Research on Online Task Scheduling Mechanism in Cloud Computing

Jintao Jiao*

Department of Computer Science and Technology, Wuyi University, 354300, China

Abstract: Mainly the online scheduling focus on resources management and distribution, aim to satisfy the request of user, pay not enough attention to service provider. In order to increase providers' revenue, this paper researches the online task scheduling mechanism and proposes an improved online task scheduling mechanism. Firstly, we present the accepting strategy of the first task based on the yield of the new task. Secondly, after considering the sunk cost and opportunity cost, we present the accepting strategy of the non first task. Then, we improve the accepting strategy, and use the balance factors to control the portion of sunk cost and opportunity cost in order to get the maximum revenue. Simulation results obtained from CloudSim shows that the proposed mechanisms effectively reduce cost of services for providers, increase the profit of providers, and promote the harmonious development of scheduling environment.

Keywords: Cloud computing, scheduling strategy, online scheduling.

1. INTRODUCTION

Cloud computing is designed to achieve the collaboration and resource sharing. But the heterogeneity and dynamic of different kinds of machine, and diversity of user' demand in cloud computing environment, make the resource management in such environment extremely complex.

The market mechanism in economics can fully inspire and arouse the enthusiasm of individual, realize the rational allocation of social resources in commodity production and exchange, then improve the social productivity. Cloud computing environment, which is composed of many highly heterogeneous and dynamic nodes, has great similarity with the reality of economic society. So there is feasibility and superiority to realize the resource management in the cloud computing environment [1].

There are two participants in cloud computing, named service providers and service requester. The existed online task scheduling strategies [2-4] are more concerned about the request of service requestor than the profit of service providers [4]. This paper focuses on the online task scheduling strategy, from the perspective of improving the service income. On the other hand, it introduce the concept of sunk costs in reference to the opportunity cost, to research the accept strategy of the provider. Specifically speaking, this paper aims to maximize the return on the service providers through study how to calculate sunk costs and opportunity cost and how to coordinate relationship between cost and opportunity cost. Compared to the task scheduling strategy in literature [4] the proposed strategy can reduce the service cost, so users are more satisfied with the service, because it can promote the scheduling environment to develop healthily and harmoniously.

2. STUDY OF ONLINE SCHEDULING

Existing task scheduling algorithms are mainly based on task priority, resources for task execution and so on. They consider mainly about the users, while the service providers are not adequately considered.

The literature [5] proposed an online task scheduling algorithm, dynamic scheduling for multiple tasks with different priorities. It gives priority to complete the higher priority task, and tries to finish the low priority task after the priority tasks are done, for the goal to make higher priority task finished before the time line. So it is more likely to complete the higher priority tasks. In the article three heuristic algorithms are proposed to select tasks which can occupy the highest priority in the dynamic environment; and for the low priority tasks, it take strategy to assign parts of resources for the low priority tasks to be completed before the bottom line even in high priority task preemption case.

For online non preemptive scheduling, we always assume the task is single, or assume that tasks require only one resource. These assumptions are not appropriate to simulate the actual situation. The literature [6] for online non preemptive scheduling model gives a new task scheduling algorithm, taking into account the task can be composed of one or more parts, and tasks may require one or more resources to complete, which make tasks can wait in different queues. This paper presents loose distribution strategy, to analyze the above situation; and simulation experiment also confirmed the validity of the algorithm. However, the literature [6] does not consider the full for the service provider.

