International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 4 798 – 804

Dynamic Resource Management Using Skewness and Load Prediction Algorithm for Cloud Computing

SAROJA V[1]
[1]P G Scholar, Department of Information Technology
Sri Venkateswara College of Engineering
Chennai, TamilNadu, India
vsaroja@svce.ac.in

SIVAGAMI V M[2]
[2]Associate Professor, Department of Information Technology
Sri Venkateswara College of Engineering
Chennai, TamilNadu, India
vmsiva@svce.ac.in

***Abstract*—** Cloud computing is the delivery of resources and services on an on-demand basis over a network. There are various technical challenges that needs to be addressed in cloud computing like Virtual Machine migration, server consolidation, fault tolerance, high availability and scalability but central issue is the load balancing, which is the mechanism of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. It also ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. The nodes are clustered based on processor type, memory speed, hard disk capacity and cost per hour. The clustered resources are allocated dynamically to the jobs by FIFO scheduling. After the resource allocation, the skewness (uneven resource utilization) is measured. The resources have been allocated dynamically which can make the maximum resources to be used efficiently so that overall utilization of resources can be improved. An effective load rebalancing algorithm with dynamic resource allocation using Clustering, Classification and VM migration has been developed to achieve overload avoidance The load is balanced by classifying the resources as hot, warm and cold spots and subsequently mitigating the hot spots to avoid overload of servers. After the elimination of hot spots, the underutilized servers are identified and its load is migrated to either warm or other cold spots and they are shut down to conserve power.

Keywords- *Cloud computing, Virtual machine migration, Load balancing, Skewness, Green computing, Hot spot, Warm spot, Cold spot, Hot spot mitigation and Cold spot mitigation*

*****

I. INTRODUCTION

Cloud computing is emerging as a new paradigm of large scale distributed computing. It has moved computing and data away from desktop and portable PCs, into large data centers[1]. It has the capability to harness the power of Internet and wide area network (WAN) to use resources that are available remotely, thereby providing cost effective solution to most of the real life requirements[2][3]. Cloud computing can be classified as a new paradigm for the dynamic provisioning of computing services supported by state-of-the-art data centers that usually employ Virtual Machine(VM) technologies for consolidation and environment isolation purposes [4]. Three markets are associated to it. Infrastructure-as-a-Service (IaaS) designates the provision of IT and network resources such as processing, storage and bandwidth as well as management middle ware. Platform-as-a-Service (PaaS) designates programming environments and tools supported by cloud providers that can be used by consumers to build and deploy applications onto the cloud infrastructure. Software-as-a-Service (SaaS) designates hosted vendor applications. IaaS, PaaS and SaaS all include self-service (APIs) and a pay-as-you-go billing model.

Today, there are more than a hundred million computing devices connected to the Internet and many of them are using cloud computing services daily. According to the IDC's anticipation , the SaaS (Software As A Service) market reached $13.1 billion in revenue at 2009 will grow to $40.5 billion by 2014 at a compound annual growth rate(CAGR) of 25.3%[5][6][7]. These networked devices submit their requests to a service provider and receive the results back in a timely manner without the involvement of the service complexity related to information storage and process, interoperating protocols, service composition, communications and distributed computation, which are all relied on the network and the backend servers to offer desirable performance. In a cloud computing environment, users can access the operational capability faster with internet application [8], and the computer systems have the high stability to handle the service requests from many users in the environment. Cloud computing involving distributed technologies to satisfy a variety of applications and user needs. Sharing resources, software, information via internet are the main functions of cloud computing with an objective to reduced capital and operational cost, better performance in terms of response time and data processing time, maintain the system stability and to accommodate future modification in the system.

Load Balancing is done with the help of load balancers where each incoming request is redirected and is transparent to client who makes the request. [9][10] Based on predetermined parameters, such as availability or current load, the load balancer uses various scheduling algorithm to determine which server should handle and forwards the request on to the selected server. To make the final determination, the load balancer retrieves information about the candidate server's health and current workload in order to verify its ability to respond to that request. If the selected server is already loaded with more task then migrating some tasks running in that server into some other server to achieve overload avoidance. Load balancing solutions can be divided into software-based load balancers and hardware-based load balancers. Hardware-based load balancers are specialized boxes that include Application

Specific Integrated Circuits (ASICs) customized for a specific use. They have the ability to handle the high speed network traffic where as Software-based load balancers run on standard operating systems and standard hardware components.

