

Research Article

A Dynamic Resource Scheduling Method Based on Fuzzy Control Theory in Cloud Environment

Zhijia Chen, Yuanchang Zhu, Yanqiang Di, and Shaochong Feng

Department of Electronics and Optics, Mechanical Engineering College, Shijiazhuang 050003, China

Correspondence should be addressed to Zhijia Chen; youshenshui@163.com

Received 12 January 2015; Revised 26 May 2015; Accepted 2 June 2015

Academic Editor: Kalyana C. Veluvolu

Copyright © 2015 Zhijia Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The resources in cloud environment have features such as large-scale, diversity, and heterogeneity. Moreover, the user requirements for cloud computing resources are commonly characterized by uncertainty and imprecision. Hereby, to improve the quality of cloud computing service, not merely should the traditional standards such as cost and bandwidth be satisfied, but also particular emphasis should be laid on some extended standards such as system friendliness. This paper proposes a dynamic resource scheduling method based on fuzzy control theory. Firstly, the resource requirements prediction model is established. Then the relationships between resource availability and the resource requirements are concluded. Afterwards fuzzy control theory is adopted to realize a friendly match between user needs and resources availability. Results show that this approach improves the resources scheduling efficiency and the quality of service (QoS) of cloud computing.

1. Introduction

Cloud computing is an Internet-based computing service mode which makes it possible to obtain the configurable shared resources (network server storage applications and services, etc.) on demand. By the virtualization of hardware resources and the full sharing of virtual resources, combined with the diversity and dynamic characteristics of resources, cloud computing is able to process massive amounts of data and accomplish users' various needs, which include computing, storage, and graphic processing. The loosely coupling and the deeply sharing nature of virtualization technology make it difficult to measure the parameters and resources utilization information of every virtual machine (VM) precisely when the virtual machine is operating. Moreover, the imprecision of users requirement is a serious challenge to improving the quality of service (QoS) of cloud computing.

In the typical IaaS (Infrastructure as a Service) model of cloud computing, users submit resource requirements according to their requirements [1] which include hardware configuration such as the total CPU cores, memory size, GPU memory size, and storage size and software configuration such as operating system type and version and application

software. Cloud computing data centre allocates resources to users through the requisition they submitted. When the system works, the following situations may occur: (1) the virtual machine is overloaded and causes low performance, even downtime; (2) the virtual machine is underloaded and causes resources underutilization; and (3) different tasks lay emphasis on different types of resources. For example, user A weighs video animation playing against mathematics computing, while user B weighs mathematics computing against video animation playing. Then the GPU load of user A's virtual machine will be heavier than that of the user B's virtual machine, while the CPU load of the user A's virtual machine will be relatively lighter than that of the user B's virtual machine. The situations above cause some resources to be overloaded while some underloaded. The imbalance problem will result in low resource utilization rate and finally affect the QoS of cloud computing.

Nowadays, users have exigent requirements for resource scheduling technology. Commonly, users are not skilled in submitting precise and proper CPU, bandwidth, or memory requirements, but they can have fuzzy requirements based on the task attitude. How to satisfy these requirements gives expression to cloud computing friendliness, which means

the ability to understand and satisfy various users' imprecise requirements and enables system to understand users' fuzzy demands. It is also an important standard of the QoS of cloud computing. The virtualization technology enables hardware resources including CPU, memory, and GPU to be shared by multiusers simultaneously. This characteristic needs resource scheduling method to consider resource allocation overall. In addition, resources should be allocated to maximize the benefits of all resources rather than individual one [2].

Some characters of cloud computing can be utilized to schedule resources to satisfy users' requirements. As the user demand for cloud computing resources typically shows a long and continuous characteristic, the users' needs of resource are possible to predict. Based on the feature of moving average method [3], it is possible to predict the resource demand. Meanwhile, in resource scheduling, extra difficulty caused by user imprecise needs can be solved by fuzzy logic theory. Fuzzy logic can model the human mind through rigorous logical reasoning and build a "human-friendly system" [4] between users and cloud computing system. The dynamic resource scheduling can be optimized by feedback control. We can adjust the configuration based on the user needs and the performance of current resource dynamically.

In this paper, we present a dynamic resource scheduling method in data centre based on fuzzy control theory. The main contributions of this paper are concluded as follows: (1) a cloud computing resource scheduling system with feedback control is established. (2) Based on users' long-term and continuous resource requirements, a model of resource demand predicting is established. (3) Fuzzy control theory is introduced to the resource scheduling process to construct the resource scheduling system. The system divides user requirements and available resources into several fuzzy levels and processes user fuzzy requirements more friendly, which improves the QoS of cloud computing.

The rest of this paper is organized as follows. Section 2 discusses the current resource scheduling method and related research work. In Section 3, a prediction model of resource requirements is established. Moreover, the relationship between user resources requirement and resource supply is modeled. Section 4 presents a new dynamic fuzzy scheduling strategy. User requirements and available resources are both divided into corresponding fuzzy levels. The fuzzy requirements-resources matching system is designed and the algorithm is described in detail. In Section 5, experiments are conducted and the results are analyzed. Finally, we conclude the whole paper and list the future works.

