

Resource Self-adaptive Allocation Method Based on Mixed Prediction Cloud Platform

¹ Hong Qi, ² corresponding author Honge Ren *, ¹ Guanglei Zhang

¹ Information and Computer Engineering College, Northeast Forestry University, Harbin, 150040, China

² Information and Computer Engineering College, Northeast Forestry University, Harbin, 150040, China

¹ 15145066420, fax: 0451-82190421

¹ E-mail: qhown@sohu.com

Abstract

There are some problems in the existing cloud platform resource allocation methods, such as low rates of resource utilization and the lack of accurate prediction of the trend changes of resources, etc. To solve these problems, MPRA (Mixed Prediction Based Resource Allocation) was proposed. According to the periodic and non-periodic characteristics of service resources demand, MPRA first adopts FFT(Fast Fourier Transform) to judge the periodic characteristics. For resource allocation without periodic characteristics, it uses Markov process to predict, and obtains the higher resource utilization and prediction accuracy, thus, to ensure the user experience. The experimental results show that MPRA can accurately predict the change trend of service resource requirements, and then can allocate the virtual machine resources self-adaptively according to the prediction results. Obviously, it has improved the virtual machine resources utilization, reduced the occupation number of physical machines and effectively reduced the violation times in SLA (Service-level Agreement).

Keywords: Virtual machine; Resource allocation; Fast Fourier Transform; Markov Chain

1. Introduction

The cloud computing mode of traditional fixed resource allocation and payment on the basis of configuration has not been able to satisfy the increasingly diverse end users and personalized needs. However, the pattern of dynamic resource allocation paid on demand has caused the extensive concern of the industry[1]. At present, the coarse-grained resource allocation is a typical method adopted by cloud service providers[2-3], such as Amazon, Google, etc. This method takes the virtual machine as the scheduling unit. According to the load changes of virtual machine, it increases or decreases the number of virtual machines on demand so as to achieve the purpose of load adaptive. However, the traditional way of the allocation of resources easily causes scattered distribution of the virtual machine, too much occupation of physical machine resources and serious waste of resources[4-6]. As the booming of cloud users, especially the mobile users, this resource allocation method not only seriously hindered the raise of the resource utilization rate, but also caused the rapid increase of operating costs[7-10].

It has become a hot research direction that solving the problem of too much occupation of physical machine resources and allocating the computing resources reasonably by optimizing the placement of the virtual machine. Document[11] abstracts virtual machine placement in cloud platform to containerization and regards the CPU resources and memory resources as two dimensions of case. Then

it uses the heuristic algorithm of Choco to seek the optimal scheme of the placement of virtual machine and solve the problem of scattered placement of virtual machine through the virtual machine transfer technology. But this method only takes the two dimensions of CPU and memory into account. As the dimension increases, the complexity of algorithm becomes excessive. In terms of this limitation, Literature[12] proposed a multi-parametric heuristic algorithm. Considering CPU, network bandwidth, memory resources, and other factors, it realized the solution of multidimensional containerization in an acceptable time complexity. This method only aims at increasing resource utilization of cloud platform as the goal without considering the characteristics of the task and user preferences. Considering this limitation, Literature[13] presented a task-oriented resource allocation method. Through the analysis of the task matrix, we can figure out corresponding user preferences and resource requirements, so that resources can be allocated in accordance with the list of requirements, and utilize the inducing deviation matrix to improve the accuracy of resource allocation.

The typical algorithm usually adjusts the number of virtual machine according to the dynamic change of demand. But if new virtual machine joins, scores of minutes are needed for allocating resources. To solve this problem, the literature[14] put forward an online real time data mining system to predict the virtual machine resource request. It can pre-configure virtual machine according to the prediction results to guarantee response time for new applications demand. In order to reduce SLA, the literature[15-16] presented a dynamic virtual machine resource adjustment algorithm based on prediction respectively. The former realized prediction using amount on the virtual machine memory through the autoregressive method. As well as the latter realized the virtual machine resource adjustment through the prediction of periodic change of load, but the prediction accuracy is not high.

However, the existing research still has some defects such as the existing algorithms of resource prediction and allocation did not classify the difference of service resources, as well as their prediction accuracy is not high, and their allocation method is unreasonable. So the overall resource utilization was limited. To solve these problems, this paper introduced the adaptive allocation method which could distinguish the service resource demand characteristics. The main work was as follows: It proposed the cloud platform resource allocation mode named MPRA based on hybrid prediction. By means of the periodic characteristics of service demand judged by FFT, service demand could be treated differently based upon if they have periodic characteristics or not. The system resource utilization could be improved and the breach of SLA could be reduced ultimately.