1874-4443/14 2014 Bentham Open

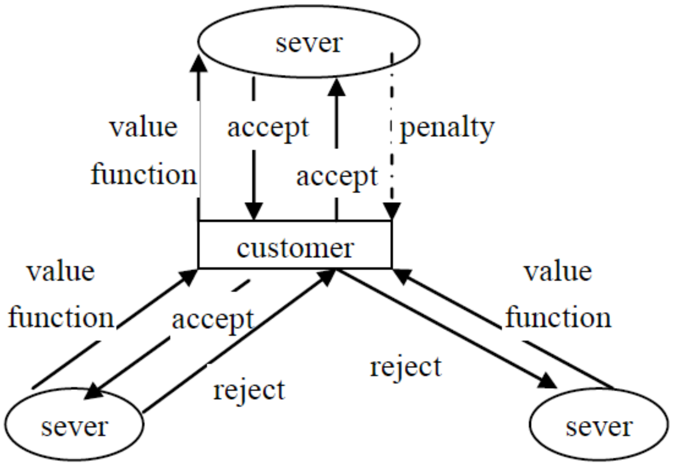

Fig. (1). Model of task scheduling.

The literature [7] proposes a new task scheduling algorithm based on single processor. The target is before reaching the tasks prescribed time limits, as much as possible the number of completed tasks submitted by users.

The literature [4] introduces the concept of opportunity cost in economics in the study of task scheduling based on market mechanism. It computes the cost of the task scheduling in detail, to decide whether to accept the task, thereby reducing the risk borne of the service provider, and service provider can get more profit. However, the literature [4] only considers the opportunity cost, does not consider other interest loss in detail.

3. TASK SCHEDULING MODEL

Task scheduling in cloud computing involves two participants, namely, service providers and service requester. The service provider provides the required services, and is responsible for scheduling tasks. For QoS users, the user can make choice according to their own conditions (such as capital, time and so on). If the user has more money, they can choose a better service provider (relatively more expensive). If the task are not completed in the effective period of time (such as deadline), the service provider need to pay compensation to the user. Payment of compensation method can effectively restrain the service, thus it can reduce the risk of paying customers.

Task scheduling model of this paper is based on the literature [4, 8], using the value function concept. However, because this chapter proposes a more comprehensive scheduling algorithm than literature [4, 8], it can extend the connotation of value function, as shown in Fig. (**1**).

This paper relates to the task value function which is a five tuple (deadline, reward, decay, bottom line, and penalty), as follows:

(1) Deadline: task time line, reflecting user's time expectations in the task. When failed to meet user's expected time, service provider has to pay for the user's compensation;

(2) Reward: the service fees users pay. When the user's time demand is satisfied, the provider is paid, without payment of compensation;

(3) Decay: the ratio of service provider to cover the user compensation. When the service provider failed to meet the user's requirement, the user's payment of damages in unit time;

(4) Bottom line: time point service provider to cover the maximum compensation, when the service provider failed to meet user's time needs, as time increases, the user is required to pay more compensation At this moment, compensation reaches the maximum value, and no longer increases. Obviously, bottom line>deadline;

(5) Penalty: the highest compensation service provider pay. When the service provider failed to meet the user's time needs, along with the time increasing, the compensation is also increasing. While the amount of compensation goes up to the penalty point, it no longer increases.

4. ONLINE TASK SCHEDULING STRATEGY BASED ON COST CACULATION

Usually, online scheduling model allocate resources for each arriving task, and implement immediately. The benefit of this scheduling approach is that it can simplify scheduling, and there is a wide range of practical applications [9].

This paper studies online scheduling cost calculation based on accepting strategy. The research hypothesis is based on the following condition:

Hypothesis 1: in online scheduling, task execution can be preempted by the other task, and once it is preempted, it need to access to resources to be executed from the beginning.

Hypothesis 2: in online scheduling, only when service provider completes the task, it is capable to get the user pay.

