

Power and Energy Management for Server Systems

Ricardo Bianchini[†] and Ram Rajamony[‡]

[†]Department of Computer Science [‡]Low-Power Computing Research Center
Rutgers University IBM Austin Research Lab
Piscataway, NJ Austin, TX
ricardob@cs.rutgers.edu rajamony@us.ibm.com

Abstract

Power and energy consumption are key concerns for Internet data centers. These centers house hundreds, sometimes thousands, of servers and supporting cooling infrastructures. Research on power and energy management for servers can ease data center installation, reduce costs, and protect the environment. Given these benefits, researchers have made important strides in conserving energy in 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

Data centers house the server infrastructure that supports most Internet services, such as Web hosting and e-commerce services. In fact, data centers typically host clusters of hundreds, sometimes thousands, of servers. Each server is comprised by a combination of software and hardware that processes requests coming from remote clients or other servers. Common examples are Web and application servers running on off-the-shelf hardware. In this paper, we address power and energy management for data centers by focusing solely on techniques that are explicitly tailored to servers and their workloads.

Power and energy consumption have become key concerns in data centers. The peak power consumption dictates the required cooling infrastructure. Systems with high peak power demands require complex and expensive cooling configurations to efficiently move heat away from the server components and thereby avoid reliability problems. In fact, providing proper cooling is becoming ever more challenging, as power densities have been rising to unprecedented levels due to increasing performance demands, decreasing form factors, and tighter packing. For example, the now popular “blade” clusters pack a large number of (single-board) computers into the same volume of traditional (multi-board) server hardware. High peak power requirements also translate into large and expensive uninterruptible power supplies and backup power generators, both of which are necessary in case of a power outage. The cost of power and cooling equipment has been estimated at U\$ 52800 for each rack of a typical data center over a 10-year lifespan [1].

The energy consumption of a data center dictates its electricity costs. These costs are particularly high for a large and/or dense server cluster in a heavily air-conditioned room. Take the example of a single high-performance 300-Watt server. In a year, such a server consumes 2628 KWh of energy. A cooling unit with a common Energy Efficiency Rating of 12 cools 12000 British Thermal Units (BTU) for 1 KWh; 1 KWh = 3414 BTU. This unit would then consume 748 KWh in cooling this server for the year. Assuming electricity costs of U\$ 0.10/KWh, the total energy cost for this single server would be U\$ 338/year (this cost does not account for the energy consumed by air circulation and power delivery subsystems). Fortunately, servers typically do not operate at their maximum power consumptions as in this example. Nevertheless, the cost of

electricity is still significant. For a typical data-center rack, this cost has been estimated at US\$ 22800 over 10 years [1].

Thus, the power and energy requirements of server systems play an important role in determining both the installation and operational costs of data centers. In fact, data from several sources (e.g., [1, 13]) show that power and energy costs represent a significant fraction of the cost of data centers. For example, [1] estimates that power equipment, cooling equipment, and electricity together are responsible for 63% of the total cost of ownership of the physical IT infrastructure of a data center. 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. Furthermore, air pollution from diesel generators activated when the electrical grid is unstable or unavailable may cause multiple health problems.

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 loads 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 2.

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 [3, 19, 20] have examined energy management strategies in server clusters. These efforts tackled the high “base” power of traditional server hardware (i.e., the power consumption when the system is powered on but idle), by dynamically reconfiguring (or shrinking) the cluster to operate with fewer nodes under light load. Other efforts [8, 7] tackled the high 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 efforts [2, 4, 10, 18, 25] addressed the energy consumption in the storage subsystem. We discuss these previous efforts in more detail in section 3.

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 limit the power consumption of each server. We discuss these two current efforts and some of the remaining challenges of this research topic in section 4.

2 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 CrusoeTM) 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-IVTM) 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.

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.

Exploiting mechanisms in battery-operated devices. 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., [14, 21]). 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., [5, 16]) or based on high-level information [12, 23]. The length of inactivity periods depends on the cost of the mode transitions.

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 large main memories and interconnection switches, for which few 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.

