

Cellular Particle Swarm Scheduling Algorithm for Virtual Resource Scheduling of Cloud Computing

Hao Yuan, Changbing Li and Maokang Du

China Key Lab of Electronic Commerce and Modern Logistics of Chongqing University of Posts and Telecommunications, Chongqing, China
E-mail: yln2446@21cn.com

Abstract

The virtual resource scheduling is an important topic in the field of cloud computing. Based on particle swarm scheduling algorithm, this paper introduces cellular automata theory to construct a new cellular particle swarm scheduling algorithm. This approach through mathematical modeling for virtual resource scheduling of cloud computing and complete the final search configuration based on directional optimization objective function. Experimental results show that the proposed method has more excellent scheduling performance, in the case of changes in resources, can be also kept stable scheduling balance.

Keywords: *Cloud computing, Cellular automata, Particle swarm scheduling, scheduling balance*

1. Introduction

Virtual resource management is an important element of cloud computing, based on the virtual machine technology and technology of resource virtualization, play a central role in supporting the entire cloud computing platform. Therefore, the study of how to optimize the management of virtualized resources, to improve the resource utilization of cloud computing and reduce energy consumption as well as the expansion of mobile internet services has an important meaning [1, 2].

In cloud computing, virtual resource scheduling problem can be attributed to assign a set of service resources sequence to the resource pool, in fact it is a multi-objective optimization problems with restrictive conditions [3]. PSO (particle swarm optimization, PSO) is a class of typical methods of solving multi-objective optimization, the origins of such methods is people's research on evolution of the social behavior patterns of birds [4]. After the emergence of particle swarm optimization, it is widely used in function optimization, control of fuzzy system, neural network training and other engineering fields. For discrete space optimization problems especially NP problem in combinatorial optimization, a binary particle swarm optimization method is extracted (Binary Particle Swarm Optimization, BPSO)[5]. For the traveling salesman problem (Traveling Salesman Problem, TSP), continuous particle swarm optimization algorithm is used to assign the tasks of distributed computers [6].

Cellular automaton (Cellular Automaton, CA), is a method emergence to simulate the process of self-proliferation of biological cells [7]. Cellular automaton is a discrete model has using in the feasibility theory, mathematics and theoretical biology. The three main characteristics of cellular automata are parallel computing, locality and consistency [8, 9]. Usually, we use the entropy to describe its evolution process; cellular automata can be

divided into smooth type, cycle type, chaotic and complex type.

Bo designed multi-objective integrated ant colony optimization

Manuscript received January 1, 201X; revised June 1, 201X; accepted July 1, 201X.
Copyright credit, corresponding author:

scheduling algorithm. As the same to grid computing environment is, the cloud has a certain complexity. Number of virtual resources in cloud environment is huge, so it has uncertainty. In order to measure the uncertainty, the concepts of entropy uses the ant colony algorithm, update of pheromone, and accelerate the improvement of the solution set of the convergence speed of algorithm [10].

In order to make the scheduling time is minimized, Gu proposed the mini-max algorithm to calculate the optimal time limit to complete the task scheduling the heuristic information. At the same time, he also considers the load balancing problem and designs a load coefficient. When algorithm is operated and local pheromone is updated, it can be brought into considering an important indicator of pheromone spreading factor. Calculating the pheromone, it is considered the current including its neighboring nodes of node, which purpose is to make the ant cooperation more closely and improve the performance of the optimal solutions [11].

Mark model was presented based on particle swarm optimization, and it is used in performing the mapping between tasks and resources, so as to realize the deadline specified by the user, so that the costs of the minimum to complete the task. It presented a binary integer program, appropriate resources choose between different cloud resource providers [12].

This article combines cellular automata with particle swarm optimization, proposes a cellular particle swarm optimization, using in virtual resource scheduling for cloud computing. Cellular particle swarm optimization combines the self-growth and search of cellular automata, rapid convergence of particle swarm optimization algorithm, can take advantage of cellular and cellular neighbors to enrich the behavior of particles, maintaining population diversity, while ensuring cellular evolution occurs only in a limited part, to avoid premature convergence, making the particle swarm remains optimization capability in the search process, which can quickly search the global optimal solution.

The performance of scheduling strategy on cloud computing has an important influence, so in the grid heterogeneous environment, the workflow task assignment and scheduling problem has been done a lot of research. A lot of work focuses on the use of heuristic algorithm to complete the workflow scheduling [13-16].

