

A Heuristic Scheduling Algorithm based on PSO in the Cloud Computing Environment

Shengjun Xue^{1,2}, Wenling Shi^{1,2} and Xiaolong Xu^{3,4}

¹ School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

² Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, China

³ State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China

⁴ Department of Computer Science and Technology, Nanjing University, Nanjing, 210023, China
shiwenling92@163.com

Abstract

In the face of a large number of task requests submitted by users, the data center in the cloud need not only to finish these massive tasks but also to satisfy the user's service demand. How to divide the resources of the system reasonably and schedule these tasks efficiently is a problem that need be solved in the cloud computing. The task scheduling of workflow is a kind of scheduling model for the most part researched in cloud computing. The scheduling goal is always decided by user's QoS (Quality of Service) and the scheduling goal of existing scheduling algorithm is always single. Based on the price model a scheduling algorithm is proposed which can realize the multiple targets in this article, and the scheduling algorithm is Service Cost Optimization based on Particle Swarm Optimization (PSO-SC). PSO-SC algorithm can adapt to dynamic cloud environment, and it not only shortens the completion time of tasks but also minimizes user's cost of task when schedule tasks. In this paper, experiment and analyze the task scheduling of workflow under the cloud computing environment, and the experimental results show that the algorithm proposed in this paper has very good scheduling performance and can reach the goal of task scheduling of tasks.

Keywords: clouding computing, PSO (Particle Swarm Optimization), task scheduling, service costs

1. Introduction

Cloud Computing is usually linked to the service of the Internet. With the development about high-speed networks, the Internet has connected to the world [1]. Not only the network broadband has been greatly improved that you can transfer large amounts of data, but also the users can enjoy these new technologies and various forms of service according to their needs at any time and any place. These services include various kinds of hardware and software services [2] such as getting computing power, storage space and network bandwidth, etc. Classification of resources, demand of executive capacity and resource sharing, and advances of various technologies have promoted the emergence about Cloud Computing [3]. It also allows users to enjoy the high-performance software resources, hardware resources, computing resources and service resources brought by advent of Cloud Computing Model.

Cloud computing is an emerging technology where information technology resources are provided to users in a set of a unified computing resources on a pay per use basis.

Cloud Computing has experienced a three-stage development: Distributed Computing, Parallel Computing and Grid Computing [4]. Cloud Computing, which combines the computing model of data sharing and the computing model of service sharing, is the commercial realization of these scientific concepts, and it has laid the foundation for computing model of next generation. It provides a kind of model to use and charge [5], and it also has some dynamic, scalable and virtual resources. Cloud Computing is inseparable from the network supports, because of the limitlessness of the network. Cloud Computing can easily share resources and information on a large number of computing devices by the Internet [6], and it can also build a pool of computing resources with these resources and information and offer services to users through a unified management and scheduling of network.

Cloud Computing offers a new model, which is charged by the on-demand resources the users access, to services providing and resources delivering [7]. It has a wide-range of computing services and resources such as servers, memory, cpu, bandwidth, etc. [8]. It also has some application services such as e-commerce and social platforms that function on the entire network. Under this service based on the mode of providing resource, users can go to consume when they need and pay the proper fees for the services that they use. Therefore, price model based on the booking pricing is proposed [9] in this paper. The price model is that the providers who offer cloud services assess a service subscribed by the user, and give a fixed price. Compared with per unit pricing and tiered pricing, the price is more favorable and the most widely used one among the price models. Of course, this favorable price has preconditions, which are that the user has to pay part of the cost by way of signing a contract in accordance to the amount of computing resources or services as well as the amount of resource that will be used estimated during the next period of time.

In this paper, we will present a method based on Particle Swarm Optimization (PSO) to execute task scheduling. Because customers' demand needs to meet as much as possible in the cloud environment, choice of a strategy of task scheduling of workflow is very important [10,11]. Workflow is a computational model of working process, which represents the logics and rules that how to organize the front and rear works together in the working process and calculates it in the computer with the proper model [12]. The main problem that the workflow should resolve is that try the best to achieve the goal of a service, and transmit information, the resources or tasks automatically according to specific rules using computers between plurality of participants. In this paper, we proposed a Service Cost Optimization Algorithm based on Particle Swarm Optimization (PSO-SC) [13] algorithm to achieve minimum cost of users.

The paper is organized as follows: the second section is the introduction to the scheduling of workflow based on the price model, including the introduction to price models and the workflow. The third section describes the scheduling algorithm proposed in the paper. The fourth section carries on experiments and analysis of the proposed scheduling algorithm. The fifth section makes a summary of this article.

