*Reprinted from the*

# Proceedings of the Linux Symposium

July 23rd–26th, 2008
Ottawa, Ontario
Canada

## Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium, Thin Lines Mountaineering*
C. Craig Ross, *Linux Symposium*

## Review Committee

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium, Thin Lines Mountaineering*
Dirk Hohndel, *Intel*
Gerrit Huizenga, *IBM*
Dave Jones, *Red Hat, Inc.*
Matthew Wilson, *rPath*
C. Craig Ross, *Linux Symposium*

## Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*
Gurhan Ozen, *Red Hat, Inc.*
Eugene Teo, *Red Hat, Inc.*
Kyle McMartin, *Red Hat, Inc.*
Jake Edge, *LWN.net*
Robyn Bergeron
Dave Boutcher, *IBM*
Mats Wichmann, *Intel*

# Applying Green Computing to clusters and the data center

Andre Kerstens
*SGI*
kerstens@sgi.com

Steven A. DuChene
*SGI*
sduchene@sgi.com

## Abstract

Rising electricity costs and environmental concerns are starting to make both the corporate IT and scientific HPC worlds focus more on green computing. Because of this, people are not only thinking about ways to decrease the initial acquisition costs of their equipment, but they are also putting constraints on the operational budgets of that same equipment. To address this challenge, we use both commercial and open-source Linux tools to monitor system utilization and closely track the power usage of those systems. The results of our monitoring are then used to make real-time decisions on whether systems can be put to sleep or shutdown altogether. In this paper we show how to use the Ganglia monitoring system and Moab scheduling engine to develop a methodology that guarantee the most efficient power usage of your systems by helping Moab make intelligent decisions based on real-time power data and incoming workload.

## 1 Introduction

In the last five years, corporate and research data centers have grown significantly due to the increasing demand for computer resources. Not only has the power used by these computer systems roughly doubled over this period, but also the energy consumed by the cooling infrastructure to support these computer systems has increased significantly. In addition to the resulting increase in data center capital and operational costs, this expanding energy use has an impact on the environment in the form of carbon-dioxide emissions that are created as an unwanted by-product of the electricity generation. In their Report to Congress on Server and Data Center Energy Efficiency [1], the Environmental Protection Agency (EPA) estimates that servers and data centers consumed about 61 billion kilowatt-hours in 2006 (1.5 percent of the total US electricity consumption) and that this will double by the year 2011 (an annual growth rate of 9 percent). Recent findings by the Uptime Institute [2] show that the EPA numbers are probably too conservative and that the annual growth rate from 2006 to 2011 is more likely to be 20 to 30 percent. No matter who is right in the end, it is obvious that measures have to be taken to limit the increase in power consumption. On a global level, organizations like the Green Grid[SM] have started defining metrics for energy efficiency. They are developing standards and measurements methods to determine data center efficiency against these metrics. On a local level, we can increase the efficiency of our data centers and reduce operating costs by decreasing the power and cooling loads of our computing infrastructure.

To achieve this objective, we propose to use intelligent monitoring and control of data center resources. With a combination of open-source and commercial Linux software like Ganglia [3] and Moab [4], we will be able to monitor system utilization as well as closely track the power usage of those systems. The information collected are used to make real-time decisions on whether systems can be put to sleep, run in a lower power mode, or shutdown altogether. The rest of this paper is organized as follows: After some history on green computing efforts in Section 2, we discuss the details of our methodology in Section 3, and show a case study in Section 4. We conclude the paper in Section 5.

## 2 Green Computing

Wikipedia defines Green Computing as the study and practice of using computing resources efficiently [5]. This comes down to designing computer systems that optimize the performance of the system compared to the cost of running (i.e., electricity) and operating it (i.e., power distribution and cooling).

In the past decade there have been some disconnected efforts by government, academic, and corporate facilities as well as data center managers and system vendors

to increase the energy efficiency of servers and other computing resources. On a national and global level, organizations such as Green Grid℠ [6], ASHRAE [18], and Climate Savers Computing Initiative℠[19] are defining metrics for energy efficiency and are developing methods to measure energy efficiency against these metrics. For example, Green Grid℠ has defined the Power Usage Effectiveness (PUE) metric [7] that enables data center operators to estimate the energy efficiency of their data centers and determines if energy efficiency improvements need to be made.