2. Backgrounds

A. Particle Swarm Optimization (PSO)

Similar to other evolutionary algorithms, PSO also uses the concept of "population" and "evolution", and operate according to the fitness of each particle[10]. PSO algorithm uses a "speed - position" search model, a solution to the solution space, on behalf of each particle which based on pros and cons of the extent of the fitness function is judged, but fitness function. The objective function are decided by

Assuming particle swarm scale of n , each particle j having plurality of characteristic attributes. Suppose there is a l -dimensional search space, the j particles can be represented by an l -dimensional vector, denoted by $Y_j = \{y_{j1}, y_{j2}, y_{j3}, \dots, y_{jl}\}$. The dimensions and velocity of the particle is designated as $V_j = \{v_{j1}, v_{j2}, v_{j3}, \dots, v_{jl}\}$. Particles in the search space autonomous flight. On the one hand, the particle will choose the position and velocity according to own experience of the flight; on the other hand, according to other particles can be detected around the particles within a range of conditions, adjusting its position and velocity in order to maintain the balance of the swarm.

In the process of particles flying, recording experienced the best position vector as $P_j = \{p_{j1}, p_{j2}, \dots, p_{jl}\}$; best optimum position vector of position of the current particles

in the swarm, denoted by $H = \{H_1, H_2, \dots, H_l\}$. Assuming t represents the current t -th generation offspring, the particle j velocity changes, denoted by v_{jk}^t , position changes denoted as y_{jk}^t , where k represents the step size. Using random multipliers, we can obtain the following equation:

$$v_{jk}^t = v_{jk}^{t-1} + d_1 s_1 (P_{jk}^{t-1} - y_{jk}^{t-1}) + d_2 s_2 (H_k^{t-1} - y_{jk}^{t-1}) \quad (1)$$

$$y_{jk}^t = y_{jk}^{t-1} + v_{jk}^t \quad (2)$$

Here, d_1, d_2 is called the acceleration factor or learning factor, which is usually in the range $[0, 2]$, $s_1, s_2 \sim U(0,1)$ represents two independent random functions. d_1 represents step size to adjust the particle flying to own optimal position, d_2 represents step size to adjust to the global particle flying to optimal position, rational choice of d_1, d_2 can ensure rapid convergence and not fall into a local optimum. High flight speed of particle would allow discussion about local optimum, in order to control the speed of the particles, must establish the speed limit v_{max} . Too large v_{max} will easy to make the particles away from the optimal solution, too small will prematurely fall into local optimum. Assuming the search space of $[-y_{max}, y_{max}]$, then if $v_{max} = i \times y_{max}$, where there are $0.1 \leq i \leq 1.0$.

B. Cellular Automata Theory

A standard cellular automaton usually consists of four parts: cellular, cellular status, the neighbors, a status update rules. The mathematical expressed as:

$$B = (O, e, T, L, g) \quad (3)$$

Here, O represents cellular space, e is the dimension of the cellular space in cellular automata; T represents limited, discrete state set of cellular; N represents a collection of all cellular of neighborhood; g represents partial map or local rules; positions within the e -dimensional cellular space of all cellular can be used to represent by an integer Z^e .

The set of all outlets where cellular distributed is called cellular space. Because theoretically cellular space in each dimension is infinite upward, in order to simulate and calculate on your computer, you need to define the boundary conditions, including periodic type, reflective type, and set point type.

Definitions 1: Let e representatives dimension of space, i behalf of cellular state, T is a collection of discrete states of cellular, r behalf of neighborhood radius. Z is a set of integers, represents one-dimensional space, t represents time. Assuming $e = 1$, for the one-dimensional cellular space, Z is discrete distribution on T , denoted as T^Z . One-dimensional cellular automaton will change state in the timeline, and its evolution recorded as:

$$G : T_t^Z \rightarrow T_{t+1}^Z \quad (4)$$

This evolutionary process was determined by the local evolution rules of cellular of g called local rules. For one-dimensional space, cellular and neighborhood denoted as T^{2r+1} , local rules denoted as:

$$G : T_t^{2r+1} \rightarrow T_{t+1} \quad (5)$$

For local rules of g , its set of inputs and outputs of functions are finite sets. In fact, it is a limited reference table. Independently applied above local rules of cellular for the cellular space, you can get the global evolution.

$$G(c_{i+1}^j) = f(c_i^{j-r}, \dots, c_i^j, \dots, c_i^{j+r}) \quad (6)$$

Here, c_i^j on behalf of the position of j of the cellular.