2. Related Work

Problems of carrying out task scheduling have been extensively studied in the distributed system. Because the issue of task scheduling is difficult, some heuristic algorithms have been proposed to solve this problem. The document [14] has proposed ant colony optimization algorithm for executing task scheduling, which balances the load of whole system and minimizes the completion time of tasks. The document [15] has put forward a genetic algorithm with double adaptability, which can not only find the scheduling results with short completion time of tasks in total, but also the average completion time of these tasks is short. The document [16] presents an optimization algorithm based on genetic algorithm to schedule the independent and divisible tasks. The

algorithm, which can adapt to different computing and memory requirements, also is used in heterogeneous systems to carry out the dynamic scheduling, but it does not take into the efficiency of global optimization account. The document [17] has introduced a new scheduling algorithm based on two traditional scheduling algorithms called Min-Min and Max-Min. This algorithm is able to complete the task scheduling within the expected time, and it owns much more significant performance compared to its two basic heuristic algorithms. The document [18] has proposed the algorithm of task allocating based on the principles of particle swarm optimization, which solves the problem of task allocating by discussing the searching strategy of the algorithm. Comparing the algorithm with the genetic algorithm, it proves that the PSO has more advantages. The document [19] presented an application about heuristic scheduling based on the PSO, and it has fully thought about the computing costs and the transportation costs.

The scheduling goal of traditional PSO is to achieve the minimum of the total completion time of tasks without taking into the service costs of the users accounts, thus the PSO-SC is improved to achieve this goal in this paper. The first concept of PSO was proposed by Eberhart and Kennedy in 1995, whose definition was summed up by studying the behavior of looking for the food of birds [20]. We take the foraging behavior of birds as an example to understand the principles of PSO. This behavior can be described specifically as: A flock of birds look for the food in a particular area and the food is the only one. The birds do not know the exact position of the food but they can determine the distance to their food. How can they find the food full out? The most effective way is to find the surroundings of the currently nearest bird to the food.

In the PSO, the particles in the multi-dimensional searching space have no size and quality that we can think of them as individuals with its own position and velocity in the population. These particles fly and search in the solution space and adjust their position and speed through the experience they have gained in flight and communication with their peers [21]. Every particle adjusts constantly their speed and position of the flight according to the optimal value found by it and the optimal value found by the entire population of all the particles during the iterative process, and eventually produces a positive feedback mechanism of group optimization. An adaptability function need to be designed when use PSO algorithm and get close to the optimal region on the basis of the adaptive value and ultimately find the optimal solution in the solution space.

3. Task Scheduling of Workflow based on Price Model

3.1. Model of Task Scheduling of Workflow

Because some of the tasks submitted by users have the interdependent relation, the workflow is act as the objective of the research in this paper to solve the problem about the scheduling of related works in the cloud environment. Initialize a set of workflow randomly. All the tasks in the workflow model [19] have the interdependent and constraint relations. Each task has its parent tasks and subtasks in addition to the first task and the end task-the first task only has the subtasks, and the end task only has the parent task. These tasks can be executed only when their parent tasks have been executed, and when they have been executed their subtasks can be executed.

In this paper, a set of workflow is initialized randomly, and the number and the length of tasks in the workflow are stochastic. The relationship between tasks is also random, that is which task is subtask and which task is the parent task is random. The following matrix $A_{[i,j]}$ can be used to represent the relationship between tasks. $A_{[i,j]}=0$ means that the task T_i and task T_j do not have interdependent relation, and $A_{[i,j]}=1$ indicates the interdependent relation between the task T_i and task T_j , that the task T_i has a directed edge which point to the task T_j .

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

According to the relationship between the tasks in matrix $A_{[i,j]}$, the workflow model which describes the workflow is drew as Figure.1. Each task has a degree which can be divided into in-degree and out-degree, and the parent task directs to the task itself is called in-degree. A one-dimensional array $T_j[n]=\{t_a, t_b, \dots, t_k\}$ can be used to record the in-degree of each task, and the subscript n represents the value of in-degree. The t_a, t_b, \dots, t_k represents the parent tasks of task T_j . According to the matrix $A_{[i,j]}$, it can conclude that $t_0[0], t_1[1]=\{t_0\}, t_2[1]=\{t_0\}, t_3[1]=\{t_0\}, t_4[1]=\{t_1, t_2, t_3\}$. After each execution of a task T_i , the in-degree of subtasks T_i of task T_j will subtract one, and the task T_j can be performed only when the value of the subscript n is 0.