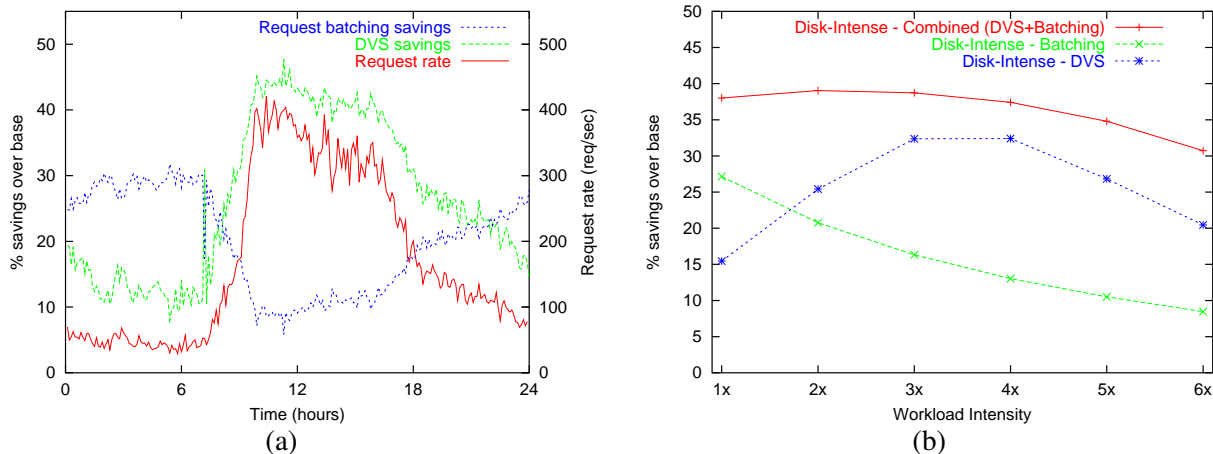


Figure 1: (a) Energy savings for 90th-percentile response time goal of 50ms for Finance, superposed with the request rate. (b) Energy savings for 90th-percentile response time goal of 50ms for Disk-Intense over a range of intensities.

3 Previous Work: Energy Management Techniques for Servers

The previous work has focused almost exclusively on energy management techniques. Next, we divide these techniques into two groups: local (section 3.1) and cluster-wide techniques (section 3.2). Local techniques are implemented independently by each server, whereas cluster-wide techniques involve multiple servers. We further organize the discussion around the different types of servers in data centers. In particular, we highlight three tiers of servers: front-end Web servers, application servers, and storage/database servers.

3.1 Local Techniques

Front-end servers: DVS and request batching. Elnozahy et. al. describe three techniques designed to reduce energy consumption in Web servers [7]. The techniques employ two power management mechanisms: DVS and a new mechanism introduced in their paper called request batching. The first technique extends recently introduced task-based DVS policies (e.g., [9]) for use in server environments with many concurrent tasks. The DVS policy conserves the most energy for intermediate load intensities.

The second technique uses request batching to conserve energy during periods of low load 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. 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 load intensities.

Finally, the third technique uses both DVS and request batching mechanisms to reduce processor energy usage over a wide range of load intensities. When there are no pending requests, the combined technique places the processor in deep-sleep 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 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-