2. MPRA : the Model of Cloud Platform Resource Allocation based on Mixed Predictions

We supposed $V = \{v_1, \dots, v_m\}$ as the current cloud platform system allocated m virtual machines. The amount of resource allocation s_i for virtual machine v_i . 1 physical machine cloud present in platform system by setting $p = \{p_1, \dots, p_l\}$. $R(v, p)$ The mapping relationship of virtual machine set V and set P physics engine were presented by $R(v, p)$.

The primary goal of the MPRA is to record service demand according to the time because any service has a corresponding resource requirement. When certain amount of sample data was obtained, we can determine for each service resource demand cycle. Thereby, periodic and non periodic service could be divided. For having a periodic characteristic resources demand, FFT should be used to get its shortest cycle; otherwise, Markov process should be used to predict.

Let $X=\{X(I) , \dots, X(L)\}$ represent the L service provided by cloud platform. The resources required interval of service $X(l)$ was $[X_{\min}^{(l)}, X_{\max}^{(l)}]$. MPRA distributed according to the resource peak demand for the service, because the lack of historical records and priori knowledge for initial access of cloud platform. Then MPRA acquired w resource samples, which are resource utilization samples $U^{(l)} = \{u_1^{(l)}, \dots, u_w^{(l)}\}$. Triggering resource demand forecasting process: did FFT transform to $U(l)$. The part of amplitude which is significantly larger than the average value represents the request cycle of resources. If a plurality of amplitude was larger than the average value, the minimum frequency was selected as the cycle frequency. That is to get the maximum period $1/f_d^{(l)}$ of the resource request. For main frequency $f_d^{(l)}$, the cycle window size was as follows:

$$Z^{(l)} = (f_d^{(l)})^{-1} \times r \tag{1}$$

r represented the sampling rate. Assuming the service resources of all types had the same sample rate, thus time series was dispersed into $Q = \lfloor w/Z^{(l)} \rfloor$ sub-time series:

$$\begin{cases} U_1^{(l)} = \{u_1^{(l)}, \dots, u_{Z^{(l)}}^{(l)}\} \\ U_2^{(l)} = \{u_{Z^{(l)}+1}^{(l)}, \dots, u_{2Z^{(l)}}^{(l)}\} \\ \dots \\ U_Q^{(l)} = \{u_{(Q-1)Z^{(l)}+1}^{(l)}, \dots, u_{QZ^{(l)}}^{(l)}\} \end{cases} \tag{2}$$

The similarity test was done on two arbitrary combination of windows based on split periodic window. If Pearson correlation coefficient of any combination of windows U_i and U_j were close to 1 and mean value of approximately was equal, it could determine that the service resource request was in accordance with the cycle $Z(l)$ having a regular change. Afterward, using DWT (Dynamic Time Warping) to obtain a periodic sequence of resource demand of the nearest service $X(l)$. The goal was to get the shortest map distance between the current resources demand sequence and it. So it could get the resource requirements $F^{(l)}(t)$ about service $X(l)$ at t time.

Nevertheless, when the service $X(l)$ does not have the characteristics of cycle time, we predicted resource requirements at t time by using Markov chain. First of all divided the resource demand state to service $X(l)$. Taking I as the interval to divide $[X_{\min}^{(l)}, X_{\max}^{(l)}]$ evenly. That is divided them into $n = \lfloor (X_{\max}^{(l)} - X_{\min}^{(l)}) / I \rfloor$ interval. The goal is to distinguish resource demand state of the same service under different conditions. Requirements for each resource interval represented a status of resource or service. Each state of the resource demand value was the mean of the interval. Therefore, service resource requirements change of $X(l)$ could be viewed as a time sequence, namely $X(l)(t)$ ($t=1, 2, \dots, w$). The resource demand of each time point belonged to different resource requirements status. Then it calculated transition probability matrix of Markov process by using statistical method.

$$P^{(l)} = [p_{ij}^{(l)} | i=1, \dots, n; j=1, \dots, n] \tag{3}$$

$p_{ij}^{(l)} = m_i^{(l)} / m_{ij}^{(l)}$ was the transition probability from state $s_i^{(l)}$ to $s_j^{(l)}$. $m_i^{(l)}$ represented the number of times the state in different period of time appear. $m_{ij}^{(l)}$ delegated times from $s_i^{(l)}$ to $s_j^{(l)}$. Therefore, we could get the service of the state transition probability matrix.

