

Comparison of Energy Aware Load Balancing Algorithms In Cloud Computing

Mr. Jayant Adhikari,
Department of M. Tech (Computer Science and
Engineering) Tulsiramji Gaikwad-Patil,
adhikari.jayant@gmail.com

Prof. Sulabha Patil,
College of Engineering and Technology, RTMNU,
Nagpur, sulabhavpatil@gmail.com

Abstract— Cloud Computing has widely been adopted by the industry or organization though there are many existing issues like Load Balancing, Virtual Machine Consolidation, Energy Management, etc. which have not been fully implemented. Central to these issues is the issue of load balancing, that is required to distribute the excess dynamic local workload equally to all the nodes in the whole Cloud to achieve a high user satisfaction. We propose DT-PALB (Double Threshold Power Aware Load Balancing) method to deploy the virtual machines for power saving purpose. DT-PALB is an extension from original PALB (Power Aware Load Balancing). Simulations and experiments have been conducted to verify our algorithms. The average power consumption is used as performance metrics and the result of PALB is used as baseline. Results show that DT-PALB we proposed can reduce the number of power-on physical machine and average power consumption compare to other deploy algorithms with power saving.

Index Terms— Cloud Computing, Virtual machine, Consolidation, Energy-Aware Scheduling, Load Balancing

I. INTRODUCTION

Load balancing is one of the central issues in cloud computing [3][7]. It is a mechanism that distributes the dynamic local workload evenly across all the nodes in the whole cloud to avoid a situation where some nodes are heavily loaded while others are idle or doing little work. It helps to achieve a high user satisfaction and resource utilization ratio, hence improving the overall performance and resource utility of the system.

Local cloud implementations are becoming popular due to the fact that many organizations are reluctant to move their data to a commercialized cloud vendor. There are several different implementations of open source cloud software that organizations can utilize when deploying their own private cloud. Some possible solutions are OpenNebula [18] or Nimbus [20] or cloudbus[19]. This architecture is built for ease of scalability and availability, but does not address the problem of the amount of power a typical architecture like this consumes.

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. 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. To address this problem and drive Green Cloud computing, data center resources need to be managed in an energy-efficient manner. In particular, 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 main objective of this work is to present our vision, discuss open research challenges in energy-aware resource management, and develop efficient policies and algorithms for virtualized data centers so that Cloud computing can be a more sustainable and eco-friendly mainstream technology to drive commercial, scientific, and technological advancements for future generations.

The rest of this paper is organized as follows. Section II related energy-aware scheduling algorithms in clouds. Section III describes energy-aware cloud architecture. Section IV presents our power model in cloud simulator and section V describe our double threshold power aware load balancing algorithm. Section VI describes experiment setup. Section VI describes result and Section VIII conclusion and future works.

II. RELATED WORKS

Ching-Chi Lin et. al [3] is proposed as an extension to the Round-Robin method. Dynamic Round-Robin method uses two rules to help consolidate virtual machines. The first rule is that if a virtual machine has finished and there are still other virtual machines hosted on the same physical machine, this physical machine will accept no more new virtual machine. Such physical machines are referred to as being in "retiring" state, meaning that when the rest of the virtual machines finish their execution, this physical machine can be shutdown. The second rule of *Dynamic Round-Robin* method is that if a physical machine is in the "retiring" state for a sufficiently long period of time, instead of waiting for the residing virtual machines to finish, the physical machine will be forced to migrate the rest of the virtual machines to other physical machines, and shutdown after the migration finishes. This waiting time threshold is denoted as "retirement threshold". A physical machine that is in the retiring state but cannot finish all virtual machines after the retirement threshold will be forced to migrate its virtual machines and shutdown.

Jeffrey M. Galloway et al proposed an algorithm[4] has three basic sections. The balancing section is responsible for determining here virtual machines will be instantiated. It does this by first gathering the utilization percentage of each active compute node. In the case that all compute nodes n are above 75% utilization, PALB instantiates a new virtual machine on the compute node with the lowest utilization number. It is worth mentioning in the case where all compute nodes are over 75% utilization, all of the available compute nodes are in operation. Otherwise, the new virtual machine (VM) is booted on the compute node with the highest utilization (if it can accommodate the size of the VM). The upscale section of the algorithm is used to power on additional compute nodes (as long as there are more available compute nodes). It does this if all currently active compute nodes have utilization over 75%. The downscale section is responsible for powering down idle compute nodes.