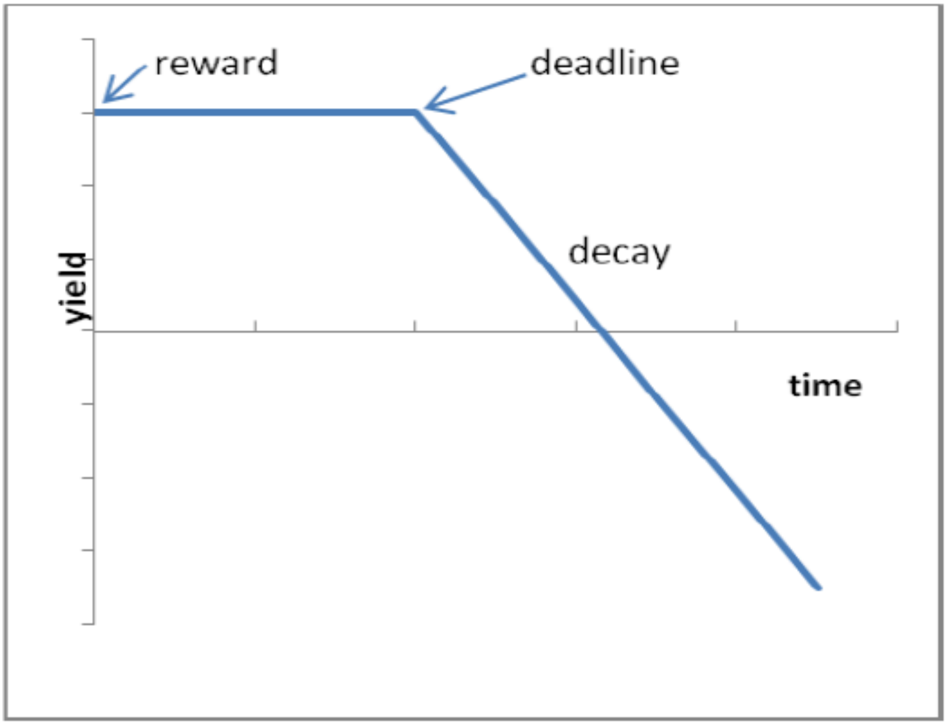

Fig. (2). Service income varies with time when maximum compensation is not set.

Hypothesis 3: task completed by service provider can meet the user's QoS requirement.

In the online scheduling, we can set the priority of task, in which when the service provider receives a new task, if the implementation of the new task will gain more income, you can temporarily stop the current task execution, and re-awaken the task to continue execution at the completion of the new task. The hypothesis consistent with the characteristics of online scheduling, and is conducive for the service provider to access higher income, so the hypothesis is reasonable.

4.1. Accepting Strategy of the First Task

In the idle state, as the service provider receives the first task value function, because the service provider's resources are idle, preemption of resources does not exist between tasks. At this time, service provider receives the task returns, and determines whether to accept the task according to the value function of the calculation task.

The value function of the five tuple in penalty can be set to 0. That is the task does not exist the highest compensation, and compensation increases as the lag time increased, until the task was completed. In this case the bottom line is +∞. Fig. (**2**) shows the service income varies with time when the highest compensation does not exist.

$$\mathrm{yield_i} = \mathrm{reward_i} \ ,$$

$$\text{task running time service provider expected} \leq \text{deadline} \quad (1)$$

$$\mathrm{yield_i} = \mathrm{reward_i} - (\mathrm{delay_i} * \mathrm{decay_i}) \ ,$$

$$\text{task running time service provider expected} > \text{deadline} \quad (2)$$

Fig. (**3**) shows the service income varies with time when the task set a maximum compensation which means penalty≠0. The service revenue ($\mathrm{yield_i}$) calculation shows as formula 3, formula 4 and formula 5, and $\mathrm{penalty_i}$ is the maximum compensation of task i when timeout.

$$\mathrm{yield_i} = \mathrm{reward_i},$$

$$\text{task running time service provider expected} \leq \text{deadline} \quad (3)$$

$$\mathrm{yield_i} = \ \mathrm{reward_i} - (\mathrm{delay_i} * \mathrm{decay_i}) \ ,$$

$$\text{deadline} < \text{ask running time service provider expected} < \text{bottomline} \quad (4)$$

$$\mathrm{yield_i} = -\ \mathrm{penalty_i},$$

$$\text{task running time service provider expected} \geq \text{bottomline} \quad (5)$$

Therefore, the first task accepting strategy is shown as follows. First the service provider estimates the execution time of the task when receives the value function of five tuple, then according to the formula 1 and formula2 and formula 3, formula 4 and formula 5 calculates the revenue of the task. If $\mathrm{yield_i}$>0, the task will be accepted.