2. Related Works

Currently, methods for resource scheduling in cloud computing are widely researched. For different targets, different resource scheduling methods were presented. These methods improved the efficiency of resource utilization and task processing to some extent. Some methods were inherited from grid task scheduling algorithm. For example, Ghodsi et al. [5] presented a max-min balancing algorithm with constraints, which was mainly used in cloud centre scheduling with strict hardware restrictions. Fang et al. [6] developed

the double-layer task scheduling mechanism based on load balancing. In this mechanism, one layer is from user to virtual machine, while another is from virtual machine to host resources. This method improved the utilization ratio and met users demand. Some researches introduced bionics to resource scheduling, such as genetic algorithm, ant colony algorithm, leapfrog algorithm, and particle swarm algorithm. To lower users' time and cost, Yan et al. [7] designed a resource scheduling strategy based on particle swarm algorithm. Results show that the strategy improves system responsiveness and resource utilization effectively. Huang et al. [8] introduced an improved resource scheduling strategy based on ant colony algorithm. The traditional ant colony algorithm feature is random, has slow convergence, and is easy to fall into local optimum. To overcome the drawbacks of ant colony algorithm, genetic algorithm was integrated with it. As a result, the convergence rate and calculation accuracy were both improved. Some researchers optimized cloud computing resource scheduling from the perspective of market rules and economics. Wu et al. [9] proposed a market-oriented hierarchical scheduling strategy. Through the service layer scheduling and task layer scheduling, minimum cost under the constraint of QoS was achieved and service efficiency was improved. Xu et al. [10] added Berg model into cloud computing resource scheduling. They constructed dual fairness constraints: one was to classify user tasks by QoS preferences and establish common expectations equation, and the other was to judge the fairness of resource allocation by defining resource fairness equation. By adopting this theory, efficiency of user task execution had been improved and fairness was better precipitated. Some scholars also have carried out researches from other aspects. Ma et al. [11] introduced ontology theory into cloud computing. According to the demand raised by users, user tasks were assigned to the most appropriate resources to meet the service levels agreement (SLA). Based on quantifying user's application preference and user effectiveness in multidimensional QoS, Sun et al. [12] proposed a multidimensional optimization objective function, which improved the availability and load balancing of resource scheduling system. Byun et al. [13] proposed a time-sharing resource scheduling mechanism, using the least amount of resources within a user-defined time to finish the task. There are also some researchers using fuzzy logic and expert system theory to optimize resources scheduling. To address the problem of scheduling experts in ensemble systems in the case of limited resources and minimally affected overall system performance, Rattanathamrong [14] proposed an architecture named as Elastic Ensemble Scheduling (EES) manager. The experiment results show that, with the EES manager scheduled, experts in a resource-constrained ensemble system are able to produce system outputs similar to those of the system with sufficient resources. In [15], researchers used fuzzy clustering to improve the resources utilization of complex hardware/software system. The method was tested in an embedded real-time system and it was able to schedule task resources efficiently. In [16], a fuzzy logic feedback-based resource scheduler was proposed to solve the problem of resource scheduling in real-time ensemble systems. It provided real-time execution of

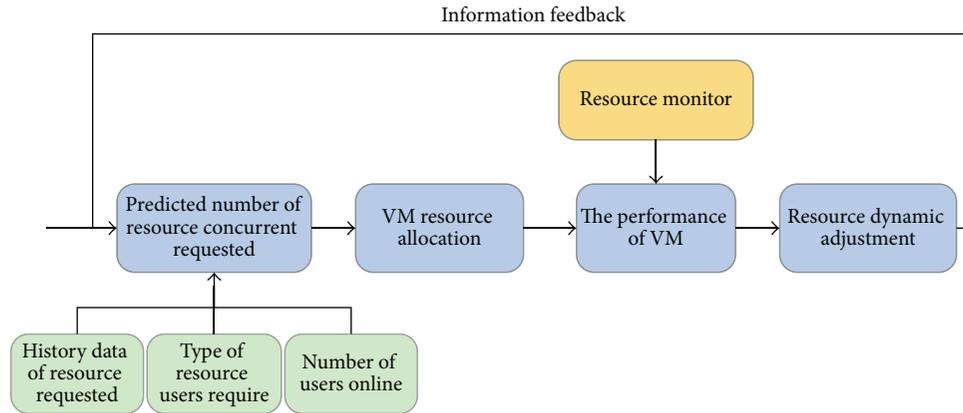


FIGURE 1: Dynamic configuration of system resources.

all relevant experts while minimizing the impact of limited resources and uncertainties on the system performance. By considering a fuzzy modeling of the workload inspired from the fuzzy approach, Masmoudi and Hait [17] proposed a technique that can keep uncertainty at all steps of the modeling and solving procedure. The method was applied to civil helicopter maintenance project and proved effective.

From different angles and by different methods, the above studies realize efficient resource scheduling and improve the task execution efficiency. The basic QoS standards such as cost, bandwidth, and completion time are met, but some extended QoS standards such as system friendliness and user experience were not mentioned. Whereas those extended QoS standards play a very important role for a customer to decide whether or not to use cloud computing many users generally cannot give a precise requirement in accordance with the attitude of the tasks. They commonly generate an imprecise requirement according to the character of the task. The ability to satisfy these requirements is an important manifestation of the friendliness of cloud computing. How to improve the system friendliness is the research emphasis of this paper. The paper introduces fuzzy logic and feedback control system into cloud computing resource scheduling strategy and constructs the user requirements model and resource availability model. According to the imprecise demands that users submit, corresponding resources are matched for the user needs, which can make use of the resource effectively and execute user tasks efficiently.

3. System Architecture and Related Models

3.1. System Architecture. Figure 1 shows the structure of system resource dynamic allocation based on information feedback. The amount of resources concurrently requested is the total amount of resources including CPU, GPU, and memory that users require. The total amount of resources concurrently requested can be predicted by the historical data of the requested resource, the type of resources, and the number of simultaneously online users. The resource managing system monitors the resource utilization of the virtual machines in real time. According to the relationship among

the characters of users' tasks, resource configuration, and the performance of virtual machine, the relationship model was constructed. Based on the model above, we dynamically adjust resource allocation of each virtual machine to modify the performance of virtual machine and improve the resource utilization. Using feedback control method, we send the adjustment parameter to the monitoring centre and optimize the allocation model. By this model, we ensure the availability and efficiency of cloud computing resources: avoid overload when resource requirements are on the peak and ensure service availability; reduce the occupation of resource when resource requirements are at trough and avoid wasting of resource.

