Power Aware Load Balancing for Cloud Computing

Jeffrey M. Galloway, Karl L. Smith, Susan S. Vrbsky

Abstract—**With the increased use of local cloud computing architectures, organizations are becoming aware of wasted power consumed by unutilized resources. In this paper, we present a load balancing approach to IaaS cloud architectures that is power aware. Since the cloud architecture implemented by local organizations tends to be heterogeneous, we take this into account in our design. Our Power Aware Load Balancing algorithm, PALB, maintains the state of all compute nodes, and based on utilization percentages, decides the number of compute nodes that should be operating. We show that our solution provides adequate availability to compute node resources while decreasing the overall power consumed by the local cloud by 70% - 97% compared to using load balancing techniques that are not power aware.**

Keywords- Distributed Computing, Load Balancing, Power Management, Virtualization

I. Introduction

Cloud computing architectures are becoming a dominant contender in the distributed systems paradigm. Using this architecture, customers are given access to resources provided by a cloud vendor as described in their Service Level Agreement (SLA). Clouds use virtualization technology in distributed data centers to allocate resources to customers as they need them. Generally, clouds are deployed to customers giving them three levels of access: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). The jobs can differ greatly from customer to customer. Each commercial vendor has a targeted customer and specific markets they wish to saturate.

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 debates on whether moving data to the public cloud would benefit small organizations. Beyond the question of benefit to the organizations utilizing public clouds, there are also issues with trust, security and legality. Some organizations may not trust a third party with their information and/or software. Other organizations may not be comfortable allowing a third party to be responsible for the security of the cloud.

Manuscript received July 5, 2011, revised August 15, 2011.

Jeffrey M. Galloway is with the Department of Computer Science at the University of Alabama, Tuscaloosa, AL, 35401 USA (phone: (205) 348-6363; e-mail: jmgalloway@crimson.ua.edu).

Karl L. Smith is with the Department of Computer Science at the University of Alabama, Tuscaloosa, AL, 35401 USA (e-mail: smith102@crimson.ua.edu).

Susan S. Vrbsky is an Associate Professor in the Department of Computer Science at the University of Alabama, Tuscaloosa, AL, 35401 (e-mail: vrbsky@cs.ua.edu).

Finally there are many organizations that work with data which they cannot legally store off site due to restrictions on the data. In cases such as these, the organization can opt to implement a cloud "in-house".

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 [1] or Nimbus [2]. However, the most common open source local cloud stack, and the one we will discuss primarily in this paper, is Eucalyptus [3], provided by Eucalyptus Systems, Inc. This is an IaaS cloud implementation that ultimately gives users virtual machines to undefined job types. A typical Eucalyptus cloud is composed of a front-end cloud controller, a cluster controller for controlling compute nodes, a virtual machine image repository, a persistent storage controller, and many compute nodes. 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.

Organizations that wish to build local clouds do so using commodity hardware. This may mean that the cloud is made up of several different hardware set ups. Even when a cloud is initially built using one type of hardware, the nature of a cloud often means it will be expanded by adding new and different hardware throughout the course of its lifetime. In terms of scalability, the amount of compute nodes will generally increase rapidly over time. Given this heterogeneous nature, the nodes used will have different energy consumption footprints. The administrative cloud components (cloud, cluster, and storage controllers) need to be continuously operating for users to access the resources provided by the compute nodes. This is not true of the compute nodes. Depending on the amount of requests given by users, it is not necessary to have all compute nodes operating at any given time.

Current research in the area of cloud computing load balancing focuses on the availability of resources. The specific load balancing approach depends on the type of resource offered. Since requests can be more specifically defined while using SaaS, the load balancing techniques used in this case may not be applicable to clouds offering IaaS architectures.

In this paper, we propose a load balancing algorithm that could be applied to the cluster controller of a local cloud that is power aware. This load balancer maintains the utilization of all compute nodes and distributes virtual machines in a way that is power efficient. The goal of this algorithm is to maintain availability to compute nodes while reducing the total power consumed by the cloud.