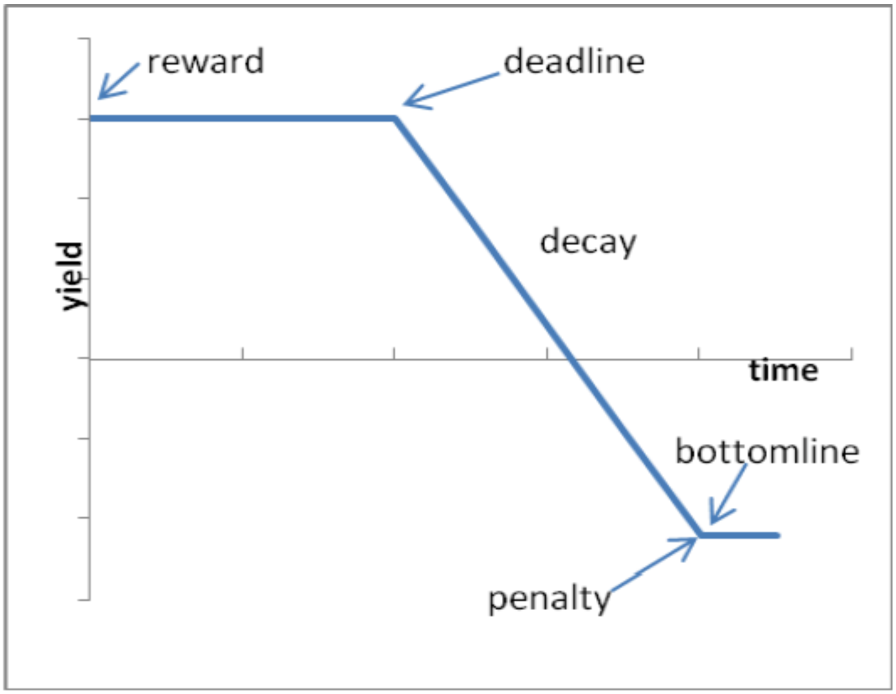

Fig. (3). Service incomes vary with time when the maximum compensation is set.

4.2. Accepting Strategy of Non First Task

While the service provider is performing a task, the new arriving task is called "non first task". In this case, the service provider calculates the sunk cost and opportunity cost, and determines whether to accept the new task or not.

(1) Accepting strategy based on the sunk cost

Sunk cost is the cost that have been paid and non recoverable. When the service provider is in the implementation of task i-1, the resources such as calculation, storage and bandwidth had already been put into, so there have been some sunk costs. Define the task execution time as sinking time before it is preempted. When the highest compensation is not set to task i-1 which is in the implementation, if service provider decides to accept task i and preempt task i-1, the cost is shown in Formula 6. sc_{i-1} is the sunk cost already put into task i-1.

$$\mathrm{cost}_i = \mathrm{sc}_{i-1} + \mathrm{delay}_{i-1} * \mathrm{decay}_{i-1} \quad (6)$$

When the highest compensation is set to task i-1 being performed and service provider decides to accept task i and preempt task i-1, the cost is shown in Formula 7 and Formula 8. s_time $_{i-1}$ is the sunk time of task i-1.

$$\mathrm{cost}_i = \mathrm{sc}_{i-1} + \mathrm{delay}_{i-1} * \mathrm{decay}_{i-1}, \mathrm{deadline}_i + \mathrm{deadline}_{i-1} + \mathrm{s\_time}_{i-1} \leq \mathrm{bottomline}_{i-1} \quad (7)$$

$$\mathrm{cost}_i = \mathrm{sc}_{i-1} + \mathrm{penalty}_{i-1}\mathrm{deadline}_i +,$$

$$\mathrm{deadline}_{i-1} + \mathrm{s\_time}_{i-1} > \mathrm{bottomline}_{i-1} \quad (8)$$

If $\mathrm{yield}_i - \mathrm{cost}_i > 0$, the service provider can benefit from the acceptance of task i.

