

Efficient Resource Management for Cloud Computing Environments

Andrew J. Younge, Gregor von Laszewski, Lizhe Wang
Pervasive Technology Institute
Indiana University
Bloomington, IN USA
Email: {ajyounge,gvonlasz,wanglize}@indiana.edu

Sonia Lopez-Alarcon, Warren Carithers
Rochester Institute of Technology
Rochester, NY USA
Email: slaec@rit.edu, wrc@cs.rit.edu

Abstract—The notion of Cloud computing has not only reshaped the field of distributed systems but also fundamentally changed how businesses utilize computing today. While Cloud computing provides many advanced features, it still has some shortcomings such as the relatively high operating cost for both public and private Clouds. The area of Green computing is also becoming increasingly important in a world with limited energy resources and an ever-rising demand for more computational power. In this paper a new framework is presented that provides efficient green enhancements within a scalable Cloud computing architecture. Using power-aware scheduling techniques, variable resource management, live migration, and a minimal virtual machine design, overall system efficiency will be vastly improved in a data center based Cloud with minimal performance overhead.

Index Terms—Cloud Computing; Green Computing; Virtualization; Scheduling

I. INTRODUCTION

For years visionaries in computer science have predicted the advent of utility-based computing will reign champion. This concept dates back to John McCarthy's vision stated at the MIT centennial celebrations in 1961.

"If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The computer utility could become the basis of a new and important industry."

Only recently has the hardware and software been available to support the concept of utility computing on a large scale.

The concepts inspired by the notion of utility computing have recently combined with the requirements and standards of Web 2.0 [1] to create Cloud computing [2], [3]. Cloud computing is defined as, "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet."

As new distributed computing technologies like Clouds become increasingly popular, the dependence on power also increases. Currently it is estimated that data centers consume 0.5 percent of the world's total electricity usage [4] and if current demand continues, is projected to quadruple by 2020.

In 2005, the total energy consumption for servers and their cooling units was projected at 1.2% the total U.S. energy consumption and doubling every 5 years [5], [6]. The majority of the energy used in today's society is generated from fossil fuels which produce harmful CO_2 emissions. Therefore, it is imperative to enhance the efficiency and potential sustainability of large data centers.

One of the fundamental aspects of virtualization technologies employed in Cloud environments is resource consolidation and management. Using hypervisors within a cluster environment allows for a number of standalone physical machines to be consolidated to a virtualized environment, thereby requiring less physical resources than ever before. While this improves the situation, it often is inadequate. Large Cloud deployments require thousands of physical machines and megawatts of power. Therefore, there is a need to create an efficient Cloud computing system that utilizes the strengths of the Cloud while minimizing its energy footprint.

In order to correctly and completely unify a Green aspect to the next generation of Distributed Systems, a set of guidelines needs to persist. These guidelines much represent a path of sustainable development that can be integrated into data center construction and management as a whole. While the framework provided in this paper represents many promising ways to reduce power consumption, true sustainable development also depends on finding a renewable and reliable energy source for the data center itself. When combined, many of today's limits in the size of data centers will begin to deteriorate.

This paper is organized as follows. Section II investigates previous research in minimizing power consumption with emphasis on Clouds. Section III presents a novel Green Cloud framework and its components, while Section IV and V detail certain components of the framework. In Section VI we assemble the components discussed and evaluate the potential energy savings. Finally, we conclude and provide insight for future work.

II. RELATED RESEARCH

In order to accurately depict the research presented in this article, the topics of Cloud computing, Grid computing, Clusters and Green computing will be reviewed.

A. Clouds

Cloud computing is becoming one of the most explosively expanding technologies in the computing industry today. It enables users to migrate their data and computation to a remote location with minimal impact on system performance [7]. Typically this provides a number of benefits which could not otherwise be realized. These benefits include:

- *Scalable* - Clouds are designed to deliver as much computing power as any user wants. While in practice the underlying infrastructure is not infinite, the cloud resources are projected to ease the developer's dependence on any specific hardware.
- *Quality of Service (QoS)* - Unlike standard data centers and advanced computing resources, a well designed Cloud can project a much higher QoS than typically possible. This is due to the lack of dependence on specific hardware, so any physical machine failures can be mitigated without the user's knowledge.
- *Specialized Environment* - Within a Cloud, the user can utilize custom tools and services to meet their needs. This can be to use the latest library, toolkit, or to support legacy code within new infrastructure.
- *Cost Effective* - Users find only the hardware required for each project. This greatly reduces the risk for institutions who may be looking to build a scalable system. Thus providing greater flexibility since the user is only paying for needed infrastructure while maintaining the option to increase services as needed in the future.
- *Simplified Interface* - Whether using a specific application, a set of tools or Web services, Clouds provide access to a potentially vast amount of computing resources in an easy and user-centric way. We have investigated such an interface within Grid systems through the use of the Cyberaide project [8], [9].

