Power and Energy Management for Server Systems [*]

Ricardo Bianchini[†] and Ram Rajamony[‡]

[†]Department of Computer Science
Rutgers University
Piscataway, NJ
ricardob@cs.rutgers.edu

[‡]Low-Power Computing Research Center
IBM Austin Research Lab
Austin, TX
rajamony@us.ibm.com

Technical Report DCS–TR–528, June 2003

Abstract

Power and energy consumption have recently become key concerns for high-performance servers, especially when they are deployed in large cluster configurations as in data centers and Web hosting facilities. Research on power and energy management for these servers can ease installation, reduce costs, and protect the environment. Given these benefits, researchers have made important strides in conserving energy in these servers. Inspired by this initial progress, researchers are delving deeper into this topic. In this paper, we detail the motivation for this research, survey the previous work, describe a few ongoing efforts, and discuss the challenges that lie ahead.

1 Introduction

Power and energy consumption have always been critical concerns for laptop and hand-held devices, as these devices generally run on batteries and are not connected to the electrical power grid. As a result, a tremendous amount of research has been directed towards low-power and low-energy design and conservation (e.g., [11, 16, 19, 20, 23, 35]).

In recent years, researchers have realized that power and energy management are also critical for servers, even though these systems are connected to the electrical power grid. The reason for this new focus is that each improvement in server hardware performance comes at increasing power and energy consumption costs. Even worse, these servers are often replicated to form large server clusters, such as those that support most data centers, hosting centers, and a multitude of Internet companies.

In this paper, we address power and energy management for such server systems. More specifically, we define a "server system" (or simply, a "server") as a combination of software and hardware that services requests coming from remote clients. Power and energy management techniques that are not explicitly directed to these servers and their workloads are beyond the scope of the paper.

Both power and energy consumption are of concern in server systems. The energy consumption of a data center's servers, interconnects, and cooling infrastructure dictates the center's electricity costs. These costs can be significant for a large and/or dense server cluster in a heavily air-conditioned room. In contrast, the system's peak power consumption dictates the required cooling configuration and backup power infrastructure. Systems with low peak power requirements can depend on in-server heat dissipation techniques to move the heat to the point where it can be dissipated by the data center's cooling infrastructure. Systems with

[*]This work has been supported in part by NSF under grants # EIA-9986046, EIA-0224428, and CCR-0238182 (CAREER award) and by DARPA under contract F33615-00-C-1736.

high peak power demands require complex cooling configurations to efficiently move heat away from the server components, as well as higher capacity cooling systems. High peak power requirements also translate into large uninterruptible power supplies (UPSs) and backup power generators, both of which are necessary in case of a power outage. High power densities also pose serious cooling and reliability problems. The use of "blade systems" in server clusters exacerbates this power density problem, since large numbers of blades can be packed into a small volume. Thus, the power and energy requirements of server systems play an important role in determining both the fixed and variable (operational) costs of data centers.

Data from several sources (e.g., [27, 33]) show that power and energy costs represent a significant fraction of the cost of data and hosting centers. Perhaps more importantly however, power and energy management can help protect the environment, since most power-generation technologies (such as nuclear and coal-based generation) have a negative impact on the environment. We discuss the motivation for the research on power and energy management for servers further in the next section.

Given the immediate potential benefits of this research and the extensive previous work on power and energy management for battery-operated devices, a natural approach would be to leverage this previous work. Unfortunately, it is not always ideal (or even possible) to leverage the management techniques used for battery-operated devices in the context of servers. In particular, management techniques for server systems have to take into account the high consumption of system components, such as power supplies, disk arrays, and interconnection switches, that are not present in battery-operated devices. Furthermore, the intensity of busy server workloads often make it infeasible to move components to low-power states (e.g., by turning them off). We detail the different facets of power and energy management for servers in section 3.

Realizing the differences between portable and server class workloads and operating environments, researchers have developed management strategies tailored specifically for servers. A few research efforts [7, 30, 31] have examined energy management strategies in server clusters. These efforts tackled the high power supply losses of current servers, by dynamically reconfiguring (or shrinking) the cluster to operate with fewer nodes under light load. Other efforts [3, 13, 14] tackled the high amounts of energy consumed by server CPUs. Their approach was to conserve energy by using either dynamic voltage scaling or request batching under light load. Finally, a few research efforts [6, 8, 17, 29] addressed the energy consumption in the storage subsystem of data-intensive servers. We discuss these previous efforts in more detail in section 4.

Even though these efforts have made important strides in conserving energy in high-performance servers, there is still much to be done. Our groups are currently addressing two issues in particular: (1) how to conserve power and energy in heterogeneous server clusters comprised of a combination of traditional and blade servers; and (2) how to exploit information from the service level, such as request priorities established by service-level agreements, to increase the benefits of request batching. We discuss these two current efforts and some of the remaining challenges of this research topic in section 5.

2 Motivation

High power consumption. Power consumption is an important concern in the context of servers as it directly influences their cooling requirements. In fact, a medium to large number of servers racked closely together in the same room, as is usually the case of server clusters, requires sophisticated racks and heavy-duty air conditioning systems. For instance, as of Spring/2001, at each Google site there were 40 racks, each of which with 80 PCs, for a total power consumption of nothing less than 180 KWatts [28]. (Today, each Google site consumes significantly more than this, but more detailed information is not publicly available.) Cooling such a large installation is an expensive challenge. Moreover, power consumption also influences the required investments in backup cooling and backup power generation for server clusters that can never be unavailable, such as those of Internet companies.

