An Active Resource Provision for Cloud Computing Environment

Naveen Chandra Gowda* **Shivakumar B**** **Venkatesh D*****

*PG Student, Department of Computer Science & Engineering, Gates Institute of Technology, Gooty, Anantapur, AP

**Asst. Prof. Department of Computer Science & Engineering, Gates Institute of Technology, Gooty, Anantapur, AP

***Prof.&Dean, Department of Computer Science & Engineering, Gates Institute of Technology, Gooty, Anantapur, AP

Abstract

Cloud computing is all that allocation of resources and services to the client requests on-demand basis over a network. There exist various challenges that need to be addressed in cloud computing like Virtual Machine migration, server consolidation, high availability and scalability but main central issue is the load balancing. The load balancing is the mechanism of distributing the load among all the nodes of a distributed system in order to improve the resource utilization and job response time, by taking care of the situation where few of the nodes are loaded heavily while other nodes are kept idle or doing very little amount of work. We also ensure that each processor in the system perform approximately the equal amount of task a instant of time. The skewness is measured for uneven utilization of resources of server. An effective load rebalancing algorithm with dynamic resource allocation using 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 that satisfies the green computing.

Keywords – Cloud Computing, Virtual Machine Migration, Load Balancing, Skewness, Green Computing

1. INTRODUCTION

Cloud computing is emerging as a new technology of large scale distributed computing. It transferred computing and data from desktop PCs and portable systems into large data centers as a cloud[1]. It also provides the capability to increase the power of Internet and wide area network to use resources that are available remotely, so as to provide cost effective solution to most of the real life requirements[2][3]. Cloud computing can be classified as a IaaS, PaaS and SaaS. 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 by using the services of Virtual Machines (VM).

In a cloud computing environment, users can access the operational capability faster with internet application, 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 Balancers perform the load balancing where each incoming request is redirected and is transparent to client who makes the request.[4][5] 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.

Green computing is implemented to achieve not only efficient processing and utilization of a computing infrastructure, but it is also to minimize energy consumption[6]. In green computing, data center resources are to be managed in an energy efficient manner. Cloud resources need to be allocated to satisfy Quality of Service (QoS) requirements specified by users via Service Level Agreements (SLAs) and also to reduce energy usage. The rest of the paper is organized as follows.

2. EXISTING SYSTEM

One of the first works, in which power management has been applied at the data center level[7], i.e. 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.

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. The proposed approach can be applied to multi-application mixed-workload environments with fixed SLAs.

Virtual machine monitors (VMMs) is a technology to map the virtual machines (VMs) in cloud to physical resources [8]. The mapping is hidden from the cloud users, i.e. user will not know place where of their

ISSN: 2278 – 1323

VM instances run. It is the cloud provider that makes sure the underlying network with all the physical machines (PMs) has sufficient re- sources to meet their needs. The change of mapping between VMs and PMs, when applications are running is made possible by VM live migration technology[9]. However, a policy issue remains same as how to decide the mapping process accurately so that the resource needs of VMs are met when the number of PMs used is minimized. This is challenging task when resource demands of VMs are heterogeneous due to the different set of applications they run on it and vary with time due to the variations in workload. The capacity of the PMs can be heterogeneous as it must handle multiple generations of hardware co-exist in a data center[10].

Drawbacks in existing system are,

- Over-provisioning.
- Do not know the place of running VM instances.
- Overloaded leads to degraded performance.
- Energy consumption by idle PMs.
- Uneven utilization of server.
- No prediction of future resource usage.

3. PROPOSED SYSTEM

In this paper, we try to the design and implementation of an active resource provision that achieves goals. The three goals are Resource Balance, Overload avoidance and Green Computing.

- *Resource Balance:* Mix the workload with different resource requirements together so that the overall utilization of server capacity is improved.
- *Overload Avoidance*: The capacity of a PM should be as sufficient as to satisfy the resource demands of all VMs running on it. Otherwise, the PM is overloaded and can lead to degradation in performance of its VMs.
- *Green computing:* The number of PMs used must be minimized as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off to save energy.