If $\mathrm{yield}_i - \mathrm{cost}_i = 0$, the service provider will neither benefit nor loss from the acceptance of task i, which equivalents to have not perform the task , but the time already spend is wasted.

If $\mathrm{yield}_i - \mathrm{cost}_i < 0$, the service provider will loss to accept task i.

$\mathrm{yield}_i + \mathrm{yield}_{i-1} - \mathrm{cost}_i > 0$ indicates that the income by executing the task i-1 subsidies cost of task i, and there is still residual income.

$\mathrm{yield}_i + \mathrm{yield}_{i-1} - \mathrm{cost}_i < 0$ indicates that the income by executing task i-1 completely subsidies cost of task I, and there is still loss.

When the non first task arrived and the service provider does not consider the execution efficiency but only pursuit the explicit revenue, the service provider calculates the value of $\mathrm{yield}_i - \mathrm{cost}_i$, accept the task i while $\mathrm{yield}_i - \mathrm{cost}_i > 0$.

(2) Accepting strategy based on the task of sunk cost and opportunity cost

The accepting strategy based on sunk cost only considers explicit cost, but when the service provider decides to accept task i and preempt task i-1, there will be opportunity cost.

Opportunity cost is the loss of potential gain from other alternatives when one alternative is chosen. When task i preempts other tasks, we define the opportunity cost (oc_i) as Formula 9 and Formula 10.

$$\mathrm{oc}_i = \sum_{j=0, j\neq i}^{n} \mathrm{decay}_j * \left(\mathrm{bottomline}_j - \mathrm{deadline}_j\right),$$

$$0 < \mathrm{deadline}_i \leq \mathrm{deadline}_j \quad (9)$$

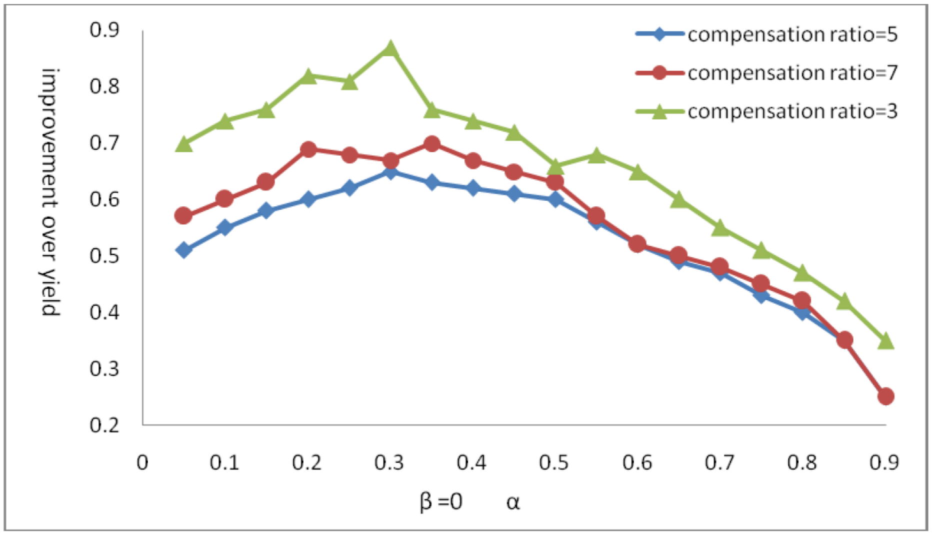

Fig. (4). Relationship between income and opportunity cost.

$$\mathrm{oc_i} = \sum_{j=0,j\neq i}^{n} \mathrm{decay_j} * \left(\mathrm{bottomline_j} - \mathrm{deadline_i}\right),$$

$$\mathrm{deadline_j} < \mathrm{deadline_i} \leq \mathrm{bottomline_j} \qquad (10)$$

If bottom line of task j does not exist, the estimation completion time of task i can be used instead of bottom line.