Power density is also becoming an important issue for server systems. As microprocessor feature sizes decrease, the same die area must now dissipate more heat. This increased power density can also lead to reliability problems if the heat is not quickly dissipated [34]. The use of “blade systems” in server clusters exacerbates this power density problem. A blade server typically crams components onto a small single-board footprint, enabling large numbers of servers to be packed into a small volume. In addition to increasing the peak power requirement, such dense packing also leads to increased demands on the data center’s cooling infrastructure.

Taking a broader perspective, power management is an important goal in that most power-generation technologies (such as nuclear and coal-based generation) have a negative impact on the environment. The Energy Star program [1] of the Environmental Protection Agency was the first to promote power-efficient computers due to environmental considerations. Unfortunately, power efficiency had not become an issue for the server market until recently.

High energy consumption. Besides power consumption, energy consumption is also important for servers. Both the computational and the air conditioning infrastructures consume energy. This energy consumption is reflected in the electricity bill, which can be significant for a large and/or dense server cluster in a heavily air-conditioned room. Take the example of a single 200-Watt server, such as the IBM 1U x300. In a year, such a server consumes 1752 KWh ($200 \times 24 \times 365$) of energy. A cooling unit with a common Energy Efficiency Rating (EER) of 12 cools 12000 British Thermal Units (BTU) for 1 KWh; 1 KWh = 3414 BTU. This unit would then consume 498 KWh ($1752 \times 3414/12000$) in cooling this server for the year. This amounts to 22% of the total energy. Assuming electricity costs of U$ 0.08/KWh, the total energy cost for this single server would be U$ 180/year.

Given that data centers, hosting centers, and other Internet companies often involve thousands of these servers, as well as storage servers and networking infrastructure to connect the server cluster to the Internet, it is easy to see that the cost of electricity can become significant. For instance, a Google site in Spring/2001 consumed no less than 130 MWh per month (1560 MWh or U$ 125K per year, at U$ 0.08/KWh), just with the server infrastructure.

As another example, consider a hosting center. Hosting centers typically bundle energy costs into the cost they charge their costumers. For instance, the average price to rent a full rack of space (about 3ft × 3ft × 6ft) at a hosting center is approximately US$1000 per month (e.g., [10, 25]). For this price, the hosting center provides physical space, power, backup power (UPS), cooling, and support services (offices). Bandwidth is typically an extra charge, and ranges widely from $200 to $650 per month per T1 line (1.544Mbps). While the amount of power provided per rack varies from one hosting center to another, all centers provide at least 4 KWatts of power per rack, with some delivering up to 7 KWatts. Thus, with energy costs averaging 8 cents per KWh, rack energy usage alone could account for up to 23% to 40% of colocation *revenue* (excluding bandwidth). Similar arguments apply for managed hosting (where the hosting center controls and owns the equipment and provides the service) as well. Consequently, there is a strong incentive to minimize server energy usage in hosting centers.

High consumption even in next-generation server clusters. In light of the key requirement in server clusters, i.e. acceptable performance, these systems can either be comprised by fewer high-end servers or more numerous low-end, possibly blade, servers. In the first scenario, the increasing complexity (and power consumption) of high-end server hardware and software will only increase the importance of management research in the future. However, we believe the second scenario to be highly probable also, given that blade clusters are typically easier to manage than traditional clusters. The current wide variety of industry efforts to produce blade systems seems to corroborate this observation.

Nevertheless, the role that clustered blades will play in the marketplace is still unclear. One of the roles blades might play is to allow new server clusters to retain the same capacity of traditional clusters, but in less physical space. This will alleviate the power and energy consumption problem by a one-time, fixed factor, but will increase the power density of these installation. However, one of the main uses of clustered blades will likely be to increase system capacity in the same amount of space as traditional clusters. In this latter scenario, the power and energy consumption problem will become even more serious. Again, power density will become a serious problem as well.

3 Background

Power management mechanisms. Essentially, power management is done by transitioning hardware components back and forth between high and low-power states or modes. In high-power mode, components are fully active and operational. The functionality associated with the low-power modes depends on the particular component. Regardless of the particular functionality, it is usually quite expensive to change power modes in terms of both energy and performance. Thus, management techniques must carefully consider the implications of mode transitions before actually effecting them. To illustrate these issues more concretely, we will focus the following discussion on two types of hardware components, namely microprocessors and disks. Similar observations can be made of other types of components.

Some current microprocessors (e.g., the Transmeta Crusoe$^{\mathrm{TM}}$ [19]) allow power management by Dynamic Voltage Scaling (DVS). DVS relies on the fact that the dynamic power consumed by the microprocessor is a quadratic function of its operating voltage. Thus, reducing the operating voltage (and consequently the operating frequency) provides substantial savings in power at the cost of slower program execution. The number of low-power modes (i.e., the number of different scaled voltages and frequencies) and the transition costs vary widely with microprocessor.