This paper is organized as follows. In section 2 we describe related work in load balancing cloud computing resources. Section 3 presents details of our cloud architecture. Section 4 describes the simulator design used to test our algorithm. Section 5 presents the experiments used with our simulator. Section 6 explains the results of our experiments. Section 7 gives our conclusions and future directions.

II. Related Work

Presently there are a number of studies involving load balancing of resources across distributed systems.

Research has been performed by [4] on load balancing of virtual machines using Eucalyptus. Their proposed solution involves moving live virtual machines across hosts. There has also been much research in towards different styles of load balancing, such as the three-level approach [5]. Another approach [6] suggested rather than minimizing power, there should be options to select different power or performance options per job.

Another area that has recently received a lot of research focus is the conservation of power in distributed computing. Many of these concepts can easily be adapted from one area of distributed computing, such as clusters or grids, to clouds. Some of the work that has been done here includes research on powering down nodes when they are not in use [7]. That research was expanded to include proper limits for when nodes should be powered up from an idle state and how long machines should remain idle [8]. There has also been work on allowing machines to enter and exit idle states faster [9].

Research has also been done in the area of low power cloud computing. One solution [10] consists of utilizing adaptive load prediction to allocate tasks to the optimal components within a datacenter. Another proposed solution [11] suggests utilizing diskless storage to reduce the energy consumption of the cloud. In our work, machines are turned completely off, instead of placed into a low power mode.

While server processors can provide a wide dynamic power range, the range of other components is much smaller, e.g. 25% for disk drives and 15% for networking switches [12]. Disk devices have a latency and energy penalty for transition from an inactive to an active mode. Nevertheless, recent work [13] [14] considers powering down disk storage during non-peak hours. In [13], a subset of the disks are powered down, but all data items are still available in the system. The goal is to have little impact on the performance of the system during normal operation. Virtual nodes and migration are utilized in [14] so that a small number of disks can be used without overloading them.

The work presented in this paper proposes a virtual machine load balancing technique that is power efficient. Depending on the virtual machine request size, it is placed on a compute node that is powered on if that compute node has the resources to host it. Otherwise, a compute node that is powered off will be switched on to host the virtual machine. We are specifically interested in the power savings from switching off unused compute nodes. Our simulator takes incoming job requests in the form of virtual machines, load balances them across compute nodes, and returns the power consumed across those compute nodes over a given time period.

III. Cloud Architecture

There are five main components to the Eucalyptus cloud infrastructure.

1. Cloud Controller – Front-end interface for users. Knows overall status of cloud resources and controllers.
2. Cluster Controller – Administers networking resources and compute nodes. PALB algorithm will be implemented here.
3. Walrus Controller – Storage for virtual machine images to use by compute nodes.
4. Storage Controller – Persistent storage device which can be mounted inside active virtual machines.
5. Compute Nodes – Provides the execution environment for virtual machines in the cloud.

Scalability in terms of computing resources in the cloud generally comes from additional compute nodes. Administrative nodes have to be continuously powered on for the cloud to operate. These areas can also be scaled, but since they are required to be operational at all times, we do not consider them in the power savings analysis. We will be focusing on the compute nodes to deliver costs savings while keeping availability high.

Given the heterogeneous nature of local cloud architectures, each of the servers used to create the cloud could have different power consumption footprints. To take this into consideration, we profiled 15 different commodity machines. These were typical mid-tower desktop grade machines aged from 3 months to 5 years old. Each machine had a single hard drive for consistency.

IV. Simulator Design

This IaaS cloud simulator was written for ease of deployment of multiple load balancing techniques used for comparison. This software was written and executed on an actual compute node built specifically for our cloud.

A. Job Scheduler

The job scheduler is used to simulate requests from users for virtual machine instances. Unlike clusters and grids, clouds have sporadic job schedules. We chose a schedule that closely resembles requests for cloud resouces.