3. Cellular Particle Swarm Optimization

A. Multicomputing Backpack Modeling

Virtual of cloud computing resource scheduling, equivalent to a multi knapsack problem, the set of feasible fields is l dimensional Euclidean space, which any elements can be referred to cellular, all the elements that make up cellular space [11, 12]. Using the binary coding, which uses a matrix $Y (n \times l)$, where $Y_{jk} = 1$, it means that the resource vector j mapped onto a virtual machine k ; when $Y_{jk} = 0$, represents the vector resource j does not mapped to the virtual machine k .

Definitions 2: Assume that the resources vector coding of $T_j = 1, 2, 3, \dots, n$, VM encoding is $V_j = 1, 2, 3, \dots, l$, the cellular can be expressed as (c_1, c_2, \dots, c_n) , wherein $c_j \in \{V_j\}$. All cellular composition of cellular space N .

Definitions 3: More types neighborhood

$M_{more} = \{C_Y \mid \text{difference } (C_Y - C_X) \leq r, C_X, C_Y \in M\}$ Here, difference $(C_Y - C_X) \leq r$ is the difference between the two cellular sequences, if no difference it is 0, when there are differences it is up to 1, r represents the degree of difference, in this paper r is set to 1.

Definitions 4: position and velocity of particles update equation. Assuming the solution space of e dimension, the j particles in the i generation of the evolution of the position $y_j^i = (y_{j1}^i, y_{j2}^i, \dots, y_{je}^i)^T$, the optimum position of the particle has been searched for, $p_j^i = (p_{j1}^i, p_{j2}^i, \dots, p_{je}^i)^T$ the speed is, $v_j^i = (v_{j1}^i, v_{j2}^i, \dots, v_{je}^i)^T$, the best location for the entire population through is, $h_j^i = (h_{j1}^i, h_{j2}^i, \dots, h_{je}^i)^T$, and update velocity and position of each particle shown the following formula:

$$V_{jk}^{i+1} = wv_{jk}^i + d_1s_1(p_{jk}^i - y_{jk}^i) + d_2s_2(h_{jk}^i - y_{jk}^i) \quad (7)$$

$$v_{jk}^{i+1} = \begin{cases} v_{\max}, v_{jk}^{i+1} > v_{\max} \\ v_{jk}^{i+1}, v_{\min} \leq v_{jk}^{i+1} \leq v_{\max} \\ v_{\min}, v_{jk}^{i+1} \leq v_{\min} \end{cases} \quad (8)$$

$$y_{jk}^{i+1} = \begin{cases} 1, s_3 < \text{sig}(v_{jk}^{i+1}) \\ 0, \text{other} \end{cases} \quad (9)$$

Here, w representative of the inertia weight to balance global and local search; c_1, c_2 is a learning factor, where c_1 represents the effect of memorized optimum position of the

particle trajectory on the flight speed, it is called "cognitive factor"; c_2 represents effects of memorized optimal position of particle swarm on the flight velocity, called "sociological factors"; s_1, s_2, s_3 are the random numbers on $[0, 1]$ interval,

$sig(x)$ represents an ambiguity function, with $sig(x) = \frac{1}{1 + e^{-x}}$, v_{jk}^{i+1} determines the component position of the particle takes the probability of 0 or 1.

B. Objective Optimization Function

In order to reduce the difficulty of scheduling optimization, the virtual resource scheduling model of this paper emphasizes the orientation optimization of a resource, namely reasonable expense other surplus resources, making a particular resource to achieve balancing optimal load. The algorithms learn standard deviation thinking from mathematics equilibrium degree of resource scheduling is numerical and proposed uniform distribution policy.

Assuming current here is a given encoding matrix $Y(n \times l)$, then the set $\{1, 2, \dots, i\}$ for resource-specific resource R , and then there is:

(1) Global load degree of resource:

$$\mathcal{G}_{R_global} = \frac{\sum_{j=1}^n A_{jR}}{\sum_{k=1}^l B_{kR}} \quad (10)$$

(2) Load degree of virtual machine V for resources

$$R : \mathcal{G}_{R_V} = \frac{\sum_{j=1}^n A_{jR} Y_{jV}}{B_{VR}} \quad (11)$$

(3) Scheduling equilibrium degree of resources R :

$$H_R = \sqrt{\frac{1}{l} \sum_{k=1}^l (\mathcal{G}_{R_k} - \mathcal{G}_{R_global})^2} \quad (12)$$