The contributions to be made in our algorithm are,

- We develop an active resource allocation system that avoids the overload occur in the system effectively by minimizing the number of servers used.
- We implement the concept of "skewness" in order to measure the uneven utilization of a server. By minimizing skewness, we can able to improve the overall usage of servers in the face of multidimensional resource constraints.
- We design a load prediction algorithm that can capture the future resource requirements of applications exactly without looking inside the VMs.

4. SYSTEM DESIGN

A. System Architecture

The System architecture is shown in the figure 4A. Each PM runs the Virtual Machine Monitor (VMM) which supports a privileged domain 0 and one or more one or more applications such as Web server, remote desktop, DNS, Mail, Map/Reduce, etc. We assume all PMs share same backend storage.

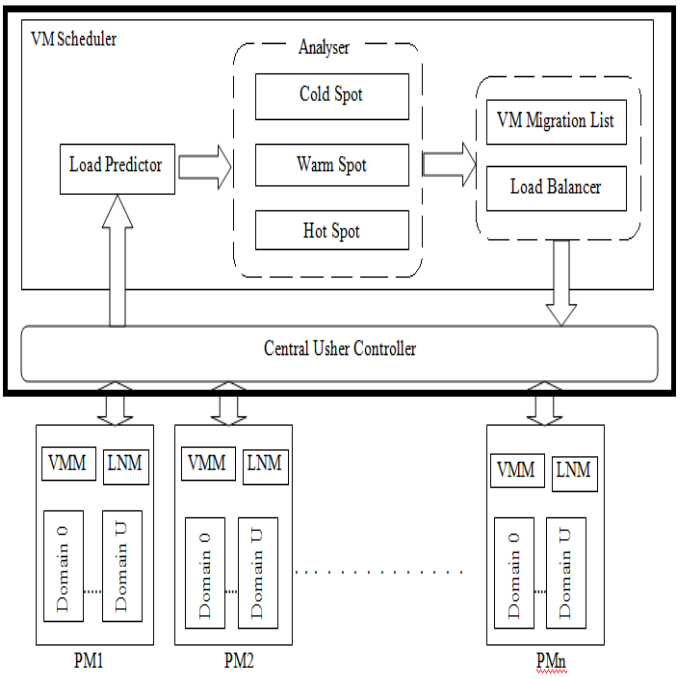

Figure 4A. System Architecture

Central Usher Controller Framework manages the multiplexing of VMs to PMs. The system is implemented as a set of plug-ins to Central Usher Controller. A local node manager (LNM) runs in each node on domain 0 is to collect the usage statistics of resources for each VM on that node. The CPU and network usage can be calculated by monitoring the scheduling events in PMs. The usage of memory within a VM is not visible to the VMM. The shortage of memory in a VM is observed by its swap activities. We implemented a working set prober (WS Prober) on each VMM to estimate the working set sizes of VMs running on it. The statistics collected at each PM are forwarded to the central usher controller where our VM scheduler runs. The VM Scheduler is invoked periodically and receives from the LNM the resource demand history of VMs, the capacity and the load history of PMs, and the current layout of VMs on PMs.

The scheduler has several components. The predictor is to predict the future resource needs of VMs and load in the future usage of PMs based on previous statistics. The Analyzer distinguishes between, hot spot solver in our VM Scheduler, i.e detected if the resource usage of any PM is above the hot threshold. If so, then few VMs running on them will be migrated away to reduce the load. The cold spot solver checks if the average utilization of APMs is below the cold threshold or green computing threshold. If so, then some of those PMs could be turned off to save energy. It identifies the set of PMs whose utilization is below the cold threshold and then attempts to migrate away all their VMs. It then compiles a migration list of VMs and passes it to the Central Usher Controller for execution.