On this basis, MPRA predicted the short term resource demand though constructing a finite state Markov chain model. Assuming that the Markov chain is homogeneous, any system state probability in t time could be calculated by using C-K equation. Possible states of the service are forecasted as follows:

$$\pi_t^{(l)} = \pi_{t-1}^{(l)}(P^{(l)}) = \pi_{t-2}^{(l)}(P^{(l)})^2 = \dots = \pi_0^{(l)}(P^{(l)})^t \tag{4}$$

$\pi_0^{(l)}$ and $\pi_t^{(l)}$ take as probability distribution of $X(l)$ at initial time and t time. According to the current state $\pi_0^{(l)}$, probability of the system in state $s_j^{(l)}$ could be inferred after t time. Thus resource demand forecast value could be achieved at t time. In summary, resource demand forecast for service $X(l)$ at t time.

$$R^{(l)}(t) = \left\{ \begin{matrix} F^{(l)}(t) \\ \pi_0^{(l)}(P^{(l)})^t \end{matrix} \right. \quad (5)$$

3. Adaptive Allocation Algorithm of Resources

Resource demand change trend of virtual machine can be obtained by the above demand forecasting. Let S_i be a collection of service on a virtual machine named vi . Therefore, the resources demand forecast of virtual machine vi in t time is:

$$\bigcup_{i=1}^m S_i \subseteq \mathbf{X} \quad (6)$$

$$R(t, v_i) = \sum_{x^{(l)} \in S_i} R^{(l)}(t) \quad (7)$$

Assume virtual machine vi and its amount of resource allocation si . The current physical machine of resources can be allocated R and resource usage expectation $E(R(t, v_i))$ of virtual machine vi in t time. $D(R(t, v_i))$ is Variance, U_{add} is a offset and a linear function about variance $D(R(t, v_i))$. It need compensating according to the prediction of fluctuation when allocating resource. The greater the volatility is, the greater the value of compensation. All virtual machines of the same physical machine were arranged in a queue to adjust. It is arranged in a queue named $Q1$ to adjust. The specific resource allocation algorithm is shown in Figure 1.

In the algorithm RA, first for $Q1$ traversal, if formula (8) was established, put vi into $Q2$ queue, ascending sort to $Q2$. The total complexity of traversal to $Q1$ and sort to $Q2$ is $O(n)$. If the remaining resources of physical machine R could meet the demand of adjustment, si was adjusted to $E(R(t, v_i)) + U_{add}$. Otherwise, this virtual machine was put to the queue which was waiting named $Q3$. So the complexity of traversal $Q2$ was $O(n)$. Finally, poll to the remaining resources of physical machine. If R could meet the first element of $Q3$ in the resource requirements, the resource must be adjusted. Therefore the RA algorithm's time complexity is $O(n)$.

$$S_j < E(R(t, v_i)) + U_{add} \quad (8)$$

If the current allocation quantity of resources exceeded the prediction value, resource should be recycled timely. The specific recovery algorithm RR(Resource Recycle) was shown in figure 2. First, traversal to $Q1$. If the current allocation of resources beyond the forecast value, put virtual machine vi to queue $Q4$. Descending $Q4$ in accordance with $si - E(R(t, v_i)) + U_{add}$. Then poll to test whether or not empty. If it was empty, there is no treatment. Otherwise adjusted resource of the first element in the queue $Q4$ to $E(sX) + U_{add}$. The poll will go on after a short while. So algorithm RR's time complexity is $O(n)$.