Jobs in this experiment come as requests from users for virtual machines. There are five different sized virtual machines and users are not limited in the size or number of instances they can request. An error is returned to the user if there is not enough space to host a requested virtual machine.

B. 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 PALB
balance:
    for all active compute nodes $j \in [m]$ do
        $n_j \leftarrow$ current utilization of compute node j
    end for
    if all $n_j > 75\%$ utilization   //all available nodes are active
        boot vm on most underutilized $n_j$
    end if
    else
    boot vm on most utilized $n_j$
    end else
upscale:
    if each $n_j > 75\%$ utilization
        if $n_j < m$
            boot compute node $n_{j+1}$
        end if
    end if
downscale:
    if $vm_i$ idle > 6 hours or user initiated shutdown
        shutdown $vm_i$
    end if
    if $n_j$ has no active vm
        shutdown $n_j$
    end if
```

Figure 1: PALB algorithm

The PALB 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 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 threshold of 75% utilization was chosen since when 25% of the resources are available, at least one more virtual machine can be accommodated using three out of five available configurations.

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. If the compute node is using less than 25% of its resources, PALB sends a shutdown command to that node.

C. Compute Nodes

In order to determine the power consumption to be used in our experiments, power consumption was measured from 15 commodity computers that could be used as compute nodes in a local cloud setup. A Watt-meter was used to gather data on these machines at idle and 100% utilization.

After the power profiling data was gathered from the different machines, they were averaged to give an approximate representation of the power requirement of a typical desktop machine. A scaling factor was then used to determine the power consumption at intermediate intervals between idle and 100% utilization. Figure 2 illustrates the power requirements needed by a single compute node in our cloud setup. The average power consumed by the commodity hardware tested at <5% utilization was 76.16 Watts. These machines consume an average of 121.5 Watts at 100% utilization, giving a range of 45.35 Watts. The power consumption scale was calculated by dividing the difference, 45.35 Watts, by 100 to find the energy used at any utilization level between 0% and 100%.

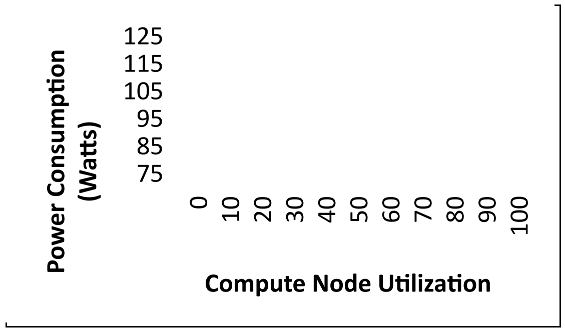

Figure 2: Power Requirements for Individual Compute Nodes

In our experiment, compute node utilization is composed of CPU and RAM used by virtual machines. Compute nodes are uniform dual-core CPU, 2 GB RAM machines. Although hard disk usage is included in our power measurements, hard disk space is not used in the computation of the utilization, since storage space far exceeds the need for the number of virtual machines that can be instantiated on these machines. KVM (Kernel-based Virtual Machine) [15] allows for 8 virtual machines per physical core. This experiment will allow for 6 virtual machines per core, giving 12 maximum virtual machines per compute node.