B. Load Prediction Algorithm

We can 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

ISSN: 2278 – 1323

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 E(t) and observed load O(t) at particular time t..

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

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

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

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

where *r* is the average utilization of all resources for server *p*. By minimizing the *skewness*, we can able to combine different kinds of workloads and hence improve the overall utilization of server resources.

D. Classification

The evaluation of resource allocation status based on the predicted future resource demands of VMs leads to,

- Hot Spot: A server is defined as a hot spot if the utilization of any of its resources is above a hot threshold. It means that the server is overloaded and hence some VMs running on it should be migrated away.
- Warm Spot: A server in a level of resource utilization that is sufficiently high for running but not so high as to risk of becoming a hot spot.
- Cold Spot: A server is defined as a cold spot if the utilizations of all its resources are below a cold threshold. It means that the server is mostly idle/under utilized and it may be turn off to save energy.

E. Hot Spot Mitigation

The purpose is to eliminate the all hot spots or 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 of temperature. VM with the highest temperature should be first migrated away. Destination server 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.

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.

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.

The two scenarios to be considered in hot spot mitigation are, (i) the VMs running in identified hot spots are migrated to warm spot servers which will not become hot by accommodating the VMs. (ii) 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.

F. Green Computing Algorithm

The main goal of green computing algorithm is to reduce the number of active servers during low load and save energy by turning off the servers when the resource utilization of servers is too low. The green computing algorithm is invoked when the average utilizations of all resources on active servers are below the green computing threshold.

The green computing algorithm works as followed,

Step 1: Sort the list of cold spots in the system based on the ascending order of their memory size.

Step 2: For each VM on a cold spot, try to find a destination server to accommodate it. The resource utilization of the destination server after accepting the VM must be below the warm threshold.

Step 3: If satisfied then VMs will be migrated to other server and update the list of cold spots and load on the related servers. After migration switch-off the source server.

Step 4: If not, do not migrate any of its VMs.

5. SIMULATIONS AND RESULTS

The performance of algorithm is evaluated using trace driven simulations. The default parameters used in the simulation are shown in Table 1.

Table 1
Parameters in our Simulation

symbol	meaning	value
h	hot threshold	0.9
c	cold threshold	0.25
w	warm threshold	0.65
g	green computing threshold	0.4

ISSN: 2278 – 1323

A. Impact of Thresholds on APMs

The effect of various threshold used in algorithm is evaluated as shown in the figure 5A. The bottom part of the figure shows the daily load variation in the system. The CPU load decreases after midnight, the memory consumption is fairly stable and network utilization stays very low. The top part of the figure shows the percentage of APMs varies with the load for different thresholds in algorithm.

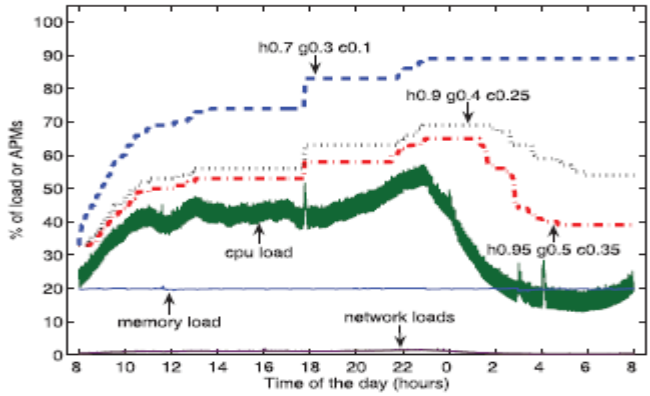

Figure 5A. Impact of thresholds on APMs

B. Impact of Load Prediction

We compare the execution of our algorithm with and without load prediction as shown in figure 5B. Without the load prediction, the algorithm uses the last observed load in its decision making. Using load prediction, it reduces the average number of hot spots during a decision run.