Table1. RA : Resource Allocation Algorithm

Algorithm: RA (Resource Allocation)
Input: $E(R(t, v_i))$, $D(R(t, v_i))$, s_i , Q_1 , R
Output: Q_3
Begin
While Q_1 do
If $(s_i < E(R(t, v_i)) + U_{add})$
Move v_i to Q_2

```

Sort  $Q_2$  order by  $(E(R(t, v_i)) + U_{add} - s_i)$  asc
While  $Q_2$  do
  If  $R - (R - (E(R(t, v_i)) + U_{add} - s_i)) > 0$ 
    Adjust  $s_i$  to  $E(R(t, v_i)) + U_{add}$ 
     $R = R - (E(R(t, v_i)) + U_{add} - s_i)$ 
  Else move  $v_i$  to  $Q_3$ 
While  $(Q_3 \neq \text{null})$ 
   $X = Q_3[0]$ 
  If  $R - (E(X) + U_{add} - s_X) > 0$ 
    Adjust  $s_X$  to  $E(X) + U_{add}$ 
     $R = R - (E(X) + U_{add} - s_X)$ 
End
Return  $Q_3$ 

```

Table 2. RR : Resource Recovery Algorithm

Algorithm: RR (Resource Recycle)
Input: $E(R(t, v_i))$, $D(R(t, v_i))$, s_i , Q_1 , R
Output: Q_4
Begin
While Q_1 do
If $(s_i > E(R(t, v_i)) + U_{add})$
Move v_i to Q_4
Sort Q_4 order by $s_i - (E(R(t, v_i)) + U_{add})$ des
While $(Q_3 \neq \text{null})$
If $Q_4 \neq \text{null}$
Adjust s_i to $E(R(t, v_i)) + U_{add}$
$R = R + s_i - (E(R(t, v_i)) + U_{add})$
Sleep t_0
End
Return Q_4

4. The Experiment and Analysis

4.1 The Virtual Machine Resource Allocation Experiment

The objective of this experiment is testing the validity of MPRA resource allocation method and observing the virtual machine load change with time which took the resource of CPU and memory as allocation of resources. Then it compared with the method of the typical fixed resource allocation. The experiment used the Xen4.0 virtual machine platform, which was deployed in the 16 core CPU, 4GB memory on the server, and the virtual machine is CentOS 64-bit version 5.2.

The process of the experiment: First we opened two virtual clients called VM1 and VM2 in the virtual machine platform. Each client virtual machine is assigned a completed physical kernel and a 512M memory for simulating physical machine which fixed resources. Then we opened up VM1_sub and VM2_sub as web servers in VM1 and VM2. Non periodic application AP_1 and a periodic application AP_2 both came from data source ClusterData2011_10. We simulated client using RUBis. It sent a HTTP request to VM1_sub and VM2_sub according to the operation data of AP_1 and AP_2. Then we observed the changes of virtual machine resources. The experimental results are shown in Figure 1-4.

