International Journal of Hybrid Information Technology
Vol. 9, No.6 (2016), pp. 403-412
http://dx.doi.org/10.14257/ijhit.2016.9.6.36

# A Task Scheduling Based on Simulated Annealing Algorithm in Cloud Computing

Xi Liu[1,a] and Jun Liu[2, b]

[1]*College of Information Science and Engineering, Yunnan University, Kunming Yunnan China*
[2]*College of Mathematics and Information Science, Qujing Normal University, Qujing Yunnan China*
[a] *lxghost@126.com* , [b]*liujunxei@126.com*

***Abstract***

*Because the task scheduling problem is np-complete problem, it is hard to find a deterministic algorithm to solve the problem of task scheduling in the cloud computing platform. Therefore this paper presents a task scheduling mechanism based on simulated annealing algorithm. The algorithm is a modern heuristic algorithm and overcome the shortcoming of the local optimum search method. The algorithm uses a greedy algorithm to generate the initial value, and heat to a sufficiently high temperature and according to certain rules to generate a new value. If the new value is better than the original value or at a certain probability can be accepted, then replace the original value with the new value until cool. Experiments proved the feasibility and effectiveness of the algorithm. Compared with traditional algorithms, this task scheduling mechanism not only meets the needs of users and improves the performance of the system.*

***Keywords****: cloud computing, virtual machine, scheduling optimization.*

## 1. Introduction

Cloud computing[1-2] has become a hot research applications and is a new computing model which is the development of distributed computing, parallel computing and grid computing. Cloud Computing is not a totally new concept. It has its root in Grid Computing paradigm, and other technologies like cluster computing, utility computing and distributed computing in general. It can be defined as utilization of computer resources be it hardware and software from any device, anytime, any where and you pay as you need. The national institute of standards and technology to define the basic characteristics of cloud computing is on-demand self-service and rapid elasticity[3]. Cloud Computing can be define in various terms. Here we present some of the key definitions of cloud: Cloud Computing is a broad array of web-based services aimed at allowing users to obtain a wide range of functional capabilities on a " pay-as-you-go" basis that previously required tremendous hardware and software investments and professional skills to acquire.

There are three basic types of Service Models depending on the service each layer provides to its upper layers [4-6]. The three models are: Infrastructure as a Service (IaaS): When user has got the maximum flexibility to deal with the Infrastructure, where he can even install his Operating System it is known as Infrastructure as a Service. In it the provider makes entire infrastructure available as a service. The end user can use the entire infrastructure on pay per usage basis. The ability to support an IaaS architecture is through a combination of some of the special characteristics of cloud computing. Platform as a Service (PaaS): A cloud platform offers an environment on which developers create and deploy applications and do not necessarily need to know how many processors or how much memory that applications will be using. In it the cloud provides software platform or middleware as service on which the



ISSN: 1738-9968 IJHIT

applications run. The user is responsible for creation, updating and maintenance of the application. Software as a Service (SaaS):This is the topmost layer of the architecture, it has all the applications residing in it. Applications are provided as a Service and these Services can be accessed by end users through Web portals. Therefore, consumers are increasingly shifting from locally installed computer programs to on-line software services that offer the same functionally.

To achieve these characteristics, the main use virtualization and its related technologies [7]. Cloud computing is mainly use virtualization technology. The current virtualization [8-9] is defined as :(1) The object of virtualization are all kinds of resources. (2) The user does not need to know the details of the hidden in the logical resources virtualize process. (3) In the user experience of virtualization, users can implement part or all of the functions in the real environment. Cloud computing services require a big data center for almost around thousand computers to server many users and host applications. Virtualization means something which is not real but gives all the facility of a real. Virtualization is a software implementation of a machine which will execute different program like a real machine. Two type of virtualization found in case of cloud computing. Para Virtualization: It means the hardware allow multiple operating systems to run on a single machine. In pare virtualization all the services are not fully available rather the services are provided partially. Full Virtualization: It means a complete installation of one machine is done on another machine.

## 2. Related Work

The load balancing model given in this article is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. Thus, this model divides the public cloud into several cloud partitions. When the environment is very large and complex, these divisions simplify the load balancing. The cloud has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy. Static and dynamic two type of load balancing approaches use in cloud computing. Static Approach: It is mainly defined in the design or implementation of system. It divides the traffic equivalently between all users. It uses only information about the average behavior of the system. These are much simpler and ignore the current state or the load of the node in the system. Dynamic Approach: In this approach considered only the current state of the system during load balancing decision. It is more suitable for widely distributed system such a cloud computing. Centralized Approach: Only s single node is responsible for managing and distribution within the whole system. Distributed Approach: Each node independently builds its own load vector. Vector collection load information of other node. All decision is made locally using local vector.