There have been a number of other attempts to manage the consumption of server systems and clusters. Rajamani and Lefurgy worked on identifying system and workload factors in power-aware cluster request distribution policies [13]. Elnozahy *et al.* have studied dynamic voltage scaling and request batching policies to reduce the energy consumption of web server farms [14]. Fan *et al.* studied the aggregate power usage characteristics of up to 15 thousand servers for different classes of applications over a period of approximately six months to evaluate opportunities for maximizing the use of the deployed power capacity of data centers and assess over-subscribing risks [15]. Chase *et al.* looked at how to improve the energy efficiency of server clusters by dynamically resizing the active server set and how to respond to power supply disruptions or thermal events by degrading service in accordance with negotiated Service Level Agreements (SLAs) [12]. Pinheiro *et al.* developed a system that dynamically turns cluster nodes on or off by considering the total load on the system and the performance implications of changing the current configuration [16].

Our goal is to expand these efforts to full data centers or clusters of data centers. For example, by moving workloads around based on the cost of power and cooling in various geographical locations or by delaying workloads to run during off-peak electricity rates, additional cost savings can be realized.

## 3 Methodology

Our method of green computing consists of three primary components:

1. Collect data to monitor resource state (power/ temperature);
2. Interfacing to power management facilities; and
3. Enabling intelligent policies to control power consumption and remove hot spots.

The following sections discuss these components in detail; they will be combined in a case study that is described in Section 4.

### 3.1 Data Collection

Most modern computer systems, if not all, contain a service processor that provides out-of-band remote management capabilities. In most open or commodity-based systems, this service processor is a Baseboard Management Controller (BMC) which provides access to Intelligent Platform Management Interface (IPMI) capabilities. Various tools to access the BMC can be used to monitor sensor information like temperature, voltages, fan speeds, and power status. The BMC also provides remote network access to power on and power off systems. The BMC operates independent of the processor and the operating system, thus providing the ability to monitor, manage, diagnose, and recover systems, even if the operating system has crashed or the server is powered down, and as long as the system is connected to a power source. Other service-processor-based, out-of-band management systems such as RSA cards, iLO, ALOM, ILOM, or DRAC implement similar feature sets using vendor-specific tool sets.

Most currently available BMCs support either IPMI 1.5 or IPMI 2.0 with common sensors of fan speeds, cpu temperatures, board temperature, cpu voltages, power supply voltages, etc. On a system with a BMC that supports IPMI 2.0, a power sensor is more likely to be present that reports watts being used by the system. The sensor data returned from `ipmitool` run against a BMC that supports IPMI 1.5 looks like this:

| | | |
|---|---|---|
| CPU Temp 1 | 29 degrees C | ok |
| CPU Temp 2 | 28 degrees C | ok |
| CPU Temp 3 | no reading | ns |
| CPU Temp 4 | no reading | ns |
| Sys Temp | 27 degrees C | ok |
| CPU1 Vcore | 1.31 Volts | ok |
| CPU2 Vcore | 1.31 Volts | ok |
| 3.3V | 3.26 Volts | ok |

| | | |
|---|---|---|
| 5V | 4.90 Volts | ok |
| 12V | 11.81 Volts | ok |
| 1.5V | 1.49 Volts | ok |
| 5VSB | 4.85 Volts | ok |
| VBAT | 3.28 Volts | ok |
| Fan1 | 13200 RPM | ok |
| Fan2 | 11200 RPM | ok |
| Fan3 | 13200 RPM | ok |
| Fan4 | 11200 RPM | ok |
| Fan5 | 13200 RPM | ok |
| Fan6 | 11100 RPM | ok |

Once the access to the sensor data from the BMC is confirmed across the cluster, the various sensor values can be pulled into a cluster monitoring system like Ganglia, Nagios, or MRTG. Since the authors of this paper are most familiar with it, Ganglia will be used to monitor and record historical data about the systems, clusters, and other data center power and cooling equipment. Ganglia has web-based data-display mechanisms as well as command-line tools to peek at the data stream. The web-based display is based around `rrdtool` just like MRTG.