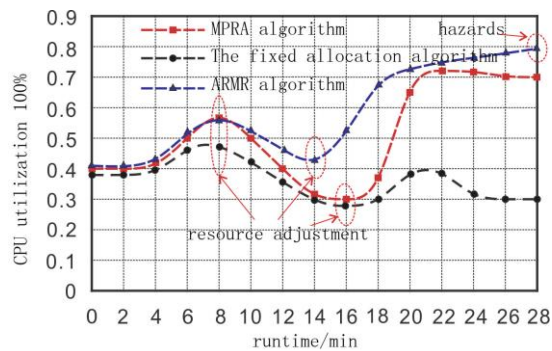


Figure 1. VM1 CPU Utilization

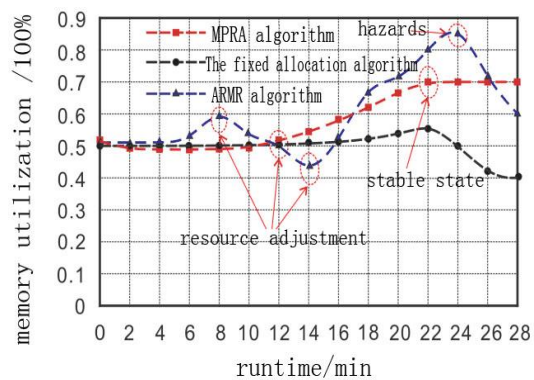


Figure 2. VM1 Memory Utilization

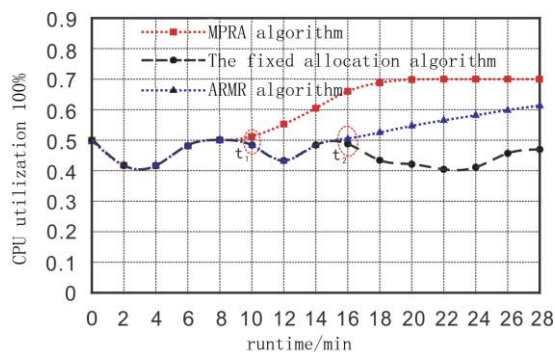


Figure 3. VM2 CPU Utilization

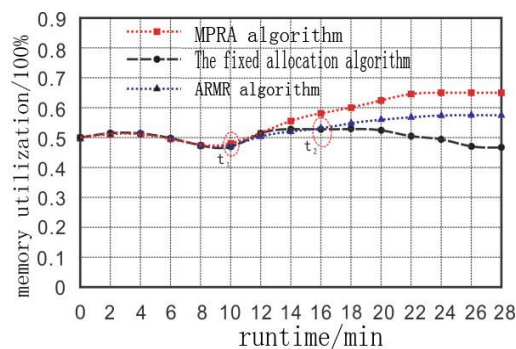


Figure 4. VM2 Memory Utilization

Figure1. and Figure2. showed the experimental result of the availability about CPU and memory(AP_1).As shown in Figure 1, the algorithm curve reflected the actual change of the service resource demand because of the CPU resources remaining unchanged during the whole experimental period in the fixed allocation algorithm. Initial CPU utilization rate was 40% and then fluctuated between 25%-50% in the fixed allocation algorithm. Thus it can be seen that the physical CPU resources allocation of the virtual machine was more adequate.

ARMR algorithm triggered the dynamic resource allocation at the 8 minutes. It reduced the number of allocation and raised the CPU availability of the virtual machine obviously. The emergence of a large fluctuation showed at the 28 minutes and it made hazards of the availability CPU of the virtual machine closing to 80%.ARMR algorithm needs to predict the service resources before the allocation of resources first. MPRA maintained a consistent change trend with the fixed allocation before not having a certain number of samples. It allocated dynamic resource at 8 minutes and 16 minutes; the ideal state showed after 22 minutes because of the utilization rate of CPU virtual machine remaining at about 70%.From here we see that the fixed allocation algorithm resulted in a great waste of physical CPU resources. While ARMA algorithm made the virtual CPU utilization close to 80%. This is due to inadequate allocation of physical CPU. It is likely to lead to the virtual machine downtime while sudden increase utilization ratio of CPU. MPRA algorithm was the most ideal for utilization ratio of CPU. It used CPU resources reasonably and avoided the CPU utilization surge problem through prediction. Figure 4 showed memory utilization of the VM1 virtual machine.

Like the CPU resource, the curve of fixed algorithm reflected the actual situation of memory utilization. It changed between 40%-55%. Therefore fixed memory allocation resource is sufficient. ARMA algorithm adjusted the memory resources at the eighth, fourteenth, twenty-sixth minutes respectively. It appeared the hazards that memory utilization rate was close to 85% at the twenty-sixth minutes because of low predictive

accuracy. The service might be breakdown. MPRA algorithm had accumulated enough prediction samples after 12 minutes. It kept the memory utilization maintaining an ideal state in the vicinity of 70%.

Figure 3 and figure 4 showed CPU and memory utilization of VM2 When AP_2 as the service request flow. In the fixed allocation method, two kinds of resource allocation are more adequate. The corresponding utilization rate is respectively between 40%-50% and 45%-55%. ARMA and MPRA were able to get the ideal utilization rate because AP_2 had the characteristics of periodicity. MPRA began to adjust the resource allocation in the T1 time and ARMA began to adjust at a later time T2. It led the overall resource utilization rate lower than MPRA.

In order to compare the prediction accuracy, the following experiment terminated the resource allocation process and collected forecast data only. The data of the previous 8 minutes were used as the prediction sample in the experiment. We observed the predicted accuracy in AP_1 and AP_2 respectively. The experimental results were showed in Figure 5 and Figure 6. Influenced by the system operation state, the data which were got before 8 minutes were not entirely consistent with Figure 1-4. It can be seen that MPRA was higher than ARMA on the prediction accuracy, no matter what it was periodic or non periodic service. But both the periodic services were higher than the non periodic forecast on prediction accuracy.