3.2. System Execution Process. Firstly, according to the predicted total number of resources the user needs, cloud computing data centre evaluates the capability of the resource pool. Then it conforms the virtual resources as virtual machine and allocates user requirements to different virtual machines according to user's address. The data centre monitors each user's virtual machine resource utilization in real time, including the running status of virtual machine, indicators of CPU management, indicators of memory management, indicators of GPU management, and indicators of process management. The data are saved to get the status of resource usage. According to the running status of virtual machine, we dynamically adjust the virtual machine resource allocation. The monitoring centre can increase or decrease the resources assigned to the user based on resource usage statistics, which frees users from the machine management and maintenance. The process in which cloud computing centre dynamically provides resources for the user is shown in Figure 2.

3.3. Resource Requirements Prediction Model. We predict users' requirements of resources combining with the characteristics of the users. In this model, we use one day as a unit and predict resource requirements hourly. When predicting, we consider the number of resource requirements in one hour is independent of each other in the same day. So, we only need to refer to the number of requested resources of the same

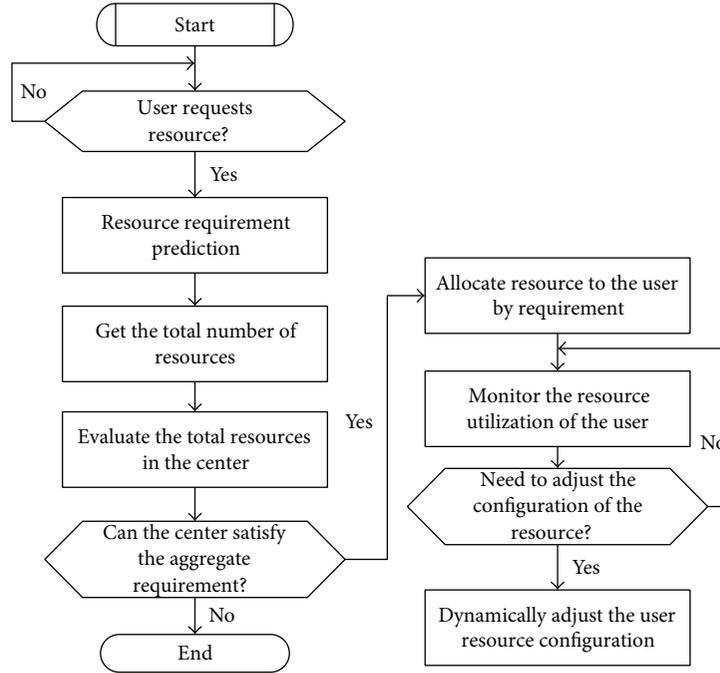


FIGURE 2: Dynamic resources supply of the cloud computing data centre.

time in history, rather than referring to the requested number of the former time in the same day. For example, if we need to predict the number of requested resources from 10:00 to 11:00, we may ignore the resources requested of other periods in the same day, such as from 9:00 to 10:00. We only need to send the concurrent requested number from 10:00 to 11:00 each day in the history to the prediction function as the input parameter. Long-term continuous operation of the cloud computing data centre can provide us with enough historical data. Thus, the prediction model will be more suitable for the resource configuration model in long-term. In view of the character of the prediction model above, to reduce the lag deviation of the prediction value and the actual value, we introduced the second moving average method. This method is particularly suitable for long-term prediction. Firstly, we calculate the first moving average value and the second moving average value at the t th time of the i th user's resource demands:

$$M_t^{(1)}(i) = \frac{x_t(i) + x_{t-1}(i) + \cdots + x_{t-(N-1)}(i)}{N}, \quad (1)$$

$$M_t^{(2)}(i) = \frac{M_t^{(1)}(i) + M_{t-1}^{(1)}(i) + \cdots + M_{t-1}^{(1)}(i)}{N}.$$

Here N is the time of moving average period. Then we introduce intermediate parameters $a_t(i)$ and $b_t(i)$. They are dependent variables of the first moving average value $M_t^{(1)}(i)$ and the second moving average value $M_t^{(2)}(i)$:

$$a_t(i) = 2M_t^{(1)}(i) - M_t^{(2)}(i), \quad (2)$$

$$b_t(i) = \frac{2}{N-1} [M_t^{(1)}(i) - M_t^{(2)}(i)].$$

i th user's requirements prediction at time $t + \tau$ can be expressed as follows:

$$x_{t+\tau}(i) = a_t(i) + \tau b_t(i). \quad (3)$$

Then the total number of the resources requested by all the m users is

$$x_{t+\tau} = \sum_{i=1}^m x_{t+\tau}(i). \quad (4)$$

From the analysis above, we can see that the prediction value at the time $t + \tau$ is decided only by N periods' values at the time t and the total number of the users. And the prediction result can be calculated at time t .

4. Dynamic Resource Scheduling Based on Fuzzy Control Theory

For cloud computing data centers, fuzzy equivalence classes can effectively improve friendliness and resource utilization of cloud computing. The meaning of friendliness is the ability to understand and satisfy various users' imprecise requirements and enable system to understand users' fuzzy demands. By ensuring the capability of understanding user's fuzzy requirements, the resources can be allocated according to the execution effect. Therefore, it is important to introduce fuzzy logic to resource scheduling. By constructing the fuzzy equivalence classes, we can adopt fuzzy control theory to schedule virtual machine resources dynamically within the limited range of QoS to achieve the purpose of improving the system friendliness and the efficiency and reducing system overhead.