There are a number of underlying technologies, services, and infrastructure-level configurations that make Cloud computing possible. One of the most important technologies is the use of virtualization [10], [11]. Virtualization is a way to abstract the hardware and system resources from an operating system. This is typically performed within a Cloud environment across a large set of servers using a Hypervisor or Virtual Machine Monitor (VMM) which lies in between the hardware and the Operating System (OS). From here, one or more virtualized OSs can be started concurrently as seen in Figure 1, leading to one of the key advantages of Cloud computing. This, along with the advent of multi-core processing capabilities, allows for a consolidation of resources within any data center. It is the Cloud's job to exploit this capability to its maximum potential while still maintaining a given QoS.

Virtualization is not specific to Cloud computing. IBM originally pioneered the concept in the 1960's with the M44/44X systems. It has only recently been reintroduced for general use on x86 platforms. Today there are a number of Clouds that offer Infrastructure as a Service (IaaS). The Amazon Elastic

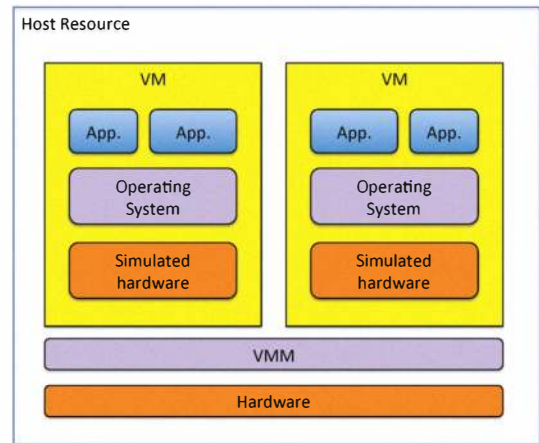


Fig. 1. Virtual Machine Abstraction

Compute Cloud (EC2) [12], is probably the most popular of which and is used extensively in the IT industry. Eucalyptus [13] is becoming popular in both the scientific and industry communities. It provides the same interface as EC2 and allows users to build an EC2-like cloud using their own internal resources. Other scientific Cloud specific projects exist such as OpenNebula [14], In-VIGO [15], and Cluster-on-Demand [16]. They provide their own interpretation of private Cloud services within a data center. Using a Cloud deployment overlaid on a Grid computing system has been explored by the Nimbus project [17] with the Globus Toolkit [18]. All of these clouds leverage the power of virtualization (typically using the Xen hypervisor) to create an enhanced data center.

B. Green Computing

The past few years has seen an increase in research on developing efficient large computational resources. Supercomputer performance has doubled more than 3000 times in the past 15 to 20 years, the performance per watt has increased 300 fold and performance per square foot has only doubled 65 times [19] in the same period of time. This lag in Moore's Law over such an extended period of time in computing history has created the need for more efficient management and consolidation of data centers. This can be seen in figure 2 [20].

Much of the recent work in Green computing focuses on Supercomputers and Cluster systems. Currently the fastest Supercomputer in the world is the IBM Roadrunner at Los Alamos National Laboratory [21], [22], which was fundamentally designed for power efficiency. However, Roadrunner consumes several Megawatts of power [23] (not including cooling) and costs millions of dollars to operate every year. The second fastest Supercomputer is Jaguar at Oak Ridge National Laboratory. While Jaguar too has a number of power saving features developed by Sandia, Oak Ridge and Cray [24] such as advanced power metering at the CPU level, 480 volt power supplies, and an advanced cooling system developed by Cray, the system as a whole still consumes almost 7 Megawatts of power.

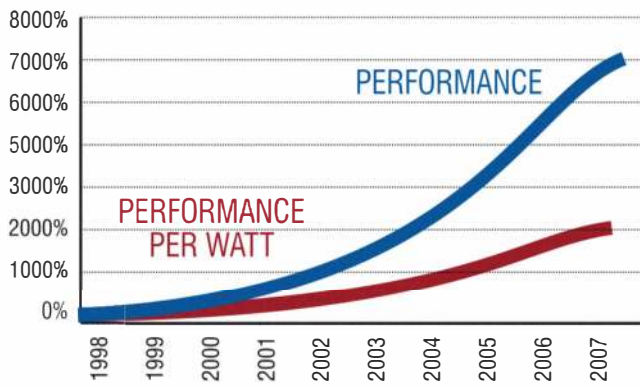


Fig. 2. Performance increases much faster than performance per watt of energy consumed [20]

One technique being explored is the use of Dynamic Voltage and Frequency Scaling (DVFS) within Clusters and Supercomputers [25], [26]. By using DVFS one can lower the operating frequency and voltage, which results in decreased power consumption of a given computing resource considerably. This technique was originally used in portable and laptop systems to conserve battery power, and has since migrated to the latest server chipsets. Current technologies exist within the CPU market such as Intel’s SpeedStep and AMD’s PowerNow! technologies. These dynamically raise and lower both frequency and CPU voltage using ACPI P-states [27]. In [28], DVFS techniques are used to scale down the frequency by 400Mhz while sustaining only a 5% performance loss.