When the service provider is in the line of duty and at the same time receives multiple task request, sunk cost and opportunity cost are helpful to decide whether to accept the new task and which one to accept. First the service provider computes the explicit cost of task i, then for all tasks that are $\mathrm{yield_i} - \mathrm{cost_i} > 0$, select the task which has the lowest sunk cost and opportunity cost to preempt task i-1.

Opportunity cost can be used to calculate the degree of urgency. When the task is urgent, the opportunity cost can give priority to find the most urgent task [4].

4.3. Task Accepting Mechanism for Maximum Revenue

In online scheduling, preemptive task will cause the sunk cost and opportunity cost and service provider can accept new task control according to the strategy introduced. However, when total cost of the sinking and opportunity caused by the multiple tasks is equal, service provider cannot judge which one to accept. For example, the service provider receives two new missions A and B, when the sunk cost A is higher than sunk cost caused by B, but opportunity cost of A is lower than opportunity cost of B, no matter accepts A or B service provider can also benefit. At this time, on the basis of the above task accepting strategy, service provider cannot determine which new tasks can gain maximum benefit.

About this problem, this section proposes a service revenue maximization task acceptance mechanism, which can be used in the situation when the sum of the sunk cost and the opportunity cost is roughly the same or the difference can be neglected. It will decide how to select one of a plurality of tasks, and achieve service revenue maximization.

Give $\mathrm{result_i}$ a definition, as the income of task i under different cost weight. The balance factors (α and β) respectively represents the proportion of income and sunk cost, the addition of proportion of income, sunk cost and opportunity cost is equal to 1. The calculation of $\mathrm{result_i}$ is shown as Formula 11, where $\mathrm{cost_i}$ means the sunk cost, $\mathrm{oc_i}$ refers to the opportunity cost, $\mathrm{expect_i}$ refers to the execution time of task i.

$$\mathrm{result_i} = (\alpha * \mathrm{yield_i} - \beta * \mathrm{cost_i} - (1 - \alpha - \beta) * \mathrm{oc_i})/\mathrm{expect_i} \qquad (11)$$

The procedure of obtaining balance factors are as follows:

Step1: After the execution of first task, the service provider adjusts the balance factor to obtain maximum value of $\mathrm{result_i}$, then records the value of balance factor.

Step2: Service provider uses balance factors to estimate $\mathrm{cost_i}$ and $\mathrm{oc_i}$ when receives multiple tasks, calculatesresult$_\mathrm{i}$, selects maximum $\mathrm{result_i}$ task to execute; after the task is completed, the actual $\mathrm{cost_i}$ and $\mathrm{oc_i}$ can be obtained; calculates present value of balance factors corresponds to the maximum value of $\mathrm{result_i}$; then gets the average value of present balance factors and the original ones as the new balance factors, applies the new balance factors into the next judgment.

5. EXPERIMENTAL SIMULATIONS AND ANALYSIS OF RESULTS

This paper uses CloudSim [10] to simulate the accepting strategies of online scheduling market, develops with Java on the platform of Eclipse. The input task data is from the literature [11-13], the arrival time and execution time are exponential distribution. We use simulation method [4] to calculate opportunity cost, make assumption that 20% task pays are higher and the left are lower. The task pay is selected randomly in [5, 5000]. The distribution ratio of task completed or not is assumed to be double peak. And the sunk cost distributes uniformly between [5, 5000].