Lowering the energy usage of data centers is a challenging and complex issue because computing applications and data are growing so quickly that increasingly larger servers and disks are needed to process them fast enough within the required time period. Green Cloud computing is envisioned to achieve not only the efficient processing and utilization of a computing infrastructure, but also to minimize energy consumption [11]. This is essential for ensuring that the future growth of Cloud computing is sustainable. Otherwise, Cloud computing with increasingly pervasive frontend client devices interacting with back-end data centers will cause an enormous escalation of the energy usage. In green computing, data center resources need to be managed in an energy-efficient manner. Cloud resources need to be allocated not only to satisfy Quality of Service (QoS) requirements specified by users via Service Level Agreements (SLAs), but also to reduce energy usage.

The rest of the paper is organized as follows. Section 2 discusses related work, followed by the proposed system design which consists of Load balancing Cloud architecture, Load prediction algorithm, Skewness algorithm and Classification of servers used in Cloud computing presented in Section 3. The proposed load balancing technique called hot spot mitigation is discussed in Section 4. Another proposed energy aware green computing algorithm called cold spot mitigation is discussed in section 5. In Section 6 we discuss our simulation and results of proposed cloud system using CloudSim. Section 7 concludes the paper with summary and future research directions.

II. RELATED WORK

One of the first works, in which power management has been applied at the data center level, has been done by Pinheiro et al. [12]. In this work the authors have proposed a technique for minimization of power consumption in a heterogeneous cluster of computing nodes serving multiple web-applications. The main technique applied to minimize power consumption is concentrating the workload to the minimum of physical nodes and switching idle nodes off. This approach requires dealing with the power/performance trade-off, as performance of applications can be degraded due to the workload consolidation. Requirements to the throughput and execution time of applications are defined in SLAs to ensure reliable QoS. The proposed algorithm periodically monitors the load of resources (CPU, disk storage and network interface) and makes decisions on switching nodes on/off to minimize the overall power consumption, while providing the expected performance. The actual load balancing is not handled by the system and has to be managed by the applications. The algorithm runs on a master node, which creates a Single Point of Failure (SPF) and may become a performance bottleneck in a large system. In addition, the authors have pointed out that the reconfiguration operations are time-consuming, and the algorithm adds or removes only one node at a time, which may also be a reason for slow reaction in large-scale environments. The proposed approach can be applied to multi-application mixed-workload environments with fixed SLAs.

Elnozahy et al. [13] have investigated the problem of power efficient resource management in a single web-application environment with fixed SLAs (response time) and load balancing handled by the application. As in [14], two power saving techniques are applied: switching power of computing nodes on/off and Dynamic Voltage and Frequency Scaling (DVFS). The main idea of the policy is to estimate the total CPU frequency required to provide the necessary response time, determine the optimal number A. Beloglazov et al. / Future Generation Computer Systems 28 (2012) 755–768 757 of physical nodes and set the proportional frequency to all the nodes. However, the transition time for switching the power of a node is not considered. Only a single application is assumed to be run in the system and, like in [12], the load balancing is supposed to be handled by an external system. The algorithm is centralized that creates an SPF and reduces the scalability. Despite the variable nature of the workload, unlike [13], the resource usage data are not approximated, which results in potentially inefficient decisions due to fluctuations. Nathuji and Schwan [15] have studied power management techniques in the context of virtualized data centers, which has not been done before. Besides hardware scaling and VMs consolidation, the authors have introduced and applied a new power management technique called “soft resource scaling”.

The idea is to emulate hardware scaling by providing less resource time for a VM using the Virtual Machine Monitor’s (VMM) scheduling capability. The authors found that a combination of ‘‘hard’’ and ‘‘soft’’ scaling may provide higher power savings due to the limited number of hardware scaling states. The authors have proposed an architecture where the resource management is divided into local and global policies. At the local level the system leverages the guest OS’s power management strategies. However, such management may appear to be inefficient, as the guest OS may be legacy or power-unaware.

III. SYSTEM DESIGN

A. *Cloud Architecture*

All the nodes in the cloud have a unique identifier. These nodes are initially clustered by k-means clustering based on the type of the Processor such as Dual Core, 4Core, 6Core etc. Nodes belonging to every cluster are further clustered based on their memory speed. These clusters are again sub-clustered based on their hard disk capacity and finally by their cost per hour.