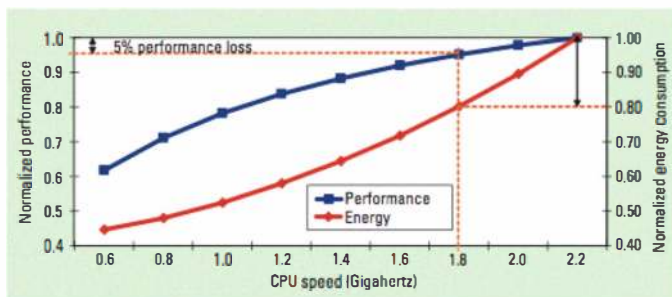


Fig. 3. Possible energy to performance trade-off. Here you can see a 18% reduction in frequency contributes to only a 5% performance loss. [28]

A power-aware Cluster supports multiple power and performance modes on processors with frequencies that can be turned up or down. This allows for the creation of an efficient scheduling system that minimizes power consumption of a system while attempting to maximise performance. The scheduler performs the energy-performance trade-off within a cluster. Combining various power efficiency techniques for data centers with the advanced feature set of Clouds could yield drastic results, however currently no such system exists.

III. GREEN CLOUD FRAMEWORK

There is a pressing need for an efficient yet scalable Cloud computing system. This is driven by the ever-increasing demand for greater computational power countered by the continual rise in use expenditures, both economical and environmental. Both business and institutions will be required to meet these needs in a rapidly changing environment.

We present a novel Green computing framework that is applied to the Cloud in order to meet the goal of reducing power consumption. This framework is meant to define efficient computing resource management and Green computing technologies can be adapted and applied to Cloud systems.

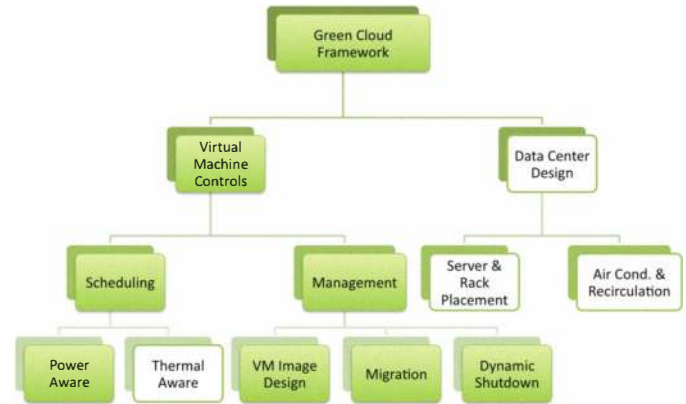


Fig. 4. Green Cloud Framework. Shaded items represent topics discussed in this paper.

Figure 4 illustrates a comprehensive *Green Cloud framework* for maximising performance per watt within a Cloud. This framework outlines the major areas are VM scheduling, VM image management, and advanced data center design.

Within the framework, there are two major areas which can lead to improvements. First, we can expand upon the baseline functioning of virtual machines in a cloud environment. This is first done with deriving a more efficient scheduling system for VMs. The Scheduling section addresses the placement of VMs within the Cloud infrastructure while minimizing the operating costs of the Cloud itself. This is typically achieved by optimising either power of the server equipment itself or the overall temperature within the data center. Due to the inherent disposability and mobility of VMs within a semi-homogeneous data center, we can leverage the ability to move and manage the VMs to further improve efficiency. The image management section attempts to control and manipulate the size and placement of VM images in various ways to conserve power and remove unnecessary bloat. Furthermore, the design of the virtual machine images can also lead to a drastic power savings.

While these operational and runtime changes can have a drastic impact, however more static data center level design decisions should also be included. Using more efficient Air Conditioning units, employing exterior “free” cooling, using completely separated hot and cold isles, or simply picking more efficient power supplies for the servers can lead to

incremental but substantial improvements. While this may be outside the scope of this paper, the integrated components of the Green Cloud framework in Figure 4 provide a sustainable development platform which has the largest potential impact factor to drastically reduce power requirements within a Cloud data center. Although the potential is great, combining each factor together in such a unified framework and deploying it to a large scale Cloud poses many challenges.

IV. VIRTUAL MACHINE SCHEDULING & MANAGEMENT

While Supercomputer and Cluster scheduling algorithms are designed to schedule individual jobs and not virtual machines, some of the concepts can be translated to the Cloud. We have already conducted such research in [29]. In many service oriented scientific Cloud architectures, new VMs are created to perform some work. The idea is similar to sand boxing work within a specialized environment.