By getting this sensor data into a monitoring tool like Ganglia, historical data becomes available for performance trend analysis and post-problem root cause analysis.

Modern data center infrastructure equipment such as Power Distribution Units (PDU), Uninterruptible Power Supplies (UPS), chillers, and Computer Room Air Conditioning (CRAC) units are IP-enabled and understand network protocols (e.g., SNMP or HTTP) for communication with other systems on the network. Using this capability, th,e electric current through a UPS or a PDU branch circuit can be measured and the temperature of the water in a chiller or the return line of a CRAC unit can be requested. Some models of rack PDUs can measure the power draw per PDU outlet. This provides an opportunity for measuring power usage of servers that do not have IPMI capabilities. Often these infrastructure devices can also be controlled over the network, but that discussion falls outside of the scope of this paper.

Adding these data sources from CRAC units or PDUs into the monitoring system provides a more complete picture of data center conditions over long periods of time or for spot analysis of daily, weekly, or monthly trends. For example, it could show that a particular set of systems or areas of a data center get unusually hot on weekends. The cause of this could be that some error in data center facilities setup is not taking into account system loads on weekends.

### 3.2 Power Management Interface

Various power states are available in a Linux system, i.e., everything from a 100% power utilization to powered-off. By being able to intelligently control the current power level based on current and future system load expectations, we can take maximum advantage of the power savings available. Within an HPC environment, where a job control system is used to process incoming workloads, the system load expectations can be fairly predictable. By placing the power state of HPC cluster client systems under control of the job control scheduler, the incoming workload can drive the power demands of the cluster. In this case, nodes can be switched off when the workload is light and switched on again when the workload is expected to go up. However, even when a system is running a job, there are power savings benefits possible by controlling the power usage of non-critical system components. Modern processors, disks, and other components can have varying states of power usage. By taking advantage of the ability to dynamically control the power state, the system is able to make adjustments when a device is idle for a significant period of time. The degree of significance here is different for a processor versus a disk or other components: processor idle times can be of milliseconds, while disk idle times are in the range of seconds.

There are currently four different processor power-saving states described in the Advanced Configuration and Power Interface (ACPI) specification [10]: C0, C1, C2, C3, and C4. Table 1 shows the power usage of the Intel Core 2 Duo processor in each of these states. Note that the lower the power-saving state, the longer it will take to wake up from that state.

Many of these numbers come from the LessWatts organization [8], which does research into power saving for Linux systems [9]. To see the current power state as well as power states supported by a system use:

```
cat /proc/acpi/processor/CPUx/power
```

where x is a number ranging from 0 to the number of CPUs in the system. For example, the output on a Core2Duo system looks as follows:

| C-State | Max Power Consumption (Watt) |
|---|---|
| C0 | 35 |
| C1 | 13.5 |
| C2 | 12.9 |
| C3 | 7.7 |
| C4 | 1.2 |

Table 1: Intel Core 2 Duo maximum power consumption in the different C-states

```
active state: C3
max_cstate: C8
bus master activity: 00000000
maximum allowed latency: 2000 usec
states:
C1: type[C1] promotion[C2] demotion[--]
latency[001] usage[00000010]
duration[0000000000000000000]
C2: type[C2] promotion[C3] demotion[C1]
latency[001] usage[181597540]
duration[00000000631489359035]
*C3: type[C3] promotion[--] demotion[C2]
latency[057] usage[1438931278]
duration[00000006636332340366]
```

With the use of a script, the active power state of the system CPUs can be monitored over time with Ganglia.

