradually increased.

ACKNOWLEDGMENT

This work is supported by the National Nature Science Foundation under grant No.61272382 and NO. 61075076, the Key Program of National 863 project under grant 2007AA041603 and 2006AA04Z259.

REFERENCES

[1] Young C L, Albert Y Z, Energy efficient utilization of resources in cloud computing systems, J Supercomput, 2012(60):268–280
[2] Ye Kejiang, Wu Zhaohui, Jiang Xiaohong, Power Management of Virtualized Cloud Computing Platform, Chinese Journal of Computers, 2012, 35(6): 1262-1285
[3] Michael Cardosa, Aameek Singh, Exploiting Spatio-Temporal Tradeoffs for Energy-Aware MapReduce in the Cloud, IEEE transactions on computers, 2012(61):1253-1265
[4] Chunlin Li, La Yuan Li, Optimal resource provisioning for cloud computing, J Supercomput, 2012(62): 989–1022
[5] Rajni, Inderveer Chana, Bacterial foraging based hyper-heuristic for resource scheduling in, Future Generation Computer Systems, 2013(29): 751–762
[6] Tong Yifei, Li Dongbo, He Yong, A QoS-based resource economic scheduling in manufacturing grid, 2012(18): 484-491
[7] Jayant Baliga, Robert W A, Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport, 2011(99): 1280-1307
[8] Dzmitry Kliazovich, Pascal Bouvry, Green Cloud: a packet-level simulator of energy-aware cloud computing data centers, J Supercomput, 2012(62): 1263–1283
[9] Beloglazov, Rajkumar Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers, 2012(24): 1397–1420
[10] Andreas Berl, Erol Gelenbe, Energy-Efficient Cloud Computing, Advance Access publication, 2009(19):1245-1261
[11] Ivan R, Hariharasudhan V, Eun K L, et al. Energy-Efficient Thermal-Aware Autonomic Management of Virtualized HPC Cloud Infrastructure. J Grid Computing, 2012, 10(1): 447–473
[12] Oinn T, Addis M, Ferris J, Senger M, Greenwood M, Carver T, Glover K, Wipat A, Li P. Taverna: A tool for the composition and enactment of bioinformatics workflows. Bioinformatics, 2004, 20(17) : 1045-1054
[13] Anton B, Jemal A, Rajkumar B. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. Future Generation Computer Systems, 2012, 28(1): 755–768
[14] Wang L, Tao J, Castellanos A C, Kramer D, Karl W. Scientific cloud computing: Early definition and experience//Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC) . Dalian, China, 2008: 826-831
[15] Lien D, Bert V. Efficient resource management for virtual desktop cloud computing. J Supercomput, 2012, 62(1): 741–767
[16] Venugopal S, Ramamohanarao K. A taxonomy of data grids for distributed data sharing, management, and processing. ACM Computing Survey, 2006, 38(1) : 12-46
[17] Doraimani S, Jamnitchi A. File grouping for scientific data management: Less on s from experimenting with real traces//Proceedings of the 17th International Symposium on High Performance Distributed Computing. Boston, MA, USA, 2008: 245-256
[18] Armbrust M, Fox A, Griffith R. A view of cloud computing .Communications of the ACM, 2010, 53(4): 49-61
[19] Zhu X, Young D, Wang Z et al. 1000 Islands: An integrated approach to resource management for virtualized date centers. Cluster Computing, 2008, 12(1): 39-61
[20] Ajiro Y, Tanaka B. Improving packing algorithms for server consolidation//Proceedings of the 33rd International Computer Measurement Group Conference. San Diego, 2007: 401-407
[21] Ghalem B, Samah B. An effective economic management of resources in cloud computing. Journal of computers, 2011,3(6):404-411
[22] Ying J, Qiuwen Z. A fuzzy comprehensive assessment system of dam failure risk based on cloud model. Journal of computers, 2013,8(4):1043-1049
[23] Feng L, Shilong M. Advance reservation-based computational resource brokering using earliest start time estimation. Journal of computers, 2012,6(7):1329-1336

Liang Hao was born in 1984 in China. In 2006, he received his B.S. degree in Department of Computer Science and Technology from Shenyang University of Technology at Shenyang. He earned his M.S. degree in 2009 in Department of Computer Science and Technology from Shenyang University of Technology at Shenyang. He is currently a Ph.D. candidate in Department of Computer Science and Technology at Harbin Institute of Technology. Her main research interests include cloud computing, network optimization and scheduling, fault tolerance computing and mobile computing.

Gang Cui was born in 1947 in China. He earned his M.S. degree in 1989 and B.S. degree in 1976, both in Computer Science and Technology from Harbin Institute of Technology at Harbin. He is currently a professor and Ph.D. supervisor in School of Computer Science and Technology at Harbin Institute of Technology. He is a member of technical committee of fault tolerant computing of the computer society of China. His main research interests include fault tolerance computing, mobile computing, software testing. Prof. Cui has implemented several projects from the National 863 High-Tech Project and has won 1 First Prize, 2 Second Prizes and 3 Third Prizes of the Ministry

Science and Technology Progress. He is a senior member of the CCF. He has published over 50 papers and one book.

Mingcheng Qu, born in 1980, he received his B.S. degree in Department of Computer Science and Technology from Harbin Institute of Technology at Harbin. He earned his M.S. degree in 2007 and Ph.D. degree in 2011 in Department of Computer Science and Technology from Harbin Institute of Technology at Harbin. He is currently a lecturer in Department of Computer Science and Technology at Harbin Instituted of Technology. His main research interests include cloud computing and grid computing.

Wende Ke was born in 1976 in China. He obtained his Bachelor in Computer Science from Wuhan Technique University of Surveying and Mapping in 1999 and his Master in Computer Science from Huazhong University of Science and Technology in 2005. Currently, he is a PhD candidate in Harbin Institute of Technology and an Associate Professor in Guangdong University of Petrochemical Technology. His research focuses on locomotion control of humanoid robot, computer architecture, etc.