A. Power-aware VM Scheduling

Currently, there are two competing types of Green scheduling systems for Supercomputers; power-aware and thermal-aware scheduling. In thermal-aware scheduling [30], jobs are scheduled in a manner that minimizes the overall data center temperature. The goal is not always to conserve the energy used to the servers, but instead to reduce the energy needed to operate the data center cooling systems. In power-aware scheduling [26], jobs are scheduled to nodes in such a way to minimize the server's total power. The largest operating cost incurred in a Cloud data center is in operating the servers. As such, we concentrate on power-aware scheduling in this paper.

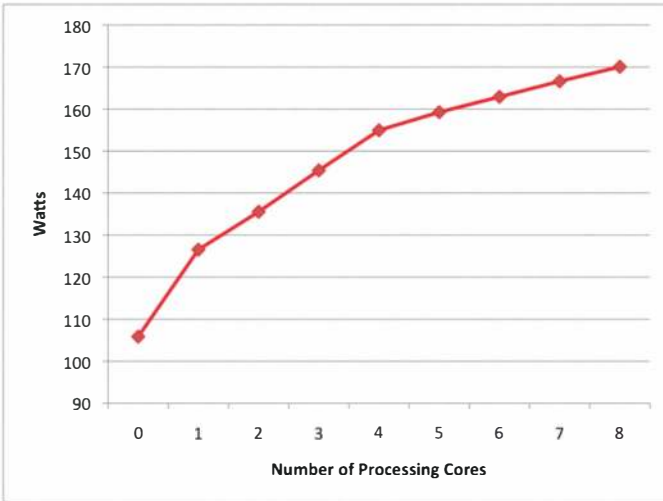


Fig. 5. Power consumption curve of an Intel Core i7 920 CPU

Figure 5 illustrates the motivation behind power-aware VM scheduling. This graphic documents our recent research findings regarding watts of energy consumed versus the number of processing cores in use. The power consumption curve illustrates that as the number of processing cores increases, the amount of energy used does not increase proportionally. In fact that change in power consumption decreases. When using

only one processing core, the change in power consumption incurred by using another processing core is over 20 watts. The change from 7 processing cores to all 8 processing cores results in an increase of only 3.5 watts.

The impact of this finding is substantial. In a normal round robin VM scheduling system like the one in Eucalyptus, the load of VMs is distributed evenly to all servers within the data center. While this may be a fair scheduler, in practice it is very inefficient. The result is that each time the scheduler distributes VMs to a processor, the power consumption increases by its greatest potential. In contrast, this research demonstrates that if the scheduler distributes the VMs with the intent to fully utilize all processing cores within each node, the power consumption is decreased dramatically. Therefore, there is a large need for an advanced scheduling algorithm which incorporates the findings in Figure 5. To meet this need we propose Algorithm 1, a new greedy-based algorithm to minimise power consumption.

Algorithm 1 Power based scheduling of VMs

```

FOR  $i = 1$  TO  $i \leq |pool|$  DO
   $pe_i = \text{num cores in } pool_i$ 
END FOR

WHILE (true)
  FOR  $i = 1$  TO  $i \leq |queue|$  DO
     $vm = queue_i$ 
    FOR  $j = 1$  TO  $j \leq |pool|$  DO
      IF  $pe_j \geq 1$  THEN
        IF check capacity  $vm$  on  $pe_j$  THEN
          schedule  $vm$  on  $pe_j$ 
           $pe_j - 1$ 
        END IF
      END IF
    END FOR
  END FOR
  wait for interval  $t$ 
END WHILE

```

Algorithm 1 is a VM scheduling algorithm that minimizes power consumption within the data center. This task is accomplished by continually loading each node with as many VMs as possible. In Algorithm 1 the pool acts as a collection of nodes remain static after initialization. While not in the algorithm, the pool can be initialized by a priority based evaluations system to either maximise performance or further minimise power consumption. At a specified interval t the algorithm runs through each VM in the queue waiting to be scheduled. The first node in the priority pool is selected and evaluated to see if it has enough virtual cores and capacity available for the new VM. If it does, it is scheduled, the pe_i is decremented by one, and this processes is continued until the VM queue is empty. When a VM finishes its execution and terminates,

it reports it back the scheduler and pe_i is increased by one to signify a core of machine i is freed.

B. VM Management

Another key aspect of a Green Cloud framework is virtual machine image management. By using virtualization technologies within the Cloud, a number of new techniques become possible. Idle physical machines in a Cloud can be dynamically shutdown and restarted to conserve energy during low load situations. A similar concept was achieved in Grid systems though the use of Condor Glide-In [31], [32] add-on to Condor, which dynamically adds and removes machines from the resource pool. This concept of shutting down unused machines will have no effect on power consumption during peak load as all machines will be running. However in practice Clouds almost never run at full capacity as this could result in a degradation of the QoS. Therefore by design, fast dynamic shutdown and startup of physical machines could have a drastic impact on power consumption, depending on the load of the Cloud at any given point in time.