Accommodations were made for the underlying compute node operating system. Each compute node uses Ubuntu 10.10 Server 64-bit operating systems. The host operating system and Eucalyptus software (including the KVM hypervisor) requires approximately 21.25% of the resources in the compute node. This approximation was computed by observing the actual memory usage of these software components on our cloud nodes. The underlying operating system was determined to need 1/8th of the CPU resources and 30% of the RAM in a 2 GB system.

D. Watt-Meter

The Watt-Meter is used to calculate the power consumption of the compute nodes over a given time period. This meter calculates the power consumption for each node in parallel and gives an output of the current load in Watts, the average load in Kilowatt-hours (kWh) and the total power consumed during a given time span. Figure 3 shows the simulator design in detail.

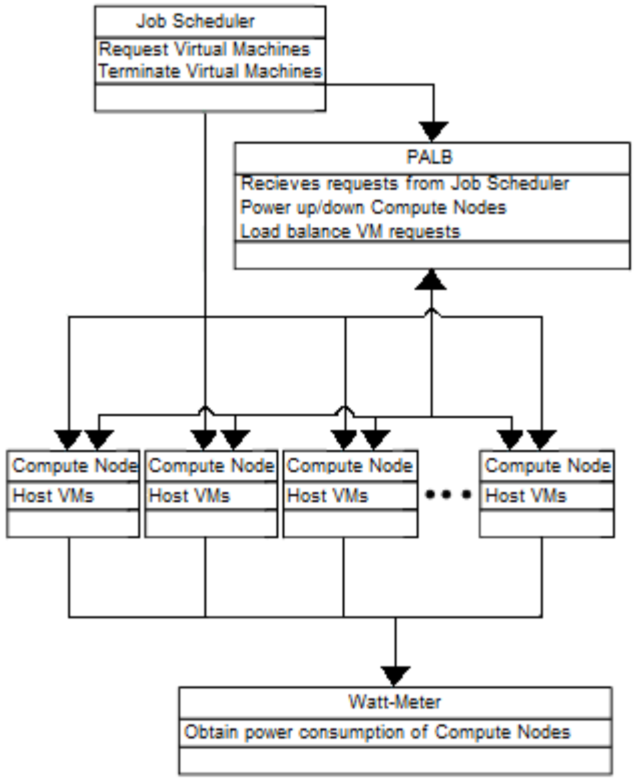

Figure 3: Simulator Design

E. *Virtual Machine Configuration*

As given in the Eucalyptus local cloud implementation, we allow the user to pick from five virtual machine sizes. The virtual machines will use Ubuntu 10.04 64-bit Server Edition with no graphical user interface. Since this is an IaaS cloud architecture, it is generally not known how the virtual machine will be used. In our study, we will assume a 100% utilization of the resources acquired by the virtual machines as long as they are operating.

In future studies, we will account for variable utilization percentages. In this design, a virtual machine will consume resources only on a single compute node. Table 1 gives the configurations options for the virtual machines in our cloud.

Table 1: Virtual machine configuration options

Vm types	Cpu	Ram	Disk	Utilization per instance
m1.small	**1**	**128**	**10**	**6.3%**
c1.medium	**1**	**256**	**10**	**12.5%**
m1.large	**2**	**512**	**20**	**25%**
m1.xlarge	**2**	**1024**	**20**	**37.5%**
c1.xlarge	**4**	**1536**	**30**	**62.5%**

V. Experiment

The experiment consisted of two similar job schedules. The first schedule consists of requests for up to 20 virtual machines. The second schedule requests up to 30 virtual machines. The job schedule distribution we used is common to requests of resources over a 24 hour period. Figure 4 shows the request schedule used in our experiment.

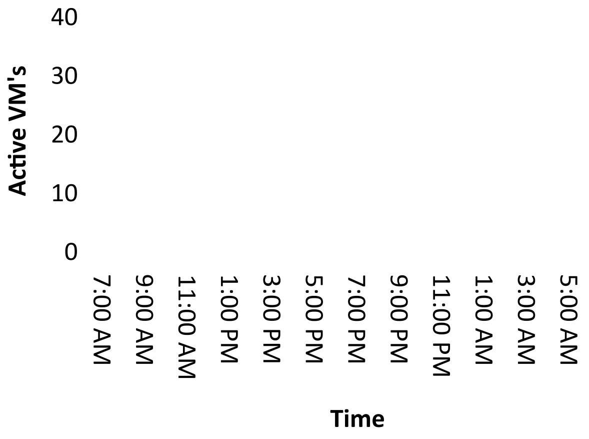

Figure 4: Virtual Machine Request Schedule

This job schedule was used in all of the runs in our experiment. Each run consisted of small, large, extra-large, or random sized virtual machines, and either 20 or 30 total unique virtual machine requests.

VI. Evaluation and Results