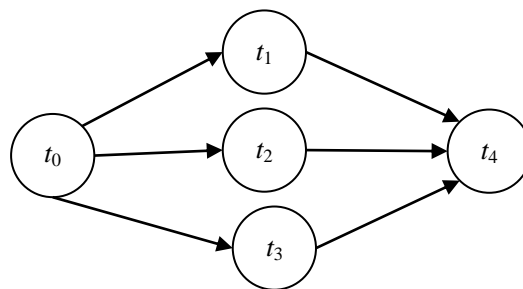


Figure 1. Workflow Model

3.2. Mathematical Model of Price Model

This paper uses the price model of reserving pricing, and the user pays a fee as deposit at first in this model. Then pay the cost of tasks according to the execution time of the virtual machine and the running costs of per unit time of the virtual machine.

Initialize a set of workflow randomly which includes n tasks and the length of each task is not same in this paper. Considering the execution time of tasks, the long tasks should be put on the virtual machine with strong execution ability as far as possible, and the running costs of per unit time of virtual machines with the different execution ability is also not same. There are n kinds of virtual machines with different execution ability provided in this article, and these virtual machines have different execution cost.

In this article, the virtual machine leased by users can be represented as VM , and the corresponding collection of the virtual machine is $VM=\{vm_1, vm_2, vm_3, \dots, vm_n\}$, where vm_i represents the i^{th} class of the virtual machine. The running cost of per unit time of the virtual machine can be represented as $Cost$ whose corresponding collection is $Cost=\{cost_1, cost_2, cost_3, \dots, cost_n\}$, where $cost_i$ represents the running cost of per unit time of the i^{th} class of the virtual machine. In the process of task scheduling, the processing speed of each processor of the virtual machine can be represented as $MIPS$ (Million Instructions Per Second), and the corresponding collection of the processing speed is $MIPS=\{mips_1, mips_2, mips_3, \dots, mips_n\}$, where $mips_i$ represents the processing speed of the i^{th} class of the virtual machine.

Because the number and the length of tasks are randomly assigned, the number and the type of the virtual machines used to execute the tasks are also randomly assigned. Suppose that there are m tasks and num_i virtual machines whose type is VM_i .

$SumLength_{ik}$ is that the total length of the tasks which are distributed in the k^{th} virtual machine with the type of VM_i . $length_m$ is the length of the m^{th} task of the virtual machine

which includes p tasks.

$$SumLength_k = \sum_{m=1}^p length_m \quad (1)$$

$time_{ik}$ is that the execution time of the k^{th} virtual machine with the types of VM_i , $mips_i$ is the processing speed of the i^{th} class of the virtual machine.

$$time_{ik} = SumLength_k / mips_i \quad (2)$$

$Sumtime_i$ is that total execution time of the VM_i types of virtual machines. num_i is the number of the VM_i types of virtual machines.

$$Sumtime_i = \sum_{k=1}^{num_i} time_{ik} \quad (3)$$

$SumCost$ is the cost of the user needs to pay for. $SetCost$ is a part of cost the users paid for before performing the task according to the contract.

$$SumCost = SetCost + \sum_{i=1}^n Sumtime_i \times cost_i \quad (4)$$

4. PSO-SC Algorithm

4.1. Task Distribution of Virtual Machines

In this article, the user submits a set of tasks to the cloud data center, and selects several of virtual machines. Because the choice of the type and the number of virtual machines will affect the implementation of the scheduling goal, service cost optimization based on particle swarm optimization is proposed in this paper.

We suppose that there are taskNum tasks and vmNum virtual machines. The virtual machine includes n types, and the total number of tasks is greater than the number of virtual machine. We suppose that there are 10 tasks and 5 virtual machines which include 5 types. The task distribution of the virtual machine is shown in Figure.2.

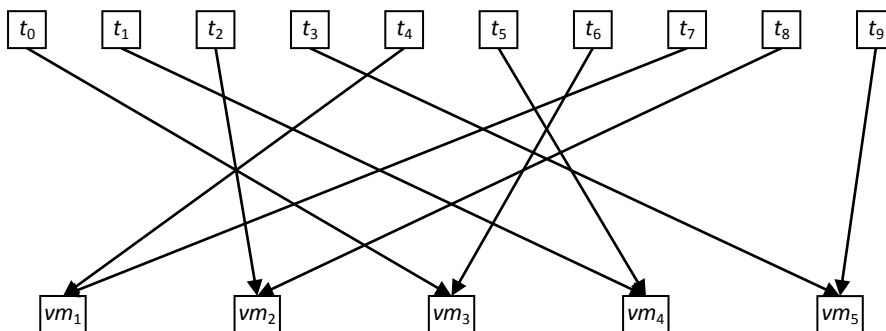


Figure 2. Task Distribution of Virtual Machines

4.2. Process of PSO-SC Algorithm

The implementation steps of the PSO-SC algorithm are shown as follows.

