International Journal of Hybrid Information Technology
Vol.8, No.2 (2015), pp.159-172
http://dx.doi.org/10.14257/ijhit.2015.8.2.14

Research on CPU Workload Prediction and Balancing in Cloud Environment

Zhulin Li[1, 2*], Cuirong Wang[3], Haiyan Lv[4] and Tongyu Xu[5]

[1]*College of information science and engineering, Northeastern University, Shenyang, Liaoning, China*
[2] *Modern educational technology center, Shenyang Agricultural University, Shenyang, Liaoning, China*
[3]*College of computer and communication engineering, Northeastern University at Qinhuangdao, Qinhuangdao, Hebei, China*
[4] *College of forestry, Shenyang Agricultural University, Shenyang, Liaoning, China*
[5]*College of information and electrical engineering, Shenyang Agricultural University, Shenyang, Liaoning, China*
Corresponding author's email: zhulin@syau.edu.cn

Abstract

Servers workload in the cloud environment should be balanced in order to achieve high efficiency and reduce resources consuming. One of the solutions is based on workload prediction, and design a proper load migration and balancing strategy. For the ease of discussion, we focus on CPU workload only in this paper. Specifically, considering the characteristics of workload, such as the strong correlation with time, we employ a time-series based two-step method to predict the CPU workload for both individual physical server and the cluster. Then, with the knowledge of the cluster workload, we design a strategy for workload migration and load balancing. Besides, we conduct extensive experiments to evaluate our method.

Keywords: *Load Balancing, CPU Workload Prediction, Time-series Analysis*

1. Introduction

With the increasing development of cloud computing [1-2] and virtualization techniques [3], computing and storage resources can be provisioned in an elastic and scalable manner, so that information technology resources can be consolidated for management, scheduling and maintenance. Typical cloud environment is built upon virtualization technique, with which different applications can run within independent spaces with the objective of satisfying the varied requirements of end users and meanwhile improving the utilization of limited resources.

ISSN: 1738-9968 IJHIT

However, due to the uncertainty of applications and the capability differences between cloud servers (or nodes), the workload of nodes in the virtualized environment is usually unbalanced. Therefore, it remains a significant issue that how to maintain the load balancing among virtual machines on nodes.

There are two main challenges along this line. First, we should be able to identify the workload of each virtual machine and each node, and detect when the workload should be adjusted. Second, if the workload needs to be adjusted, where does the extra workload go, and what if the capacity of virtual machine is mostly unused?

Some existing load balancing methods are static algorithms. However, without consideration of current workload but only history situation, static methods are hard to achieve real balance. Therefore, in this paper, we employ a dynamic method for load balancing. Although the complexity of algorithm is relatively higher, and extra overhead has to be paid for collecting workload information, dynamic methods can coordinate the capacity of servers and improve the throughout of the system provided that current status of each server in the cluster is gathered.

For the ease of discussion, we focus on CPU workload only in this paper, which can be easily extended to memory, disk and network workloads. Specifically, in this paper, we have two contributions with regards above two challenges. First, considering the characteristics of workload, such as the strong correlation with time, we employ a time-series based two-step method to predict the CPU workload for both individual physical server and the cluster. We provide two improvements to increase the precision of prediction and avoid the noises: (1) before performing prediction algorithm on the workload sequence, we introduce wavelet packet decomposition (WPD) [4] to divide the original sequence into more stable sub-sequences; (2) based on the set of sub-sequences, a revised ARIMA (autoregressive integrated moving average) [5] model is applied. Accordingly, the prediction of workload is combined through all the sub-sequences.

Second, with the knowledge of the cluster workload, we design a strategy for workload migration and load balancing. Specifically, we use double thresholds for workload migration for overload and idle load respectively, in order to balance the overall load as well as reduce the energy consumption.

The remains of this paper are organized as follows. Section 2 discusses the related work, and Section 3 describes the problem statement. In Section 4, we propose the method for CPU workload prediction, and in Section 5 we present load balancing strategy. Then, we conduct some experiments in Section 6. Finally, the paper is concluded in Section 7.

2. Related Work

The workload research in cloud computing environment is mainly focused on using load balancing techniques to adjust the workload assignment of each node and thus balance the

capacity, in order to achieve the maximum utilization of resources and provide best user response [6-7].

Basically, there are two types of load balancing methods: static and dynamic methods. Static balancing is based on the current execution and hardware information to select the best node for task assignment [8-9]. Dynamic method is typically based on historical situations and current status to make the decision [10-12].

