ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Future Load Prediction of Virtual Machines in Cloud Computing Environment

Debabrata Saddar[1], Prodip Ghosh[2]

[1]Asst. Proff. Department of CSE, University of Kalyani
[2] Asst. Proff. Department of CSE , Narula Institute of Technology, Agarpara

Abstract: In cloud computing environment, customers are allowed to scale up and down their resource usage according to their needs. Here resources are multiplexed from physical machines to virtual machines through virtualization technology. In this paper, we are trying to avoid overloading for every physical machine of an automated resources management system that uses virtualization technology for allocating resources dynamically. We develop a new algorithm for predicting the future load of each physical machine and then decide which may be overloaded next. Then we can take the necessary action to prevent the overload in the system. The experimental results support the improvements of our algorithm.

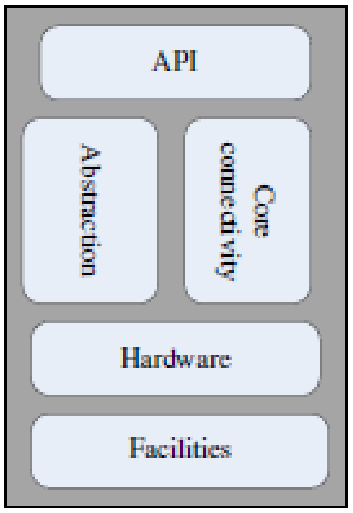
Keywords: Cloud computing, physical machine, virtual machine, virtualization technology, resources management

I. Introduction

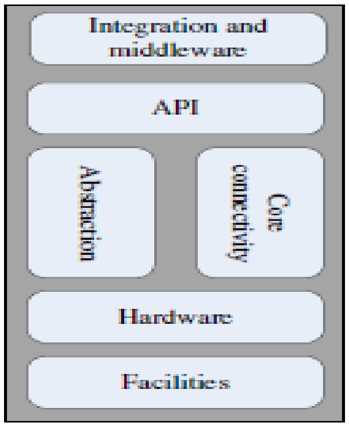
The word 'Cloud computing' is used to describe the concepts of computing in a large number of computers connected through a real-time communication network such as the Internet. In science, cloud computing is a synonym for distributed computing over a network, and means the ability to run a program or application on many connected computers at the same time [1]. It also refers to network-based services, which appear to be provided by physical server hardware, and are in fact served up by virtual hardware, simulated by software running on one or more physical machines [PMs]. Such virtual machines [VM] do not physically exist and can therefore be moved around and scaled up or down on the fly without affecting the end user.

Cloud computing has three delivery models. These are:–

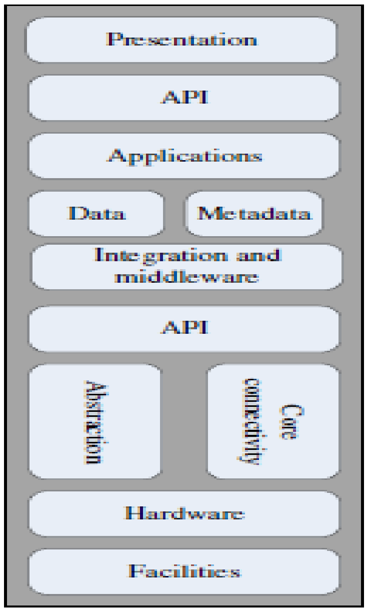
a) Software as a Service (SaaS) – This model allows the user to use applications supplied by the provider. But they have no control over the underlying cloud infrastructure.
b) Platform as a Service (PaaS) – This model allows deploying user-created or user-acquired applications using programming languages and tools supported by the provider. The users have no control on the underlying cloud infrastructure. They have control only over the deployed applications and application hosting environment configurations.
c) Infrastructure as a Service (IaaS) – This model allows the user to deploy and run arbitrary software, which can include operating system and applications. The users have limited control over the underlying cloud infrastructure such as operating systems, storage etc.



(a) Software as a Service



(b) Platform as a Service



(c) Infrastructure as a Service

Fig.1. Three delivery model of cloud computing

The above three models again deployed as following:-

a) Private cloud – This cloud is available solely for an organization.
b) Public cloud – This cloud is available to the general public and is owned by a cloud service selling organization.
c) Community cloud – This cloud is shared by several organizations.
d) Hybrid cloud – This cloud is a composition of two or more clouds.

II. Related Work

In cloud computing platform, more than one VMs get services from a PM. Hypervisors like Xen provide a mechanism for mapping these VMs to PMs and the cloud users are totally ignorant about this mapping. Now a PM should have sufficient resources to meet all the demands of all VMs running on it. Otherwise, the PM will be overloaded and it will not provide resources to some VMs. Those VMs will be idle for certain time and this situation can degrade the performance of its VMs.

Such situation can be avoided by predicting the future load of each PM. First predict the future resource demands of each VM. Then the future load of a PM is predicted by aggregating all the predicted resource demands of each VMs running on the PM. For a particular resource if it is predicted that the demand exceeds than the PM have, then the PM will be overloaded in the next time interval. In this situation some VMs should be migrated from that PM until the predicted load does not lead to overloaded system. Also the accepting PMs should not be overloaded.