Metrics for Load Balancing: Throughput: It is used to calculate the all tasks whose execution has been completed. The performance of any system is improved if throughput is high. Fault Tolerance: It means recovery from failure. The load balancing should be a good fault to tolerant technique. Migration time: It is the time to migrate the jobs or resources from one node to other nodes. It should be minimized in order to enhance the performance of the system. Response Time: It is the amount of time that is taken by a particular load balancing algorithm to response a task in a system. This parameter should be minimized for better performance of a system. Scalability: It is the ability of an algorithm to perform Load balancing for ant finite number of nodes of a system.

Task scheduling problem in the cloud computing is a NP-hard problem. Existing task scheduling algorithm of cloud computing platform such as FIFO, fair scheduling, capacity scheduling have shortcomings, that difficult to well meet the needs of practical application. Therefore, some intelligent optimization algorithm is proposed, such as inheritance algorithm, particle swarm optimization algorithm and ant colony algorithm. But this algorithm which has their own shortcomings still remains in the stage of exploration and research.

 

Walsh based on utility function use different hosts dynamic management[10-11]. Beloglazov and others use the best adaptation algorithm to deploy the virtual machine[12]. Bobroff and others use not to violate the service level of dynamic management method to predict the virtual machine[13]. Qiudongli and others use the resource requirements of vector method based on queuing theory to deploy virtual machine[14].

## 3. Virtual Machine Selection Strategy Based on Simulated Annealing Algorithm

### 3.1. Simulated Annealing Algorithm Description

There are two ways to solve real problems[15-16]. One is use the traditional method to establish the approximate model to obtain accurate solution. The other is to create a more accurate model, by using approximate algorithm to obtain approximate solutions. The latter is often in all aspects of performance than the former. Local search algorithm means: assuring the problem solution space is S, this algorithm starts with an initial value i which belongs to S. According to the specific problem areas, within the field of Si in the current solution i according to certain rules to find a new value, and i became the new value to replace the current value. Judgment algorithm meets the end condition. If this result is not satisfied then repeating this process. Or algorithm ends and the current value is used as the final value.

The key of the local search algorithm is to define the specific problems related. Therefore its performance related to the initial state and the definition of field. But the local search algorithm is easy to fall into local optimum which is difficult to achieve the global optimal. So the simulated annealing algorithm is introduced, which is a kind of solving global optimization algorithm.

Global optimization algorithms can be divided into two categories. One is a deterministic algorithm, and the other is random algorithms. The former is based on the deterministic search strategy, which in the objective function to satisfy certain constraints down to solve the problem. The latter is the introduction of random factors in the appropriate search strategy. Because of its objective function is generally not need special conditions, thus it has more extensive applicability. But because it uses a random search strategy, it can only be obtained in the probabilistic sense for global optimal solution to provide a guarantee.

The basic idea of simulated annealing algorithm is derived from the physical annealing process. The so-called physical annealing has three stages: (1) Heating stage. Its purpose is to enhance thermal motion of particles, so that it is deviating from the equilibrium position. (2) Isothermal phase. To exchange heat with the surrounding environment and the closed system of constant temperature, the system state changes always spontaneously toward the direction of reducing the free energy. When the free energy achieves the minimum, the system reaches equilibrium. (3) Cooling stage. Its purpose is to make the thermal motion of the particle weaken and become more orderly, and system energy gradually decreases thus to low-energy crystal structure.

The idea of simulated annealing algorithm is: The solid is heated to melt first, and then let it cool slowly solidified into a regular crystal. When heating the fixed, the inside of the solid particles can increase the internal energy with the increase of internal temperature. When internal energy achieves maximum, the arrangement state of the particle into a liquid disordered. This process is called smelting. When cooling, Particle solidified into a solid crystalline state with the decrease of the temperature. The particle is orderly and solidified into a solid crystalline state. This process is called annealing. When the internal energy is reduced to the minimum, and finally reached the ground state at normal temperature. According to Metropolis criterion, when the particle temperature is T, the probability of particle balance is $\exp(-\Delta E/KT)$. E is the internal energy of the temperature T, $E_1$ is a dynamic parameter and k is a Boltzman constant. The internal energy E is modeled as an

objective function f, temperature T is modeled as control parameters t, to obtain the simulated annealing algorithm to solve the combinatorial optimization problem . By the initial value i and the control parameters of the initial value t start, repeat for the current solution "to produce the new objective function →to calculate difference to determine→ whether to accept→ accept or abandonment "of the iterative process. The algorithm runs gradually decay t value. When the algorithm terminates, the current solution is approximate optimal solution.