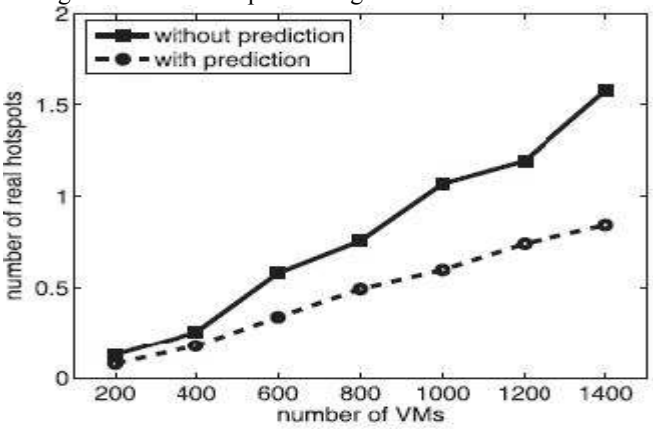

Figure 5B(i). Average number of hot spots

The number of migrations in the system with load prediction is less than that without load prediction as it avoids the unnecessary migrations due to temporary load fluctuation occurs without prediction.

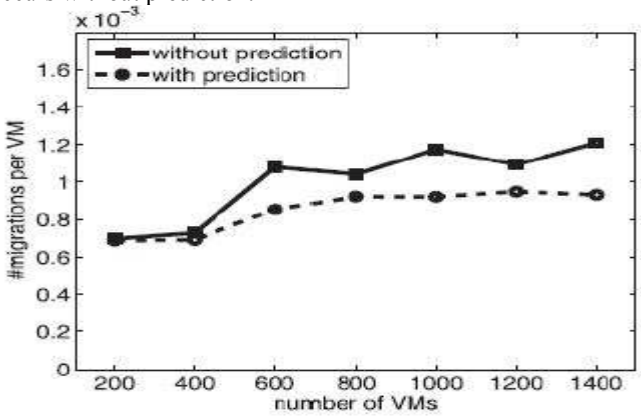

Figure 5B(ii). Number of migrations

The average number of APMs remains same with and without load prediction.

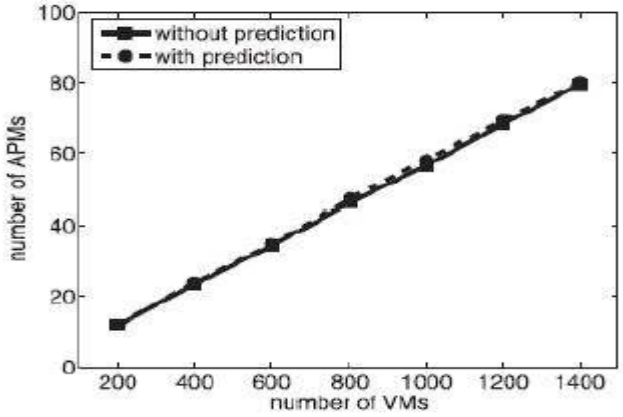

Figure 5B(iii). Average number of APMs

C. Measure of Scalability of Algorithm

The scalability of the algorithm can be measured by varying the number of VMs in the simulation as ratio of VM to PM is 10:1.

Figure5C(i) shows that the average decision time of our algorithm increases with the system size. The decision time is of two parts: hot-spot mitigating time and cold-spot mitigating time. The decision time for synthetic workload is higher than the real trace.

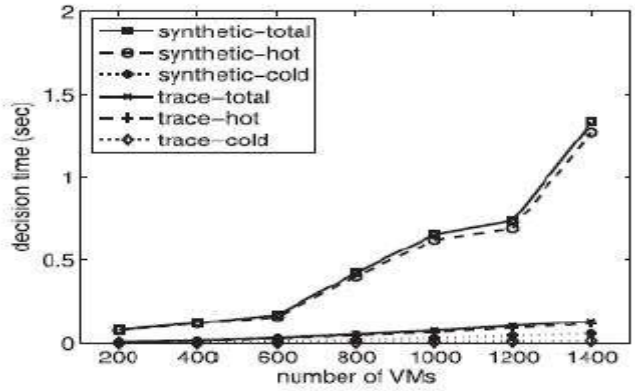

Figure 5C(i). Average Decision Time

Figure 5C(ii) shows that average number of migrations is increases linearly with the system size. The much number of migrations occurs due to hot-spots. The number of migrations for synthetic workload is higher than the real trace.