In FUSD (Fast Up and Slow Down) algorithm the future resource needs of VMs are predicted based on previous statistics [2]. This prediction is done using the following equation:

$$E(t) = \alpha * E(t-1) + (1-\alpha) * O(t), \quad 0 \le \alpha \le 1,$$

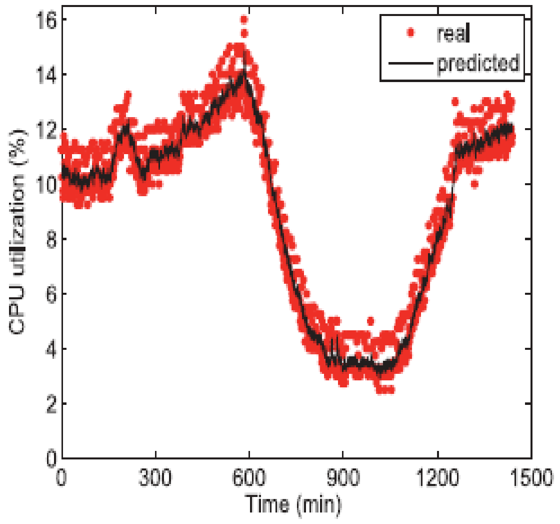
Where $E(t)$ and $O(t)$ are the estimated and the observed load at time t, respectively. α is a constant reflecting a tradeoff between stability and responsiveness.

The designers of this algorithm use the above formula to predict the CPU load on the DNS server in their university. They measure the load every minute and predict the load in the next minute. After many observations they set the value of α is 0.7 and the Fig. 2a shows the corresponding results. The dotted line in the figure represents the observed values and the curve represents the predicted values. The figure shows that the curve cuts through the middle of the dots. This indicates that the prediction is satisfactory.

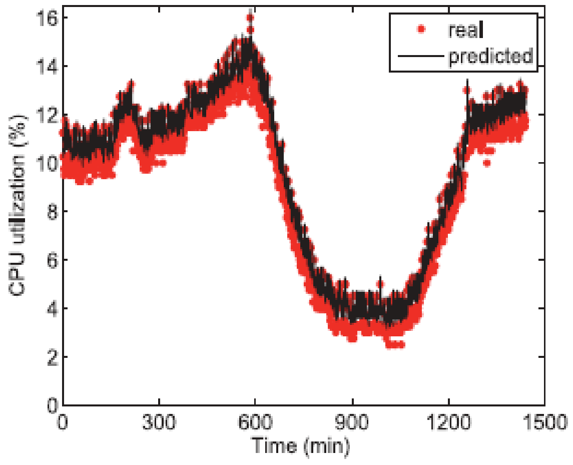
But this formula does not reflect the acceleration of resource demands. As for example, if the sequence of $O(t)$ is 20, 30, 40, and 50, then it is logical to predict the next value to be 60. Unfortunately, when α is between 0 and 1, the predicted value is always between the historic and observed value. To reflect the accelaration they set α to a negative value. The above formula is transformed as follow for $-1 \le \alpha < 0$

$$E(t) = -|\alpha| * E(t-1) + (1+|\alpha|) * O(t)$$

Hence, to measure $E(t)$ they use two parameters, ↑α and ↓α when $O(t)$ is increasing or decreasing, respectively. They select the values of these two parameters ↑α = -0.2 and ↓α = 0.7 based on field experience. Fig. 2b shows the results with these values.



(a) α=0.7



(b) ↑α = -0.2 ↓α = 0.7

Fig.2. CPU load prediction using FUSD algorithm

III. Proposed Work

In this paper we introduced another algorithm for predicting the future resource demands of VMs. This prediction is based on previous statistics like FUSD algorithm. Here the predicting formula is slightly changed. The new formula is:

$$E(t) = m * A(t-1),$$

Where $E(t)$ and $A(t)$ are the estimated and actual load at time t, respectively. m is a multiplier whose value is calculated from previous steps as follows:

$$m = \frac{A(t-1)}{A(t-2)}.$$

Though it is not possible to predict the actual future load of a VM, our algorithm predicts very closely. It also reflects both the acceleration and deceleration of resource usage. We do not need another equation for capturing the rising trends of resource usage. For example, if we get the sequence of $A(t)$ 20, 30, 40, and 50, then it is logical to predict the next value to be 60. Our algorithm predicts it as 62.5, which is close to 20.

Any future load prediction algorithm does not always give the actual result. Sometimes it gives higher value than the actual. It is treated as a high error. When the algorithm gives lower value than the actual, it will be treated as a low error. There is no problem for high error. In case of low error, there is a possibility that a PM may be overloaded in next time interval but we do not take any action to avoid it due to low prediction of the algorithm. So a better algorithm must decrease the probability of occurrence the low error. In this sense, we can tell that our algorithm is better than the existing one. The above two examples also support this.

IV. Simulation Results

We use our algorithm for prediction the percentage of CPU utilization on the DNS server in Kalyani University. We predict the CPU load in every minute by measuring the actual loads of previous two minutes. The simulation results are shown in Fig.3.