4.1. Scheduling System Model. Fuzzy logic theory has played an important role in cloud computing. Frey et al. [18] described a method using fuzzy controller to control the scale of the cloud. Through some additional information, the maximum load is predicted, so the virtual cluster scaling can be completed in advance. Ramezani et al. [19] described a prediction method based on fuzzy logic. The virtual machine workload types and the migration time of virtual machines were predicted based on fuzzy prediction method, and finally resources automatic allocation for the application was realized. The researches above mainly introduced the fuzzy theory to load forecasting, but direct resource scheduling was not involved. This paper refers to the above research ideas, applying fuzzy logic directly to the resource scheduling.

For user tasks, different tasks require different types of virtual machine resources configuration. For the network communication tasks, more bandwidth resources need to be assigned to the corresponding virtual machines. While some tasks do lots of mathematic computation, then higher computing performance virtual machine resources are needed [20]. Thus, based on the QoS requirements of user tasks, the features information of virtual machine VM_i can be abstracted into some feature information granules C_{ij} , representing CPU, GPU, memory and bandwidth information granule, and so forth. Similarly, for user tasks and resource availability, we create different classification and fuzzy equivalence classes [21]. As shown in Figure 3, when the system has multiple user tasks arriving, firstly the tasks are sorted in accordance with the tasks submission sequence and then the first come first serve (FCFS) queue is adopted. According to user tasks type, duration, and other attributes, corresponding resources are inferred through fuzzy logic by reference to the fuzzy inference knowledge base. After getting the level of the required resources, the inference results are defuzzified based on the corresponding strategy. Finally, they are converted into precise data, including the specific requirement of CPU, GPU, and memory. According to the specific needs, the system realizes dynamic scheduling of resources. The purpose of improving resource utilization and execution ratio is achieved.

4.2. Fuzzy Inference. Referring to the levels of the service provisioning capabilities of the data center and the resource requirements of the users, the data centre defines the symbol as positive or negative in accordance with the increasing or decreasing of virtual machine resources. The input variables are the tasks' requirements for resources and available resources of users' virtual machines, respectively. The output variables are the resources that need to be increased to or decreased from the corresponding virtual machines. The fuzzy values of the input variables and output variables are described as Positive Big, Positive Middle, Positive Small, Zero, Negative Small, Negative Middle, and Negative Big. In accordance with the general definition of fuzzy logic, the above seven-level service capacity and the number of resources' scheduling are abbreviated as PB, PM, PS, ZO, NS, NM, and NB by descending order. Correspondingly, according to the number of resources to be scheduled, it is defined as Positive Big, Positive Middle, Positive Small, Zero,

TABLE 1: Resource scheduling table based on fuzzy rules.

S_j	T_{ij}						
	NB	NM	NS	ZO	PS	PM	PB
NB	ZO	PS	PS	PM	PM	PB	PB
NM	NS	ZO	PS	PS	PM	PB	PB
NS	NS	NS	ZO	PS	PM	PM	PB
ZO	NM	NM	NS	ZO	PS	PM	PB
PS	NM	NM	NM	NS	ZO	PS	PM
PM	NB	NB	NM	NS	NS	ZO	PS
PB	NB	NB	NM	NM	NS	NS	ZO

Negative Small, Negative Middle, and Negative Big, which can be abbreviated to PB, PM, PS, ZO, NS, NM, and NB. Assuming the user's tasks are divided into n levels, and there are m kinds of resources types, tasks of different level request different resources. If i th level task's request for j th class resource is very high, then set $T_{ij} = PB$. If the current j th class resource availability of the virtual machine is very low (NB), set $S_j = NB$. Then the number of resources scheduled is Positive Big (PB), set as $f_{ij} = PB$ (i.e., when the current task needs high computing performance, and the CPU resource availability is low, then the status of resources to be scheduled is high). Using MATLAB fuzzy programming language, the above reasoning process can be expressed as follows:

if (T_{ij} is PB) and (S_j is NB) then (f_{ij} is PB).

Take CPU resource scheduling as an example. Suppose the requirement for CPU resources of VM_1 to be T_{11} and the available CPU resources of VM_1 to be S_1 . They are input variables. The CPU resources to be scheduled f_{11} are output variables. The domain of the two input variables is $X, Y = \{-3, -2, -1, 0, 1, 2, 3\}$. The membership function is defined as triangular function and is shown in Figure 4(a). Correspondingly, the domain of output variables is $Z = \{-3, -2, -1, 0, 1, 2, 3\}$. The membership function is defined as triangular function and is shown in Figure 4(b).

According to the above definition and the scheduling principle, to improve the resource utilization and system performance, the rules of the resource scheduling are listed in Table 1. Based on the fuzzy rules of input variables and output variables, we get the resource scheduling input-output relationship surface diagram, as shown in Figure 5.

The max-min fuzzy inference algorithm is adopted as the inference algorithm. The algorithm can be formulated as

$$\mu_z^k = \max \left[\min \left[\mu_x^k, \mu_y^k \right] \right]. \quad (5)$$

Consider the following two fuzzy rules:

R_1 : if X is A_1 and Y is B_1 then Z is C_1 .

R_2 : if X is A_2 and Y is B_2 then Z is C_2 .

As we define the membership function as triangular function, we can use graphic method to explain the max-min algorithm as the former can be more easily realized by computer programs. The graphic representation of max-min inference algorithm is shown in Figure 6.