Other microprocessors (e.g., the Intel Pentium-IV$^{\mathrm{TM}}$ processor [9]) allow power management by halting and/or deactivation. In contrast with DVS-based microprocessors, no useful work can be performed in the halted or inactive low-power modes. Halting the microprocessor stops it from executing any instructions and therefore reduces the amount of internal activity. Deactivation sends the microprocessor to an even deeper low-power mode, directly addressing the static power requirements of the microprocessor. A specific set of signals needs to be delivered in order to re-awaken the processor. Again, the transition costs vary with microprocessor but are typically in the 1-3 millisecond range.

Current disks also allow for power management through deactivation, often exhibiting multiple inactive modes. In high-power mode, or active mode, the disk is being actively used and consumes the most power. In idle mode, the disk is still spinning at its regular speed and accesses can be performed without delay. Other low-power modes involve high transition overheads, as they involve turning the spindle motor off (standby) and turning the disk interface off (sleep). The transition overheads depend on the particular disk.

Based on these mechanisms, several energy management techniques (or policies) have been proposed for battery-operated devices. When hardware components can still operate in low-power modes, the techniques typically send components to the lowest power mode that will not compromise performance excessively, provided that transition costs can be amortized (e.g., [21, 35]). When hardware components need to be deactivated for energy savings, the techniques typically send components to lower power modes after periods of inactivity (e.g., [12, 24]) or based on high-level information [20, 36]. The length of inactivity periods depends on the cost of the mode transitions.

There has also been some research exploring power management mechanisms to limit thermal dissipation in microprocessors. Brooks and Martonosi propose the use of on-chip temperature sensors, activity (performance) counters, dynamic profile analysis, or compiler inserted triggers to detect a “thermal emergency”. Once an emergency is detected, they propose using mechanisms such as frequency scaling, DVS, and decode throttling to limit short-term energy consumption [5].

Exploiting mechanisms in server systems. Unfortunately, the techniques discussed above are not always appropriate for servers. For example, busy servers often cannot afford to send their hardware components to low-power modes due to the resulting performance degradation. Even in relatively lightly loaded servers, components such as disks need to remain inactive for a long time (on the order of several tens of seconds for high-performance disks), if their mode transition overheads are to be amortized. Servers are rarely idle for such long periods of time.

To make matters worse, servers pose a few new problems: (1) they are provisioned for peak load, meaning that their hardware components typically exhibit high performance and, thus, high power consumption; (2) popular servers rely on widespread replication of resources (such as clusters of machines and disk arrays) for high availability and high bandwidth; (3) their power supplies typically exhibit high power losses, as they have to store spare capacity to deal with sudden spikes in load; and (4) server systems often involve components, such as disk arrays and interconnection switches, for which no management techniques had been proposed.

Given the characteristics of servers and their loads, power and energy management requires new ideas. Fortunately, some of these same characteristics can be exploited to manage power and energy in a different way. In particular, researchers have exploited the wide variations in the amount of load offered to servers to propose techniques such as multi-speed disks and coordinated DVS for server clusters. These load variations and the replication of resources have motivated proposals to concentrate load onto a subset of resources, so that other resources can be turned off. The request patterns of server loads, in terms of both frequency and recency of access, have motivated work on energy management for disk array-based servers. Finally, the wide-area network delays involved in accessing servers have motivated strategies that degrade response time slightly in favor of energy conservation, such as request batching. In the following section, we describe these previous works in detail.

4 Previous Work: Management Techniques for Servers

The previous work has focused almost exclusively on energy management techniques. Next, we divide these techniques into two groups: those that relate to stand-alone servers and those that focus on server clusters.

4.1 Stand-Alone Servers

Front-end servers: DVS and request batching. Elnozahy et. al. describe three techniques designed to reduce energy consumption in stand-alone Web servers [14]. The techniques employ two power management mechanisms: dynamic voltage scaling (DVS), and a new mechanism introduced in their paper called request batching. The first technique uses DVS in isolation, except that it extends recently introduced task–based DVS policies [15, 26] for use in server environments with many concurrent tasks. The DVS technique conserves the most energy for intermediate request rates (workload intensities).

The second technique uses request batching to conserve energy during periods of low workload intensity. In this technique, the incoming requests are accumulated in memory by the network interface processor, while the host processor of the server is kept in a low-power state, such as Deep Sleep [9]. The host processor is awakened when an accumulated request has been pending for longer than a *batching timeout*. Request batching conserves the most energy for low workload intensities.

Finally, the third technique uses both DVS and request batching mechanisms to reduce processor energy usage over a wide range of workload intensities. When there are no pending requests, the combined technique places the in DeepSleep mode. When the processor is activated, it is set to operate at the lowest possible operating frequency and the DVS technique takes over.

All the techniques trade off system responsiveness to save energy. However, the techniques employ the mechanisms in a feedback–driven control framework in order to conserve energy while maintaining a specified quality of service level, as defined by a percentile–level response time. The techniques are evaluated using Salsa, a Web server simulator that has been extensively validated for both energy and response time against measurements from a commodity Web server. Three day-long static Web workloads from real Web

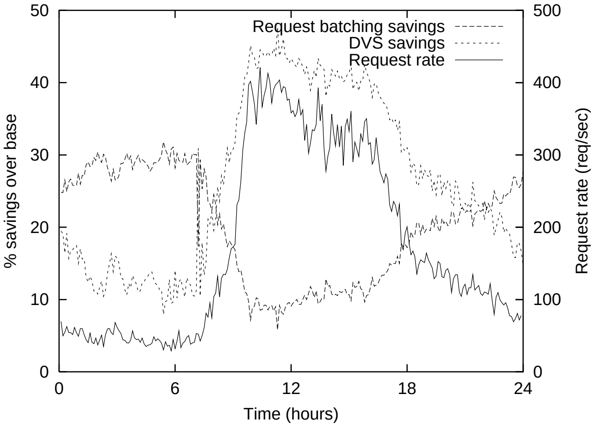

Figure 1: Energy savings with a 90th–percentile response time goal of 50ms for Finance, superposed with the request rate.