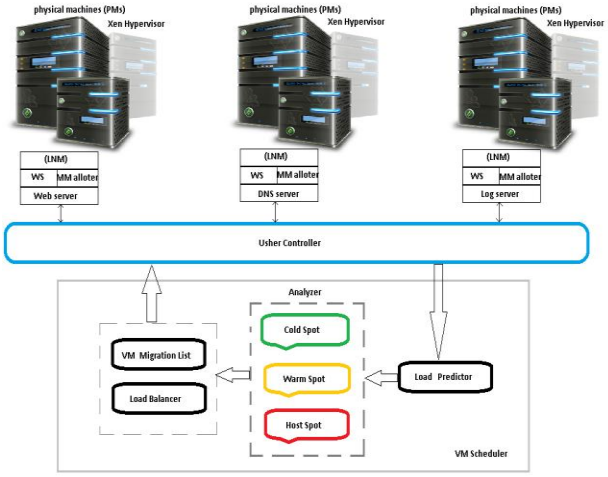

Figure1. Load balancing architecture

In each node there is a manager called Local Node Manager (LNM) is running. Its job is to monitor the node and periodically generate resource utilization report and send it to

Usher controller. Usher is a cluster management system designed to substantially reduce the administrative burden of managing cluster resources while simultaneously improving the ability of users to request, control, and customize their resources and computing environment. The proposed system model is given in figure1. The report generated by Usher controller is given as input to load predictor algorithm.

B. Load Prediction Algorithm

We need to predict the future resource needs of VMs. One of the possibility is to look inside a VM for application level statistics, e.g., by parsing logs of pending requests. Doing so requires modification of the VM which may not always be possible. Instead, we make our prediction based on the past external behaviors of VMs.

Step 1: Calculate an exponentially weighted moving average (EWMA) using a TCP-like scheme. Here the estimated load and observed load at particular time t can be calculated using the following equation (1).

$$E(t) = \alpha * E(t - 1) + (1 - \alpha) * O(t); 0 < \alpha < 1 \quad (1)$$

where α reflects a tradeoff between stability and responsiveness.

Step 2: We use the EWMA formula to predict the CPU load on the server. We measure the load every minute and predict the load in the next minute. Use the value for α as α = 0.7

Step 3: When the observed resource usage is going down, we want to be conservative in reducing our estimation. In most of the time (77%) the predicted values are higher than the observed ones. The median error is increased to 9:4% because we trade accuracy for safety. It is still quite acceptable nevertheless.

Step 4: When α is between 0 and 1, the predicted value is always between the historic value and the observed value. To reflect acceleration set α to a negative value. When α is between -1 and 0, the equation (1) can be transformed as (2) which is given below.

$$E(t) = - |\alpha| * E(t - 1) + (1 + |\alpha|) * O(t); -1 < \alpha < 0$$
$$= O(t) + |\alpha| * (O(t) - E(t - 1)) \quad (2)$$

This prediction is done based on the past external behaviours of VMs.

C. Skewness Algorithm

Skewness is the measure of uneven resource utilization of a server. Let *n* be the number of resources present in server and *ri* be the utilization of the *i*-th resource. We define the resource skewness of a server *p* as the following equation (3).

$$\text{skewness}(p) = \sqrt{\sum_{i=1}^{n} (\frac{r_i}{r} - 1)2} \quad (3)$$

where *r* is the average utilization of all resources for server *p*. In practice, not all types of resources are performance critical and hence we only need to consider bottleneck resources in the above calculation. By minimizing the *skewness*, we can combine different types of workloads nicely and improve the overall utilization of server resources.

D. Classification of Servers

The nodes had clustered based on processor type, memory speed and network bandwidth and they are further classified into types based on its resource utilization such as Hot, Warm and Cold spots.

Hot: We define a server as a hot spot if the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VMs running on it should be migrated away.

Warm: Servers having the risk of becoming a hot spot in the face of temporary fluctuation of application resource demands.

Cold: We define a server as a cold spot if the utilizations of all its resources are below a cold threshold. This indicates that the server is mostly idle/under utilized and it may be turn off to save energy.

IV. HOT SPOT MITIGATION

To reduce the temperature of the hot spots to less or equal to warm threshold. The nodes in hot spots are sorted by quick sort in the descending order. VM with the highest temperature should be first migrated away. Destination is decided based on least cold node. After every migration the status of each node is updated. This procedure continues until all hot spots are eliminated.

The VM which is removed from the identified hot spot can reduce the skewness of that server the most. For each VM in the list, if a destination server can be found to accommodate it then that server must not become a hot spot after accepting this VM. Among all such servers, we select one whose skewness can be reduced the most by accepting this VM. Note that this reduction can be negative which means we select the server whose skewness increases the least. If a destination server is found, then the VM can be migrated to that server and the predicted load of related servers was updated. Otherwise, move on to the next VM in the list and try to find a destination server for it. As long as a destination server was found for any of its VMs, it can be considered as this run of the algorithm a success and then move on to the next hot spot. Note that each run of the algorithm migrates away at most one VM from the overloaded server.