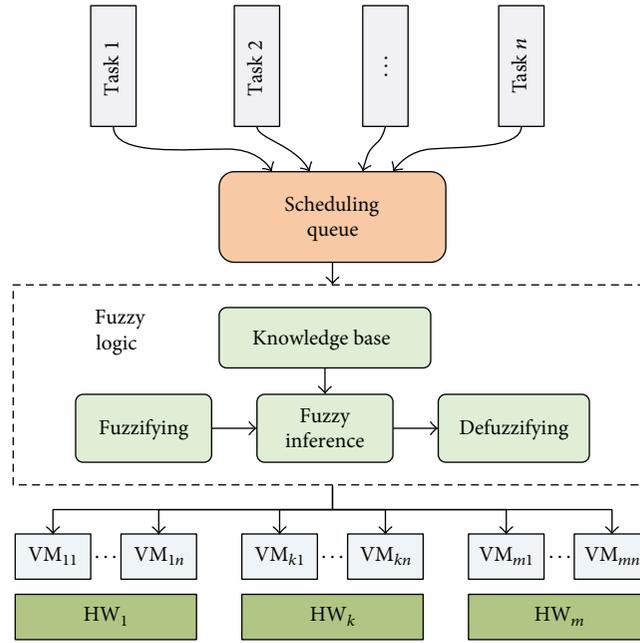


FIGURE 3: Resource scheduling process based on fuzzy logic.

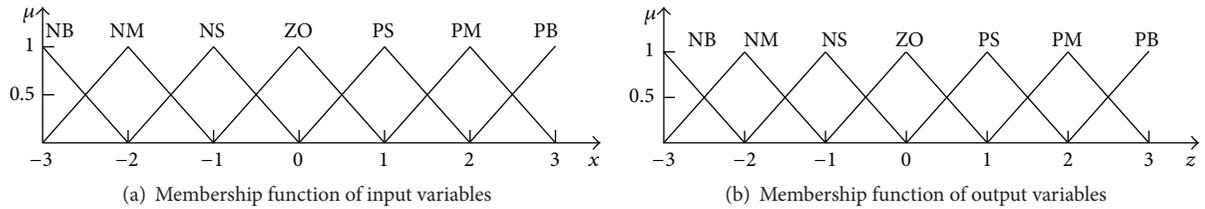


FIGURE 4: The membership functions of input and output variables.

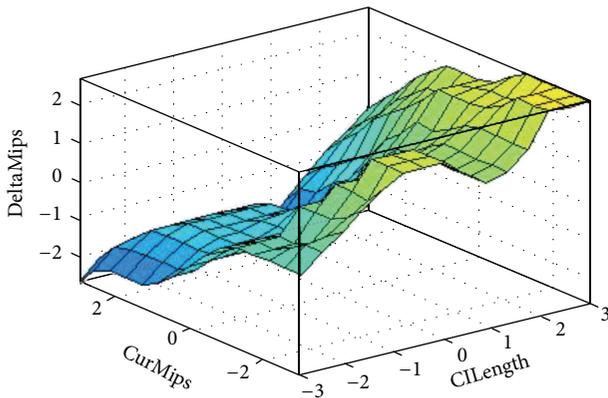


FIGURE 5: Relationship of input and output variables of resource scheduling algorithm.

In Table 1, the data are all imprecise. Based on relevant defuzzification algorithm [22], we can get the precise data after defuzzifying. In this paper, we use the centroid algorithm as the solution. Meanwhile, based on the experiment results and control experience, we modified the basic fuzzy

TABLE 2: The control enquiry table of precise data after system defuzzification.

S_j	T_{ij}						
	-3	-2	-1	0	1	2	3
-3	0	1	1	2	2	3	3
-2	-1	0	1	1	2	3	3
-1	-1	-1	0	1	2	2	3
0	-2	-2	-1	0	1	2	3
1	-2	-2	-2	-1	0	1	2
2	-3	-3	-2	-1	-1	0	1
3	-3	-3	-2	-2	-1	-1	0

inference algorithm. The precise data after system defuzzification are listed in Table 2, which is also known as “control enquiry table.” Scheduling resources by means of fuzzy logic can effectively improve the utilization of resources and the service quality.

4.3. The Flow of Resource Scheduling Algorithm. The basic execution process of fuzzy dynamic scheduling algorithm based on task demand and resource availability is as follows.

TABLE 3: The task allocation schemes.

	VM1	VM2	VM3	VM4	VM5
Length of tasks	20000–30000	30000–40000	40000–50000	50000–60000	60000–70000

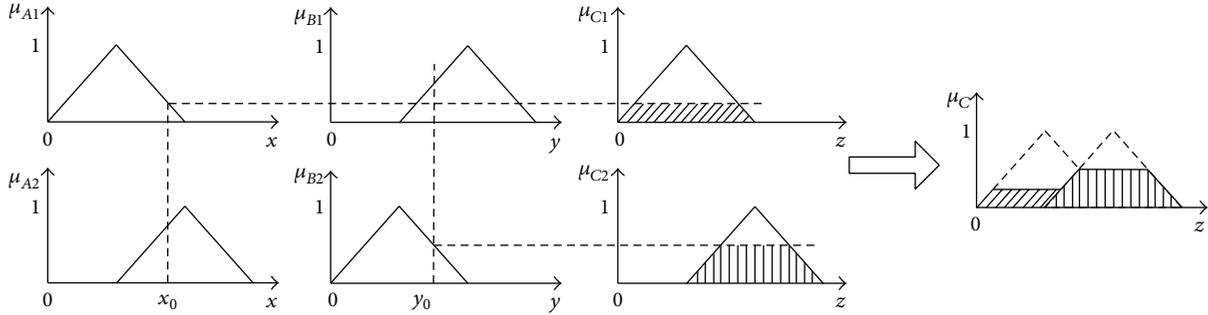


FIGURE 6: The graphic representation method of max-min inference algorithm.

Step 1. Initialize the virtual machines of data centre. Based on the prediction values of resources user demands, predetermine the amount of resources of the virtual machines, and make resource availability fuzzy classification.

Step 2. Establish the fuzzy control rule base.

Step 3. Join the task that each user submitted into FCFS scheduling queue; then the scheduler gets the corresponding task.

Step 4. According to different resource demands of the selected tasks, make fuzzy classification for those resource demands.