(1) Initialization of particle swarm, the number of tasks, the number and the type of virtual machines [22]

The total population size is parNum, and the number of tasks is taskNum. There are vmNum virtual machines which include five types that are VM_1 , VM_2 , VM_3 , VM_4 , and VM_5 .

(2) Initialization of the velocity and position of particles

The position of the i^{th} particle is presented as X_i and X_i can be presented as $X_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$. The velocity of the i^{th} particle is presented as V_i and V_i can be represented as $V_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}$. The subscript n is between 1 and taskNum, and the subscript I is between 1 and parNum. X_{ij} represents that the task t_j is assigned to the node X_i , and x_{ij} is between 1 and vmNum. v_{ij} is between $-vmNum$ and $vmNum$.

(3) Definition of adaptability function

For each strategy of task allocation, there is a fitness value to measure the merits of the allocation strategy. In this article, for finding the particles of the least amount of the task cost, we have to find the particles of a high adaptability. The bigger the fitness value, the easier it is selected. The definition of fitness function is shown as follows. $f(i)$ is the fitness function and $SumCost(i)$ is the total cost of the i^{th} particle.

$$f(i) = \frac{1}{SumCost(i)}, 1 \leq i \leq parNum \quad (5)$$

(4) Individual extremum and global extremum

The individual extreme value of the i^{th} particle is the best of fitness values which can be represented as $pb_i = \{pb_{i1}, pb_{i2}, \dots, pb_{in}\}$. The global extreme value of the group is represented as $gb_i = \{gb_1, gb_2, \dots, gb_{parNum}\}$. ParNum represents the size of the population. According to the fitness function, there are two fitness values are obtained. $pb_i(t)$ is the individual extreme value of the i^{th} particle when the number of iterations is t . $fMax(t)$ is the max of the fitness value and it also can be represented as $fMax(t) \in \{pb_1, pb_2, \dots, pb_{parNum}\}$. $gb(t)$ is the global extreme value of the i^{th} particle when the number of iterations is t .

$$pb_i(t+1) = \begin{cases} pb_i(t), f(x_i(t+1)) \leq f(pb_i(t)) \\ x_i(t+1), f(x_i(t+1)) > f(pb_i(t)) \end{cases} \quad (6)$$

$$fMax(t) = getMax\{f(pb_1(t)), f(pb_2(t)), \dots, f(pb_{parNum}(t))\} \quad (7)$$

$$gb(t) = fMax(t) \quad (8)$$

(5) Comparing fitness with individual extremum and global extremum

For each particle, if it's fitness value is greater than the individual extremum, then the fitness value is used to instead the individual extremum. If it's fitness value is greater than the global extremum, the fitness value is used to instead the global extremum.

(6) Update of the particle's speed and position [22]

Each generation of particle update their speed and location according to the following formula. The parameter t represents the number of iterations, and w is inertia weight. The parameter called c_1 and c_2 is learning factor, and the function called $Rand()$ is a random number between 0 and 1.

$$v_i(t+1) = w \times v_i(t) + c_1 \times Rand() \times (pb_i(t) - x_i(t)) + c_2 \times Rand() \times (gb_i(t) - x_i(t)) \quad (9)$$

$$x_i(t+1) = x_i(t) + v_i(t) \quad (10)$$

4.3. Implementation Steps of PSO-SC Algorithm

Figure.3 briefly describes the steps of the PSO-SC algorithm which is used to perform the execution of task scheduling of workflow.

```
Algorithm: PSO-SC
Input: a group of workflow randomly assigned, the number and the length of tasks
Output: completion time of workflow, the execution cost of workflow, the best scheme
1. initialize the  $w$ ,  $c_1$ ,  $c_2$ , population size, the number of iterations and the number of times of the
   constant iteration result;
2. while ( $t < t_{max}$ ) and ( $k < 60$ );
3.   do for each particle
4.     for each task
5.       do computing the particle's fitness value according to the formula (5);
6.       do computing the individual extremum of particles according to the formula (6);
7.       do if fitness value is greater than individual extremum;
8.         then using the fitness value to instead the individual extremum;
9.       end if
10.      do computing the global extremum according to the formula (8);
11.      do if fitness value is greater than the global extremum;
12.        then using the fitness value to instead the global extremum;
13.      end if
14.      do updating the particle's velocity according to the formula (9);
15.      do updating the particle's position according to the formula (10);
16.    end for
17.  end for
18.  record the optimal solution of the current iteration;
19.   $t=t+1$ ;
20.  if the result of the current iteration is not the same as the last
21.    then  $k=0$ ;
22.    else  $k=k+1$ ;
23.  end if
24. end while
25. find the optimal scheme and bind the virtual machine of the scheme and the corresponding task
    of workflow;
```