Jaspreet kaur [5] proposed the cloud manager estimates the job size and checks for the availability of the virtual machine and also the capacity of the virtual machine. Once the job size and the available resource (virtual machine) size match, the job scheduler immediately allocates the identified resource to the job in queue.

Y. Lua et al. [14] proposed a Join- Idle-Queue load balancing algorithm for dynamically scalable web services. This algorithm provides large scale load balancing with distributed dispatchers by, first load balancing idle processors across dispatchers for the availability of idle

processors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. By removing the load balancing work from the critical path of request processing, it effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time.

Wenhong Tian[11] introduce a dynamic and integrated resource scheduling algorithm (DAIRS) for Cloud datacenters. Unlike traditional load-balance scheduling algorithms which consider only one factor such as the CPU load in physical servers, DAIRS treats CPU, memory and network bandwidth integrated for both physical machines and virtual machines.

R. Yamini [6][7] proposed two algorithm i.e ECTC and MaxUtil follow similar steps in algorithm description with the main difference being their cost functions. In a nutshell, for a given task, two heuristics check every resource and identify the most energy efficient resource for that task. The evaluation of the most energy efficient resource is dependent on the used heuristic, or more specifically the cost function employed by the heuristic. The cost function of ECTC computes the actual energy consumption of the current task subtracting the minimum energy consumption required to run a task if there are other tasks running in parallel with that task. That is, the energy consumption of the overlapping time period among those tasks and the current task is explicitly taken into account.

III. ENERGY-AWARE CLOUD ARCHITECTURE

It is assumed that physical servers are equipped with multi-core CPUs. A multi-core CPU with n cores each having m MIPS is modeled as a single-core CPU with the total capacity of nm MIPS. This is justified since applications, as well as VMs are not tied down to processing cores and can be executed on an arbitrary core using a time-shared scheduling algorithm shown in Figure 1. The only limitation is that the capacity of each virtual CPU core allocated to a VM must be less or equal to the capacity of a single physical CPU core. The reason is that if the CPU capacity required for a virtual CPU core is higher than the capacity of a single physical core, then a VM must be executed on more than one physical core in parallel. However, automatic parallelization of VMs with a single virtual CPU cannot be assumed.

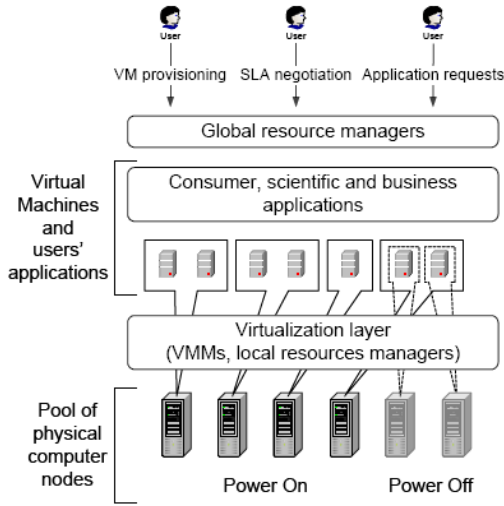


Figure 1: The system view

The software layer of the system is tiered comprising local and global managers (Figure 1). The local managers reside on each node as a module of the VMM. Their objective is the continuous monitoring of the node's CPU utilization, resizing the VMs according to their resource needs, and deciding when and which VMs should to be migrated from the node. The global manager resides on the master node and collects information from the local managers to maintain the overall view of the utilization of resources. The global manager issues commands for the optimization of the VM placement. VMMs perform actual resizing and migration of VMs as well as changes in power modes of the nodes.

IV. POWER MODEL

Power consumption by computing nodes in data centers is mostly determined by the CPU, memory, disk storage and network interfaces. In comparison to other system resources, the CPU consumes the main part of energy, and hence in this work we focus on managing its power consumption and efficient usage.