There are also some efforts on workload prediction and load management. For example, Wu *et al.* [13] designed an adaptive hybrid method to solve the performance prediction in grid computing environment. Gmach *et al.* [14] proposed a resource management strategy based on analysis of the workload pattern and demand prediction. Ganapathi *et al.* [15] used a statistical method to predict the resource requirement for job scheduling. Khan *et al.* [16] introduced a workload prediction algorithm based on hidden Markov model. Xu *et al.* [17] formulated the virtual machine assignment in cloud environment problem as a multi-objective optimization problem, and employed a GA based algorithm as the solution. Wang *et al.* [18] considered the bandwidth restriction on workload prediction, and Beloglazov *et al.* [19] focused on the green computing perspective. However, there lacks of combining the workload prediction and load balancing together. In this work, we propose an initial attempt along this line.

3. Problem Statement

As indicated by [20], the workload of servers has the following characteristics: (1) the process of host load changing over time is a stochastic process; (2) host load is strongly correlated with time, which means the the past load has a great impact on the future load; (3) the value of load fluctuate or remain stable during a certain time period; (4) Based on above observations, it is feasible to employ a time-series based method to estimate the workload of servers at specific time.

Suppose we have M physical machines (or physical nodes), and there are M_i CPUs on each physical machine i, and the j-th core on i-th physical node is denoted as (i, j). Suppose there are N virtual machines (or virtual nodes) running on physical machines, and the number of virtual nodes on i-th physical machine is N_i.

In this paper, we leverage workload detection to reduce the impact of load fluctuate on the clusters, and also precisely reflect the trend of load changes. In order to increase the accuracy, we consider the periodic characteristics of workload dynamics.

We formulate the workload changing as a time-series sequence. Let T be the changing cycle of workload. With each monitoring cycle, the workload of each core on each physical

node is collected as $t_1^c(i,j), t_2^c(i,j), ..., t_n^c(i,j)$, where n is the number of observation point, and $t_k^c(i,j); i = 1,2,...,M; j = 1,2,...,M_i$ denotes the workload on the j-th core on i-th physical node within cycle c.

Therefore, the average workload of the cluster at observation point k within cycle c can be calculated as:

$$avg_k^c = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M_i} t_k^c(i,j)}{N} . \quad (1)$$

4. CPU Workload Prediction

In this section, we discuss the algorithm of predicting CPU workload. In order to improve the precision of prediction, we first employ wavelet decomposition on the observed sequence, and then apply a revised ARIMA model on the sequences to increase the precision of prediction.

4.1. Wavelet Decomposition

The idea is that by decomposing the sequence into several sub-sequences, we get more stable sequences, and the prediction upon that would be more accurate.

A wavelet packet includes a scaling function and a mother wavelet [21]. By wavelet packet decomposition (WPD), we get a subspace with a set of scaling functions and mother wavelets. Based on multiple scaling factors j, the Hilbert space $L^2(R)$ can be decomposed into a set of orthogonal subspace, *i.e.*:

$$L^2(R) = \bigoplus_{j \in Z} W_j , \quad (2)$$

where W_j is the wavelet subspace of wavelet function $\psi(t)$.

Unify the scaling subspace V_j and wavelet subspace W_j as U_j. Suppose the level of decomposition is l. The structure of WPD is shown in Figure 1, where U_0^0 is the original space, and U_j^n is the subspace of j-th decomposition. Thus, the subspaces are:

$$U_j^n = U_{j+1}^{2n} \oplus U_{j+1}^{2n+1} , \quad (3)$$

where $j \in Z, n \in Z^+$.

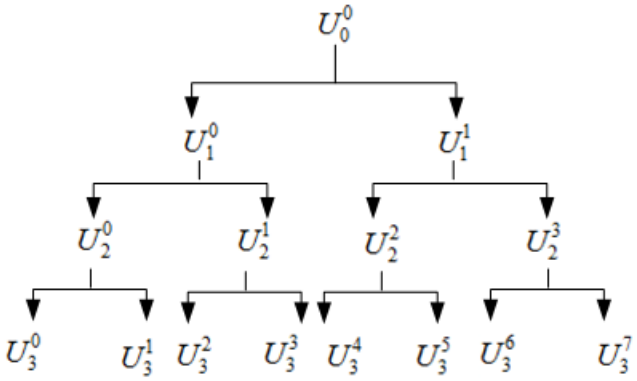

Figure1. Illustration of Tree Structure of WPD

Let U_j^n is the closure space of function $u_n(t)$, and U_j^{2n} is the closure space of function $u_{2n}(t)$,

$$\begin{cases} u_{2n}(t) = \sqrt{2}\sum_{k\in Z} h(k)u_n(2t-k) \\ u_{2n+1}(t) = \sqrt{2}\sum_{k\in Z} g(k)u_n(2t-k) \end{cases}, \tag{4}$$

where $g(k) = (-1)^k h(1-k)$.