Step 5. According to the level of resource requirement and resource availability of current virtual machine, use fuzzy logic to infer whether the current virtual machine can meet the resource requirements of the task or not. If the needs can be met, then process the tasks using the current virtual machine. Otherwise, transfer to Step 6.

Step 6. Use fuzzy control system to schedule resource dynamically according to the resource demand level. After reaching the standard, the virtual machine processes the tasks. Then the goal that not only meets the task requirements for resources but also avoids resource waste is achieved.

Step 7. Confirm that all tasks have been completed. If there are uncompleted tasks, repeat Steps 3–6.

5. Experiment Results and Analysis

Cloud computing emulator Cloudsim [23] is adopted to simulate the scheduling algorithm. Experiments were conducted in the following cases: orderly assigned strategy (OAS), load balancing strategy (LBS), greed algorithm (GA), and fuzzy control theory based resource scheduling strategy (FCTRS). Virtual machines created in virtual data centre have attributes as follows: initial speed of execution is 200, RAM is 2048,

and the bandwidth is 1000. The properties of the 20 tasks are shown as follows: tasks length ranges from 20000 to 70000; task queue is a single FCFS queue.

From [24], we sample the data that represents the network visit traffic. We use 350 days' traffic data as the test reference. The prediction results are shown in Figure 7. In Figure 7(a), the “+” curve represents the prediction value and the “*” curve represents the actual value. The two curves fit with each other. In Figure 7(b), we can see that the overall effect is promising. The maximum error is controlled in 20% and most of the error falls in $\pm 5\%$. The user's resource demands can be effectively predicted.

To analyze the effectiveness of resource allocation of FCTRS, we compare the resource distribution status of different algorithms. Here we allocate tasks with different length to different virtual machines, as shown in Table 3. In Figure 8, the execution speed of CPU in different virtual machines is shown. From the figure, we can see that, according to the length of tasks, FCTRS effectively schedules the resources to corresponding virtual machines. When the length of tasks is longer, the execution speed of the virtual machine becomes higher. The trend is more distinct than other algorithms while LBS and GA cannot respond to the length of tasks as quickly as FCTRS. Moreover, the OAS can hardly respond to the attributes of the tasks.

Figure 9 shows the task execution time using different strategies when there are five virtual machines. When OAS is adopted, the time of task execution is the longest. This is because the tasks are allocated to the virtual machines orderly and cause the resources' shortage and being insufficiently utilized. GA improves the task execution efficiency and the execution time is shorter than the time when OAS is adopted. However, the tasks are not classified, which may cause high configuration virtual machines to run simple tasks and resources not sufficiently utilized. When LBS is adopted, each processing unit adjusts the available resources according to the task length, the total execution time, and the execution time gap between different virtual machines decreases. When FCTRS is adopted, the task execution time is the shortest. It is because when FCTRS is adopted, the availability of resources

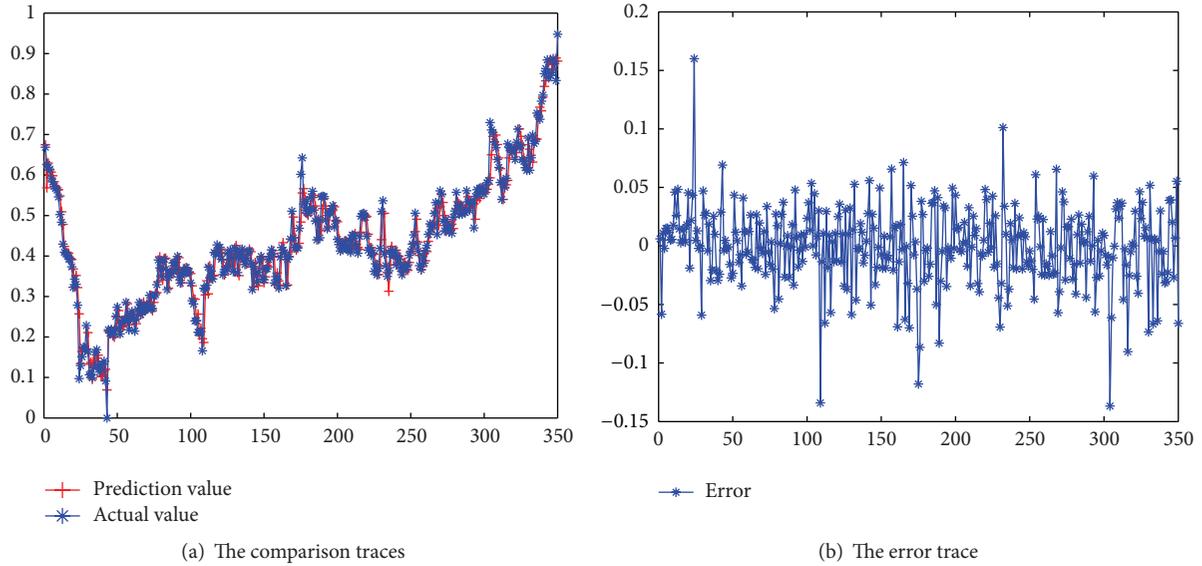


FIGURE 7: The prediction results and error.

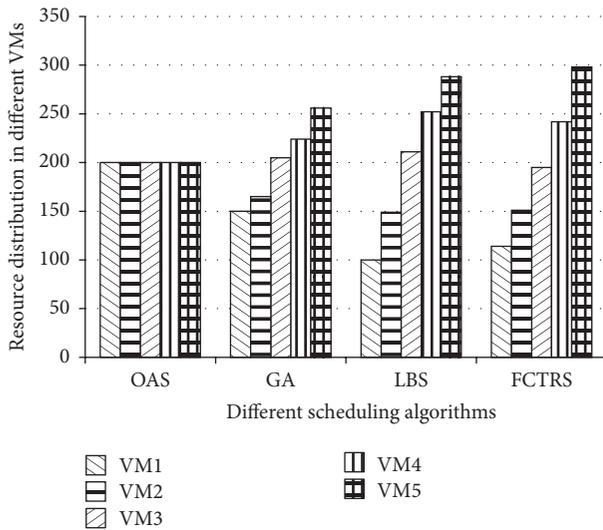


FIGURE 8: Resource distribution across the virtual machines of different algorithms.