Algorithmic approaches:- It has been experimentally determined that an ideal server consumes[11] about 70% of the power utilized by a fully utilized server. (See figure 2). Moreover, the CPU utilization is typically proportional to the overall system load. Recent studies [17][13] have shown that the application of DVFS on the CPU results in almost linear power-to-frequency relationship for a server. The reason lies in the limited number of states that can be set to the frequency and voltage of the CPU and the fact that DVFS is not applied to other system components apart from the CPU.

Moreover, these studies have shown that on average an idle server consumes approximately 70% of the power consumed by the server running at the full CPU speed.

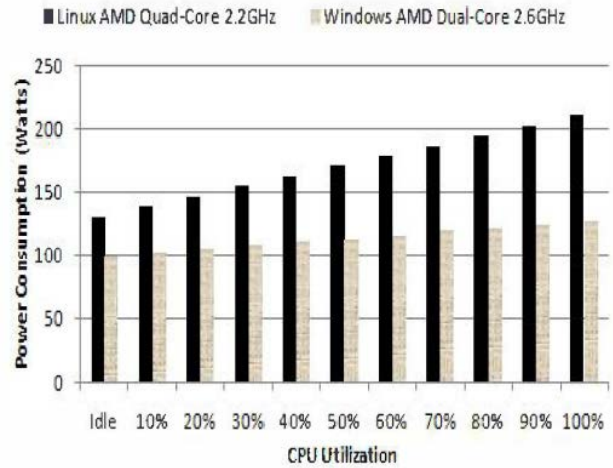


Figure 2: Power Model

This fact justifies the technique of switching idle servers to the sleep mode to reduce the total power consumption. Therefore, in this work we use the power model defined in [17].

$$P(u) = k \cdot P_{max} + (1 - k) \cdot P_{max} \cdot u \quad \dots\dots\dots(1)$$

P_{max} is the maximum power consumed when the server is fully utilized; k is the fraction of power consumed by the idle server (i.e. 70%); and u is the CPU utilization. For our experiments P_{max} is set to 250 W, which is a usual value for modern servers. For example, according to the SPEC power benchmark, for the fourth quarter of 2010, the average power consumption at 100% utilization for servers consuming less than 1000W was approximately 259 W.

The utilization of the CPU may change over time due to the workload variability. Thus, the CPU utilization is a function of time and is represented as $u(t)$. Therefore, the total energy consumption by a physical node (E) can be defined as an integral of the power consumption function over a period of time as shown in [17].

$$E = \int_{t_0}^{t_1} P(u(t))dt. \quad \dots\dots\dots(2)$$

V. DT-PALB ALGORITHM

Our algorithm is intended to be used by organizations wanting to implement small to medium sized local clouds. This algorithm should scale to larger sized

clouds because one of the main contributions of the cluster controller is load balancing compute nodes.

All of the computation included in this algorithm is maintained in the cluster controller. The cluster controller maintains the utilization state of each active compute node and makes decisions on where to instantiate new virtual machines.

Algorithm: DT-PALB

balance:

```
for all active compute nodes  $j \in [m]$  do
     $n_j \leftarrow$  current utilization of compute node  $j$ 
end for
for all  $n_j < 90\%$  and  $n_j > 40\%$  utilization
    boot vm on most underutilized  $n_j$ 
end if
if  $n_j < 40\%$  utilization
    migrate vmj to most underutilized  $n_j$ 
```

upscale:

```
if each  $n_j > 90\%$  utilization
    if  $n_j < m$ 
        boot compute node  $n_{j+1}$ 
    end if
end if
```

downscale:

```
if vmi idle or user initiated shutdown
    shutdown vmi
end if
if  $n_j$  has no active vm
    shutdown  $n_j$ 
end if
```

The DT-PALB [2] algorithm has three basic sections. The balancing section is responsible for determining where virtual machines will be instantiated. It does this by first gathering the utilization percentage of each active compute node. In the case that all compute nodes n are above 90% utilization, PALB instantiates a new virtual machine on the compute node with the lowest utilization number. It is worth mentioning in the case where all compute nodes are over 90% utilization, all of the available compute nodes are in operation. Otherwise, the new virtual machine (VM) is booted on the compute node with the highest utilization (if it can accommodate the size of the VM). If any nodes have utilization in between 90% and 40% then find most underutilized node and assign vm to that node. If any nodes have less than 40% utilization then migrate vm to other node.