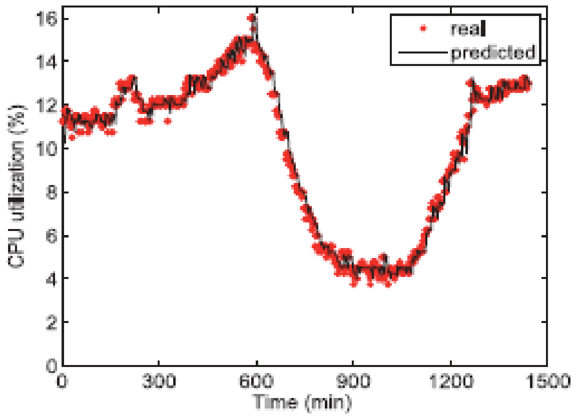


Fig.3. Simulation results of CPU load prediction

V. CONCLUSION

We have implemented an algorithm for predicting the future resource demands of VMs for avoiding the system overloading. Though our algorithm predicts satisfactorily, in future we have to be conservative in decreasing our prediction when the actual resource usage is decreasing.

VI. REFERENCE

1) Mariana Carroll, Paula Kotzé, Alta van der Merwe (2012). "Securing Virtual and Cloud Environments". In I. Ivanovetal. Cloud Computing and Services Science, Service Science: Research and Innovations in the Service Economy. Springer Science + Business Media. doi:10.1007/978-1-4614-2326-3.

2) Zhen Xiao, Weijia Song, Qi Chen." Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment". Ieee transactions on parallel and distributed systems, vol. 24, no. 6, june 2013.

3) R. Ranjan and R. Buyya. Decentralized Overlay for Federation of Enterprise Clouds. Handbook of Research on Scalable Computing Technologies, K. Li et. al. (ed), IGI Global, USA, 2009 (in press).

4) "Amazon elastic compute cloud (Amazon EC2),"http://aws.amazon.com/ec2/, 2012.

5) M. Nelson, B.-H. Lim, and G. Hutchins, "Fast Transparent Migration for Virtual Machines," Proc. USENIX Ann. Technical Conf., 2005

6) C.A. Waldspurger, "Memory Resource Management in VMware ESX Server," Proc. Symp. Operating Systems Design and Implementation (OSDI '02), Aug. 2002.

7) J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, and R.P. Doyle, "Managing Energy and Server Resources in Hosting Centers," Proc. ACM Symp. Operating System Principles (SOSP '01), Oct. 2001

8) N. Bila, E.d. Lara, K. Joshi, H.A. Lagar-Cavilla, M. Hiltunen, and M. Satyanarayanan, "Jettison: Efficient Idle Desktop Consolidation with Partial VM Migration," Proc. ACM European Conf. Computer Systems (EuroSys '12), 2012

9) A. Singh, M. Korupolu, and D. Mohapatra."Server-Storage Virtualization: Integration and Load Balancing in Data Centers," Proc. ACM/IEEE Conf. Supercomputing, 2008.

10) C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A Scalable Application Placement Controller for Enterprise Data Centers," Proc. Int'l World Wide Web Conf. (WWW '07), May 2007.

11) N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," Proc. IFIP/IEEE Int'l Symp. Integrated Network Management (IM '07), 2007.

12) R. Nathuji and K. Schwan, "Virtualpower: Coordinated Power Management in Virtualized Enterprise Systems," Proc. ACM SIGOPS Symp. Operating Systems Principles (SOSP '07), 2007.

13) T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-Box and Gray-Box Strategies for Virtual Machine Migration," Proc. Symp. Networked Systems Design and Implementation (NSDI '07), Apr. 2007.

14) P. Padala, K.-Y. Hou, K.G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant, "Automated Control of Multiple Virtualized Resources," Proc. ACM European conf. Computer Systems (EuroSys '09), 2009.

15) M. Zaharia, A. Konwinski, A.D. Joseph, R.H. Katz, and I. Stoica, "Improving MapReduce Performance in Heterogeneous Environments,"Proc. Symp. Operating Systems Design and Implementation (OSDI '08), 2008.

Mr **Debabrata Sarddar,** Assistant Professor in the Department of Computer Science and Engineering, University of Kalyani, Kalyani,Nadia, West Bengal, INDIA. He has done PhD at Jadavpur University. He completed his M.Tech in Computer Science & Engineering from DAVV, Indore in 2006, and his B.E in Computer Science & Engineering from NIT, Durgapur in 2001. He has published more than 75 research papers in different journals and conferences. His research interest includes cloud computing and mobile system and WSN.
Email: dsarddar@rediffmail.com

Mr **Prodip Ghosh,** Assistant Professor in the Department of Computer Science and Engineering, Narula Institute of Technology, Agarpara, Kolkata, West Bengal, INDIA. He completed his M.E in Computer Science & Engineering from BESU, Shibpur in 2010, and his B.E in Computer Science & Engineering from GCETTB, Berhampore in 2007. His research area is cloud computing.
Email:gprodip@yahoo.co.in