Define the basic properties of the simulated annealing algorithm: (1)Energy. Energy function is expressed as E, E=f(x), f (x) is the general function.(2) Generating function. Generating function g ($x_1$) defines the probability density function of the difference between the current point and the next point to be accessed. $x_1=x_{new}-x$. According to $x_1$, the algorithm will generate a new value. (3) Accept function. After generating a new solution by generating functions, the algorithm based on accepted function h($E_1$,T) value to determine whether to accept or give up the new value. This article uses the Boltzman probability accept function $h(E_1,T)=1/(1+\exp(E_1/(CT)))$. C is a constant system. T is the temperature. $E_1$ is the energy difference between $f(x_{new})-f(x)$. Simulated annealing algorithm based on probability way to generate new value. When the $E_1$ is negative, the algorithm tends to accept the solution. When $E_1$ is positive, the algorithm tends to accept the solution with smaller probability. (4) Simulated annealing schedule. Annealing schedule as a function of time or number of iterations control the temperature which drop from high to low.

### 3.2. Simulated Annealing Algorithm Framework

The task scheduling problem algorithm framework in cloud computing with simulated annealing algorithm is as follows:

(1) Algorithm selects an initial value of x, and set a higher initial temperature.

(2) Generating a new task scheduling E = f (x). In order to guarantee that the initial task scheduling near the optimal solution and improve the efficiency of the search algorithm simulation, initial task scheduling uses greedy strategy, using the longest task priority principle to arrange the initial task scheduling.

(3) Algorithm randomly selects three tasks m, n, k, and sets the three tasks of physical machines m, n, k. Algorithm re-secure three tasks: task m arrangement on the physical machine N, task n arrangement on the physical machine K, task k arrangement on the physical machine M.

(4) Calculating the new task scheduling: $E_{new}=f(x_{new})$.

(5) If $E_{new}<E$, $E_{new}$ is accepted as a new path. Otherwise the system at a certain probability accepts $E_{new}$ and cool down.

(6) When the end condition is satisfied, the algorithm stops. Otherwise, turn to step 3.

Task scheduling problem solving in the cloud with simulated annealing algorithm pseudo-code is as follows:

E(y): the evaluation function value in the state of y
Y(i): represents the current state
Y(i+1): represents the new state
r: used to control the speed of cooling
T: temperature of the system
T_min: The lower limit of temperature. When the T achieves T_min, the system stops the searching algorithm.

```
While(T>T_min)
{
E=E(Y(i+1))-E(Y(i));
 if(E>=0) Y(i+1)=Y(i);
 // If there is a more optimal solution, the algorithm accepts from Y(i) to Y(i+1).
 else { if(exp(E/CT))>random(0,1)
```

 

```
Y(i+1)=Y(i);}
 T=r*T;
// Reducing the temperature and annealing. 0<r<1. If r is larger, the slower cooling, and vice versa.
 i++ ;
}
```

In the algorithm, if r is too large, the search for the possibility of a global optimal solution will be higher, but the corresponding search time will be longer. If r is too small, the search process will be very fast, but the result may be a local optimal solution.

## 4. Algorithm Implementation and System Test

We implement a prototype system in this paper. System based on Windows 7 platform, using c # language, using visual studio 2010 development environment. The system uses the Timer class implements resource collection, multicast, task distribution, virtual machine operations and other operations, such as using Thread multithreading technology realize the thread concurrent execution module.

According to the four different types, this paper set the physical machine performance resources as shown in Table 1, number of tasks such as shown in Table 2. The four types of tasks of the number are distributed evenly.