Figure 3. Pseudo Code of PSO-SC Algorithm

5. Simulation Experiment and Analysis

5.1. Experimental Parameters Settings

Since the FIFO scheduling algorithm [23] is widely used in the task scheduling of workflow, the simulation software will be used in this paper to carry on the simulation experiment of these two algorithms of task scheduling in cloud environment. In order to test that the algorithm proposed in this paper is superior to the FIFO algorithm on the performance of scheduling, a workflow model is designed to test the effectiveness of the PSO-SC algorithm. In this model, the length of tasks, the numbers of tasks and the relation between tasks are random. Compare these two different scheduling algorithms in the same simulation platform. The parameter setting shown in Table 1 which references the documents [24-25] will be used in this paper, and the optimal solution can be obtained in a short period. The terminal condition of the algorithm is that the iteration has reached the largest amount of iterations and continuous 60 times the total task time and task overhead did not change.

Table 1. Main Parameters of Algorithm

Name	Value
Population size	150
Inertial factor	0.9
Learning factor c_1	2
Learning factor c_2	2
Largest number of iterations	200

5.2. Experimental Results and Performance Analysis

In this article, a data center is simulated in the simulation experiment, and in which a certain number of five kinds of virtual machines with different executive capability are defined. The serial number of the virtual machines starts from 1, and the parameters of the five kinds of virtual machines are shown in Table 2, assuming that other attribute values of resources are same.

Table 2. Parameters of Virtual Machines

Types	Speed	Cost
VM_1	320	0.02
VM_2	446	0.03
VM_3	590	0.04
VM_4	678	0.05
VM_5	869	0.06

In the experiment, the maximum number of task is 500, and the value of tasks' length is from 2000 MI (Million Instruction) to 8000 MI. The experimental tests run 100 times, and the average results of 100 experiments are taken as mapping data. In the experiment, the total number of tasks and the amount of virtual machine are set in proportional allocation, so the interval of the proportion should be found and the algorithm of this article can get better effect when the proportion in the interval. In the experiment, the proportions separately are 1:1, 1:10, 1:100, 1:200, 1:300, 1:400, 1:500, and under the different proportions, the trend of total time to complete the task and the total cost are shown as Figure.4 and 5.