This does not necessarily eliminate the hot spot, but at least reduces its temperature. If it remains a hot spot in the next decision run, the algorithm will repeat this process. It is possible to design the algorithm so that it can migrate away multiple VMs during each run. But this can add more load on the related servers during a period when they are already overloaded. It is decided to use this more conservative approach and leave the system some time to react before initiating additional migrations.

There are two scenarios are considered in hot spot mitigation. In the first scenario, the VMs running in identified hot spots are migrated to warm spot servers which will not become hot by accommodating the VMs. In the second scenario, if sufficient warm spots are not available to accommodate the VMs in the hot spot, few loads are migrated to the nodes in the cold spot also to mitigate the hot spots. The before and after migration of load from hot spot to identified warm spot and/or cold spots is given in below figures 2, 3, 4 and 5.

 ISSN: 2321-8169
798 – 804

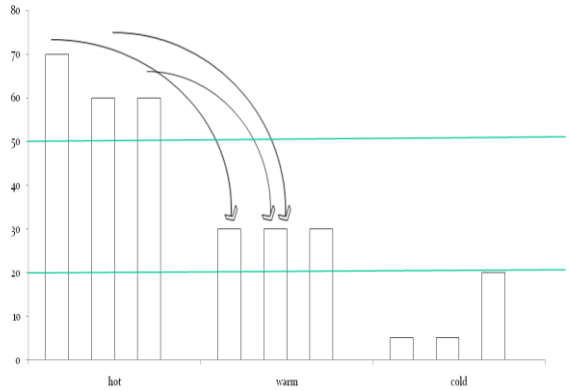

Figure2. Migration of load from hot to warm spots

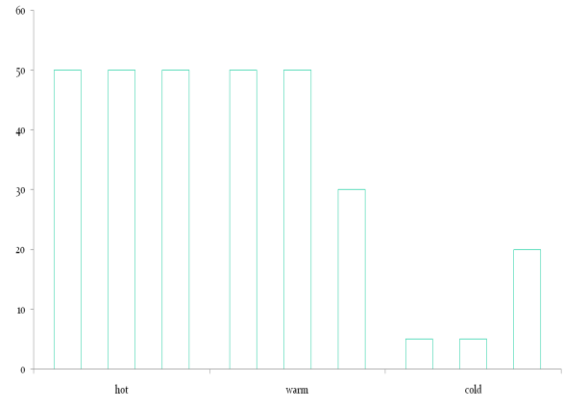

Figure3. After migration of load from hot to warm spots

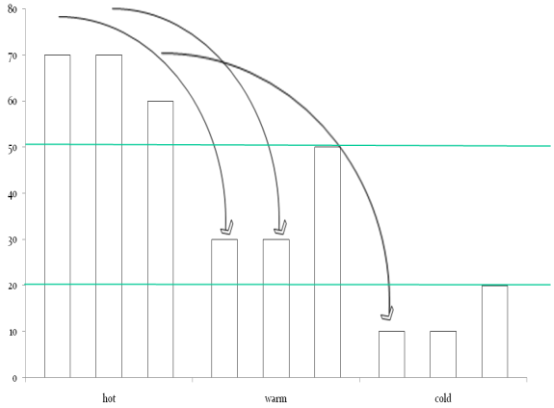

Figure4. Migration of load from hot spot to warm & cold spots

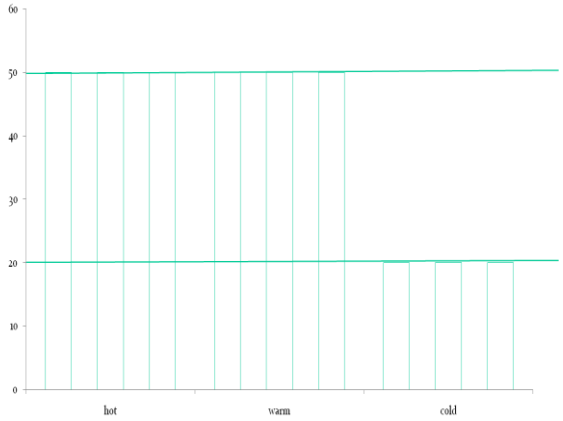

Figure5. After migration of load from hot spot to warm & cold spots

V. COLD SPOT MITIGATION

To reduce the power consumption, the servers that are under-utilized are switched off. We sort the cold spots in ascending order and select the node with the least temperature. The VMs in these least loaded cold spot servers should be migrated away to another cold spot without raising the temperature of the destination above the warm threshold. If such cold spot destinations are not available, then move the load in the cold spot to a warm spot without raising the temperature of the destination above the warm threshold. After migration source server can be switched off thus contributing to efficiency of energy.