The upscale section of the algorithm is used to power on additional compute nodes (as long as there are

more available compute nodes). It does this if all currently active compute nodes have utilization over 90%.

The downscale section is responsible for powering down idle compute nodes. If the compute node is using less than 40% of its resources, PALB sends a shutdown command to that node.

VI. EXPERIMENT SETUP

We have simulated a data center comprising 10-30 heterogeneous physical nodes. Each node is modeled to have one CPU core with the performance equivalent to 1000, 2000 or 3000 MIPS, 8 GB of RAM and 1 TB of storage. Power consumption by the hosts is defined according to the model described in Section VII. According to this model, a host consumes from 175 W with 0% CPU utilization, up to 250W with 100% CPU utilization. Each VM requires one CPU core with 250, 500, 750 or 1000 MIPS, 128 MB of RAM and 1 GB of storage. The users submit requests for provisioning of 10-100 heterogeneous VMs that fill the full capacity of the simulated data center. Each VM runs a web-application or any kind of application with variable workload, which is modeled to generate the utilization of CPU according to a uniformly distributed random variable. The application runs for 150,000 MI that is equal to 10 min of the execution on 250 MIPS CPU with 100% utilization. Initially, the VMs are allocated according to the requested characteristics assuming 100% CPU utilization. Each experiment has been run 10 times.

VII. EVALUATION AND RESULTS

A. Simulation results

To evaluate the double-threshold power aware policies it is necessary to determine the best values for the thresholds in terms of the energy consumption and QoS delivered. We have simulated our policy varying the absolute values of the lower and upper thresholds as well as the interval between them. First of all, it is important to determine which threshold has higher influence on the energy consumption. Using the graph we have analysis the relationship between the energy consumption and values of the utilization thresholds. The values of the result show that the lower threshold has higher influence on the energy consumption than the upper threshold.

This can be explained by the fact that an increase of the lower threshold eliminates the low utilization of the resources leading to higher energy savings; however, possibly increasing the number of VM migrations and SLA violations. The results showing the mean energy

consumption achieved using the double-threshold power aware (DTPALB) policy for different values of the lower utilization threshold and the interval between the thresholds are presented in Fig. 3. The graph shows that an increase of the lower utilization threshold leads to decreased energy consumption. However, the low level of energy consumption can be achieved with different intervals between the thresholds. Therefore, to determine the best interval we have to consider another factor – the level of SLA violations.

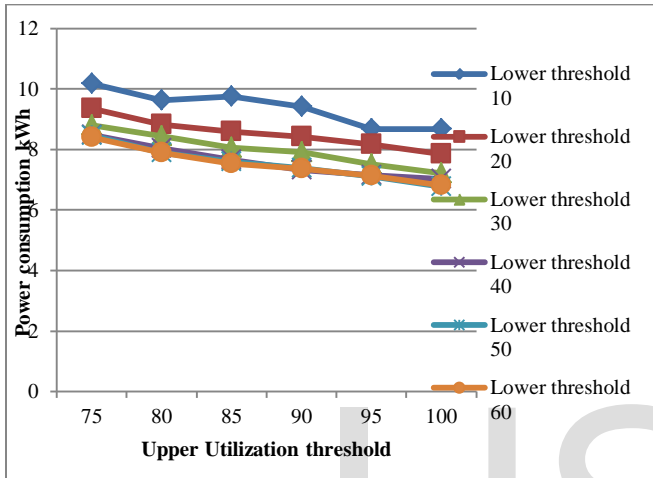


Figure 3: Power consume by DTPALB using different threshold level.

B. Comparison with PALB scheduling

We measure the power consumptions of the PALB [2] scheduling algorithms and compare the results with our proposed algorithms. Since PALB do not migrate VMs, the number of powered-on physical machines in PALB is equivalent to the number of physical machines required to process all incoming virtual machines. Figure 4 and Figure 5 shows average number of powered-on physical machines and total power consumption from the two scheduling methods. The two algorithms include one of our proposed algorithms DTPALB and PALB. From Figure 4 and Figure 5, observation can be made that the number of powered-on physical machines and the power consumption reduce significantly using our algorithms, when compared with PALB.