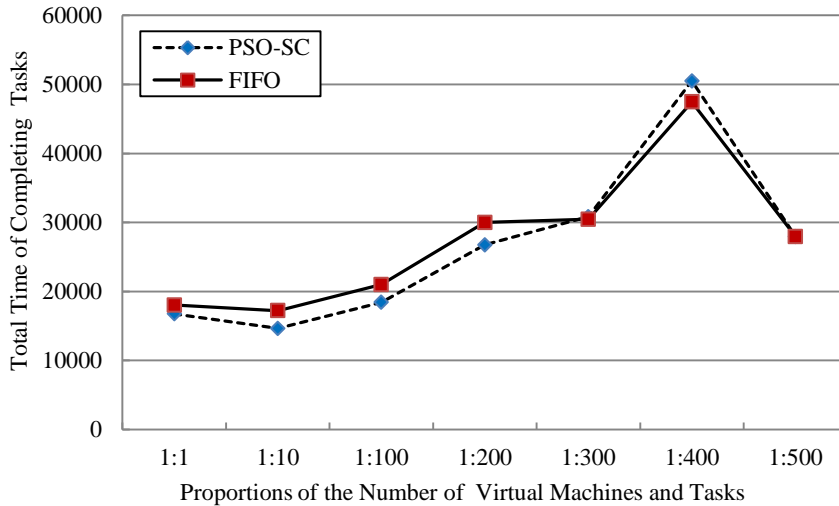


Figure 4. Total Time of Tasks of Different Proportions

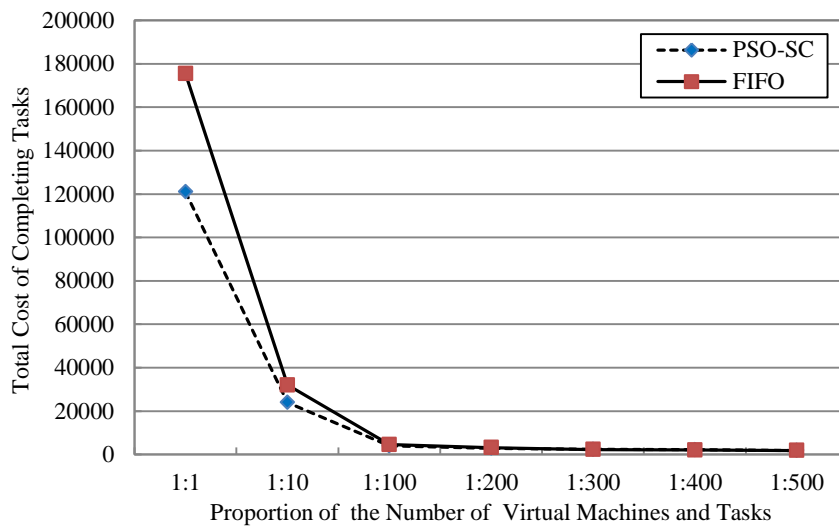


Figure 5. Total Cost of Tasks of Different Proportion

It can be seen from the Figure.4 that when the proportion is between 1 and 300, the total time to complete the task of PSO-SC algorithm has been less than FIFO algorithm. While the proportion is between 300 and 500, the total time to complete the task of FIFO algorithm is less than PSO-SC algorithm, so we select the proportion between 1 and 300. As can be seen from the Figure.5 when the proportion is between 1 and 100, the total cost of tasks of PSO-SC algorithm is less than FIFO algorithm. Though when the proportion is between 100 and 500, the total cost of tasks of the two algorithms is almost equal, as a result we select the proportion between 1 and 100. Combining Figure.4 and 5, when we select proportion of the number of tasks and the amount of virtual machines, we should choose the proportion between 1 and 100.

In the experiment, when we choose 1:5 as the proportion of the virtual machines for tasks to carry on the allocation, the allocation of the number of virtual machines of various types when executing the task is shown as in Table 3. The completion time of tasks is shown as in Figure.6, as well as the Figure.7 indicates the running time of tasks and the Figure.8 describes the total cost of tasks.

Table 3. Allocation of Virtual Machines

Number of tasks	100	200	300	400	500
VM_1	4	10	7	15	19
VM_2	3	6	9	16	18
VM_3	5	8	18	24	20
VM_4	3	7	10	15	21
VM_5	5	9	16	10	22
Sum	20	40	60	80	100