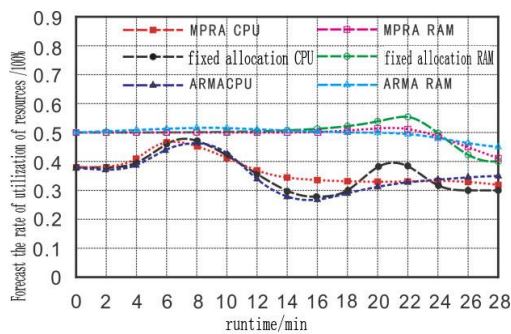


Figure.5 Forecast of Utilization of Resources on AP_1

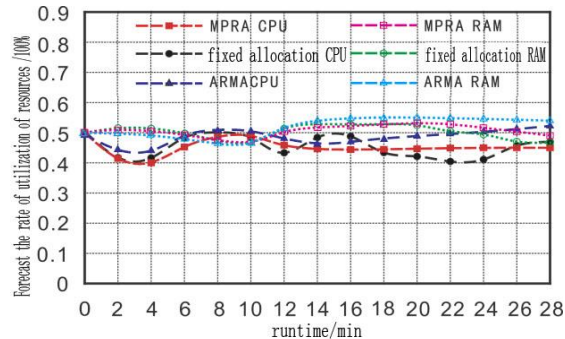


Figure.6 Forecast of Utilization of Resources on AP_1

4.2 MPRA Simulation Analysis based on CloudSim

In order to test the amount of physical machine, Virtual machine average load, The number of the violation of SLA act. MPRA method was deployed in the CloudSim simulation platform in the experiment. It was compared with FRA which is a virtual machine resource allocation model of current mainstream. ClusterData2011_1 as data was used in the experiment. It simulated the cloud platform basing on the FRA and MPRA method respectively. The average values of a monthly load condition of 25 virtual machines were recorded. With CPU as the main criterion, CPU utilization in 0%-50% is low load; CPU utilization in 50%-80% is reasonable load; CPU utilization more than 80% is high load; CPU utilization more than 90% is the breach of SLA threshold.

As shown in Figure 8, the virtual machine was assigned to 11 physical machines. With the migration of the virtual machine, the numbers of physical machines were increased under FRA method. The virtual machine which dispersed into the physical machine could lead to physical machine resources occupancy increased and the rate of utilization of resources of cloud platform reduced. The number of physical machine was not monotonic increased. At the beginning, physical machine resources' occupation was the same with the initialization. As time goes on, physical machine resources' occupation was less than the initialization.

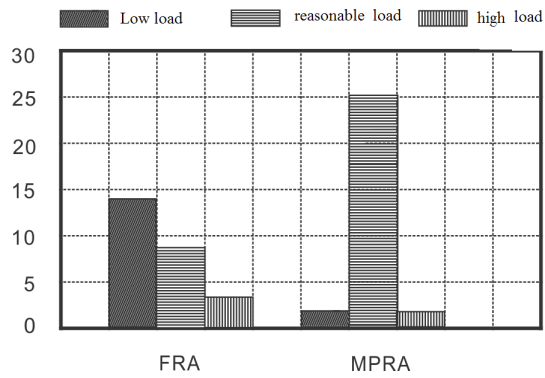


Figure7. The Average Load of Virtual Machine

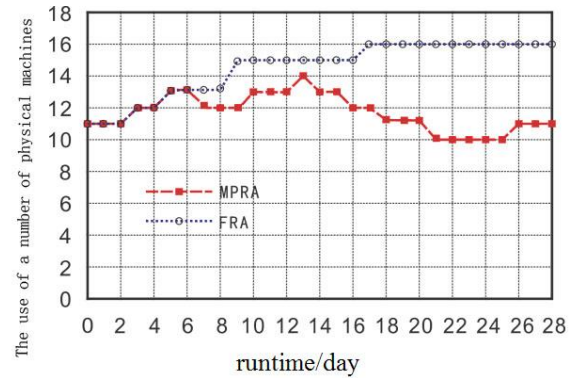


Figure8. Physical Machine Usage

Table 3 showed the statistics operation data from 0 to 27 days. Violation of SLA phenomenon didn't happen during the whole runtime by using MPRA method. While it happened 4, 3, 3, 2 times every week respectively by using FRA method. It is thus clear that SLA phenomenon can be avoided effectively by using MPRA method.

Table 3. Times Contrast of Violation of SLA

Violation of SLA	0-6	7-13	14-20	21-27
MPRA	0	0	0	0
FRA	4	3	3	2

5. Conclusion

This article put forward Mixed Prediction Based Resource Allocation (MPRA) method according to the characteristics of periodic and non periodic service resource demand. It solved the problem that the existing resource allocation method. For any kind of service resources, MPRA first used FFT to judge its cycle characteristics. It used Markov for the prediction of the process according to the resource allocation which had not periodic characteristics to obtain a higher resource utilization rate and the rate of accuracy of prediction. Therefore, the number of violations of the SLA was reduced effectively. Further research will be done to subdivide the needs of non periodic service resources in order to achieve higher prediction accuracy of forecast method.