When a low-power state is entered, it is best to stay in that state as long as possible for the greatest energy savings. Unfortunately, older Linux kernels have a regularly occurring interrupt, called a timer tick, that is used to alert the kernel when some housekeeping tasks have to be performed. This feature limited the usefulness of the lower power states (for example, C3 or C4), because the system could only stay in that state in between timer ticks (1, 4, or 10 ms). In newer Linux kernels, starting with 2.6.21 for x86 and 2.6.24 for x86_64, a tick-less timer was introduced which made it possible to keep the processor in a lower power state for a longer amount of time, at least until the next timer event occurred. Unfortunately, there is still much code around (e.g., applications, device drivers) which does not take energy efficiency into account and which triggers the kernel hundreds of times per second to wake it up and do some work. Until this situation changes (and it is indeed slowly changing due to projects like Lesswatts.org [8]), saving power on Linux using power-saving states continues to be a struggling task.

The ACPI specification also defines four system sleep states:

- S1 – Stopgrant – Power to CPU is maintained, but no instructions are executed. The CPU halts itself and may shut down many of its internal components. In Microsoft Windows, the "Standby" command is associated with this state by default.

- S3 – Suspend to RAM – All power to the CPU is shut off, and the contents of its registers are flushed to RAM, which remains on. In Microsoft Windows, the "Standby" command can be associated with this state if enabled in the BIOS. Because it requires a high degree of coordination between the cpu, chipset, devices, OS, BIOS, and OS device drivers, this system state is the most prone to errors and instability.

- S4 – Suspend to Disk – CPU power shut off as in S3, but RAM is written to disk and shut off as well. In Microsoft Windows, the "Hibernate" command is associated with this state. A variation called S4BIOS is most prevalent, where the system image is stored and loaded by the BIOS instead of the OS. Because the contents of RAM are written out to disk, system context is maintained. For example, unsaved files would not be lost following an S4 transition.

  S4 is currently not supported by the 2.4.x kernel series in Linux, but you might have good luck with SWSUSP. Some machines offer `S4_BIOS` whose support is considered to be experimental within Linux/ACPI.

- S5 – Soft Off – System is shut down, however some power may be supplied to certain devices to generate a wake event—for example, to support automatic startup from a LAN or USB device. In Microsoft Windows, the "Shut down" command is associated with this state. Mechanical power can usually be removed or restored with no ill effects.

These sleep states are activated by writing values to the file `/sys/power/state`. The current state can be queried by reading this file.

Processor voltage and frequency scaling are other techniques for managing power usage that have been available in consumer platforms for some years and only recently have been made available to server-class processors [11]. Processor voltage scaling is used

to run a processor (or processors) at a lower voltage than the maximum possible while frequency scaling is used to run a processor at a lower frequency than the maximum possible to conserve power. In Linux 2.6.x systems, frequency scaling can be controlled through the directory `/sys/devices/system/cpu/cpu0/cpufreq/`. There are several governors available to control the frequency scaling behavior of the system, e.g., conservative, on-demand, power-save, user-space, performance. The list on a specific system can be shown by viewing the file `/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors`. The governor can be changed by writing a new value into the file `/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor`. Both processor voltage and frequency scaling techniques are heavily used on notebooks and other systems containing mobile processors. Unfortunately mobile processors are not used for data center workloads normally and thus many of the power-saving features discussed above are not applicable to server platforms. On the other hand, vendors like Intel are planning to bring power management features usually found in their mobile line of processors like the Core 2 Duo to server platforms like the Xeon family of processors [9]. By passing on the power-savings features from the mobile market to the server market, a whole new range of possibilities for energy conservation is entering the data center.

In HPC clusters, the concept of shutting down nodes is a relatively new concept since most HPC clusters leave nodes running 24/7 except for scheduled maintenance outages. This is not necessary with current workload management systems. One issue is that cluster administrators often perceive shutting down nodes as a probable cause of power supply failures or hard drive spin-up failures. However, our view is that if components are going to break, and this is inevitable, they should do so in a known, controlled manner at a time that jobs are not scheduled on these resources. Deliberately taking a node down is the clearest indicator of the reliability of your cluster. Doing these tests, everything from UPS loading, to switch fabric, to control and mediation, as well as the hardware of the node itself is affected.