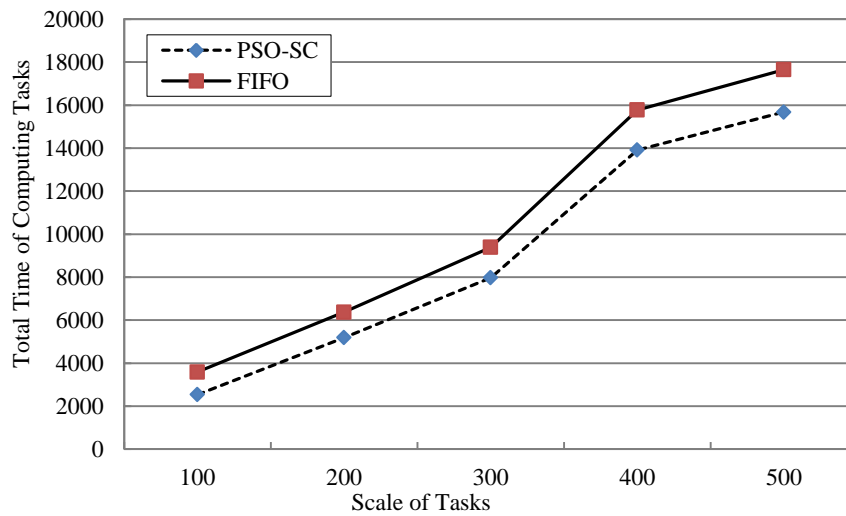


Figure 6. Task Completion Time of Two Algorithms

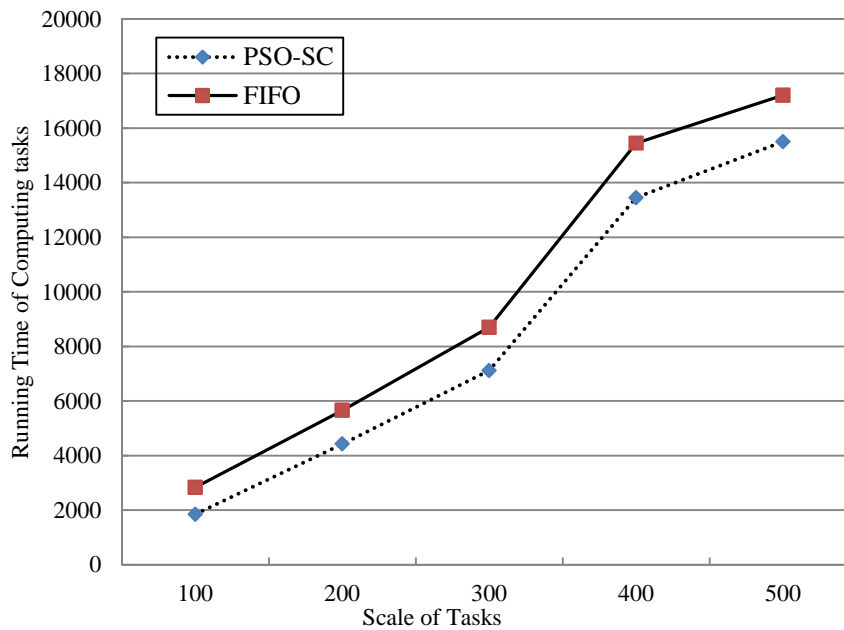


Figure 7. Task Running Time of Two Algorithms

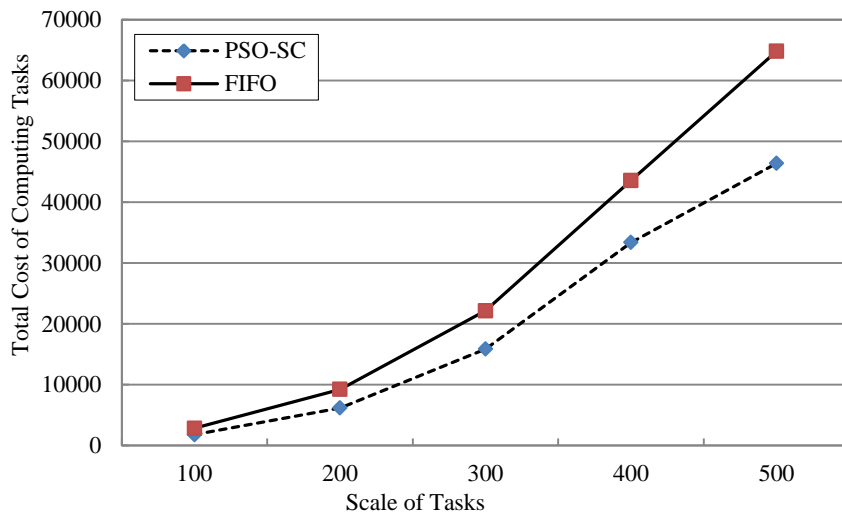


Figure 8. Task Cost of Two Algorithms

The Figure.6, Figure.7 and Figure.8 show the total time and the running time to complete the tasks, and the total cost of the task under the scheduling of these two algorithms. As can be seen from the figure, the total time, the running time and the total cost of processing tasks of FIFO algorithm are greater than PSO-SC algorithm. With the increase in the number of tasks, the gap of time and cost between the two algorithms keeps growing. It can be seen that the scheduling performance of PSO-SC algorithm is better than FIFO algorithm and has more advantages.

6. Conclusion

A workflow model based on the price model is put forward which can achieve multiple targets in this article. In this model, tasks are not independent but interrelated and they restrain each other. Because the particle swarm optimization is not only easy to fall into a local optimum and the effect of convergence is unsatisfactory but also the scheduling goal of most of scholars is relatively single when use particle swarm optimization to schedule tasks, service cost optimization algorithm based on particle swarm optimization and an objective function which considers the task costs is proposed in this paper. The result of the simulation experiment shows that the PSO-SC algorithm proposed in this paper can not only decrease the total time to complete the task as far as possible, meanwhile it can adjust the execution task on the basis of computing power of the virtual machines in the data center, and allocate these virtual machines rationally and realize the minimal goal of users' task overheads.

As for the future work, there are two interesting points that deserve further investigation. First, in this work, we assume that all tasks are mutually dependent, i.e., there are constrains between tasks, which are not enough for the relationship between tasks. Second, we assume that tasks are computationally intensive, which is not realistic for cloud systems. So, we should continue to develop the cloud computing which has played a key role in the practical application. Cloud computing transform the physical resources to dynamically scalable virtual resources by virtualization technology, as a result, the companies don't have to spend too much to buy expensive software and hardware facilities, and they can get access to the cloud resources through the way of leasing. In this way, it's helpful for small and medium enterprises to cut unnecessary cost.