C. Algorithmic Search Process

After the introduction of the above objective function for resource R , individual solutions Y define penalty function for the degree of violation of constraint as follows:

$$H(Y) = H_R$$

Thus, process of cellular particle swarm optimization can be described as follows:

(1) Determine the population size of L , the inertia weight factor of w , and learning factor of $d_1, d_2, NC \leftarrow 0$, in which NC represents the iteration step;

(2) Initializing the position and velocity of all the particles: Particle positions are randomly generated by $y_{jk}^0 = \begin{cases} 0, R(0,1) < 0.5 \\ 1, other \end{cases}$;

Particle velocity are randomly generated by $v_{jk}^0 = v_{min} + R(0,1)(v_{max} - v_{min})$;

Here, $R(0,1)$ means that random number on the interval $[0, 1]$, v_{\min} , v_{\max} represents the maximum speed and minimum speed.

(3) calculate the objective function value for each particle; record the current optimal solution;

(4) According to the definition of cellular neighbors, evolution in the range of neighbors, recording the optimal solution;

(5) Compare and update the optimal location of each particle and the optimum position of population;

(6) Set $NC \leftarrow NC + 1$;

(7) If NC is less than a predetermined number of iterations, update the position and velocity of each particle, duplication from the step (3) to step (6).

(8) If NC is reached a predetermined number of iterations, output current optimal solution.

3. Experimental Results and Analysis

In order to verify the effectiveness of the method, ie the feasibility of cellular particle swarm algorithm to solve virtual resource scheduling problem and compare the difference performance between this method and other methods have used in cloud computing, carry out the experiment and analysis of the experimental results are as follows.

A. Experimental Environment Settings

Set the virtual machine resource scheduling model as that, the number of population of 50, the maximum number of iterations is 1024, cross-factor is 0.88, and the variation factor is $1 / 1$, where 1 represents the length of the gene. Assuming resources property contains: CPU resource, memory resources and bandwidth resources, the following experimental conditions to participate in tested is in follows in Tab.1.

It is seen from table 1, types of resource vector computer are divided into 5 kinds: the first kind, CPU frequency is 1200MHz, the memory size is 4G, the bandwidth is 90M; second, CPU frequency is 1000MHz, the memory size is 3G, the bandwidth is 70M; third, CPU frequency is 700MHz, memory size of 1.5G the bandwidth, the size of 30M; fourth, CPU frequency is 2000MHz, the memory size is 2G, the bandwidth is 6M; fifth, CPU frequency is 600MHz, the memory size is 0.6G, the bandwidth is 20M. Classification of the virtual host in accordance with these three indicators is divided into 8 categories.

Table 1. Resource Vector and Virtual Host

Resource vector				Virtual host			
Serial number	CPU	RAM	Bandwidth	Serial number	CPU	RAM	Bandwidth
A	1200MHz	4G	90M	1	280MHz	360M	15M
B	1000MHz	3G	70M	2	200MHz	100M	15M
C	700MHz	1.5G	30M	3	500MHz	300M	8M
D	2000MHz	2G	60M	4	1000MHz	1G	20M
E	600MHz	0.6G	20M	5	60MHz	30M	4M
				6	80MHz	60M	8M
				7	600MHz	2G	50M
				8	160MHz	200M	15M

Select this method in this article and three other methods to compared, these three methods are: stochastic scheduling algorithm, static scheduling algorithm and sorting matching algorithm set the following three methods:

(1)Stochastic scheduling algorithm: given a random scheduling scheme $S(p_1, p_2, \dots, p_n)$.

(2)Static scheduling algorithm: given scheduling program $S(p_1, p_2, \dots, p_n)$, and there is $p_i = i \% l$.

(3)Sorting matching algorithm: for each resource vector, searching the closest virtual host of resources vector, ie with a resource vector $T_j(A_{j1}, A_{j2}, A_{j3})$ and virtual hosts $V_k(B_{k1}, B_{k2}, B_{k3})$, and the existence of k , such that $\{|A_{j1} - B_{k1}| + |A_{j2} - B_{k2}| + |A_{j3} - B_{k3}|\}$ reaches a minimum.

B. Analysis of Experimental Results

First of all, verify the effectiveness of proposed cellular particle swarm optimization scheduling algorithm. When eight virtual machines were used in realizing scheduling of the five resource response vector, the formation of ten different optimal solutions would come into beings. The optimal solution of the CPU equilibrium, memory equilibrium degree, and bandwidth balancing index, had been Listed in table 2.