We set the value of β to 0, then research the relationship between income and opportunity cost (the opportunity cost

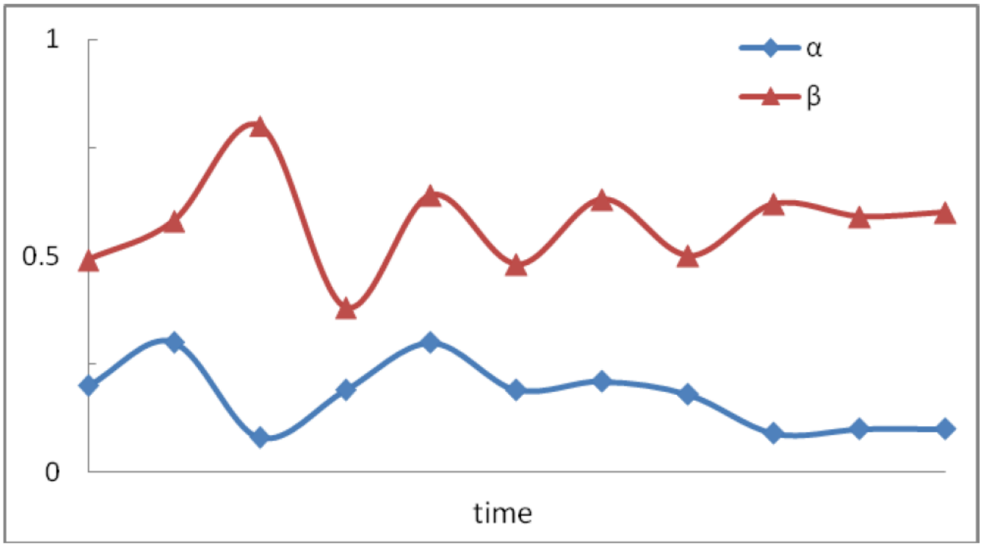

Fig. (5). Curve of α and β .

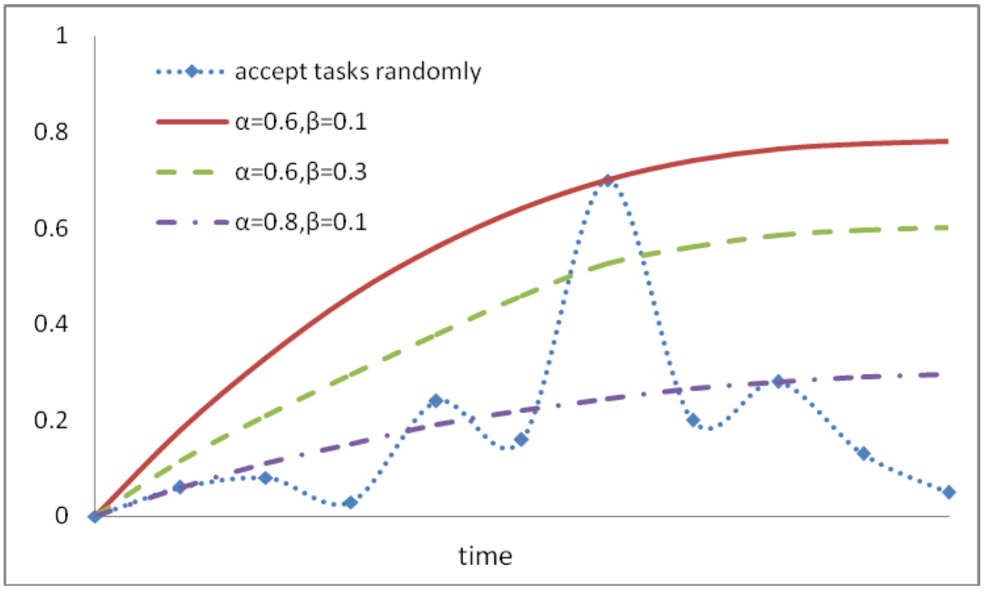

Fig. (6). Revenue per unit of time in case of different balance factors and accepting tasks randomly.

changes more violently than sunk cost, so it influents much more on income). Fig. (**4**) shows the relationship between the proportion of opportunity cost and income increment with different compensation ratio. When the compensation ratio is 3 and α is 0.3, service provider will earn the largest income. If the compensation ratio is low, the income will be high; and if the value of α is small, the opportunity cost accounts for a larger proportion, which means that the more opportunity cost is considered, the greater yield service provider will get.