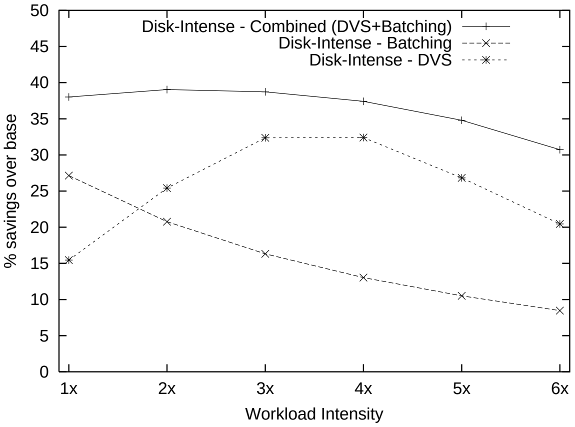

Figure 2: Energy savings with a 90th–percentile response time goal of 50ms for Disk-Intense over a range of intensities.

server systems are used to quantify the energy savings: the Nagano Olympics98 server, a financial services company site, and a disk-intensive workload. The results show that when required to maintain a 90th–percentile response time of 50ms, the DVS and request batching techniques save from 8.7% to 38% and from 3.1% to 27%, respectively, of the CPU energy used by the base system. The two techniques provide these savings for complementary workload intensities. The combined technique is effective for all three workloads across a broad range of intensities, saving from 17% to 42% of the CPU energy. Figures 1 and 2 show the impact of these techniques on Finance and Disk-Intense [14].

Storage servers: Multi-speed disks, MAID, and PDC. Carrera *et al.* [6] and Gurumurthi *et al.* [17] considered multi-speed disks for stand-alone servers. The idea behind these works is to set the speed of the disk (and, as a result, its energy consumption) dynamically, according to the load imposed on the disk. Gurumurthi *et al.* [17] introduced performance and power models for multi-speed disks, proposed a policy based on disk response time to transition speeds dynamically, and discussed multiple implementation issues. Using simulation and synthetic workloads, they showed that multi-speed disks can provide energy savings of up to 60%.

Carrera *et al.* [6] studied four disk energy management techniques, including the combination of laptop and SCSI disks and simple two-speed disks. Using a real kernel-level implementation and real Web and proxy workloads, they showed that the combination of laptop and SCSI disks can reduce energy consumption by up to 41%, but only for over-provisioned servers. Using emulation and the same workloads, they also found that two-speed disks (15K and 10Krpm in their experiments) can reduce energy consumption by about 20% for properly-provisioned servers and a range of hardware and software parameters.

Figures 3 and 4 illustrate some of their results. Figure 3 shows the server and disk throughputs for a real (but accelerated) Web trace. The figure shows a common behavior, namely an alternation of server load peaks and valleys with lighter loads on weekends. The disk loads follow the same trend, but are more bursty. Figure 4 depicts the disk power consumption for a server with a high-performance disk (labeled “Traditional”) and a server with their two-speed disk (labeled “Two-speed”). The results in this figure show that the server with a traditional high-performance disk consumes 14.8 KJ of disk energy on this workload. The two-speed results show that the two-speed disk switches to 15K rpm only three times during the whole experiment. The two-speed disk consumes 11.6 KJ of disk energy, leading to a savings of 22%.

In terms of disk array-based servers, Gurumurthi *et al.* [18] considered the effect of different RAID parameters, such as RAID level, stripe size, and number of disks, on the performance and energy consumption

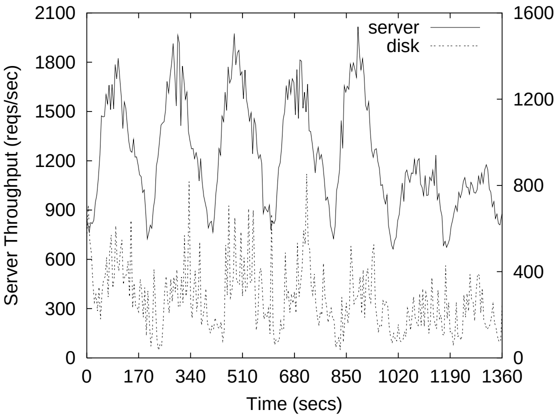

Figure 3: Server and disk throughputs.