There are two scenarios are considered in cold spot mitigation. In the first scenario the VMs in these least loaded cold spot servers should be migrated away to another cold spot without raising the temperature of the destination above the warm threshold. If such destinations are available, the VM can be migrated and source server can be switched off thus contributing to efficiency of energy. The first scenario is given in figure 6 and 7.

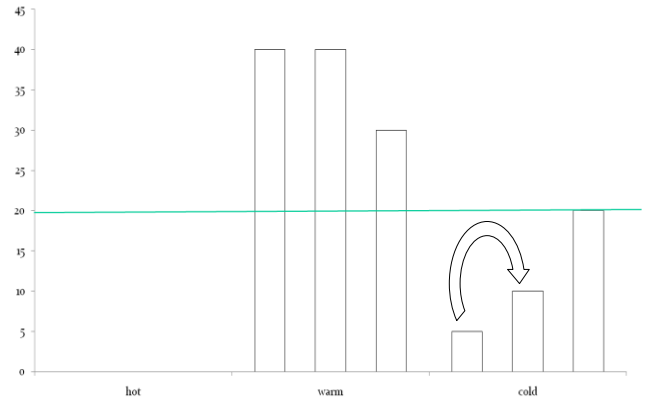

Figure6. Migration of load from one cold to another cold spot

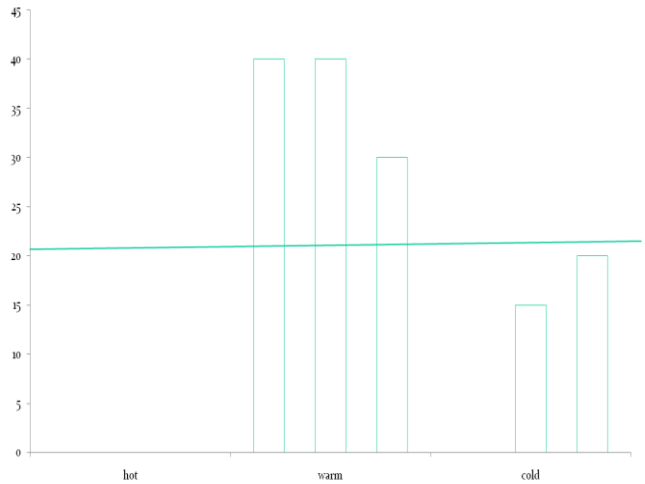

Figure7. After Migration switching off of one cold spot

 ISSN: 2321-8169
798 – 804

In the second scenario if no cold spots are available, move the load in the cold spot to a warm spot without raising the temperature of the destination above the warm threshold. If such destinations are available as in figure 8, the VM can be migrated and source server can be switched off.

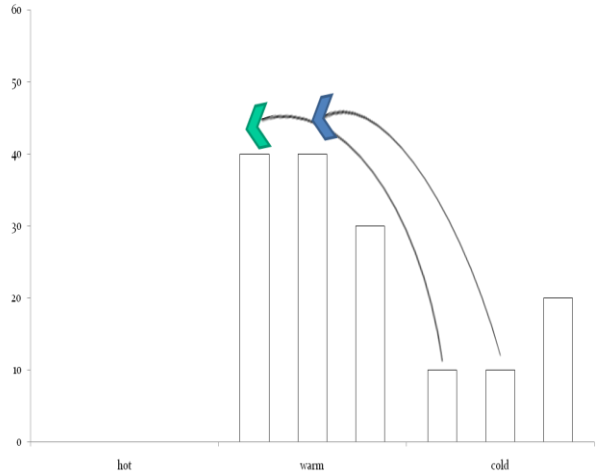

Figure8. Migration of load from cold spots to warm spots

VI. SIMULATION AND RESULTS

CloudSim allows cloud customers to test their services in repeatable and controllable environment free of cost, and to turn the performance bottlenecks before deploying on real clouds. It can provide a generalized and extensible simulation framework that enables modeling, simulation and experimentation of emerging cloud computing infrastructures and application services. It is designed for studying various resource management approaches and scheduling algorithms in cloud environment. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines and provisioning policies of resource. It implements generic application provisioning techniques that can be extended with ease and limited efforts In CloudSim, users is modeled by a Datacenter Broker, which is responsible for mediating between users and service providers depending on users' QoS (Quality of Service) requirements and deploys service tasks across Clouds. In our experiments, we have used CloudSim as a simulator for checking the performance of our improved load prediction algorithm and the green computing algorithm. CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments. We have considered Virtual Machines as resource and Cloudlets as tasks/jobs.