The following figures give the results of our experiment runs. As stated previously, only the compute nodes that are powered on are used to determine the total power consumed over the 24 hour period. The load balancer PALB is used to balance the virtual machines across the compute nodes in a way that conserves the most power.

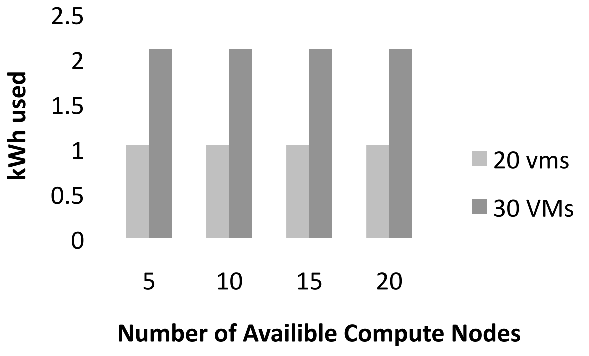

Figure 5: Comparison of small VM requests

Figure 5 gives the average kWh over a 24 hour period for a job schedule composed of small virtual machines. While requesting 20 virtual machines of this size, the power consumed remains constant at 1.04 kWh. This is due to the fact that the cloud can handle this number of requests using only two compute nodes. The remaining compute nodes in each run remain powered down. When requesting 30 small virtual machines, the cloud can accommodate this workload using only three compute nodes. The power consumed for requesting 30 small virtual machines is consistently 2.1kWh.

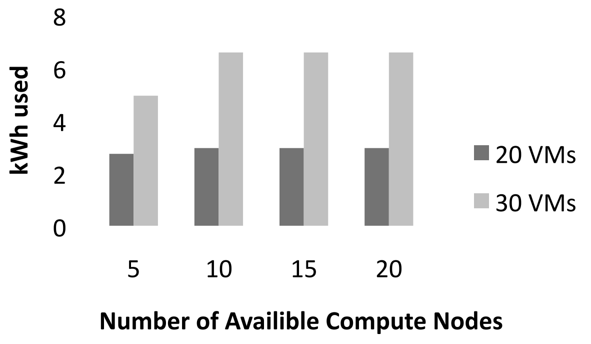

Figure 6: Comparison of large VM requests

Figure 6 shows the comparison of large sized virtual machine requests. Requesting 20 virtual machines of this size consumed 2.27 kWh with five compute nodes. The cloud could only accommodate for 15 requests of this size while having only five compute nodes. This means the remaining 5 requests were refused. The power increased to 2.95 kWh when the number of compute nodes increased to 10, 15, and 20. The cloud needed seven compute nodes to accommodate all 20 requests. The power consumed for 30 requests of large virtual machines was 4.93 kWh while having five compute nodes available. When the cloud had 10, 15, and 20 compute nodes available, the power consumption increased to 6.57 kWh. The power consumption remains the same since 30 requests of this size can be launched with 10 compute nodes.

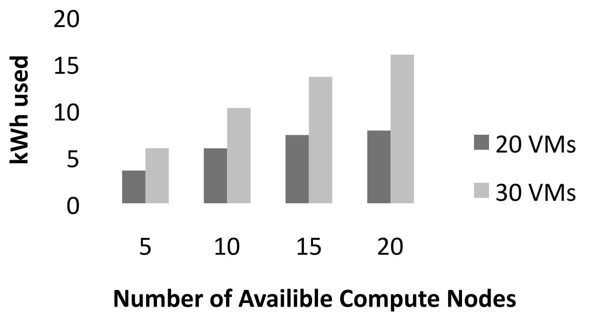

Figure 7: Comparison of extra-large VM requests

Figure 7 shows the power consumed when requesting virtual machines of extra-large size. While requesting 20 virtual machines of this size and a maximum compute node number of five, the power consumed was 3.5 kWh. When the maximum number of compute nodes was increased to 10, the power consumed was 5.88 kWh. At a maximum of 15 compute nodes, the power consumed was 7.3 kWh. When a maximum of 20 compute nodes, the power consumed increased to 7.78 kWh.

As assumed, the cloud compute nodes consumed the most power when the job scheduler demands 30 extra-large virtual machines over a 24 hour period. When the cloud is constrained to five compute nodes, it consumed 5.9 kWh. With 10 compute nodes, the power consumption was 10.2 kWh. With 15 compute nodes, the power increased to 13.53 kWh. Lastly, the highest recorded power consumption came when 30 extra-large virtual machines were requested with 20 compute nodes. The power consumed during this test was 15.9 kWh.

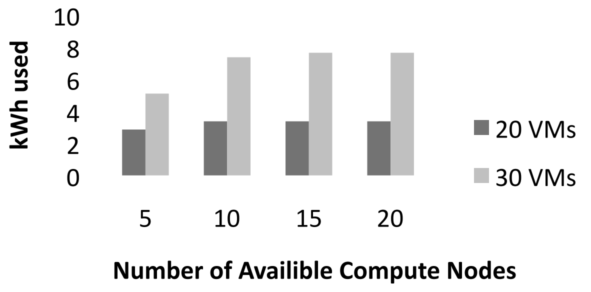