The use of live migration features within Cloud systems [33] is a recent concept. Live migration is presently used for proactive fault tolerance by seamlessly moving VMs away from failing hardware to stable hardware without the user noticing a change [34] in a virtualized environment. Live migration can be applied to Green computing in order to migrate away machines. VMs can be shifted from low load to medium load servers when needed. Low load servers are subsequently shutdown when all VMs have migrated away, thus conserving the energy required to run the low load idle servers. When using live migration, the user is completely unaware of a change and there is only a 60 to 300ms delay, which is acceptable by most standards.

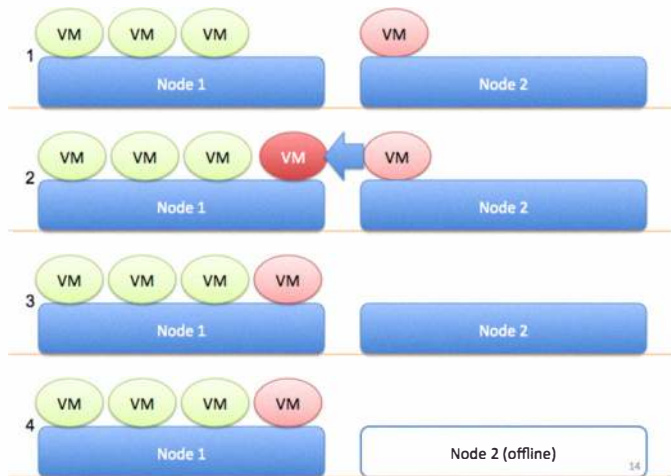


Fig. 6. Virtual Machine management dynamic shutdown technique

This process of dynamically allocating and deallocating physical machines is complimentary to our scheduling system outlines in Algorithm 1. As the scheduling algorithm executes, it will leave a number of machines idling, potentially for long

periods of time, creating an optimal state for the VM management system. The machines that are left idle can shut down as illustrated in Figure 6. When load increases, we use Wake on LAN (WOL) to start them back up. This control can be easily monitored and implemented as a daemon running on the Cloud head node or scheduler. This effectively displays the goal of the Green Cloud Framework: while any single power saving technique can be beneficial, the calculated combination of multiple techniques from a systems-level perspective can yield significant power savings when compared to their individual implementations.

V. SERVICE ORIENTED VIRTUAL MACHINE IMAGE

While scheduling and management of virtual machines within a private Cloud environment is important, one must realize what is actually being scheduled. In a normal Cloud environment like the Amazon's EC2 [12], full Operating System VMs are scheduled, often to carry out specific tasks in mass. These VM instances contain much more than they need to in order to support a wide variety of hardware software and varying user tasks. While this is ideal for a desktop based environment, it leads to wasted time and energy in a server based solution. A hypervisor provides the same virtualized hardware to each VM and each VM is typically designed for a specific task. In essence, we want the OS within the VM to act only as a light wrapper which supports a few specific but refined tasks or services, and not an entire desktop/application suite. In order to accomplish this task, we need to concentrate on two areas; VM image size and boot time.

Normal x86 hardware can vary widely, so most modern operating systems including Linux are able to detect various hardware and load modules on the fly upon startup. This is not an issue with a virtual machine environment since the hardware is standardized and known in advance. It is common for the boot to spend 15 seconds running modprobe to load only a single module. The modules in the system and many of the time consuming probing functions can be reduced upon bootup within a VM environment. In [35] considerable amount of time is saved by changing the IDE delay times for probing new hardware.

Another technique for reducing the boot time is to orchestrate the boot sequence in a more efficient way. Often many daemons and applications are loaded for general use and in the case of a lightweight VM instance, aren't needed and can be removed. This includes standalone server applications like Window managers and the X11 windowing system. This would also remove the system's disk footprint considerably to save valuable hard drive space in distributed file systems as well as network traffic when migrating the machines.

Boot time can be further improved by creating a new order which maximises both the CPU utilization and I/O throughput. The use of bootchart [36] can profile where bootup system inefficiencies occur and to allow for optimization of the boot sequence. Another useful tool is readahead [37]. Readahead profiles the system startup sequence and uses file pre-fetching techniques to load files into memory before they are

requested. Therefore an application reads directly from system memory and does not have to wait for disk seek-time.

VI. POWER CONSUMPTION ANALYSIS

In order to validate our framework it is important to investigate its feasibility within an actual virtual machine cluster environment. This section discusses the implementation of our scheduling algorithm as it is applied to the OpenNebula project in a multi-core cluster and evaluates the creation of a new lightweight VM image.

A. Scheduling Analysis

OpenNebula [14] is an open source distributed virtual machine manager for dynamic allocation of virtual machines in a resource pool. The OpenNebula core components accept user requirements via the OpenNebula interface, and then place virtual machines in compute nodes within the cluster.