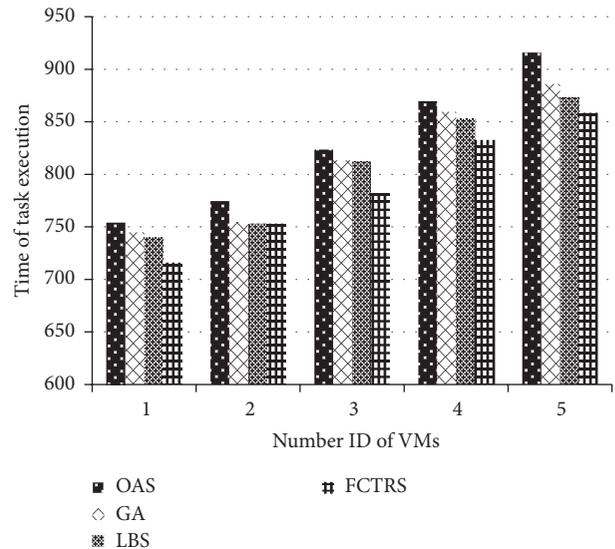


FIGURE 9: Task execution time of each processing unit adopting different virtual resource scheduling strategy.

between virtual machines is dynamically adjusted according to the level of tasks demand, the execution time is effectively shortened, and execution time gap between processing units is significantly reduced.

Figure 10 shows task execution time when tasks are allocated to corresponding virtual processing unit. When OAS is adopted, the task execution makespan is the longest. GA algorithm shortens the makespan, but the makespan is still longer than when LBS is adopted. While FCTRS is adopted, the makespan is the shortest. When LBS is adopted, the makespan of each task is more average. This is because when LBS is adopted, the load of each virtual machine is more balanced than other algorithms. However, the total execution time is still longer than FCTRS algorithm. When OAS is

adopted, the makespan of each task differentiates most. When FCTRS is adopted, the difference between the makespan of each task is moderate.

The experiment results are reasonable: (1) OAS has done little to optimize the execution efficiency, so the total execution time and the difference between each task execution time are both maximal. (2) GA allocates the current task to the highest configuration virtual machine and shortens the execution time to a certain extent. However, as the tasks' level is not classified, GA may cause resources to be insufficiently utilized and resources shortage. (3) While LBS specializes in balancing the execution time between each task, the difference of each task execution time is the lowest. (4)

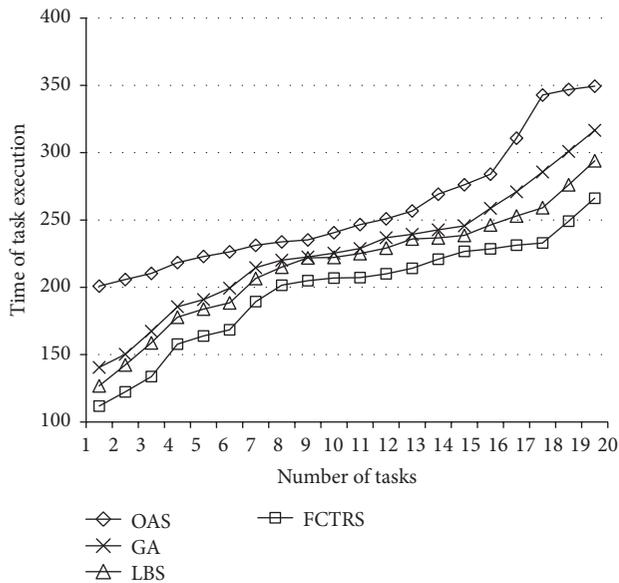


FIGURE 10: The makespan of each task using different resource scheduling strategies.

FCTRS improves the tasks execution efficiency and decreases the makespan of each task, though it is not the most suitable strategy for balancing the task load. However, FCTRS does more on ensuring the friendliness between users and the cloud computing system and improving the task execution efficiency, so each task can be processed more efficiently by dynamically scheduling resources according to the execution capability of each virtual processing unit.

Moreover, the factor of cost is introduced to measure the proposed algorithm. As shown in Figure 11, the normalization of the cost is compared when different algorithms are adopted. Make the cost index of OAS as 1, and we can see that the cost of GA is lower than OAS. As the LBS can effectively reduce the resource waste, the cost of LBS is much lower than OAS. When it comes to FCTRS, the algorithm effectively uses the information of tasks and resources, and the cost is reduced greatly.

6. Summary

This paper presents a cloud computing resource dynamic scheduling strategy based on fuzzy control algorithm. Second moving average method is adopted to predict user resource requirement. By modeling user resource requirements and resource availability, the conditions to improve the QoS of cloud computing are obtained. By introducing the fuzzy control theory into resource scheduling, dynamic and friendly resource scheduling is achieved. The experiments show that this method can effectively improve the QoS of cloud computing. It effectively improves the friendliness and user satisfaction of cloud computing.

For future work, we need to verify the method in practical cloud computing platforms and test the effect of improving the quality of service.