The sequence constructed by Equation (3) $\{u_n(t)\}(n \in Z^+)$ is called the orthogonal wavelet packet of base function $u_0(t) = \phi(t)$, which is also the scaling function, and $u_1(t) = \psi(t)$, which is also the wavelet base function.

Suppose $g_j^n(t) \in U_j^n$, which can be represented as:

$$g_j^n(t) = \sum_l d_l^{j,n} u_n(2^j t - l), \tag{5}$$

And

$$g_{j+1}^n(t) = g_j^{2n}(t) \oplus g_j^{2n+1}(t). \tag{6}$$

Therefore, we have the wavelet packet decomposition process:

$$\begin{cases} d_l^{j+1,2n} = \sum_k h_0 d_l^{j,n}(2k-l) \\ d_l^{j+1,2n+1} = \sum_k h_1 d_l^{j,n}(2k-l) \end{cases} . \quad (7)$$

After that, the WPD produces 2^l different sets of subspaces. The wavelet packet reconstruction process is:

$$d_l^{j,n} = \sum_k h_0(l-2k) d_k^{j+1,2n} + h_1(l-2k) d_k^{j+1,2n+1} . \quad (8)$$

4.2. ARIMA based Prediction

Now we have 2^l sequences. In this section, we perform regression analysis on each sequence, and then combine the results together to get the final prediction value. The overall two-step prediction process can be illustrated in Figure 2.

In this paper, we employ ARIMA model for our time-series prediction problem, which combines the advantages of both time-series and regression analysis. However, typical ARIMA model is linear and lack of precision. To this end, in this section, we propose to revise ARIMA with SVM (support vector machine) [22]. Basically, the idea is to perform ARIMA model, and then apply AVM on the residuals of ARIMA prediction results.

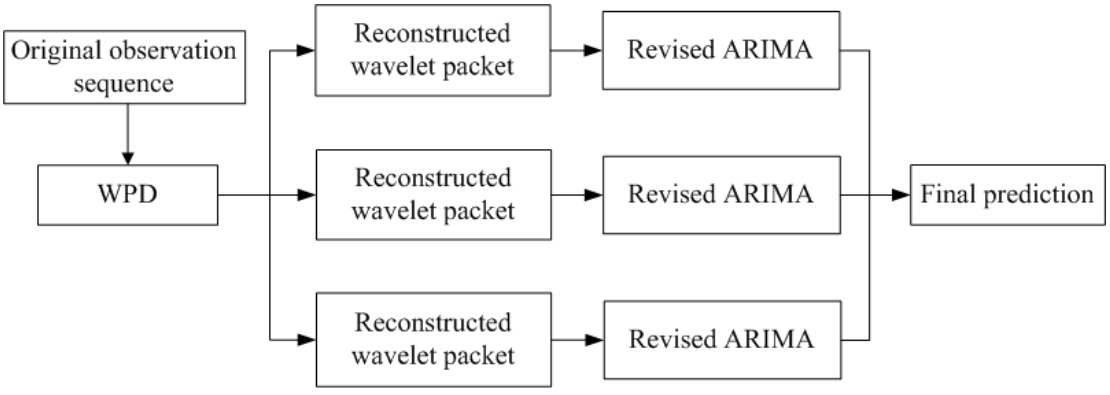

Figure 2. Two-step Prediction

First, apply ARIMA model on the workload sequence obtained from Section 4.1. Suppose a new sequence $\{x_t\}$ is obtained after d differentials, and

$$x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + ... + \varphi_p x_{t-p} + u_t - \theta_1 u_{t-1} - \theta_2 u_{t-2} - ... - \theta_q u_{t-q} , \quad (9)$$

where $\varphi_1, \varphi_2, ..., \varphi_p$ are autoregressive coefficients, $\theta_1, \theta_2, ..., \theta_p$ are average moving coefficients, $\{u_t\}$ is the white noise sequence, and $\{x_t\}$ is the autoregressive integrated moving average sequence, notated as $ARIMA(p, d, q)$.