References

- [1] Q. Zhang, L. Cheng, R. Boutaba. "Cloud computing: state-of-the-art and research challenges", *Journal of internet services and applications*, vol. 1, no. 1, (2010), pp. 7-18.
- [2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. "A view of cloud computing", *Communications of the ACM*, vol. 53, no.4, (2010), pp. 50-58.
- [3] R. Buyya, C. S. Yeo, S. Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities", *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications*, (2008) September 25-17; Dalian, China.
- [4] R. Buyya, C. S. Yeo, S. Venugopal J. Broberet, I. Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Generation computer systems*, vol. 25, no. 6, (2009), pp. 599-616.
- [5] B. P. Rimal, E. Choi, I. Lumb. "A taxonomy and survey of cloud computing systems", *Proceedings of the Fifth International Joint Conference on INC, IMS and IDC*, (2009) August 25-27; Seoul, Korea.
- [6] S. Zhang, S. Zhang, X. Chen, X. Huo. "Cloud computing research and development trend", *Proceedings of the Second International Conference on Future Networks*, (2010) January 22-24; Sanya, Hainan.
- [7] D. Ergu, G. Kou, Y. Peng, Y. Shi, Y. Shi. "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment", *The Journal of Supercomputing*, vol. 64, no. 3, (2013), pp. 835-848.
- [8] T. Dillon, C. Wu, E. Chang. "Cloud computing: issues and challenges", *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*; (2010) April 20-23; Perth, Australia.
- [9] W. Y. Lin, G. Y. Lin, H. Y. Wei. "Dynamic auction mechanism for cloud resource allocation", *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, (2010) May 17-20; Melbourne, Australia.
- [10] Y. Fang, F. Wang, J. Ge. "A task scheduling algorithm based on load balancing in cloud computing", *Web Information Systems and Mining*. Springer Berlin Heidelberg, (2010), pp. 271-277.
- [11] M. Xu, L. Cui, H. Wang, Y. Bi. "A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing", *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications*, (2009) August 10-12; Chengdu, China.
- [12] A Vouk M. "Cloud computing—issues, research and implementations", *CIT. Journal of Computing and Information Technology*, vol. 16, no. 4, (2008), pp. 235-246.
- [13] Y. Jiang, T. Hu, C. C. Huang, X. Wu. "An improved particle swarm optimization algorithm", *Applied Mathematics and Computation*, vol. 193, no. 1, (2007), pp. 31-239.
- [14] K. Li, G. Xu, G. Zhao, Y. Dong, D. Wang D, "Cloud task scheduling based on load balancing ant colony optimization", *Proceedings of the Sixth Annual on Chinagrid Conference (ChinaGrid)*, (2011) August 22-23; Liaoning, China.
- [15] J. F. Li, J. Peng. "Task scheduling algorithm based on improved genetic algorithm in cloud computing environment", *Jisuanji Yingyong/ Journal of Computer Applications*, vol. 31, no. 1, (2011), pp. 184-186.
- [16] C. Zhao, S. Zhang, Q. Liu, J. Xie, J. Hu. "Independent tasks scheduling based on genetic algorithm in cloud computing", *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing*, (2009) September 24-26; Beijing, China.
- [17] K. Etmnani, M. Naghibzadeh. "A min-min max-min selective algorithm for grid task scheduling", *Proceedings of the 3rd IEEE/IFIP International Conference in Central Asia on Internet*, (2007) September 26-28; Tashkent, Uzbekistan.
- [18] A. Salman, I. Ahmad, S. Al-Madani. "Particle swarm optimization for task assignment problem", *Microprocessors and Microsystems*, vol. 26, no. 6, (2002), pp. 363-371.
- [19] S. Pandey, L. Wu, S. M. Guru, R. Buyya. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments", *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, (2010) April 20-23; Perth, Australia.
- [20] M. Clerc. "Particle swarm optimization", *John Wiley & Sons*, (2010).
- [21] R. Poli, J. Kennedy, T. Blackwell. "Particle swarm optimization", *Swarm intelligence*, vol. 1, no. 1, (2007), pp. 33-57.
- [22] G. Venter, J. Sobieszczanski-Sobieski. "Particle swarm optimization", *AIAA journal*, vol. 41, no. 8, (2003), pp. 1583-1589.
- [23] N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand. "Achieving 100% throughput in an input-queued switch", *IEEE Transactions on Communications*, vol. 47, no. 8, (1999), pp. 1260-1267.
- [24] G. L. Chen, W. Z. Guo, Y. Z. Chen. "A PSO-based intelligent decision algorithm for VLSI floor planning", *Soft Computing*, vol. 14, no. 12, (2010), pp. 1329-1337.
- [25] I. C. Trelea. "The particle swarm optimization algorithm: convergence analysis and parameter selection", *Information processing letters*, vol. 85, no. 6, (2003), pp. 317-325.

Authors



Shengjun Xue, He was born in China on 1956. He is a Professor and now works at Nanjing University of Information Science & Technology. He served as dean of the school of Computer and Software and his research direction include cloud computing and big data.



Wenling Shi, She was born in China on 1992. She received her Bachelor's degree in Software Engineering at Nanjing University of Information Science & Technology in 2014. She is a graduate student now and she is interest in cloud computing.



Xiaolong Xu, He is currently working towards the PHD degree at the Department of Computer Science and Technology, Nanjing University, China. He received his Bachelor's degree in Software Engineering in 2010 and Master's degree in Computer Science in 2013. His research interests include cloud computing, green computing and big data.