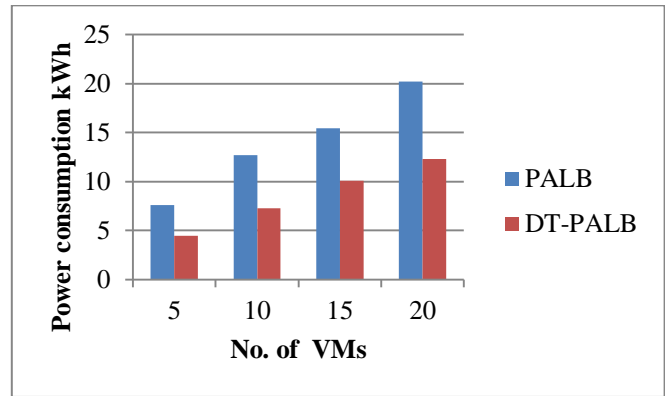


Figure 4: Power consume by PALB and DTPALB with 10 hosts

In comparison in Figure 4, when the cloud has 10 compute nodes and 5 small virtual machines are requested, DT-PALB consumes 58.60% of the energy consumed for PALB with the same parameters. When requesting 10 small virtual machines while having 10 available compute nodes, DT-PALB only uses 57.42% of the energy consumed by PALB. Using the requests for 15 virtual machines and 10 available compute nodes, DT-PALB uses 65.43% of the energy consumed by PALB. Requesting 20 virtual machines with 10 compute nodes available, DT-PALB consumes 60.91% of the energy used by PALB. As can be observed from Figure 4, the DT-PALB reduces the average number of powered on physical machines by 39.40%.

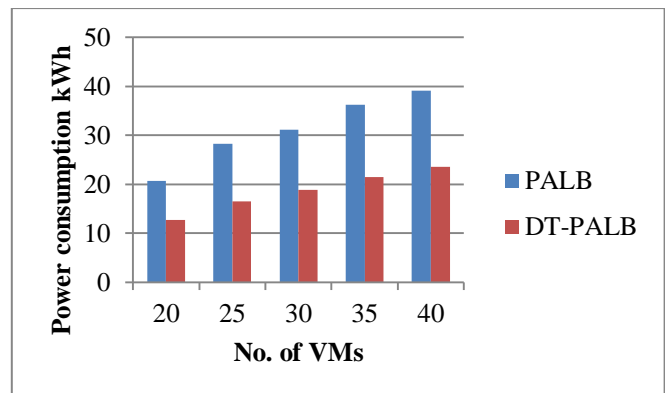


Figure 5: Power consume by PALB and DTPALB with 30 hosts

In Figure 5, when the cloud has 30 compute nodes and 20 small virtual machines are requested, DT-PALB consumes 61.65% of the energy consumed for PALB with the same parameters. When requesting 25 small virtual machines DT-PALB only uses 58.23% of the energy consumed by PALB. Using the requests for 30 virtual machines DT-PALB uses 60.72% of the energy consumed by PALB. Requesting 35 virtual machines with 30 compute nodes available, DT-PALB

consumes 59.33% of the energy used by PALB. Requesting 40 virtual machines with 30 compute nodes available, DT-PALB consumes 60.23% of the energy used by PALB. As can be observed from Figure 5, the DT-PALB reduces the average number of powered on physical machines by 39.98%.

VIII. CONCLUSION AND FUTURE WORKS

Cloud Computing has widely been adopted by the industry or organization though there are many existing issues like Load Balancing, Virtual Machine Consolidation, Energy Management, etc. which have not been fully implemented. Central to these issues is the issue of load balancing, that is required to distribute the excess dynamic local workload equally to all the nodes in the whole Cloud to achieve a high user satisfaction. We propose DT-PALB method to deploy the virtual machines for power saving purpose. DTPALB is an extension from original PALB.

Experiments conducted on 10 and 30 nodes test bed have shown that DTPALB is able to reduce energy consumption by the compute nodes by up to 40% with a limited application performance impact as shown in previous section. From the presented results we can conclude that the usage of the DTPALB policy provides the best energy savings with the least SLA violations and number of VM migrations among the evaluated policies for the simulated scenario. Moreover, the results show the flexibility of the DTPALB algorithm, as the thresholds can be adjusted according to the SLAs requirements.