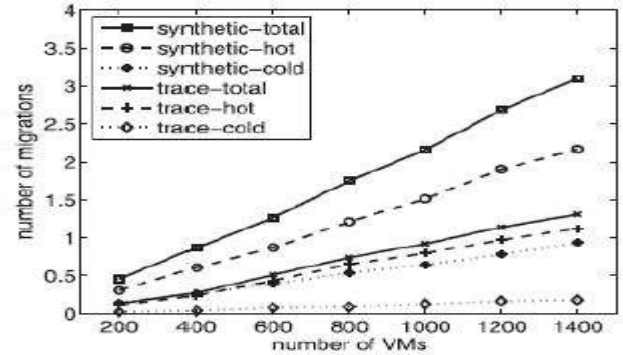

Figure 5C(ii). Average Number of migrations

ISSN: 2278 – 1323

Figure 5C(iii) shows the average number of migrations per VM in each decision run is constant.

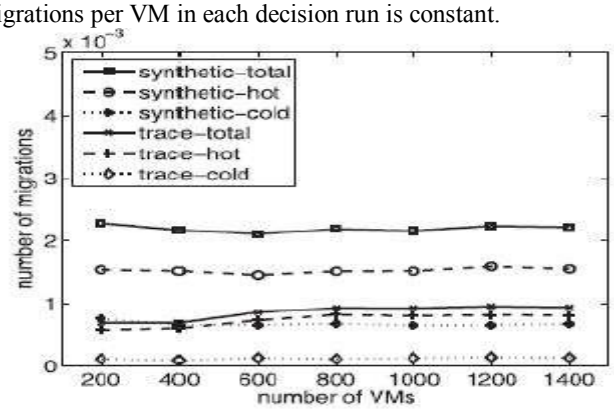

Figure 5C(iii). Number of migrations per VM

D. Effectiveness of the Algorithm

Consider three PMs and five VMs in a system. When overload occurred in PM1, our algorithm migrates any of the VM from PM1 to PM2or PM3 to resolve the overload.

At 480sec, VM3 migrated from PM1 to PM3 and then at 1500sec it is released to PM2.

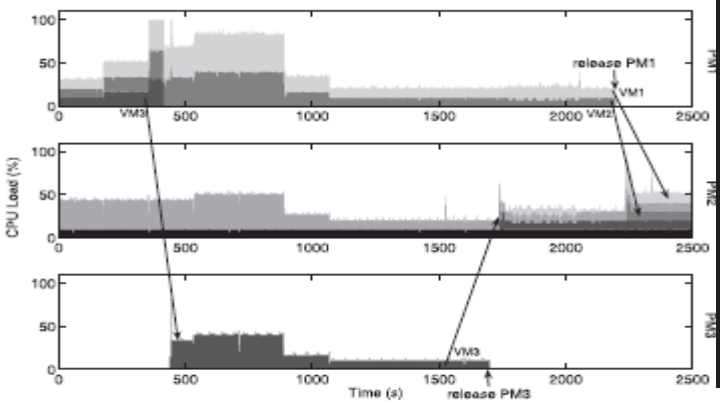

Figure 5D. Effectiveness of Algorithm

E. Resource Balance

The main goal of the skewness is to mix the workload with different resource requirements together so that the overall utilization of server capacity is improved.

The figure 5E shows the usage of CPU and network dimensions in PMs. Migration occurred due to the CPU overload and problem with the network dimensions.