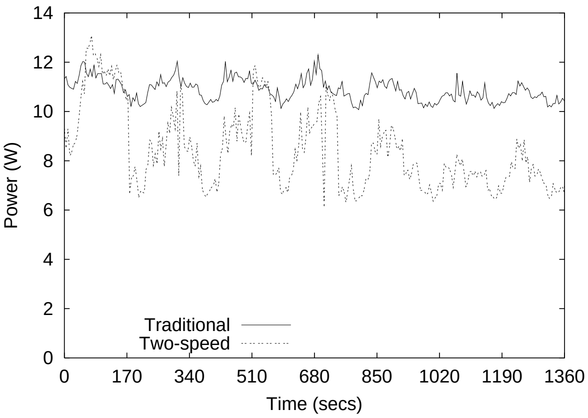

Figure 4: Power of traditional and two-speed disks.

of stand-alone database servers running transaction processing workloads.

The Massive Array of Idle Disks (MAID) [8] has been proposed as a replacement for old tape backup archives with hundreds or thousands of tapes. Because only a small part of the archive would be active at a time, the idea is to copy the required data to a set of “cache disks” and spin down all the other disks. All accesses to the archive should then check the cache disk(s) first. Cache disk replacements are implemented using an LRU policy. Replaced data can simply be discarded if it is clean, since its original copy is still stored on one of the non-cache disks. Replaced data that is dirty has to be written back to the corresponding non-cache disk.

Popular Data Concentration (PDC) [29] has been proposed specifically as an energy management strategy for disk array-based servers. PDC is inspired by the heavily skewed file access frequencies of several server workloads (e.g., [4, 22]), where a relatively small number of files is accessed frequently, whereas a large number of files is accessed rarely. The idea behind PDC is to concentrate the most popular disk data (i.e., those that most frequently miss in the main memory cache) by migrating it to a subset of the disks. This concentration should skew the disk load towards this subset, while other disks become idle longer and more often. These other disks can then be sent to low-power modes to conserve energy. More specifically, the goal of PDC is to lay data out across the disk array so that the first disk stores the most popular disk data, the second disk stores the next set of most popular disk data, and so on; the last disk stores the least popular disk data. Because data popularity can change over time, PDC may have to be applied periodically during the lifetime of the server.

PDC and MAID are based on the same general observation, which is that concentrating load on certain resources (and thereby increasing their utilization) increases their power consumption less than linearly, given the fixed power consumed just by having the resource on-line. Furthermore, PDC and MAID have the same objective, namely to increase idle times by moving data around the disk array and spinning disks down. As a result, both techniques sacrifice the access time of certain files in favor of energy conservation. However, instead of relying on file popularity and migration like PDC, MAID relies on temporal locality and copying to conserve energy.

Pinheiro and Bianchini [29] presented a quantitative comparison of PDC and MAID when applied to a file server for a wide range of workload and server parameters. Their simulation results showed that it is only possible to conserve energy during periods of very low load on the server, such as during the night or on weekends, regardless of the approach to conservation. They also found that the PDC-based file server can deal more gracefully with increases in file system coverage and decreases in file popularity than the MAID-based server. In contrast, the MAID-based server benefits the most from increases in temporal locality.

Hot servers: Throttling. Bellosa has developed an operating system technique to keep average power consumption, and therefore thermal dissipation, in server systems within limits [2]. He uses an infrastructure called Joule Watcher to infer energy consumption levels using the processor performance counters [2]. At periodic intervals, the energy consumption over the past interval is compared to the desired consumption level. If the processor consumed more energy than permitted, halt cycles are introduced to temporarily place it in a low-power state. This throttling technique has been implemented in the Linux operating system and has been shown to maintain the average power consumption a Web server within specified goals.

4.2 Server Clusters

Front-end server clusters: Cluster reconfiguration and CVS. Pinheiro *et al.* [30, 31] and Chase *et al.* [7] concurrently proposed similar strategies for managing energy in the context of front-end server clusters. Pinheiro *et al.* called the technique Load Concentration (LC). The basic idea behind LC is to dynamically distribute the load offered to a server cluster so that, under light load, some hardware resources can be idled and put in low-power modes, e.g., can be turned off. Like in PDC and MAID, the key here is that this type of concentration is usually beneficial in terms of overall power consumption. Under heavy load, the inverse operation should be performed, i.e., we should reactivate components and redistribute the load to alleviate or eliminate any performance degradation.

Pinheiro *et al.* developed an LC-based cluster of front-end Web servers (as well as an LC-based cluster of cycle servers). Because the hardware and data resources of front-end servers are replicated at all nodes and the power supply loss in traditional servers is very high (about 48 Watts for their servers), the system dynamically turns entire nodes on and off, in effect reconfiguring the cluster. The key component of their system is an algorithm that considers the total load imposed on the cluster and the expected throughput and power consumption of different cluster configurations. This cluster reconfiguration algorithm is run by a "master" node that receives load information from all other nodes.

Figure 5 illustrates the behavior of their system for a 7-node cluster running a real Web trace. The figure plots the evolution of the cluster configuration and offered loads on each resource, as a function of time in seconds. The load on each resource is plotted as a percentage of the nominal throughput of the same resource in one node. The parameters for the cluster reconfiguration algorithm are set to guarantee quick reaction to fluctuations in load, so that the system can tackle significant increases in load without performance degradation. The figure shows that for this workload the network interface is the bottleneck resource throughout the whole execution of the experiment (140 minutes). The traffic directed to the server initially increases slowly, triggering the addition of a node, before increasing substantially and triggering the addition of several new nodes in quick sequence. The traffic then subsides, until another period of high traffic occurs, which is followed by a substantial decline in traffic.