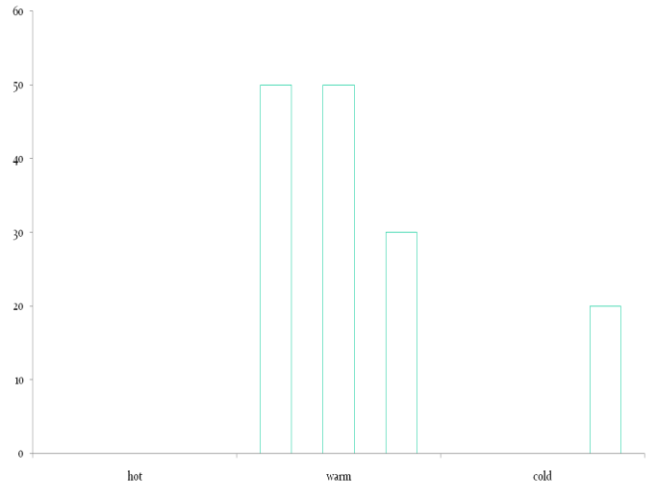

Figure9. After Migration switching off two cold spots

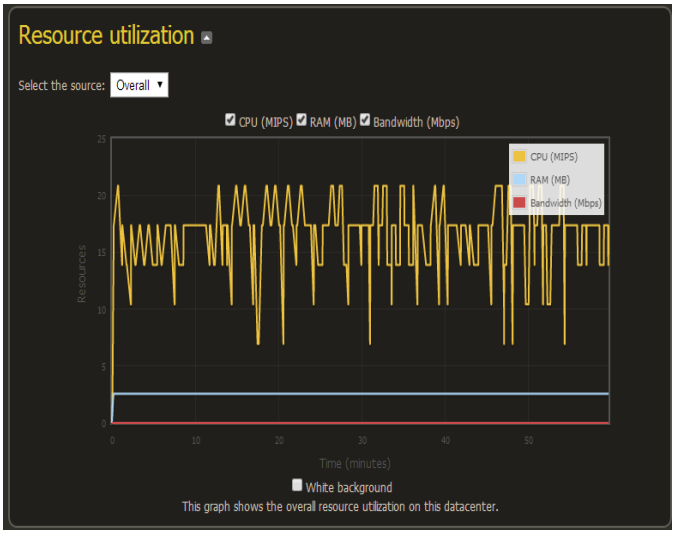

Figure10. Overall resource utilization on selected datacenter

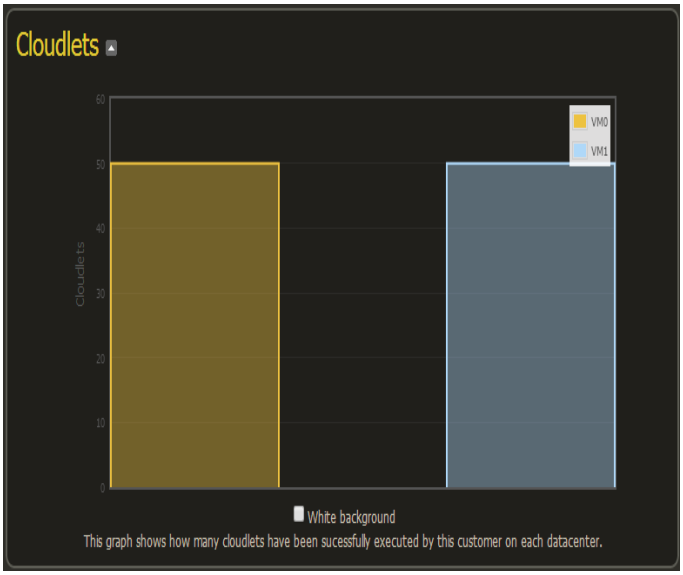

Figure11. Graph shows how many cloudlets have been successfully executed by selected customer on each datacenter