Fig. (**5**) shows the curve of balance factor α and β based on the experimental results. At the beginning, tasks are performed, and the new task preempts the original one, the service provider continues to adjust its balances factor, so the balance factors go up and down. As time goes by, more tasks are completed, then the balance factors reach a status of balance, they basically don't change any more. At this time, the service provider has got maximum result_i under the action of balance factors, also reduced the service cost and maximized revenue.

Due to the balance factors are constantly changing throughout the accepting strategy, the effects of balance factors on service revenue cannot be seen directly. In the experiment we chose two special states, one is$\alpha = 0.8, \beta = 0.1$; another is$\alpha = 0.6, \beta = 0.3$, and assume the value of balance factors does not change; then simulate the experiment again, get the value of result_i. When the service provider finally tends to equilibrium state, $\alpha = 0.6, \beta = 0.1$. Fig. (**6**) shows the service revenue under the three conditions. In order to compare more intuitively, we also simulate the income per unit of time when service provider accept the task randomly. Finally, we can see the importance of the balance factors from the results.

CONCLUSION

This paper presents an online task scheduling strategy based on cost computing in the cloud computing environment; first elaborates the first task scheduling strategy of service provider and introduces the sunk cost; then analyzes the non first task scheduling strategy under the premise of calculation of sink cost and opportunity cost; finally compares the relationship between sink cost and opportunity cost, analyses the impact of sunk cost and opportunity cost for the service revenue. Simulation results show that the proposed mechanisms effectively reduce cost of services for providers, increase the profit of providers, and promote the harmonious development of scheduling environment.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This research was partially supported by the National Nature Science Foundation of China (No.61173032), the Tianjin Higher Education Science and Technology Development Fund (No.20100806) and the Natural Science Foundation of Tianjin, China (No. 13JCYBJC15500).

REFERENCES

[1] K. Subramoniam, M. Maheswaran, and M. Toulouse, “Towards a micro-economic model for resource allocation in grid computing systems”, In: *Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering*, 2002, pp. 782-785.

[2] B. Chun, *Market-based cluster resource management*, Berkeley: University of California, 2001.

[3] B.N. Chun, and D.E. Culler, “User-centric performance analysis of market-based cluster batch schedulers”, In: *2nd IEEE International Symposium on Cluster Computing and the Grid*, 2002.

[4] D.E. Irwin, L.E. Grit, and J.S. Chase, “Balancing risk and reward in a market-based task service”, In: *13th IEEE International Symposium on High performance Distributed Computing*, 2004.

[5] C. Chiyu, and S. Bhattacharya, “Dynamic scheduling of real-time messages over an optical network”, In: *6th IEEE International Conference Computer Communications and Networks*, 1997.

[6] K. Ho, J.H. Rice, and J. Srivastava, “Real-Time scheduling of multiple segment tasks”, In: *14th IEEE Annual International Computer Software and Applications Conference*, 1990.

[7] R. Abbott, and H. Garcia-Molina, “Scheduling real-time transactions: A performance evaluation”, *ACM Transact. Database Syst.*, vol. 17, no. 3, pp. 513-560, 1992.

[8] H. Song, S.B. Yang, and X.Q. Liu, “A quality-driven algorithm for task scheduling in grid market,” *J. Graduate School Chinese Acad. Sci.*, vol. 28, no. 1, pp. 86-93, 2011.

[9] J. Sgall, “On-line scheduling--a survey”, *Lect. Note Comput. Sci.*, vol. 1442, pp. 196-231.

[10] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, and R. Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”, *Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2011.

[11] A. Downey, and D. Feitelson, “The elusive goal of work-load characterization”, *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, no. 4, pp. 14-29, 1999.

[12] V. Lo, J. Mache, and K. Windisch, “A comparative study of real work load trace sand synthetic workload models for parallel job scheduling”, In: *Proceedings of the 4th Work-shop on Job Scheduling Strategies for Parallel Processing*, 1998.

[13] H. Shan, L. Oliker, and R. Biswas, “Job super scheduler architecture and performance in computational grid environments”, In: *Proceedings of Supercomputing*, 2003.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014