Figure 6 presents the power consumption of the whole cluster for two versions of the same experiment, as a function of time. The lower curve (labeled "Dynamic Configuration") represents the power-aware version of the system, in which the cluster configuration is dynamically adapted to respond to variations in resource demand. The higher curve (labeled "Static Configuration") represents a static cluster configuration fixed at 7 nodes. As can be seen in the figure, the dynamic system can reduce power consumption significantly for most of the experiment. Calculating the area below the two curves, we find that the dynamic server saves 38% in energy. Similar results were accrued for the clustered cycle servers.

Rajamani and Lefurgy have studied how to improve the energy saving potential of the cluster reconfiguration technique by using spare servers and history information about peak server loads [32]. They also modeled the key system and workload parameters that influence the cluster reconfiguration technique.

Elnozahy *et al.* [13] evaluated different combinations of cluster reconfiguration and dynamic voltage scaling for clusters in which power supplies are more efficient (i.e., power supply losses are relatively low). Their work proposed independent voltage scaling (IVS) and coordinated voltage scaling (CVS). In IVS, each server node makes its own independent decision about what voltage and frequency to use, depending on the

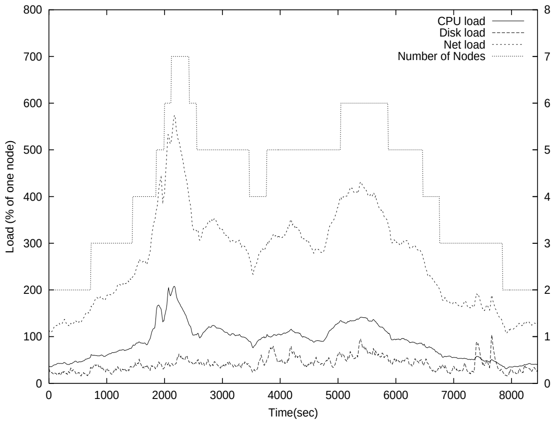

Figure 5: Cluster evolution and per-resource offered loads.

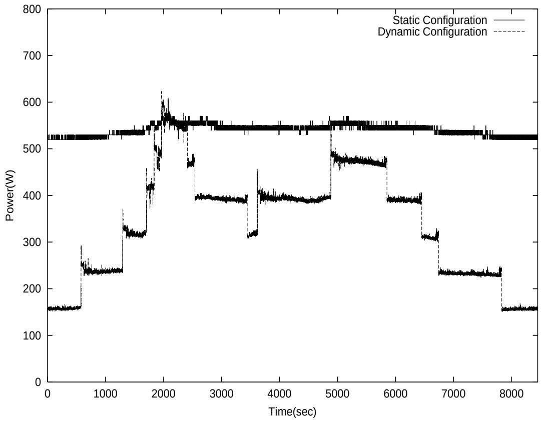

Figure 6: Power under static and dynamic cluster configurations.

load it is receiving. In CVS, nodes coordinate their voltage and frequency settings in order to optimize the overall energy consumption. They showed that CVS conserves slightly more energy than IVS. However, the additional benefits achieved by CVS come at the expense of increased implementation complexity. They also showed that, depending on workload, either CVS or cluster reconfiguration is the best technique in the presence of low supply losses. For the workloads they examined, the combination of these techniques is always the best approach.

5 Current and Future Work

5.1 Exploiting Service-Level Information

The previous work on power and energy management for servers has not considered exploiting service-level information to increase savings. An example of useful service-level information is request priorities.

Request priorities can help keep resources on low-power modes longer and more often. For example, we can prevent a low-priority request from activating a resource (a disk or the CPU, say) by blocking the request until the activation can be amortized over multiple of these requests or until a timeout expires. In effect, this strategy trades off higher non-premium request response times for lower energy. The reason why this tradeoff is acceptable is two-fold: (1) as we mentioned before, the latency of the wide-area network overwhelms relatively short delays at servers; and (2) non-premium requests usually do not provide response time guarantees to clients.

Researchers at the IBM Low-Power Computing Research Center and at the Rutgers DARK Lab are exploiting request priority information to increase the CPU energy savings achievable by request batching [14]. In particular, they are assessing the range of priority distributions for which service-level information provides gains and quantifying these gains. When non-premium requests dominate, it is possible to aggressively increase the batching timeout, thereby conserving more energy. Under such a scenario, request batching should be able to conserve significant energy over a much wider range of workload intensities.

5.2 Heterogeneous Server Clusters

The previous work on power and energy management for server clusters has focused solely on homogeneous systems. However, real-life clusters are almost invariably heterogeneous in terms of the performance, capacity, and power consumption of their hardware components. The reason for this is simple and at least two-fold: (1) failed or misbehaving components are usually replaced with different (more powerful) ones, as cost/performance ratios for off-the-shelf components keep falling; and (2) any necessary increases in performance or capacity, due to expected increases in offered load, are also usually made with more powerful components than those of the existing cluster. In essence, the server cluster is only homogeneous (if at all) when first installed. Finally, blade servers are starting to make their way into existing large server clusters. It is unreasonable to expect that these blade servers will replace all the traditional servers of existing large server clusters in one shot. This replacement is more likely to occur in multiple stages, producing clusters that at least temporarily will include nodes of widely varying characteristics.