Figure 8: Comparison of random sized VM requests

Figure 8 shows the comparison of random sized virtual machine requests over the 24 hour period. Each new virtual machine request in this run is larger than the previous, returning to the smallest size once an extra-large request is made. With a 20 virtual machine request schedule and having a maximum of five compute nodes, the cloud consumed 2.86 kWh. When the maximum number of compute nodes was increased to 10, 15, and 20, the power consumed was 3.37 kWh. With a 30 virtual machine request schedule and a maximum of five compute nodes, 5.1 kWh of power was consumed. When the maximum number of compute nodes was increased to 10, the power consumed was 7.36 kWh. Lastly, when the number of compute nodes increased to 15 and 20, the power consumed was 7.64 kWh.

The results obtained by using our PALB algorithm would obviously bring higher power saving over conventional load balancing algorithms such as the "round robin" approach used by Eucalyptus. In this approach, the cluster controller assigns virtual machine requests to compute nodes sequentially. This effectively balances the load across the compute nodes, but leaves these nodes in an "on" state with usually low utilization. Figure 8 shows the power consumption of the round robin load balancing algorithm.

A. *Comparison of PALB and Round Robin Load Balancers*

The major performance difference between our PALB algorithm and the round robin approach is that compute nodes that are idle using PALB are powered down. The round robin approach is effective in load balancing across the available compute nodes, but such a relatively simple load balancer consumes a large amount of unnecessary power.

In comparison, when the cloud has 5 compute nodes and 20 small virtual machines are requested, PALB consumes 11% of the energy consumed for round robin with the same parameters. When requesting 20 small virtual machines while having 20 available compute nodes, PALB only uses 2.8% of the energy consumed by round robin. Using the requests for extra-large virtual machines and five available compute nodes, PALB uses 29.5% of the energy consumed by round robin. Requesting 20 extra-large virtual machines with 20 compute nodes available, PALB consumes 20.6% of the energy used by round robin.

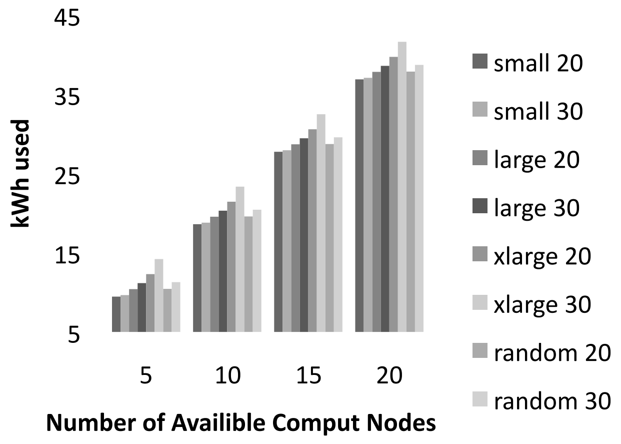

Figure 9: Comparison of round robin VM requests

VII. Conclusion and Future Work

With the increased use of local cloud computing architectures, organizations are becoming aware of wasted power consumed by unutilized resources. We introduced a load balancing algorithm that balances resources across available compute nodes in a cloud with power savings in mind. Since the cloud architecture implemented by local organizations tends to be heterogeneous, we take this into account in our design. Our Power Aware Load Balancing algorithm, PALB, maintains the state of all compute nodes, and based on utilization percentages, decides the number of compute nodes that should be operating.