 ISSN: 2321-8169
798 – 804

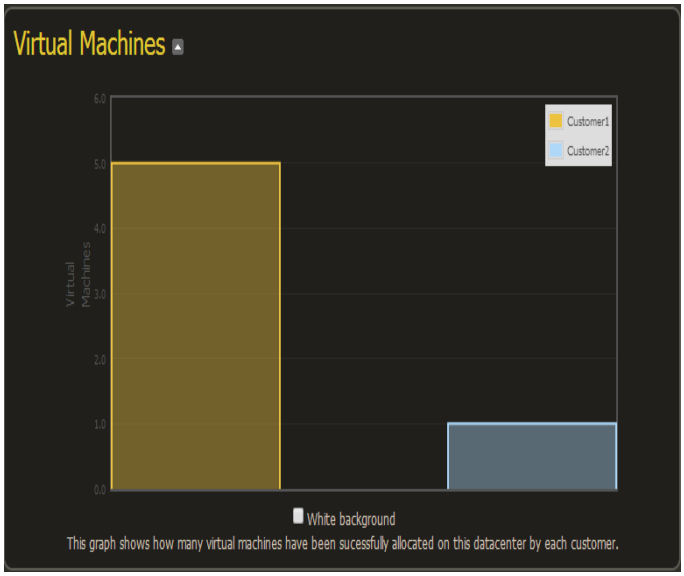

Figure12. Graph shows how many VMs have been successfully allocated on selected datacenter by each customer

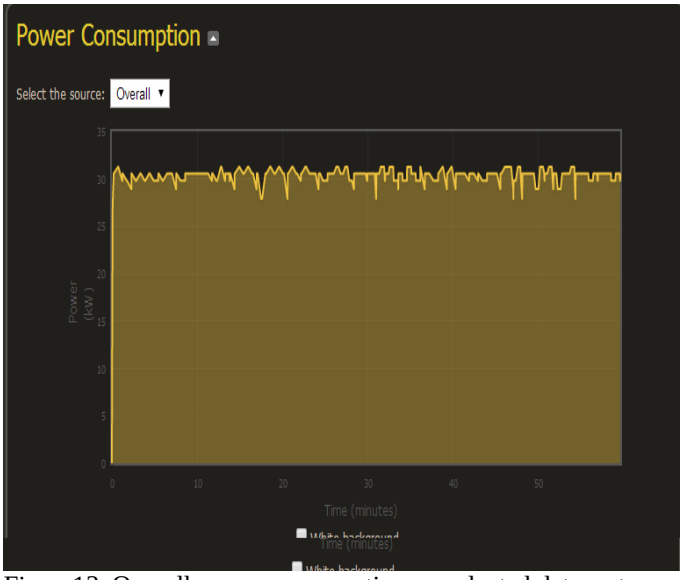

Figure13. Overall power consumption on selected datacenter

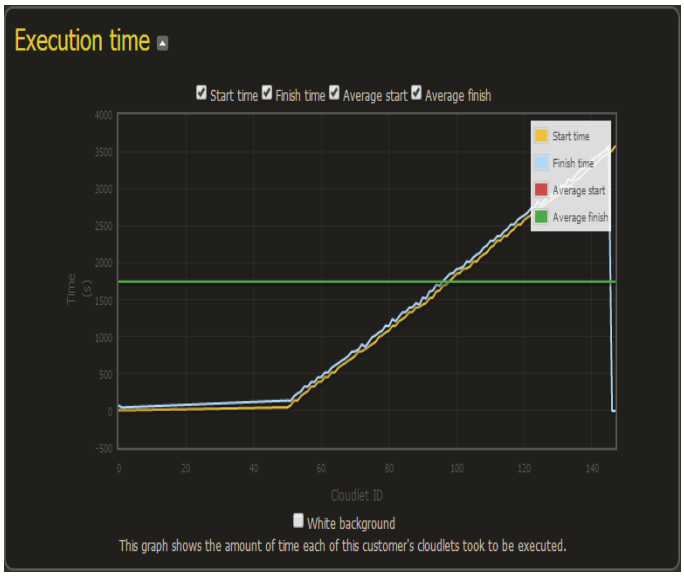

Figure14. Graph shows the amount of time each of the selected customer's cloudlets took to be executed