Heterogeneity raises the interesting problem of how to distribute the clients' requests to the different cluster nodes for best performance. Furthermore, heterogeneity must be considered if we want to conserve energy through cluster reconfiguration [7, 30, 31], raising the additional problem of how to configure the cluster for an appropriate tradeoff between energy conservation and performance.

The DARK group at Rutgers is developing a cluster-based server that can adjust the cluster configuration and the request distribution to optimize power, energy, throughput, latency, or some combination of these metrics. The particular optimization function can be defined by the system administrator; as an example, they are selecting the ratio of cluster-wide power consumption and throughput, so that the system can dynamically produce the lowest power consumption per request at each point in time. Designing such a server is a non-trivial task when nodes are highly heterogeneous, and becomes even more complex when nodes communicate (e.g., when subsets of nodes have different functionalities as in multi-tier e-commerce servers, or when nodes cooperate to share resources such as CPUs, main memory caches, and/or disk storage).

To tackle this design task, they are developing analytical models that use information about the expected load on the cluster to predict the overall throughput and power consumption, as a function of the request distribution. Using the models and an iterative optimization algorithm, their system can evaluate a large space of configurations and distributions to find the one that minimizes the power/throughput ratio for each level of offered load intensity.

5.3 Future Challenges

Power and energy modeling and prediction. As server hardware and software become more power and energy-efficient, future management techniques will need the ability to more carefully evaluate (or predict the effect of) their potential actions. This essentially means that analytical modeling of power and energy consumption will become even more important than it is now. The challenge is that modeling the power and energy consumed by complex servers is not straightforward. Modeling power is potentially simpler provided that one understands the details of the power behavior of the hardware components. In contrast, modeling energy is much harder in that it also involves modeling server performance. With accurate models of power, energy, and performance, the management technique can evaluate the benefits of different settings for the components' power modes or different load distributions, before actually taking any actions.

Peak power management. The work we described in section 4 has only addressed energy management, i.e., it does not reduce the maximum power that can be consumed by the server system. However, power management is critical in the presence of power delivery limitations. A common example of such a limitation affects server clusters during power outages. Backup power generation systems are usually provisioned to provide just a fraction of the power needed by the whole server installations. During the outage, one would like to provide the best possible performance under this fixed and smaller power budget. However, it is challenging to determine what this performance is and what configuration of components' power modes can actually achieve it. The modeling and prediction research that we just mentioned can help determine what configuration can provide the best behavior under this and other constraints.

Temperature issues. Given the high power consumption and thermal dissipation of large clusters of densely packed servers, it is important to design equipment room cooling and ventilation systems to avoid overheat and hardware reliability problems. Even for properly designed cooling and ventilation systems, it may be appropriate to monitor temperatures in different parts of the system and shift load around to achieve the most even temperature distribution. The challenge here is to map and understand the air flow in equipment rooms, achieve accurate temperature monitoring, and tie it all into a system-wide workload balancing framework.

Main memory. Servers often have extremely large main memories to optimize performance. These memories are accessed frequently, since hardware caches are usually much smaller than the working sets of real servers. Thus, the memory energy consumption is an issue that must be dealt with. The challenge in conserving memory energy is in properly laying data out across the memory banks, so that the low-power states can be used.

Network interfaces and cluster interconnects. As far as we know, high-bandwidth network interfaces and cluster interconnects have received no attention so far in the literature. Nevertheless, a high-performance switch can consume a significant amount of power. Our measurements of a 32-port Gigabit Ethernet switch in the Rutgers DARK Lab show that it consumes more than 700 Watts when completely idle. A complete understanding of the power and energy consumption of server clusters clearly requires addressing these components. The challenge here is that the internal architecture of these interconnects is often not described in the public literature, making the task of accurately modeling them extremely complex.

6 Conclusion

In this paper we discussed the technical, financial, and environmental incentives for managing power and energy in server systems. Based on these incentives, researchers in both academia and industry have started to address this topic in the scientific literature. We detailed these contributions as well as some of the ongoing work in this area. Finally, we outlined several key challenges that must be addressed in the future in order to build more power and energy-conscious server systems.

References

[1] Environmental Protection Agency. Energy Star for Your Small Business. http://www.energystar.gov/.

[2] Frank Bellosa. The Case for Event-Driven Energy Accounting. Technical Report TR-I4-01-07, Computer Science 4, University of Erlangen-Nurnberg, 2001.

[3] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony. The Case for Power Management in Web Servers. In Graybill and Melhem, editors, *Power-Aware Computing*. Kluwer Academic Publishers, January 2002.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proceedings of IEEE InfoCom'99*, pages 126–134, March 1999.

[5] D. Brooks and M. Martonosi. Dynamic Thermal Management for High-Performance Microprocessors. In *Seventh International Symposium on High-Performance Computer Architecture (HPCA)*, January 2001.

[6] E. V. Carrera, E. Pinheiro, and R. Bianchini. Conserving Disk Energy in Network Servers. In *Proceedings of the 17th International Conference on Supercomputing (ICS'03)*, June 2003.

[7] J. Chase, D. Anderson, P. Thackar, A. Vahdat, and R. Boyle. Managing Energy and Server Resources in Hosting Centers. In *Proceedings of the 18th Symposium on Operating Systems Principles*, October 2001.