The OpenNebula scheduler is an independent component that provides policies for virtual machine placement. We choose the OpenNebula project because of this compartmentalized design as it allows for integration of our custom scheduling algorithm. The default scheduler provides a scheduling policy based on rank, which allocates compute resources for virtual machines. Scheduling algorithm 1 is implemented by modifying the OpenNebula scheduler to reflect the desired results in Section IV.

pool. This one machine operates at the full 170 Watts, however all other machines idle at 105 Watts, resulting in a pool power consumption of 485 Watts. Therefore, using our power based scheduling algorithm, we conserve 12% of the system’s power on only 4 machines on a normal load, as seen in Figure 7. While this experiment is only with 4 nodes, the results are clearly scalable to a large cluster and will be magnified as higher node core counts are deployed. If the live migration and shutdown strategy is also deployed, some servers could be dynamically shutdown to further conserve energy.

B. VM Image Analysis

In order to evaluate the performance of our VM image design, we must create a prototype. There are two paths available to build such a VM OS image. The first is a bottom up approach where a basic Linux kernel is built upon to reach the minimal feature set needed. This requires an entirely new distribution from scratch. While this may be the “cleanest” way, it would require a large development team and is therefore infeasible for this project. The other option involves a top-down approach of taking a common distribution and removing certain components from it, making for a lighter and faster sub-distribution. This route is more practical as it does not require reinventing the wheel and the option to keep components such as a package management system and a large distribution library are maintained.

Following the second approach, a custom Linux image was created to illustrate the possibility of a fast and lightweight VM OS. Starting with Ubuntu Linux version 9.04 Jaunty, all unnecessary packages were removed, including the Gnome window manager and X11. By removing these multitude of packages, the system image is reduced from 4Gb to only 636Mb. This minimization speeds up migration of the image from one server to another as there is less network traffic during the movement phase. A number of other packages, libraries and boot level daemons were also removed from the startup process. At the final stage, the image is a minimal Linux installation with only the bare necessity components. One thing that was left was the Synaptic package management system, so if any tools or libraries are needed it is a trivial process to have them installed on the system. While the package management system does take up some room, it is well worth the added extendability it provides to the system. A number of kernel modules were also removed from the 2.6.28-11 kernel to speed up the kernel init and modprobe processes as much as possible.

To test the speed of the custom image, both it and a basic Ubuntu 9.04 installation were moved to a VMWare server with 2.5Ghz Intel Core 2 Duo and 4GB of ram. The standard Ubuntu image booted from BIOS in 38 seconds. With our custom VM image, boot time was reduced dramatically to just 8 seconds. By comparing the boot charts in figures 8 and 9, we can see there is a drastic change in boot time, resulting in the boot time decreased by 30 seconds. Instead of a large amount of I/O blocking, all disk I/O is done at once towards the beginning, allowing for much higher utilization of

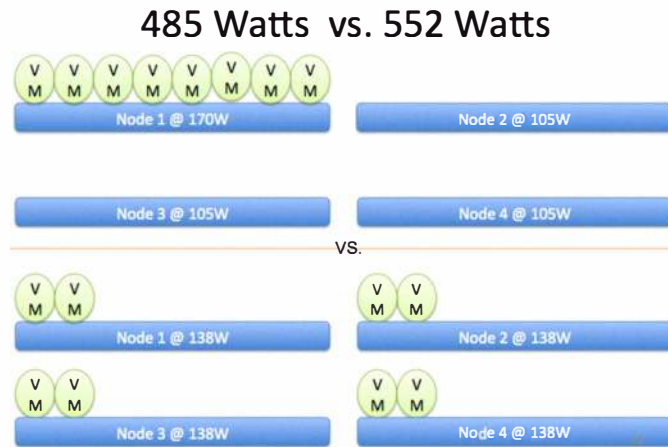


Fig. 7. Illustration of Scheduling power savings

To evaluate the energy savings of Algorithm 1, we consider the following small OpenNebula pool of just 4 servers. Each server within the pool is a 2.6Ghz Intel Core i7 920 with 12GB of RAM. We assume each server can hold 8 VMs as it has 8 virtual cores. At idle, they consume 105 Watts of power and under 100% load they consume 170 Watts (see Figure 5). If we execute the default OpenNebula scheduler to schedule 8 virtual machines each running CPU-bound tasks, each server would gain 2 VMs and would consume 138 Watts with a total pool power consumption of 552 Watts. However when Algorithm 1 is used, all the VMs are scheduled to the first machine in the

the CPU. While a boot time of 8 seconds is a considerable improvement, we can do better. The kernel still takes a full 2 seconds to load, however with some improvements a second or more could possibly be saved.