Our future work will include implementing the new energy aware load balancing algorithm for local cloud. We will also be implementing load balancing algorithms including three level thresholds for virtual machine consolidation. Also, keeping to the energy savings load balancing mindset, we will be studying the effects of persistent storage load balancing across multiple storage nodes.

References

[1]. Mr. Jayant Adhikari, Prof. Sulabha Patil, "Load Balancing The Essential Factor In Cloud Computing", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 1 Issue 10, December- 2012.

[2]. Mr. Jayant Adhikari, Prof. Sulabha Patil, "Double Threshold Energy Aware Load Balancing In Cloud Computing", 4th IEEE International Conference on Computing, Communication and Networking Technology, 4-6 July 2013.

[3]. Ching-Chi Lin, Pangfeng Liu, Jan-Jan Wu. "Energy-Efficient Virtual Machine Provision Algorithm for Cloud System", IEEE 4th International Conference on Cloud Computing, 81-88, 09/2011.

[4]. Jeffrey M. Galloway, Karl L. Smith, Susan S. Vrbsky. "Power Aware Load Balancing for Cloud Computing", Proceedings of the World Congress on Engineering and Computer Science 2011 Vol. IWCECS 2011, October 19-21, San Francisco, USA.

[5]. Jaspreet kaur "Comparison of load balancing algorithms in a Cloud" 2012, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, pp.1169-1173.

[6]. R. Yamini, "Power Management In Cloud Computing Using Green Algorithm", IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM-2012), March 30, 31, 2012, pp-128-133.

[7]. R. Yamini, "Energy Aware Green Task Assignment Algorithm In Clouds", International Journal For Research In Science And Advance Technology, Issue-1, Volume-1, pp-23-29.

[8]. Anton Beloglazov, Rajkumar Buyya, "Managing Overload Host For Dynamic Consolidation Of Virtual Machines Cloud Data Centers Under Quality Of Service Constraints", IEEE Transaction On Parallel And Distributed Systems, 2012.

[9]. Zehua Zhang, Xuejie Zhang, "A Load Balancing Mechanism Based On Ant Colony And Compel Network Theory In Open Cloud Computing Federation", IEEE-International Conference On Automation, May 2010, pp-240-243.

[10]. Makhlof Hadji, Djamel Zeglache, "Minimum Cost Maximum Flow Algorithm For Dynamic Resource Allocation In Cloud", IEEE-Fifth International Conference In Cloud Computing, Aug-2012, pp-876-882.

[11]. Wenhong Tian, Yong Zhao, Minxian Xu, Chen Jing, "A Dynamic And Integrated Load Balancing Scheduling Algorithm For Cloud Data Center", Proceeding of IEEE CCIS Feb 2011, pp-311-315.

[12]. Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid "Availability and Load Balancing in Cloud Computing", 2011 International Conference on Computer and Software Modeling IPCSIT vol.14, IACSIT Press, Singapore.

[13]. Rajkumar Buyya, Anton Beloglazov, and Jemal Abawajy, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges", Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010), Las Vegas, USA, July 12-15, 2010.

[14]. Trieu C. Chieu, Hoi Chan, "Dynamic Resource Allocation Via Distributed Decisions In Cloud Environment", Eight IEEE International Conference on e-Business Engineering, Sept-2011, pp-125-130.

[15]. Sivadon Chaisiri, Bu-Sung Lee, "Optimization Of Resource Provisioning Cost In Cloud Computing", IEEE transaction on services computing, vol. 5, No. 2 June 2012

[16]. Ayman G. Fayoumi, "Performance Evaluation Of A Cloud Based Load Balancer Severing Pareto Traffic", Journal of Theoretical and Applied Information Technology, 15th October 2011. Vol. 32 No.1

[17]. Anton Beloglazov and Rajkumar 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 (MGC 2010, ACM Press, New York, USA), In conjunction with ACM/IFIP/USENIX 11th International Middleware Conference 2010, Bangalore, India, November 29 - December 3, 2010.

[18]. OpenNebula <http://opennibula.org/>

[19]. Cloudbus <http://www.cloudbus.org/>

[20]. Nimbus <http://www.nimbus.com/>