[8] D. Colarelli and D. Grunwald. Massive Arrays of Idle Disks for Storage Archives. In *Proceedings of SC'2002*, November 2002.

[9] Intel Corporation. Pentium 4 technical specifications, 2002. http://developer.intel.com/design/pentium4/-datashts/24988703.pdf.

[10] MZima Corporation. Mzima promotions, June 2003. http://mzimahosting.com/prodserv/promotion1.html.

[11] Fred Douglis and P. Krishnan. Adaptive Disk Spin-Down Policies for Mobile Computers. *Computing Systems*, 8(4):381–413, 1995.

[12] Fred Douglis, P. Krishnan, and Brian Marsh. Thwarting the Power-Hungry Disk. In *Proceedings of the 1994 Winter USENIX Conference*, 1994.

[13] E. N. Elnozahy, M. Kistler, and R. Rajamony. Energy-Efficient Server Clusters. In *Proceedings of the 2nd Workshop on Power-Aware Computing Systems*, February 2002.

[14] M. Elnozahy, M. Kistler, and R. Rajamony. Energy Conservation Policies for Web Servers. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems*, March 2003.

[15] K. Flautner, S. Reinhardt, and T. Mudge. Automatic Performance Setting for Dynamic Voltage Scaling. In *Proceedings of the 7th ACM Int. Conf. on Mobile Computing and Networking (MOBICOM)*, July 2001.

[16] Jason Flinn and M. Satyanarayanan. Energy-aware adaptation for mobile applications. In *Proceedings of the 17th Symposium on Operating Systems Principles*, pages 48–63, 1999.

[17] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: Dynamic Speed Control for Power Management in Server Class Disks. In *Proceedings of the International Symposium on Computer Architecture*, June 2003.

[18] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, H. Franke, N. Vijaykrishnan, and M. J. Irwin. Interplay of Energy and Performance for Disk Arrays Running Transaction Processing Workloads. In *Proceedings of the International Symposium on Performance Analysis of Systems and Software*, March 2003.

[19] T. Halfhill. Transmeta Breaks the x86 Low-Power Barrier. In *Microprocessor Report*, February 2000.

[20] T. Heath, E. Pinheiro, J. Hom, U. Kremer, and R. Bianchini. Application Transformations for Energy and Performance-Aware Device Management. In *Proceedings of the 11th International Conference on Parallel Architectures and Compilation Techniques*, September 2002. Best student paper award.

[21] C-H. Hsu and U. Kremer. The Design, Implementation, and Evaluation of a Compiler Algorithm for CPU Energy Reduction. In *Proceedings of the ACM SIGPLAN Conference on Programming Languages, Design, and Implementation (PLDI'03)*, June 2003.

[22] S. Jin and A. Bestavros. GISMO: A Generator of Internet Streaming Media Objects and Workloads. *ACM SIGMETRICS Performance Evaluation Review*, 29(3), November 2001.

[23] Alvin R. Lebeck, Xiaobo Fan, Heng Zeng, and Carla S. Ellis. Power Aware Page Allocation. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IX)*, pages 105–116, 2000.

[24] Kester Li, Roger Kumpf, Paul Horton, and Thomas Anderson. A Quantitative Analysis of Disk Drive Power Management in Portable Computers. In *Proceedings of the 1994 Winter USENIX Conference*, pages 279–291, 1994.

[25] XMission L.L.C. Xmission colocation account and services, June 2003. http://www.xmission.com/business/colocation.html.

[26] Jacob R. Lorch and Alan Jay Smith. Improving Dynamic Voltage Scaling Algorithms with PACE. In *ACM SIGMETRICS 2001*, June 2001.

[27] J. Mitchell-Jackson. Energy Needs in an Internet Economy: A Closer Look at Data Centers. Master of Science Thesis from the Energy and Resources Group, University of California at Berkeley. July, 2001.

[28] D. Patterson. CS252 Lecture Slides. University of California at Berkeley, Spring 2001.

[29] E. Pinheiro and R. Bianchini. Energy Conservation Techniques for Disk Array-Based Servers. Technical Report DCS-TR-525, Department of Computer Science, Rutgers University, June 2003.

[30] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath. Dynamic Cluster Reconfiguration for Power and Performance. In L. Benini, M. Kandemir, and J. Ramanujam, editors, *Compilers and Operating Systems for Low Power*. Kluwer Academic Publishers, 2003. To appear.

[31] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath. Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems. In *Proceedings of the International Workshop on Compilers and Operating Systems for Low Power*, September 2001.

[32] K. Rajamani and C. Lefurgy. On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, March 2003.

[33] The Industry Standard. Down on the Server Farm, February 2001. http://www.thestandard.com/article/display/0,1151,22095,00.html.

[34] Ram Viswanath, Vijay Wakharkar, Abhay Watwe, and Vassou Lebonheur. Thermal Performance Challenges from Silicon to Systems. *Intel Technology Journal*, August 2000.

[35] Mark Weiser, Brent Welch, Alan Demers, and Scott Shenker. Scheduling for Reduced CPU Energy. In *Proceedings of the 1st Symposium on Operating System Design and Implementation*, 1994.

[36] A. Weissel, B. Buetel, and F. Bellosa. Cooperative I/O – A Novel I/O Semantics for Energy-Aware Applications. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, December 2002.