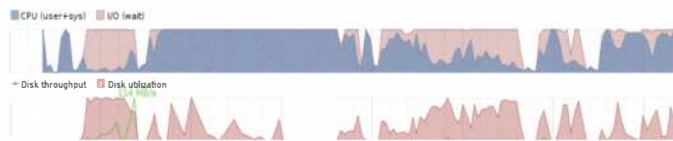


Fig. 8. Bootup chart of the default Ubuntu Linux VM image

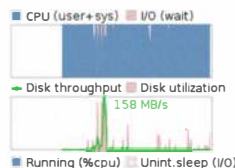


Fig. 9. Bootup chart of Minimal Linux VM image

Consider a VM which is started on a machine that requires 250 watts of power. Spending 30 seconds on booting up the VM results in 2.08 wh or .002 Kwh of energy used. While this savings of .002Kwh or 30 seconds doesn't seem like much, its effects are actually quite significant. In just a small private Cloud system, it is common for 100 VMs to be created every hour, depending on system load and utilization. As such, there is over 1750 Kwh wasted per year. Thus these changes in the VM image lead to hundreds or thousands of dollars in savings. Furthermore, the power savings realized of using lightweight VM images on a 10 Megawatt facility where thousands of VMs are started every minute equate to tens or hundreds of thousands of dollars a year.

VII. CONCLUSION AND FUTURE WORK

As the prevalence of Cloud computing continues to rise, the need for power saving mechanisms within the Cloud also increases. In this paper we have presented a novel Green Cloud framework for improving system efficiency in a data center. To demonstrate the potential of our framework, we have presented new energy efficient scheduling, VM system image, and image management components that explore new ways to conserve power. Though our research presented in this paper, we have found new ways to save vast amounts of energy while minimally impacting performance.

Not only do the components discussed in this paper compliment each other, they leave space for future work. Future opportunities could explore a scheduling system that is both power-aware and thermal-aware to maximise energy savings both from physical servers and the cooling systems used. Such a scheduler would also drive the need for better data center designs, both in server placements within racks and closed-loop cooling systems integrated into each rack. While a number of the Cloud techniques are discussed in this paper, there is a growing need for improvements in Cloud infrastructure, both

in the academic and commercial sectors. We believe Green computing will be one of the fundamental components of the next generation of Cloud computing technologies.

REFERENCES

- [1] B. Alexander, "Web 2.0: A New Wave of Innovation for Teaching and Learning?" *Learning*, vol. 41, no. 2, pp. 32–44, 2006.
- [2] R. Buyya, C. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08, IEEE CS Press, Los Alamitos, CA, USA)*, 2008.
- [3] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in *Grid Computing Environments Workshop, 2008. GCE'08*, 2008, pp. 1–10.
- [4] W. Forrest, "How to cut data centre carbon emissions?" Website, December 2008. [Online]. Available: <http://www.computerweekly.com/Articles/2008/12/05/233748/how-to-cut-data-centre-carbon-emissions.htm>
- [5] J. Koomey, "Estimating total power consumption by servers in the US and the world," *Final report. February*, vol. 15, 2007.
- [6] "Report to congress on server and data center energy efficiency - public law 109-431," Webpage, July 2007. [Online]. Available: www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf
- [7] L. Wang, G. von Laszewski, A. Younge, X. He, M. Kunze, and J. Tao, "Cloud computing: a perspective study," *New Generation Computing*, vol. WangLYAH, to appear in 2010.
- [8] G. von Laszewski, A. Younge, X. He, K. Mahinthakumar, and L. Wang, "Experiment and Workflow Management Using Cyberaide Shell," in *4th International Workshop on Workflow Systems in e-Science (WSES 09) in conjunction with 9th IEEE International Symposium on Cluster Computing and the Grid*. IEEE, 2009. [Online]. Available: <http://cyberaide.googlecode.com/svn/trunk/papers/09-gridshell-ccgrid/vonLaszewski-ccgrid09-final.pdf>
- [9] G. von Laszewski, F. Wang, A. Younge, X. He, Z. Guo, and M. Pierce, "Cyberaide JavaScript: A JavaScript Commodity Grid Kit," in *GCE08 at SC'08*. Austin, TX: IEEE, Nov. 16 2008. [Online]. Available: <http://cyberaide.googlecode.com/svn/trunk/papers/08-javascript/vonLaszewski-08-javascript.pdf>
- [10] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. L. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, New York, U. S. A., Oct. 2003, pp. 164–177.
- [11] VMware, "Understanding Full Virtualization, Paravirtualization, and Hardware Assis," VMware, Tech. Rep., 2007. [Online]. Available: http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf
- [12] Amazon, "Elastic Compute Cloud." [Online]. Available: <http://aws.amazon.com/ec2/>
- [13] D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus Open-source Cloud-computing System," *Proceedings of Cloud Computing and Its Applications*, 2008.
- [14] J. Fontan, T. Vazquez, L. Gonzalez, R. S. Montero, and I. M. Llorente, "OpenNEBula: The Open Source Virtual Machine Manager for Cluster Computing," in *Open Source Grid and Cluster Software Conference*, San Francisco, CA, USA, May 2008.
- [15] S. Adabala, V. Chadha, P. Chawla, R. Figueiredo, J. Fortes, I. Krsul, A. Matsunaga, M. Tsugawa, J. Zhang, M. Zhao, L. Zhu, and X. Zhu, "From virtualized resources to virtual computing Grids: the In-VIGO system," *Future Generation Comp. Syst.*, vol. 21, no. 6, pp. 896–909, 2005.
- [16] J. Keahey, D. Irwin, L. Grit, J. Moore, and S. Sprenkle, "Dynamic virtual clusters in a grid site manager," in *12th IEEE International Symposium on High Performance Distributed Computing, 2003. Proceedings*, 2003, pp. 90–100.
- [17] K. Keahey, I. Foster, T. Freeman, X. Zhang, and D. Galron, "Virtual Workspaces in the Grid," *Lecture Notes in Computer Science*, vol. 3648, pp. 421–431, 2005. [Online]. Available: http://workspace.globus.org/papers/VW_EuroPar05.pdf
- [18] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *International Journal of Supercomputer Applications*, vol. 11, no. 2, pp. 115–128, 1997, <ftp://ftp.globus.org/pub/globus/papers/globus.pdf>.