### 3.3 Scheduling and Control

The power management functionality discussed in Section 3.2 is not necessarily passed through to the control of the workload scheduler. We will start by doing coarse-grained control through Moab, the workload scheduler we chose in this paper.

Moab uses IPMI or similar capabilities to monitor temperature information, check power status, power-up, power-down, and reboot compute nodes. This information can be utilized to make intelligent scheduling decisions that can save energy, limit or remove hot spots in the data center, and open up the possibility of implementing chargeback structures for users, groups, or departments based on energy usage.

In this context, the first action is to specify a pool of idle nodes that are accessed by the scheduler when the workload of a cluster changes. To achieve this, the scheduler utilizes workload prediction algorithms to determine when idle nodes are required to run a queued workload and switches them on accordingly by taking into consideration the time the node needs to boot up. Initially all nodes in the pool will be idle and switched on for instant workload response, but when nodes have not been requested after a specified time, the nodes go to the off state and power down. The status of the idle pool is transparent to the end users and workload. If service level agreements (SLAs) are in place, the idle pool can be dynamically adjusted based on the requested quality of service (QoS) level. To maximize the number of nodes that are powered off, and if memory bandwidth requirements allow it, jobs can be densely packed together on nodes instead of running all on separate nodes. Checkpoint/restart job migration strategies can be part of this scheme to make sure that power consumption is minimized.

A second method of energy conservation is to utilize cost-aware scheduling. The scheduler needs to be made aware of expensive peak and cheaper off-peak electricity rates and the times these rates are in effect. Only time-critical workloads are scheduled during the more expensive time periods where the peak electricity rate is in effect, while other, less time-critical workloads are deferred to off-peak time periods. This type of scheduling can be extended to incorporate summer and winter rates which are used by many utility companies to provide seasonal discounts to their customers. If an organization operates in multiple geographically dispersed data centers or co-locations, one could go a step further and migrate workloads to the least expensive data center based on the time of day at those locations and the electricity rates that are in effect in these locations. Tak-

ing advantage of any significant rate differences from these distinct locations, countries, and even continents, additional cost savings can be realized. From a user perspective, as long as the input and output data sets are readily available, it would not make a difference if their workload is run in New York or in Hong Kong, but from a energy cost perspective it makes a significant difference.

Gartner's research, performed in 2007, shows that 47% of data centers in the US will run out of power by 2008 [17]. For such data centers, Moab can be instructed to utilize daily limits based on watts per user or group. Again, these can vary for different times of a day and different seasons.

Moab's ability to learn application behavior can be utilized to find out on which systems certain applications use minimum power with acceptable performance. The data that is gathered during the learning process can then be used to create the system mapping which defines the optimal performance/power ratio for each application. This mapping can subsequently be used during workload scheduling.

Not only can Moab make scheduling decisions to optimize power usage in a data center, but it also can be configured to make sure that any localized hot spots are minimized in a computing facility. Hot spots occur when certain systems or nodes are highly utilized, but do not receive sufficient cooling to keep temperature at an acceptable level. It is important that any server or data center issues leading to recurring hot spots are investigated as soon as possible, because over-temperature events in a server can be directly correlated to failure rates of systems. A node that runs hot all the time has a larger probability of component failures then a node that is kept within its specified temperature range. Most modern server systems have service processors that allow real-time remote monitoring of temperatures and fan speeds. This data can be used to set limits which can then be used to make intelligent scheduling decisions to resolve or minimize the effects of hot spots. For example, several parts of a workload can be distributed over certain nodes to best balance the heat dissipation in the data center. If a raised floor environment is used to supply server racks with cool air, often the systems that are mounted higher in a rack receive less cool air and thus run hotter. A scheduling policy can be implemented, based on data gathered during a learning process, that will always schedule workloads with lower processor utilization (and thus generate less amounts of heat) on nodes that are located in these higher temperature areas of a rack or data center.

In this section we have discussed a number of ways to schedule workload with the goal of lowering power consumption of and removing hot spots in a data center. The next section shows a detailed case study that uses a number of these methods in a data center facility.