Suppose the estimated value using ARIMA is $\hat{x}_t$, and the observation is x_t. The residual of ARIMA prediction result are calculated as:

$$r_t = (\hat{x}_t - x_t)^2 . \tag{10}$$

Second, apply SVM model on the residuals $\{r_t\}$. We use RBF kernel function for SVM. Suppose the kernel function is $K(x_i, x_j)$, and the nonlinear fitting function is:

$$f(x) = w\varphi(x) + b = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) K(x_i, x_j) + b . \tag{11}$$

where α, α^* are Lagrange factors.

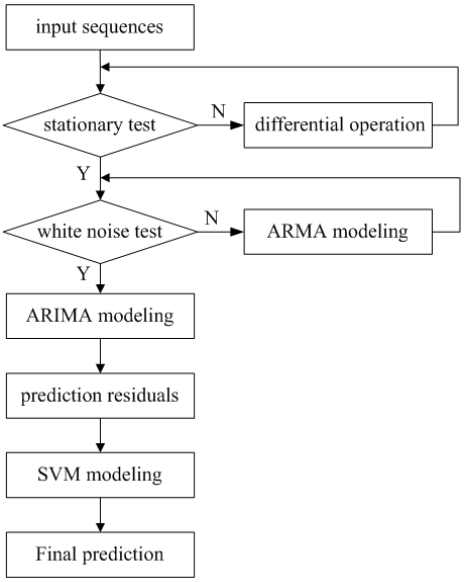

Figure3. Revised ARIMA with SVM

Modify the ARIMA prediction based on SVM fitting results. The procedure of revised ARIMA is shown in Figure 3.

5. Load Balancing Strategy

Now that we know about the workload situation of the nodes in the cluster, we are able to further conduct load balancing operations.

We define the triggering rules of workload migration as follows:

(1) When the cluster workload exceeds the upper threshold, new virtual machines need to be added to the cluster to share the extra workload, and the extra workload should be migrated to the newly added nodes.

(2) When the cluster workload is lower than the lower threshold, it means that some virtual machines are not fully utilized, and the cluster can be reduced for the sake of green computing.

(3) When the individual workload exceeds the upper threshold, it means that node is overloaded, and extra workload should be migrated to other nodes.

In this way, the size of cluster can be increased or reduced due to the workload changes, and the capacity of each node is fully explored.

The overall framework can be illustrated as Figure 4. Specifically, the task management module is responsible for receiving tasks from users. The workload prediction module estimates the workload of the cluster, as described in Section 4. Then, based on the prediction results, the controller module controls the virtual machines in the cluster using above rules.

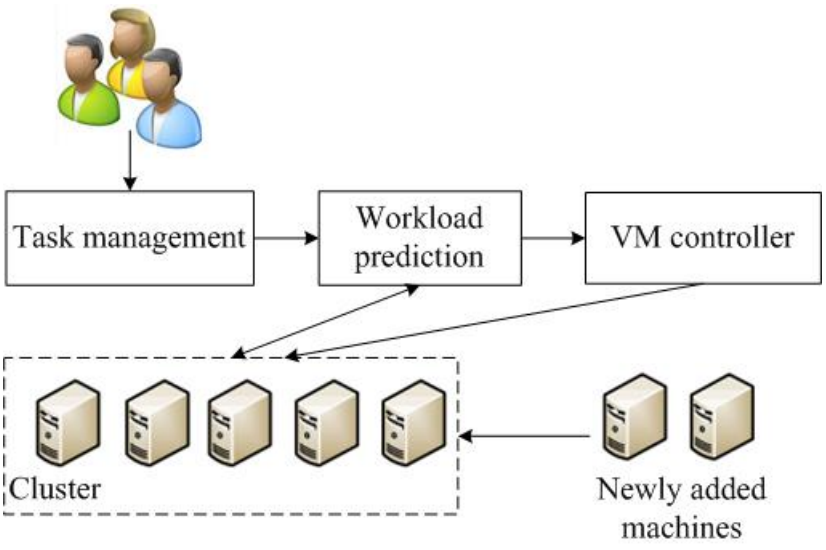

Figure 4. Illustration of Workload Prediction and Load Balancing

6. Experiment

In this section, we conduct some experiments to evaluate our method. We use CloudSim [23] 3.0 to simulate the cluster. The total number of nodes available in the cluster is 30. The operation system is Linux, and the CPU workload can be obtained by vmstat command. Figure 5 gives one example of the observed CPU workload.