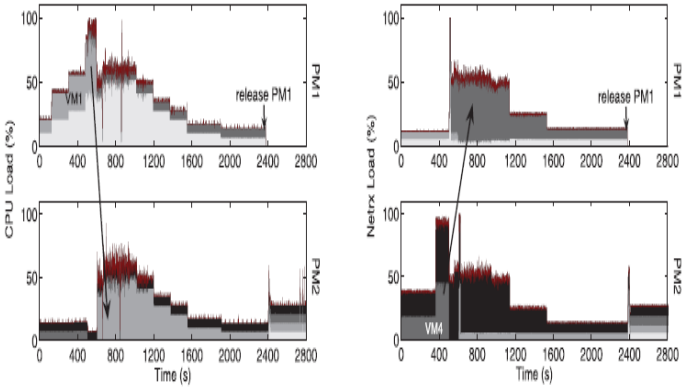

Figure 5E. Resource Balance

F. Overall Resource Utilization

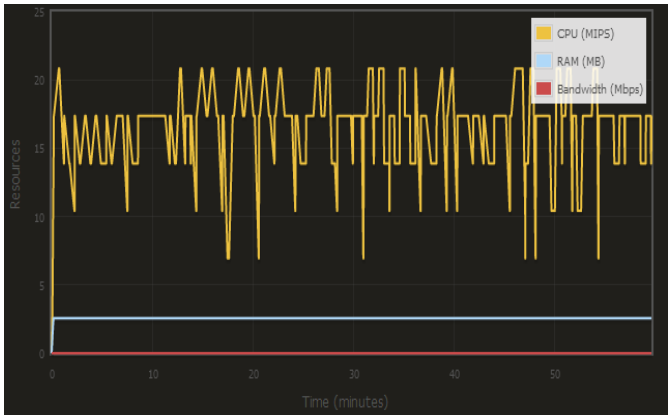

Figure 5F. Overall resource utilization on selected datacenter

G. Overall Power Consumption

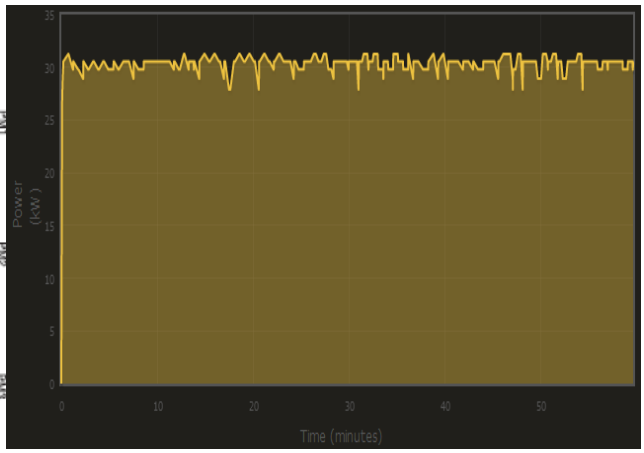

Figure 5G. Overall power consumption on selected datacenter

6. CONCLUSIONS & FUTURE WORK

We have presented the design, implementation, and evaluation of a resource provision system for cloud computing services. We implemented load prediction algorithm that can avoid the occurrences of under provisioning and over provisioning by allocating the resources dynamically. The overall utilization of server resources can be improved by minimizing skewness. The proposed system can also 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 it can be extended as to divide the whole process of VMs to instances and perform the load balancing and reduce the energy required.

7. 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

ISSN: 2278 – 1323

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] 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

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

[6] 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.

[7] 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.

[8] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R.Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," Proc. ACM Symp. Operating Systems Principles (SOSP '03), Oct. 2003.

[9] 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.

[10] Zhen Xiao, " Dynamic Resource Allocation Using Virtual Machines for Cloud Environment", IEEE transactions on Parallel and Distributed Systems, Vol. 24, No. 6, June 2013

ISSN: 2278 – 1323