## 4 Case Study

Let's consider this hypothetical scenario: Doodle Computing Inc. owns a co-location data center with four large cluster systems which are used by many manufacturing companies to run Computer Fluid Dynamics (CFD) and Finite Element Analysis (FEA) studies to design and optimize their products. Figure 1 shows the layout of the data center in detail. As shown in the figure, the cluster racks are laid out in a hot/cold aisle configuration to create an effective separation of hot and cold air. The Computer Room Air Conditioning (CRAC) units are placed at the end of the hot aisles to provide a short distance for the hot waste air to return to the unit. The cold aisles contain perforated floor grates to allow the cold air from underneath the raised floor to come up to the air intake of the servers in the racks.

Doodle Computing Inc. currently owns four different clusters: A, B, C, and D, all with approximately the same processor speeds, but with different power efficiencies. Table 2 shows the number of nodes/blades each cluster contains and how much power each of these nodes/blades consumes.

| Cluster | Number of Nodes | Power Usage / Node (Watts) |
|---|---|---|
| A | 1040 | 700 |
| B | 560 | 500 |
| C | 280 | 500 |
| D | 2688 | 1000 |

Table 2: Cluster Properties

Clusters B and C are some years older and contain a mix of 1U and 2U nodes with dual single-core processors per node. Cluster A is only one year old and consists of high-density blades with two dual-core processors each. As shown in the figure, one of the racks of cluster A runs a little hot (depicted by the red dot). Cluster D