- [19] W. chun Feng and K. W. Cameron, "The Green500 List: Encouraging Sustainable Supercomputing," *IEEE Computer*, vol. 40, no. 12, pp. 50–55, 2007.
- [20] M. Wagner, "The efficiency challenge: Balancing total cost of ownership with real estate demands," Cherokee International, Tech. Rep., 2008.
- [21] K. Barker, K. Davis, A. Hoisie, D. Kerbyson, M. Lang, S. Pakin, and J. Sancho, "Entering the petaflop era: the architecture and performance of Roadrunner," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*. IEEE Press Piscataway, NJ, USA, 2008.
- [22] J. Dongarra, H. Meuer, and E. Strohmaier, "Top 500 supercomputers," website, November 2008.
- [23] S. Sharma, C.-H. Hsu, and W. chun Feng, "Making a case for a green500 list," in *IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006)/ Workshop on High Performance - Power Aware Computing*, 2006.
- [24] J. Laros III, K. Pedretti, S. Kelly, J. Vandyke, K. Ferreira, C. Vaughan, and M. Swan, "Topics on Measuring Real Power Usage on High Performance Computing Platforms," in *IEEE Cluster*, 2009.
- [25] C.-H. Hsu and W. chun Feng, "A Feasibility Analysis of Power Awareness in Commodity-Based High-Performance Clusters," in *CLUSTER*, 2005, pp. 1–10.
- [26] C. Hsu and W. Feng, "A power-aware run-time system for high-performance computing," in *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*. IEEE Computer Society Washington, DC, USA, 2005.
- [27] D. Bodas, "Data Center Power Management and Benefits to Modular Computing," in *Intel Developer Forum*, 2003. [Online]. Available: http://www.intel.com/idf/us/spr2003/presentations/S03USMODS137_OS.pdf
- [28] W. Feng, X. Feng, and R. Ge, "Green Supercomputing Comes of Age," *IT PROFESSIONAL*, vol. 10, no. 1, p. 17, 2008.
- [29] G. von Laszewski, L. Wang, A. J. Younge, and X. He, "Power-aware scheduling of virtual machines in dvfs-enabled clusters," in *IEEE Cluster 2009*, IEEE. New Orleans, Louisiana: IEEE, 08/2009 2009.
- [30] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 11, pp. 1458–1472, 2008.
- [31] S. Sarkar and I. Sfiligoi, "GlideCNAF: A Purely Condor Glide-in Based CDF Analysis Farm," CDF/DOC/COMP UPG/PUBLIC/7630P, Tech. Rep., 2005.
- [32] M. J. Litzkow, M. Livny, and M. W. Mutka, "Condor - A Hunter of Idle Workstations," in *Proceedings of the 8th International Conference on Distributed Computing Systems (ICDCS)*. San Jose, California: IEEE Computer Society, Jun. 1988, pp. 104–111. [Online]. Available: <http://www.cs.wisc.edu/condor/>
- [33] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *In Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, May 2005.
- [34] A. B. Nagarajan, F. Mueller, C. Engelmann, and S. L. Scott, "Proactive fault tolerance for hpc with xen virtualization," in *ICS '07: Proceedings of the 21st annual international conference on Supercomputing*. New York, NY, USA: ACM, 2007, pp. 23–32.
- [35] T. Bird, "Methods to improve bootup time in Linux," in *Proc of the Ottawa Linux Symp*, 2004.
- [36] Z. Mahkovec, "Bootchart," Webpage, 2005. [Online]. Available: <http://www.bootchart.org/>
- [37] H. Hoyer and K. Zak, "Readahead," Webpage. [Online]. Available: <https://fedorahosted.org/readahead/>