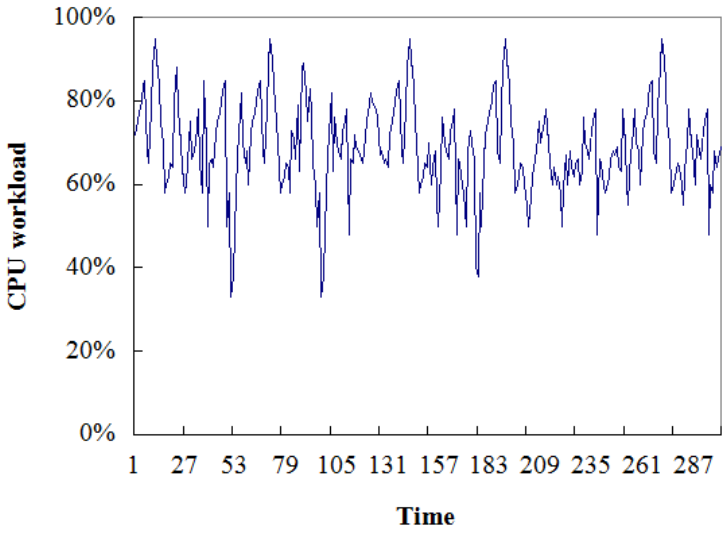

Figure 5. Observed Workload Sequence of One Node

We compare our prediction model with AR, ARMA and ARIMA, and the results are shown in Figure 6. We have the following observations. (1) If the workload is generally stable, all methods can predict the workload correctly enough, and ARIMA is even slightly better than the proposed method. (2) If the workload changes suddenly, our method can capture the burst more precisely; that is, the prediction curve using our proposed is the most fit to the observed sequence.

Besides, we give the mean squared error (MSE) values for each prediction algorithm in Table 1, which is calculated as:

$$MSE = \sqrt{\sum_{t=1}^{n} (\hat{x}_t - x_t)^2 / n} \quad . \tag{12}$$

Each value is the average of 10 times running in the cluster. We can see that the MSE of our method is obviously smaller than others, although the prediction time cost is slightly higher. Therefore, we conclude that our two-step prediction method can estimate the CPU workload effectively, which provides evidence for load balancing.

Table1. Comparison of Different Methods

Method	Running times	Average prediction time (ms)	Average workload (%)	MSE
AR	10	1.796	0.58	0.021
ARMA	10	1.821	0.61	0.013
ARIMA	10	1.980	0.63	0.009
Proposed	10	2.014	0.75	0.004

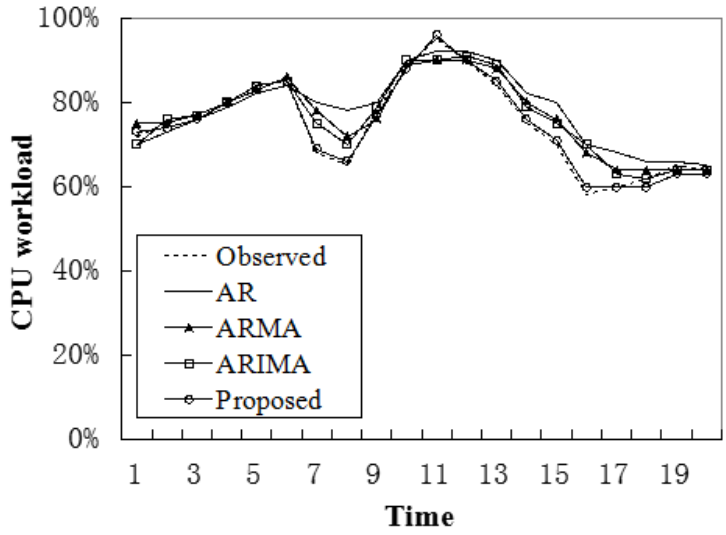

Figure 6. CPU Workload Prediction Results

Besides, we compare our load balancing strategy with ACT (availability check technique) [24], which is a resource co-allocation protocol and tries to reduce the conflicts that happen between co-allocators when they try to allocate multiple resources simultaneously.