**Table 1. Physical Machine Resource Information**

| Physical Machine | CPU Idle Rate | Memory Idle Rate | Disk Idle Rate | Bandwidth Idle Rate |
|---|---|---|---|---|
| IP1 | X1 | Y1 | Z1 | Q1 |
| IP2 | X2 | Y2 | Z2 | Q2 |
| … | … | … | … | … |
| IP3 | X3 | Y3 | Z3 | Q3 |

**Table 2. Task Number Information**

| Experiment | High CPU Usage | High Memory Usage | High Disk Usage | High Bandwidth Usage |
|---|---|---|---|---|
| Experiment 1 | 50 | 50 | 50 | 50 |
| Experiment 2 | 125 | 125 | 125 | 125 |
| Experiment 3 | 250 | 250 | 250 | 250 |
| Experiment 4 | 375 | 375 | 375 | 375 |

In order to verify the performance of the algorithm proposed in this paper, we use this algorithm compares the traditional method. The experimental results are shown in Figure 1,2,3 and 4.

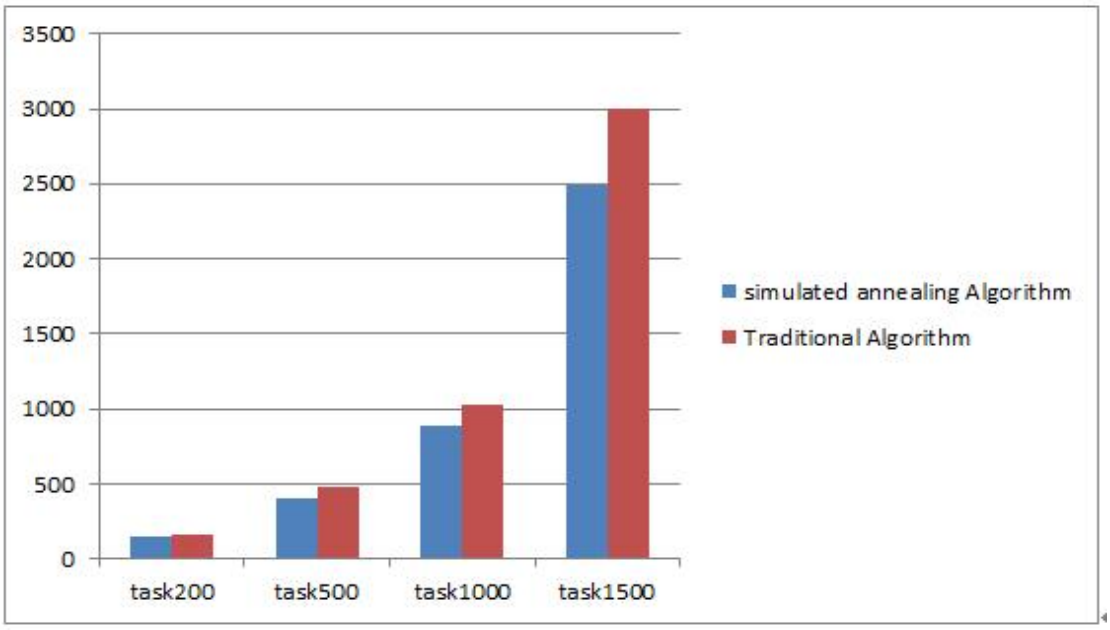


**Figure 1. Simulated Annealing Algorithm is Compared with the Traditional Algorithm of Task Completion Time**

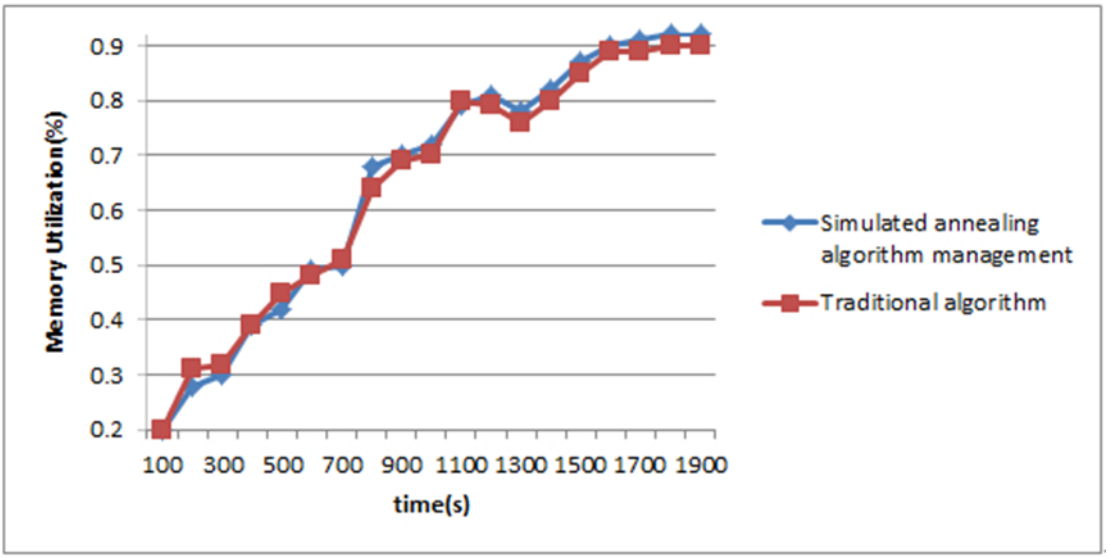


**Figure 2. Simulated Annealing Algorithm Management is Compared with the Traditional Algorithm of Task Completion Time**

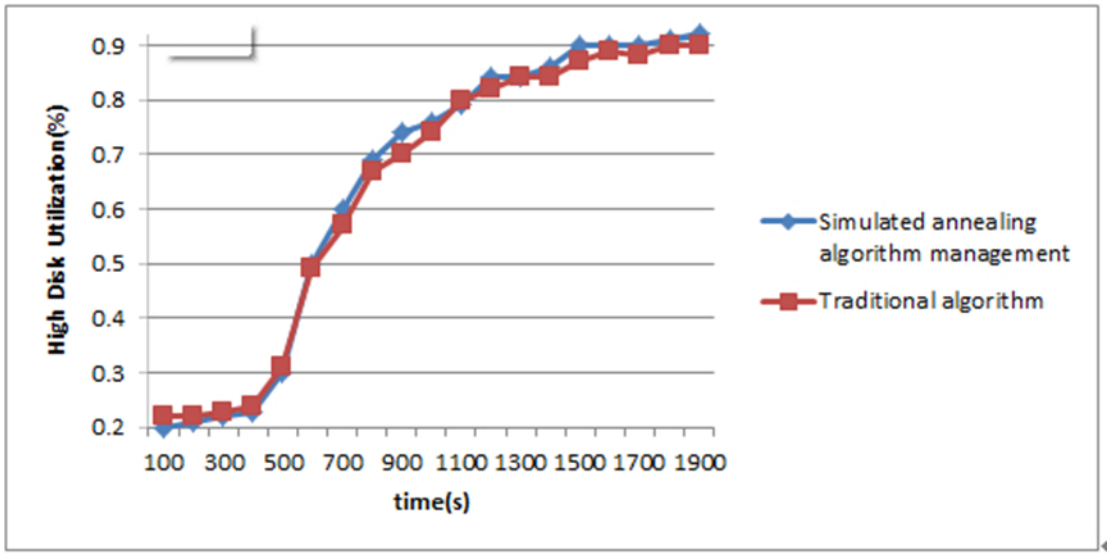


**Figure 3. Simulated Annealing Algorithm Management Is Compared with the Traditional Algorithm of Task Completion Time**