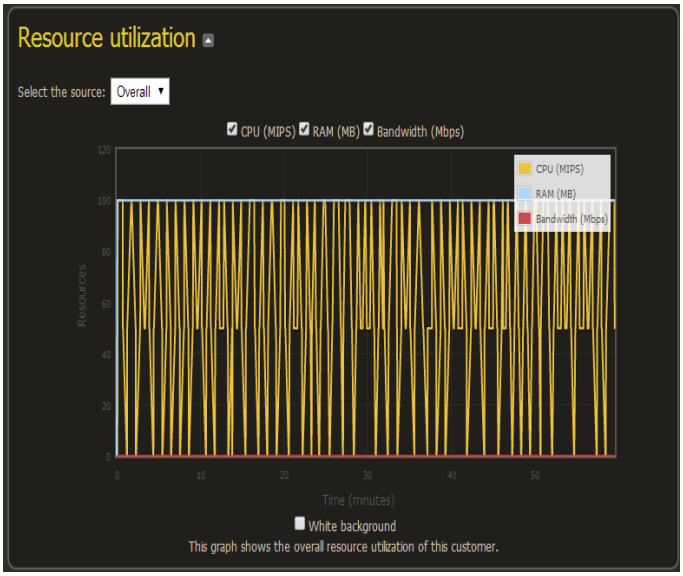

Figure15. Overall resource utilization of selected customer

VII. CONCLUSIONS & FUTURE WORK

The solution for allocating data center resources dynamically based on application demands have designed along with load prediction algorithm. The occurrence of over provisioning and under provisioning will be avoided by allocation of resources dynamically using load prediction algorithm. The overall utilization of server resources can be improved by minimizing skewness. The proposed system optimize the number of servers actively in use by green computing. The main goal of proposed system is overload avoidance and energy efficiency. In future the proposed work can be extended by the development of software platform that supports energy efficient management and allocation of datacenter resources.

REFERENCES

[1] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling And Simulation Of Scalable Cloud Computing Environments And The Cloudsim Toolkit: Challenges And Opportunities," Proc. Of The 7th High Performance Computing And Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.

[2] Nidhi Jain Kansal, "Cloud Load Balancing Techniques : A Step Towards Green Computing", IJCSI International Journal Of Computer Science Issues, January 2012, Vol. 9, Issue 1, No 1, , Pg No.:238-246, ISSN (Online): 1694-0814

[3] R. P. Mahowald, Worldwide Software As A Service 2010–2014 Forecast: Software Will Never Be Same ,In, IDC, 2010

[4] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, Xen and the art of virtualization, in: Proceedings of the 19th ACM Symposium on Operating Systems Principles, SOSP 2003, Bolton Landing, NY, USA, 2003, p. 177.

[5] Rich Lee, Bingchiang Jeng "Load Balancing Tactics In Cloud" International Conference On Cyber Enabled Distributed Computing And Knowledge Discovery, 2011

[6] A. Bhadani , And S. Chaudhary , "Performance Evaluation Of Web Servers Using Central Load Balancing Policy Over Virtual Machines On Cloud", Proceedings Of The Third Annual ACM Bangalore Conference (COMPUTE), January 2010.

[7] http://www.ca.com/~/media/Files/whitepapers/turnkey_clouds_turnkey_profits.pdf.

[8] K.D. Devine, E.G. Boman , R.T. Hepahy, B.A.Hendrickson, J.D. Teresco, J. Faik,J.E. Flaherty,L.G. Gervasio," New Challenges In Dynamic Load Balancing, Applied Numerical Mathematics,52(2005)133-152.

[9] Mishra , Ratan , Jaiswal, Anant,P"Ant Colony Optimiza tion: A Solution Of Load Balancing In Cloud",April 2012, International Journal Of Web & Semantic Technology;Apr2012, Vol. 3 Issue 2, P33

[10] Eddy Caron , Luis Rodero-Merino "Auto-Scaling , Load Balancing And Monitoring In Commercial And Open-Source Clouds " Research Report ,January2012

[11] R. Buyya, A. Beloglazov, J. Abawajy, Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges, in: Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2010, Las Vegas, USA, 2010.

[12] E. Pinheiro, R. Bianchini, E.V. Carrera, T. Heath, Load balancing and unbalancing for power and performance in cluster-based systems, in: Proceedings of the Workshop on Compilers and Operating Systems for Low Power, 2001, pp. 182–195.

[13] E. Elnozahy, M. Kistler, R. Rajamony, Energy-efficient server clusters, Power- Aware Computer Systems (2003) 179–197.

[14] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, R.P. Doyle, Managing energy and server resources in hosting centers, in: Proceedings of the 18th ACM Symposium on Operating Systems Principles, ACM, New York, NY, USA, 2001, pp. 103–116.

[15] R. Nathuji, K. Schwan, Virtualpower: coordinated power management in virtualized enterprise systems, ACM SIGOPS Operating Systems Review 41 (6) (2007) 265–278.