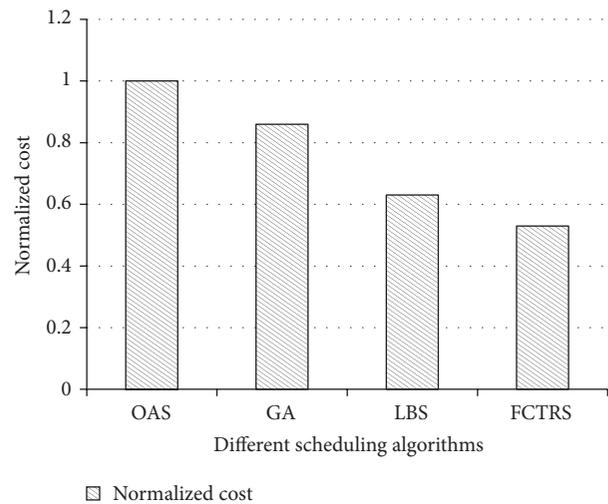


FIGURE 11: Comparison of the normalized price using different resource scheduling strategies.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The project is supported by the Equipment Pre-Research Fund (no. 9140A04030213JB34001).

References

- [1] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing—the business perspective," *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, 2011.
- [2] X.-L. Shi and K. Xu, "Utility maximization model of virtual machine scheduling in cloud environment," *Chinese Journal of Computers*, vol. 36, no. 2, pp. 252–262, 2013.
- [3] G. Zhao, H. Yu, T. Ji et al., "Adaptive resource provisioning for cloud computing," *Telecommunications Science*, no. 1, pp. 31–37, 2012.
- [4] Z. Jing, *Fuzzy Control and Systems Theory*, Mechanical Industry Press, Beijing, China, 2005.
- [5] A. Ghodsi, M. Zaharia, S. Shenker, and I. Stoica, "Choosy: max-min fair sharing for datacenter jobs with constraints," in *Proceedings of the 8th ACM European Conference on Computer Systems*, pp. 365–378, April 2013.
- [6] Y. Fang, F. Wang, and J. Ge, "A task scheduling algorithm based on load balancing in cloud computing," in *Web Information Systems and Mining: International Conference, WISM 2010, Sanya, China, October 23–24, 2010. Proceedings*, vol. 6318 of *Lecture Notes in Computer Science*, pp. 271–277, Springer, Berlin, Germany, 2010.
- [7] W. Yan, W. Jinkuan, and W. Cuirong, "Research on resource scheduling of cloud based on improved particle swarm optimization algorithm," in *Proceedings of the International Conference on Brain Inspired Cognitive Systems*, pp. 118–125, 2013.

- [8] L. Huang, H.-S. Chen, and T.-T. Hu, "Survey on resource allocation policy and job scheduling algorithms of cloud computing," *Journal of Software*, vol. 8, no. 2, pp. 480–487, 2013.
- [9] Z. Wu, X. Liu, Z. Ni, D. Yuan, and Y. Yang, "A market-oriented hierarchical scheduling strategy in cloud workflow systems," *Journal of Supercomputing*, vol. 63, no. 1, pp. 256–293, 2013.
- [10] B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," *Advances in Engineering Software*, vol. 42, no. 7, pp. 419–425, 2011.
- [11] Y. B. Ma, S. H. Jang, and J. S. Lee, "Ontology-based resource management for cloud computing," in *Intelligent Information and Database Systems: Third International Conference, ACIIDS 2011, Daegu, Korea, April 20–22, 2011, Proceedings, Part II*, vol. 6592 of *Lecture Notes in Computer Science*, pp. 343–352, Springer, Berlin, Germany, 2011.
- [12] D.-W. Sun, G.-R. Chang, F.-Y. Li, C. Wang, and X.-W. Wang, "Optimizing multi-dimensional QoS cloud resource scheduling by immune clonal with preference," *Chinese Journal of Electronics*, vol. 39, no. 8, pp. 1824–1831, 2011.
- [13] E.-K. Byun, Y.-S. Kee, J.-S. Kim, and S. Maeng, "Cost optimized provisioning of elastic resources for application workflows," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1011–1026, 2011.
- [14] P. Rattanathamrong, *Real-time scheduling of ensemble systems with limited resources [Ph.D. thesis]*, University of Florida, Gainesville, Fla, USA, 2011.
- [15] M. Luo, K. Zhang, L. Yao, and X. Zhou, "Research on resources scheduling technology based on fuzzy clustering analysis," in *Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '12)*, pp. 152–155, May 2012.
- [16] P. Rattanathamrong and J. A. B. Fortes, "Fuzzy scheduling of real-time ensemble systems," in *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS '14)*, pp. 146–153, IEEE, Bologna, Italy, July 2014.
- [17] M. Masmoudi and A. Haït, "Project scheduling under uncertainty using fuzzy modelling and solving techniques," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 135–149, 2013.
- [18] S. Frey, C. Luthje, V. Huwwa et al., "Fuzzy controlled QoS for scalable cloud computing services," in *Proceedings of the 4th International Conference on Cloud Computing, GRIDs, and Virtualization*, pp. 150–155, 2013.
- [19] F. Ramezani, J. Lu, and F. Hussain, "An online fuzzy decision support system for resource management in cloud environments," in *Proceedings of the IFSA World Congress and NAFIPS Annual Meeting*, pp. 754–759, Edmonton, Canada, June 2013.
- [20] P. Qi and L.-S. Li, "Task scheduling algorithm based on fuzzy quotient space theory in cloud environment," *Journal of Chinese Computer Systems*, vol. 34, no. 8, pp. 1793–1797, 2013.
- [21] H. Han, Q. Deyui, W. Zheng, and F. Bin, "A Qos guided task scheduling model in cloud computing environment," in *Proceedings of the 4th International Conference on Emerging Intelligent Data and Web Technologies*, pp. 72–76, September 2013.
- [22] Y. J. Huang, T. C. Kuo, and H. K. Lee, "Fuzzy-PD controller design with stability equations for electro-hydraulic servo systems," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '07)*, pp. 2407–2410, October 2007.
- [23] 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.
- [24] Data Flow Statistics and Analysis, 2014, <http://tongji.cnzz.com/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