Using PALB, organizations wanting to build local clouds using Eucalyptus would be able to save on energy costs. This is due to the fact that Eucalyptus does not account for power consumption when applying its default load balancing technique. Depending on the job schedule distribution and virtual machine request size, organizations can save 70% - 97% of the energy consumed compared to using load balancing techniques that are not power aware.

Our future work will include implementing this on our local cloud “Fluffy”. This cloud is a standard Eucalyptus build with independent nodes for each component of the cloud. We will also be implementing other load balancing algorithms for comparison to PALB. 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] OpenNebula cloud software. opennebula.org.

[2] Nimbus cloud software. www.nimbusproject.org.

[3] Eucalyptus Systems cloud software. www.eucalyptus.com.

[4] Jinhua Hu, Jianhua Gu, Guofei Sun, and Tianhai Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment," in *3rd Internation Symposium on Parallel Architectures, Algorithms and Programming*, pp. 89-96.

[5] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, and Shun-Sheng Wang, "Towards a Load Balancing in a Three-Level Cloud Computing Network,".

[6] Hein Nguyen Van, Frederic Dang Tran, and Jean-Marc Menaud, "Performance and Power Management for Cloud Infrastructures," in *2010 IEEE 3rd International Conference on Cloud Computing*, 2010.

[7] Susan Vrbsky, M Lei, Karl Smith, and S Byrd, "Data Replication and Power Consumption in Data Grids," in *2nd IEEE Internation Conference on Cloud Computing Technology and Science* , 2010.

[8] Karl Smith, Michael Galloway, and Susan Vrbsky, "Reactive Power Management for Distributed Systems," in *ACM Southeaster Conference*, 2011.

[9] David Meisner, Brian T Gold, and Thomas F Wenisch, "The PowerNap Server Architecture," *ACM Transaction on Computer Systems*, vol. 29, no. 1, February 2011.

[10] KranthiManoj Nagothu, Brian Kelley, Jeff Prevost, and Mo Jamshidi, "Ultra Low Energy Cloud Computing Using Adaptive Load Prediction," in *World Automation Congress*, 2010.

[11] Che-Yuan Tu, Wen-Chieh Kuo, Wei-Hua Teng, Yao-Tsung Wang, and Steven Shiau, "A Power-Aware Cloud Architecture with Smart Metering," in *39th Internation Conference on Parallel Processing Workshops*, 2010.

[12] L Barroso and U Holzle, "The Case for Energy-Porportional Computing," *IEEE Computer*, vol. 41, no. 12, pp. 33-37, Dec 2007.

[13] D Harnik, D Naor, and I Segal, "Low Power Mode in Cloud Storage Systems," in *IEEE International Symposium on Parallel & Distributed Processing*, 2009, pp. 1-8.

[14] K Hasebe, T Niwa, A Sugiki, and K Kato, "Power-Savings in Large-Scale Storage Systems with Data Migration," in *2nd IEEE Internation Conference on Cloud Computing Technology and Science*, Indianapolis, 2010.

[15] KVM. www.linux-kvm.org.