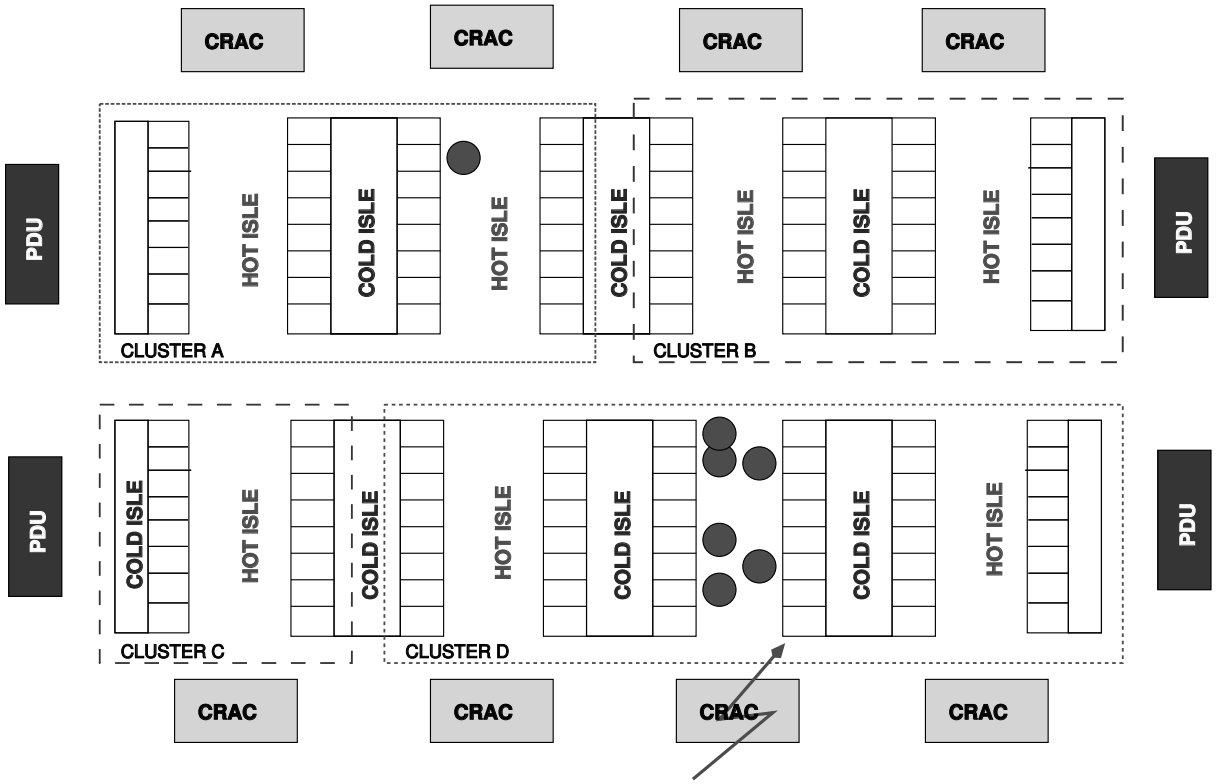


Figure 1: Doodle Computing Inc.'s co-location data center layout

is the newest cluster in the data center and consists of very high density blades with four quad-core processors each. Due to the high power usage of the nodes and the fact that the CRAC unit serving that aisle is broken and does not cool the air sufficiently any more, there are a number of hot spots in the second hot aisle of that cluster (see Figure 1 for details). The monthly energy bill for Doodle Computing Inc. of approximately $160,000 has led the management team to believe that savings should be possible if the data center can more intelligently manage the resources. Doodle Computing Inc. has standardized on Moab and TORQUE as their scheduler and resource managers. The servers and cluster nodes of clusters A, B, and D have a BMC interface and provide temperature and power information to the outside world, while cluster C only reports temperature. To make it possible to monitor the power usage of this cluster, the facilities department has installed IP-enabled rack PDUs in the racks of cluster C. This type of PDU can report on the power usage of every outlet and submit that information as a reply to an HTTP request.

There are two different electricity rates in effect in the location where Doodle Computing Inc. operates their co-location data center: $0.10/kWh from 7 AM to 7 PM and $0.05/kWh from 7 PM to 7 AM. By implementing several of the green policies that were discussed in Section 3.3 of this paper, significant power savings can be achieved and the observed problem with hot spots in the Doodle Computing Inc. co-location data center is resolved. We will show in the remainder of this section how this is done.

Moab has knowledge of the power usage of compute nodes in all of the clusters and thus it can make intelligent decisions where the incoming workload has to be scheduled to optimize power usage. For our scenario, let's consider a job that runs optimally on 64 processors or processor cores, has an approximate running time of 4 hours, and whose result has to be available within 24 hours of submission. For such a job, Moab can calculate the respective cost of running the job on each cluster as follows:

- Cluster A 16 nodes of four cores, each is needed to run the job. This represents a power usage of 16*700 = 11.2 kW. For a four-hour run, 4*11.2

= 44.8 kWh is needed. The cost of this run is 44.8*$0.10 = $4.80.

- Cluster B and C 32 nodes of two processors, each is needed to run the job. This represents a power usage of 32*500 = 16 kW. For a four-hour run, 4*16 = 64 kWh is needed. The cost of this run is 64*$0.10 = $6.40.
- Cluster D for nodes of 16 cores, each is needed to run the job. This represents a power usage of 4*1000 = 4 kW. For a four-hour run, 4*4 = 16 kWh is needed. The cost of this run is 16*$0.10 = $1.60.

It is obvious that the job most efficiently runs on cluster D. If a node on cluster D is not available, cluster A is the next choice, followed by clusters B or C. Because the result of the job only has to be available in 24 hours, even more cost savings can be made by scheduling this job at a different time of the day, e.g., after 7 PM when the resource cost is lower. By running the job at night, the cost on cluster D decreases to a mere 80 cents ($2.40 on cluster A and $3.20 on cluster B and C).

For all clusters, idle pools are created for nodes that have been switched off. Moab can instruct clusters to switch off any compute nodes that have not been utilized in the past hour and for which no reservations or jobs in the queue exist. For example, assume that at 9 PM cluster A is 50 percent utilized, cluster B is 40 percent utilized, cluster C is not utilized at all, and cluster D is 80 percent utilized. Also assume that reservations exist that utilize all of the clusters at 100% starting from 7 o'clock the next morning. In such a situation, half of the nodes of cluster A (520 nodes) can be switched off and move to the idle pool until 7 AM. This represents a cost saving of: 520*700W*10h*0.05 = $182. In addition, 60 percent of the nodes in cluster B (336 nodes) can be moved to the idle pool which represents a cost saving of 336*500W*10h*0.05 = $84. All the nodes in cluster C (280 nodes) can be moved to the idle pool for a saving of 280*500W*10h*0.05 = $70. Last but not least, 20% of the nodes in cluster D (538 nodes) can be moved to the idle pool for a cost saving of 538*1000W*10h*0.05 = $269. This means that a total of $605 is saved by switching off compute nodes and move them into the idle pool for 10 hours.