We use CPU utilization rate to measure the efficiency of each node. The larger the utilization rate is, the more efficient the node is, and therefore, the more balance the system is. Figure 7 shows the utilization rate of two methods. We can observe that compared to ACT, our proposed method can achieve higher utilization rate.

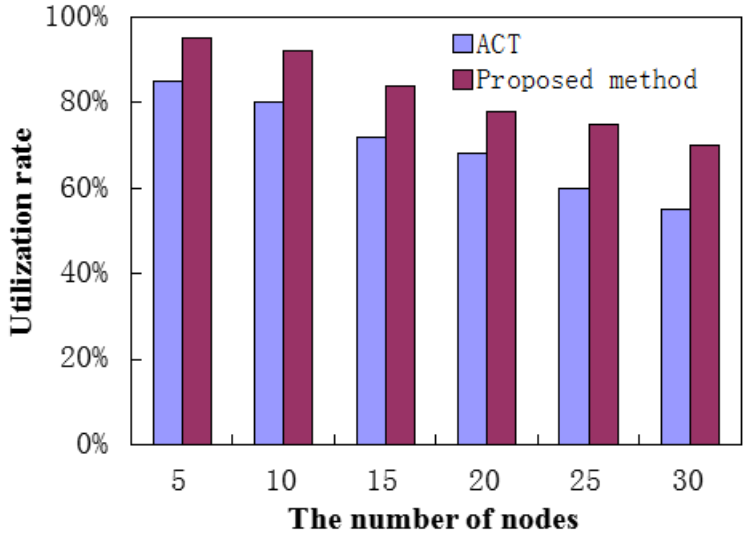

Figure 7. Utilization Rate

As mentioned earlier, the size of our cluster can be increased or reduced due to the computing demands. Figure 8 shows the size of cluster over time in our experiment, which shows that our load balancing method can make the best use of each machine and release unused resources for green computing.

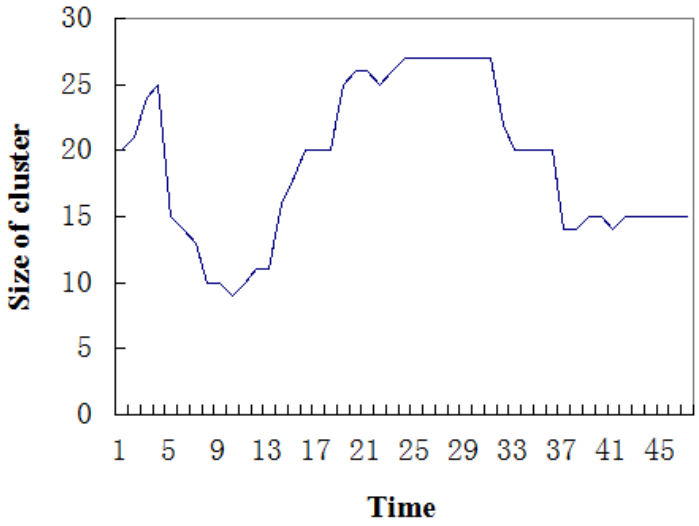

Figure 8. The Size of Cluster over Time

7. Conclusion

In this paper, we discussed the problem of predicting workload in the cloud cluster, and proposed to employ a time-series method revised by SVM model to increase the prediction precision. Based on the estimated workload, we define some rules for load migration and load balancing. However, we simplify the workload as CPU workload only, without consideration of network bandwidth between nodes. In future, we would like to further explore other elements in cluster workload estimation.

Acknowledgments

Liaoning information platform of rural science and technology service and the University of Agricultural Technology Extension demonstration of research. Science and technology projects of Liaoning Province.2013301004-7.

References

[1] M. Armbrust, "A view of cloud computing", Communications of the ACM, vol. 53, no. 4, (**2010**), pp. 50-58.

[2] [2] Mell, Peter, and Timothy Grance. "The NIST definition of cloud computing (draft)." NIST special publication, vol. 800, no. 145, (**2011**), p. 7.

[3] Y. Xing and Y. Zhan, "Virtualization and cloud computing", Future Wireless Networks and Information Systems, Springer Berlin Heidelberg, (**2012**), pp. 305-312.

[4] E. Sami, S. Yildirim and M. Poyraz, "Energy and entropy-based feature extraction for locating fault on transmission lines by using neural network and wavelet packet decomposition", Expert Systems with Applications, vol. 34, no. 4, (**2008**), pp. 2937-2944.