Table 2. Experimental Results of Cellular Particle Swarm Optimization

Solution sequence	equilibrium degree of CPU	equilibrium degree of RAM	equilibrium degree of Bandwidth
BBCDDEAB	0.2801	0.2185	0.1733
BBCDDEAB	0.2997	0.2403	0.1346
BBCDDEAB	0.3199	0.2355	0.1502
BBCDDEAB	0.2703	0.2290	0.2302
BBCDDEAB	0.2988	0.2438	0.1901
CBDDEBAB	0.3004	0.2702	0.1800
CBDDEBAB	0.2775	0.2238	0.2805
DBCDEBAB	0.3103	0.2858	0.1903
CBDDEBAB	0.3302	0.2771	0.2012
DBCDEBAB	0.2944	0.2790	0.2339

For the practical application of cloud computing, the bandwidth resources are generally constrained cloud performance bottlenecks. Therefore, the optimization strategy of bandwidth resources is the most important optimization strategy. For video calls and other application scenarios, time delay requires higher, demand for cloud computing resources also corresponding increase, then optimized for CPU resources becomes more critical. Seen for different applications, it can determine the optimal primary and secondary strategy. To compared results of this method with stochastic optimization scheduling, the static scheduling algorithm and sorting matching algorithm, as shown in Table 3.

Table 3. Comparison of this Method and Other Methods

Algorithm	Strategy	Solution sequence	CPU	RAM	Bandwidth
Present method	Balanced optimization	BBCDBEAB	0.2720	0.2188	0.1734
Stochastic scheduling	Balanced optimization	CBDDADA A	0.4201	0.3229	0.4202
Static scheduling	Balanced optimization	ABCDEABC	—	—	—
sorting matching	Balanced optimization	DACDAEBA	0.3233	0.3657	0.2451

As can be seen, balanced scheduling of virtual resources in cloud computing is at the premise of a directional optimizing, rational configure a variety of resources although some resources need to sacrifice, but to ensure that the load is running for the best results. As can be seen from the comparison results of Table 3, balanced optimization index of cellular particle swarm optimization scheduling algorithm was significantly higher than other methods. Among them, the static scheduling algorithm can not produce a feasible solution.

To further investigate the performance of these four methods in the resource dynamic process, the following experimental were designed. In cloud computing, resource ratio refers to the ratio of total resources of user requests a and cloud resources can provide, it can reflect the current load and peak level of cloud. To do this, set the following experimental conditions, the maximum number of iterations is 1024 times, the range of resource ratio control in the [0.2, 0.9] , further comparison of the dynamics of balanced scheduling of four methods, shown in Figure 1.

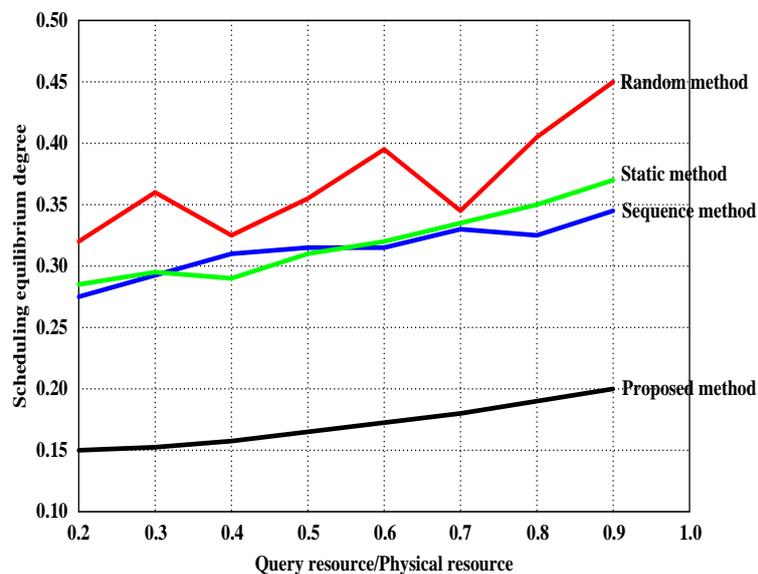


Figure 1. Balanced Schedule Degree of Different Resources Ratio

Not difficult to see from Figure 1, along with changes occurred in cloud resources ratio, under conditions of different resources ratio, the proposed cellular particle swarm scheduling algorithm showed more excellent performance than the other three methods, resource scheduling is more balanced. With the next increase in resources ratio, and the number of iterations remain unchanged, scheduling balance of this method will decline.