Acknowledgement

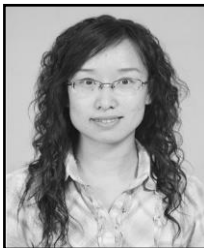
The work supported by “the Fundamental Research Funds for the Central Universities”, DL12BB09; “Natural Science Foundation of Heilongjiang Province of china”, ZD201203; The National Natural Science Foundation of China,31370566

References

- [1] M. Armbrust, A. Fox, R. Griffith, "A view of cloud computing", Commun. ACM, vol. 53, no. 4, pp. 50-58, (2010).
- [2] R. Buyya, C. S. Yeo, S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities", Proceedings of High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on, (2008).
- [3] R. Buyya, C. S. Yeo, S. Venugopal, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, vol. 925, no. 6, (2009), pp. 599-616.
- [4] B. Grobauer, T. Walloschek, E. Stocker, "Understanding Cloud Computing Vulnerabilities", Security &

- Privacy, IEEE, vol. 9, no. 2, (2011), pp. 50-57.
- [5] D. Quiang, Y. Yuhong, A. V. Vasilakos, "A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing", Network and Service Management, IEEE Transactions on, vol. 9, no. 4, (2012), pp. 373-392.
- [6] M. N. O. Sadiku, S. M. Musa, O. D. Momoh, "Cloud Computing: Opportunities and Challenges", Potentials, IEEE, vol. 33, no. 1, (2014), pp. 34-36.
- [7] L. Jenn-Wei, C. Chien-Hung, J. M. Chang, "QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing Systems", Cloud Computing, IEEE Transactions, vol.1, no. 1, pp. 101-115, (2013).
- [8] S. K. Mong, "Agent-Based Cloud Computing", Services Computing, IEEE Transactions, vol. 5, no. 4, (2012), pp. 564-577.
- [9] N. Roy, A. Dubey, A. Gokhale, "Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting", Proceedings of Cloud Computing (CLOUD), 2011 IEEE International Conference, (2011).
- [10] S. M. Parikh, "A survey on cloud computing resource allocation techniques", Proceedings of Engineering (NUICONE), 2013 Nirma University International Conference, (2003).
- [11] F. Hermenier, X. Lorca, J.-M. Menaud, "Entropy: a consolidation manager for clusters", Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, (2009); ACM: Washington, DC, USA.
- [12] V. C. Eneakaroha, I. Brandic, M. Maurer, "SLA-Aware Application Deployment and Resource Allocation in Clouds", Proceedings of Computer Software and Applications Conference Workshops (COMPSACW), (2011); pp. 298-303.
- [13] D. Ergu, G. Kou, Y. Peng, "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.
- [14] A.. Khan, X. Yan, T.Shu, "Workload characterization and prediction in the cloud: A multiple time series approach", Proceedings of Network Operations and Management Symposium (NOMS), (2012).
- [15] W. Zhao, Z. Wang, Y. Luo, "Dynamic memory balancing for virtual machines", SIGOPS Oper. Syst. Rev., vol. 43, no. 3, (2009), pp. 37-47.
- [16] D. Manasche, M. N. Bannani, "Autonomic Virtualized Environments", Proceedings of Autonomic and Autonomous Systems, 2006. ICAS '06. 2006 International Conference, (2006).

Authors



Hong Qi, 13. 10. 1978, China, she is a Doctoral student in reading at Northeast Forestry University. Currently, she is a teacher at Northeast Forestry University, China. Her research interests include image processing and pattern recognition.



Hong-e Ren, 23. 11. 1962, China, she received the Ph.D. degree from Northeast Forestry University, China in 2009. Currently, she is a professor and doctoral supervisor at Northeast Forestry University, China. Her research interests include pattern recognition and intelligent control, modern information technology and information security.



Guanglei Zhang, 28.11. 1978, China, he is a Doctoral student in reading at Harbin Engineering University. Currently, he is a teacher at Northeast Forestry University, China. His research interests include system engineering and intelligent control.