[5] J. Saboia and L. Maurity, "Autoregressive integrated moving average (ARIMA) models for birth forecasting", Journal of the American Statistical Association, vol. 72, no. 358, (**1977**), pp. 264-270.

[6] X. Yi and C. Weng, "Load balance approach to save power on cloud datacenter", Jisuanji Kexue yu Tansuo, vol. 6, no. 4, (**2012**), pp. 327-332.

[7] G. Ping and L. Qi, "Load balancing scheduling algorithm based on classifying the server by their load", Journal of Huazhong University of Science and Technology: Nature Science Edition, vol. 40, (S1), (**2012**), pp. 62-65. (in Chinese).

[8] T. Schroeder, S. Goddard and B. Ramamurthy, "Scalable web server clustering technologies", Network, IEEE, vol. 14, no. 3, (**2000**), pp. 38-45.

[9] Y. Zhao and W. Huang, "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud", INC, IMS and IDC, 2009. NCM'09, Fifth International Joint Conference on. IEEE, (**2009**).

[10] Huu, T. Truong and C.-K. Tham, "An Auction-Based Resource Allocation Model for Green Cloud Computing", Cloud Engineering (IC2E), 2013 IEEE International Conference on. IEEE, (**2013**).

[11] Z. Zhang and X. Zhang, "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation", Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on. vol. 2, IEEE, (**2010**).

[12] S.-C. Wang, "Towards a load balancing in a three-level cloud computing network", Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. vol. 1, IEEE, (**2010**).

[13] Y. Wu, "Adaptive workload prediction of grid performance in confidence windows", Parallel and Distributed Systems, IEEE Transactions on vol. 21, no. 7, (**2010**), pp. 925-938.

[14] D. Gmach, "Workload analysis and demand prediction of enterprise data center applications", Workload Characterization, 2007. IISWC 2007. IEEE 10th International Symposium on. IEEE, (**2007**).

[15] A. Ganapathi, "Statistics-driven workload modeling for the cloud", Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on. IEEE, (**2010**).

[16] A. Khan, "Workload characterization and prediction in the cloud: A multiple time series approach", Network Operations and Management Symposium (NOMS), 2012 IEEE. IEEE, (**2012**).

[17] J. Xu and J. AB Fortes, "Multi-objective virtual machine placement in virtualized data center environments", Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom). IEEE, (**2010**).

[18] M. Wang, X. Meng and L. Zhang, "Consolidating virtual machines with dynamic bandwidth demand in data centers", INFOCOM, 2011 Proceedings IEEE. IEEE, (**2011**).

[19] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers", Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science. ACM, (**2010**).

[20] P. A. Dinda, "The statistical properties of host load", Scientific Programming, vol. 7, no. 3, (**1999**), pp. 211-229.

[21] Z. Sun and C. C. Chang, "Structural damage assessment based on wavelet packet transform", Journal of structural engineering, vol. 128, no. 10, (**2002**), pp. 1354-1361.

[22] T. Van Gestel, "Least squares support vector machines", Singapore: World Scientific, vol. 4, (**2002**).

[23] R. N. Calheiros, "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, (**2011**), pp. 23-50.

[24] D. Azougagh, J.-L. Yu and S.-R. Maeng, "Resource co-allocation: A complementary technique that enhances performance in grid computing environment", Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on. vol. 1, IEEE, (**2005**).

Authors

Zhulin Li, He was born in 1976. He received the M.S. degree from college of information science and engineering, Northeastern University, Shenyang, China. Now he is a Senior Experimentalist, and his research interests include task scheduling of CPU, secure programming.

Cuirong Wang, She was born in 1963. She received the Ph.D. degree from the school of information science and engineering, Northeastern University, Shenyang, China in July 2004. Now, she is a professor and a research manager of the wireless sensor network and next generation network technology group. Her research interests include Routing Protocol, Network Security and Wireless Sensor Networks.

Haiyan Lv, She was born in 1979. She received Ph.D. degree from Chinese Academy of Agricultural Sciences. Now, she is a lecturer, and her main research fields are Remote Sensing and Geographical Information System.

Tongyu Xu, He was born in 1967. He received the Ph.D. degree from the college of information and electrical engineering, Shenyang Agricultural University, Shenyang, China in June 2006. Now he is a professor and a doctoral supervisor. His research interests are Agricultural Internet of things.