4. Conclusion

For virtual resource scheduling problem in cloud computing, this paper proposes a cellular particle swarm optimization scheduling methods. This approach put cellular automata theory and particle swarm optimization scheduling algorithm together in order to fully play the advantages of both. This method first conduct mathematical modeling in cloud computing of virtual resource scheduling problem, after given orientation optimized objective function and configure the search algorithm. Through validation experiments and contrast test of other methods can be seen that this method have more excellent performance of cloud computing virtual resource scheduling, in the case of changes in resources ratio schedule can be kept stable equilibrium.

5. Acknowledgements

This work is supported by national natural science foundation of China (No. 61003256); Chongqing university of posts and telecommunications natural science foundation of China (No. A2011-21)

References

- [1] F. Thomas, J. Alexandru and P. Sabri, "ASKALON, a tool set for cluster and grid computing", *Concurrency and Computation Practice and Experience*, vol. 17, (2005), pp. 143-169.
- [2] E. Barrent, E. Howley and J. Duggan, "A learning architecture for scheduling workflow applications in the cloud", *IEEE European Conference on Web Services*, (2011), pp. 83-90.
- [3] C. A Coello, G. T. Pulido and E. M. Montes, "Current and future research trends in evolutionary multi-objective optimization", *Information Processing with Evolutionary Algorithms*, (2005), pp. 213-231.
- [4] J. Yu and R. Buyya, "Workflow scheduling algorithms for grid computing", *Metaheuristics for Scheduling in Distributed Computing Environments*, vol. 146, (2008), pp. 173-214.
- [5] R. Sakellariou, H. Zhao, E. Tsiakkouri, "Scheduling workflows with budget constrains", *Integrated Research in Grid Computing Springer*, (2007), pp. 189-202.
- [6] M. Rahman, S. Venugopal and R. Buyya, "A dynamic critical path algorithm for scheduling scientific workflow applications on global grids", *In the Proc of the 3rd IEEE International Conference on e-Science and Grid Computing*, pp. 35-42.2007.
- [7] H. M. Fard, R. Prodan, J. J. D. Barrionuevo, "A multi objective approach for workflow scheduling in heterogeneous environments", *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, (2012), pp. 300-309.
- [8] J. Tordsson, R. Montero, R. M. Vozmediano, "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers", *Future Generation Computer Systems*, vol. 28, (2012), pp. 358-367.
- [9] D. Warneke, O. Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud", *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, (2011), pp. 1045-9219.
- [10] X. Bo , Q. Guan and K. Chen, "Multi-Agent Coalition Formation Based on Quantum-behaved Particle Swarm Optimization", *Journal of Information & Computational Science*, vol. 7, (2010), pp. 1059-1064.
- [11] J. Gu, J. Hu, T. Zhao, *et al.*, "New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment", *Journal of Computer*, vol. 7, (2012), pp. 42-52.
- [12] M. Stillwell, D. Schanzenbach, F. Vivien, *et al.*, "Resource allocation algorithms for virtualized service hosting platforms", *Journal of Parallel and Distributed Computing*, vol. 70, (2010), pp. 962-974.
- [13] S. Ostermann, A. Iosup and N. Yigitbasi, "A performance analysis of EC2 cloud computing services for scientific computing", *Cloud Computing*, (2011), pp. 115-131.
- [14] J. Dejun, G. Pierre, C. H. Chi, "EC2 performance analysis for resource provisioning of service-oriented applications", *Service-oriented Computing. ICSOC/ServiceWave 2009 Workshops*, Springer Berlin/Heidelberg, (2010), pp. 197-207.
- [15] H. Qi, A. Gani, "Research on mobile cloud computing: review, trend and perspectives", *Digital Information and Communication Technology and its Application*, (2012), pp. 195-202.
- [16] W. Song and X. Su, "Review of mobile cloud computing", *IEEE 3rd Int Communication Software and Networks*, (2010), pp. 1-4.

Authors



Hao Yuan, Associate Professor, Chongqing University of Posts and Telecommunications. Born in 1970, Mr Yuan graduated and got master degree from Chongqing University, and his main research interests are computational intelligence and Cloud Computing.



Changbing Li, Associate Professor, Chongqing University of Posts and Telecommunications.

Born in 1970, Mr Li graduated and got doctor degree from Chongqing University, and his main research interests are computational intelligence and Cloud Computing.



Maokang Du, Professor, Chongqing University of Posts and Telecommunications. Born in 1969, Mr Du graduated and got bachelor degree from Xinan University, and his main research interests are computational intelligence and Cloud Computing.