To solve the hot-spot problems, Moab is provided with node temperature limits and instructed to assess the temperature output of the node BMCs on a regular basis. This way, if it finds that the temperature of the nodes in the second hot aisle of cluster D is too high, the workload can be migrated to cooler nodes in the same cluster. If D nodes are not be available, the workload can be migrated to cluster A, B, or C. Moab can react even more proactively if it has access to data from the CRAC units so it can react to any problems with those units. In this case, the workload can be migrated more promptly making sure that hot spots do not get a chance to occur and nodes in that hot aisle are switched off or turned to sleep if the CRAC unit has an extended problem. The hot spot found in cluster A can be taken care of in a similar way.

## 5 Conclusion

In this paper we present several techniques for monitoring and controlling power consumption and temperature. Through a case study we show how these tools can be practically deployed in a data center facility. In particular we show how real-time monitoring and intelligent scheduling of workload can be efficiently utilized to lower the energy cost of data centers. In the scenario we have used, we also show ways to limit or remove temperature hot spots.

## References

[1] EPA Report to Congress on Server and Data Center Energy Efficiency, 2007, `http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf`

[2] Uptime Institute, `http://uptimeinstitute.org`

[3] Ganglia, `http://ganglia.sourceforge.net`

[4] MOAB, `http://www.clusterresources.com/pages/products/moab-cluster-suite.php`

[5] Wikipedia Green Computing, `http://en.wikipedia.org/wiki/Green_computing`

[6] The Green Grid, `http://www.thegreengrid.org`

[7] Green Grid Metrics: Describing datacenter power efficiency, 2-20-2007, http://www.thegreengrid.org/gg_content/Green_Grid_Metrics_WP.pdf

[8] Less Watts: Saving Power with Linux on Intel Platforms, http://www.lesswatts.org/

[9] Less Watts Whitepaper, http://oss.intel.com/pdf/lesswatts_whitepaper.pdf

[10] ACPI specification Version 3.0b, 10/6/2006, ACPI Working Group, http://www.acpi.info/spec.htm

[11] Intel Corporation. Dual-Core Intel Xeon Processor LV and ULV Datasheet. http://download.intel.com/design/intarch/datashts/31139101.pdf, September, 2006.

[12] J.S. Chase, D.C. Anderson, P.N. Thakar, and A.M. Vahdat, Managing Energy and Server Resources in Hosting Centers, Proc. 18th Symp. Operating Systems Principles, ACM Press, 2001, pp. 103–116.

[13] K. Rajamani and C. Lefurgy, On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters, Proc. IEEE Intl Symp. Performance Analysis of Systems and Software, IEEE CS Press, 2003, pp. 111–122.

[14] M. Elnozahy, M. Kistler, and R. Rajamony, Energy Conservation Policies for Web Servers, Proc. 4th Usenix Symp. Internet Technologies and Systems, Usenix Assoc., 2003, pp. 99–112.

[15] X. Fan, W. Weber, and L.A. Barroso, Power Provisioning for a Warehouse-sized Computer, Proc. of the ACM International Symposium on Computer Architecture, San Diego, CA, June 2007.

[16] E. Pinheiro, R. Bianchini, E.V. Carrera, and T. Heath, Dynamic Cluster Reconfiguration for Power and Performance, Compilers and Operating Systems for Low Power, Kluwer, 2003, pp. 75–94.

[17] Gartner: 2007 Data Center Conference Poll Results for Power and Cooling Issues.

[18] ASHRAE, http://www.ashrae.org/

[19] Climate Savers Initiative, http://www.climatesaverscomputing.org/