**Figure 2. CPU, Memory, Disk and Bandwidth Utilization**

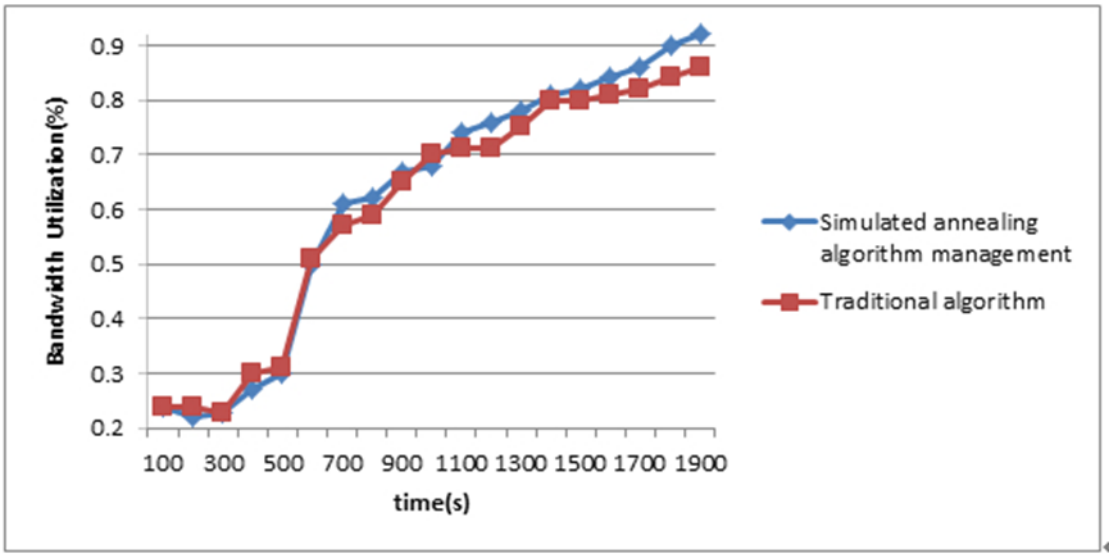


**Figure 4. Simulated Annealing Algorithm Management is Compared with the Traditional Algorithm of Task Completion Time**

Experimental data show that, compared with the traditional algorithm, simulated annealing algorithm is significantly lower total completion time. When the number of nodes is small, the gap is small. As the number of nodes increases, the gap is more obvious.

## 5. Conclusions

This paper presents a task scheduling mechanism based on simulated annealing algorithm. The algorithm is a general, efficient, robust and feasible approximation algorithm. It is not necessarily can find global optimal solution, but can more quickly find the optimal approximate solution of the problem. But as a result of simulated annealing algorithm parameters have important influence on the algorithm convergence speed, and the initial value has important influence on the efficiency of the algorithm, simulated annealing algorithm need a high enough initial temperature, annealing of slow speed, a large number of iterations and enough disturbance frequency under the same temperature can be as much as possible to search the global optimal solution. In order to solve the number of constraints and model perturbation range, other research presented VFSA[13] improved algorithm. The simulated annealing algorithm and deterministic algorithms combine to improve the efficiency and success rate, which is a problem that needs to continue the further study.

## Acknowledgements

The work was supported by Chinese Natural Science Foundation Grant No.11361048, Yunnan NSF Grant No.2010CD086.

## References

[1] M. Ambrust, A. Fox and R. Griffith, "Above the Clouds:A Berkeley View of Cloud Computing[EB/OL]/', http://www.eecs.berkeley.edu/pubs/techrpts/2009/EECS-2009-28. pdf.
[2] K. Chen and W. Zheng, Journal Software, vol.5, no.20, **(2009)**.
[3] J. Mell and T. Grance, "National Institute of Standards and Techology", vol. 3, no.25, **(2011)**.
[4] R. Buyya, C. Yeo And S.Venugopal, "Vision.hype and reality for delivering it services as computing utilities", vol. 3, no.11, **(2008)**.
[5] K. Danielson, Retrievesd February, vol. 5, no. 22, **(2010)**.
[6] H.Stevens and C.Pettey, Gartner Newsroom, vol. 3, no. 16, **(2010)**.
[7] R. Buyya and C. S. Yeo, "Future Generation Computer System", vol. 2, no.25, **(2009)**.
[8] R. E. Ambust, Virtualization & Cloud Computing [EB/OL]. [2011-10-20]. http://www.vmware.com/solutions/ cloud-computing/index.html.
[9] J. Y. Yuan,Wuhan. Huazhong University of Science and Technology, vol. 3, no.28, **(2008)**.
[10] E. W. Walsh and J. O. Kephart, Proceedings of 1 st IEEE International Conference On Autonomic Computing, vol. 2, no. 17, **(2004)**.
[11] T. Gerald, D. Rajarshi and W. E. Wiliam, Proceedings of the Second International Conference on Autonomic Computing, vol. 1, no.2, **(2005)**.
[12] R. Anton and B. Rajkumar, Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud, and Grid Computing, vol. 1, no.10, **(2010)**.
[13] B. Noram, K. Andrzej and B. Kirk, International Symposium on Integrated Network Management, vol. 2, no.22, **(2007)**.
[14] Y. D. Li, H. T. Liu and S. A. Bin, Computer Engineering and Design, vol. 6, no.32, **(2011)**.
[15] P. M. Pardal, H. E. Pomeijn and H. J.Tuyh, Comput. and Appl. Math, vol. 2, no.24, **(2009)**.
[16] L. Inbger, Math. Conput. Modeling, vol. 4, no.12, **(1989)**.

 

# Authors

**Xi Liu**, Doctor, College of Information Science and Engineering, Yunnan University, Kunming Yunnan China

**Jun Liu,** Professor, College of Mathematics and Information Science, Qujing Normal University, Qujing Yunnan China

 