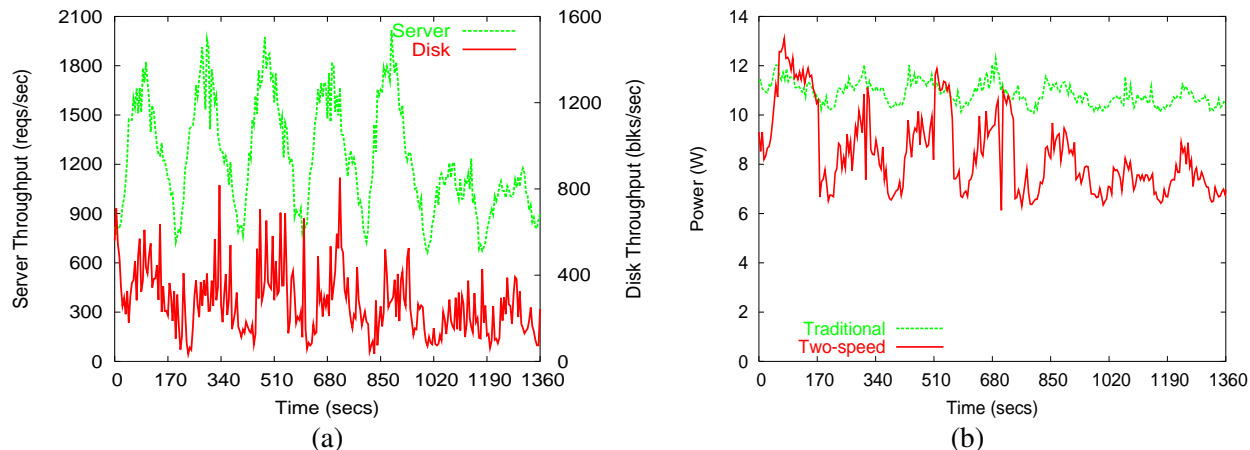


Figure 2: (a) Server and disk throughputs. (b) Power of traditional and two-speed disks.

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 load 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. Figure 1 shows the impact of these techniques on Finance and Disk-Intense [7].

Storage servers: Multi-speed disks, MAID, and PDC. Carrera *et al.* [2] and Gurumurthi *et al.* [10] considered multi-speed disks for servers. These papers showed that significant energy savings can be accrued by dynamically adjusting speeds according to the load imposed on the disk. Gurumurthi *et al.* [10] 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.* [2] studied four disk energy management techniques, including combining laptop and SCSI disks and using 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.

Figure 2 illustrates some of their results. Figure 2(a) 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 2(b) 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.* [11] considered the effect of different RAID parameters, such as RAID level, stripe size, and number of disks, on the performance and energy consumption of database servers running transaction processing workloads.

For the same types of workloads, Zhu *et al.* [24] considered storage cache replacement techniques that selectively keep blocks from certain disks in the main memory cache to increase their idle times, so that the disks can stay in low-power mode for a longer period of time. Recently, Zhu *et al.* [25] studied a more elegant energy-aware storage cache replacement policy, in which dynamically adjusted memory partitions

are used for caching data from different disks.

The works discussed so far did not involve data movement, which could provide further energy savings. The Massive Array of Idle Disks (MAID) [4] and Popular Data Concentration (PDC) [18] do apply data movement. MAID was 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. Replaced data that is dirty has to be written back to the corresponding non-cache disk.

PDC was proposed specifically as an energy management strategy for disk array-based servers. PDC was inspired by the heavily skewed file access frequencies of several types of server workloads, in which 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: concentrating load on certain resources (and thereby increasing their utilization) increases their power consumption by only a fraction of the fixed power consumed by simply 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 sending some disks to low-power modes. 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 [18] presented a quantitative comparison of PDC and MAID when applied to a file server with an array of conventional or two-speed disks, for a wide range of workload and server parameters. Their simulation results for arrays of conventional disks show that PDC and MAID can only conserve energy when the load on the server is extremely low. When two-speed disks are used, both PDC and MAID can conserve up to 30-40% of the disk energy with only a small fraction of delayed requests. Overall, they found that PDC is more consistent and robust than MAID; the behavior of MAID is highly dependent on the number of cache disks. Furthermore, PDC achieves these properties without the overhead of extra disks. However, the PDC energy savings degrade substantially for long migration intervals.

3.2 Cluster-Wide Techniques

Front-end server clusters: Cluster reconfiguration and CVS. Pinheiro *et al.* [19] and Chase *et al.* [3] 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. Under heavy load, resources should be reactivated and the load should be redistributed to eliminate any performance degradation. Because the disk data of Web servers are replicated at all nodes and the “base power” of traditional server hardware (i.e., the power consumption when the system is powered on but idle) is high, their systems dynamically turn entire nodes on and off, in effect reconfiguring the cluster.

Pinheiro *et al.* developed an LC-based cluster of front-end Web servers (as well as an LC-based cluster of cycle servers). Figure 3(a) illustrates the behavior of their system for a 7-node cluster running a real (but

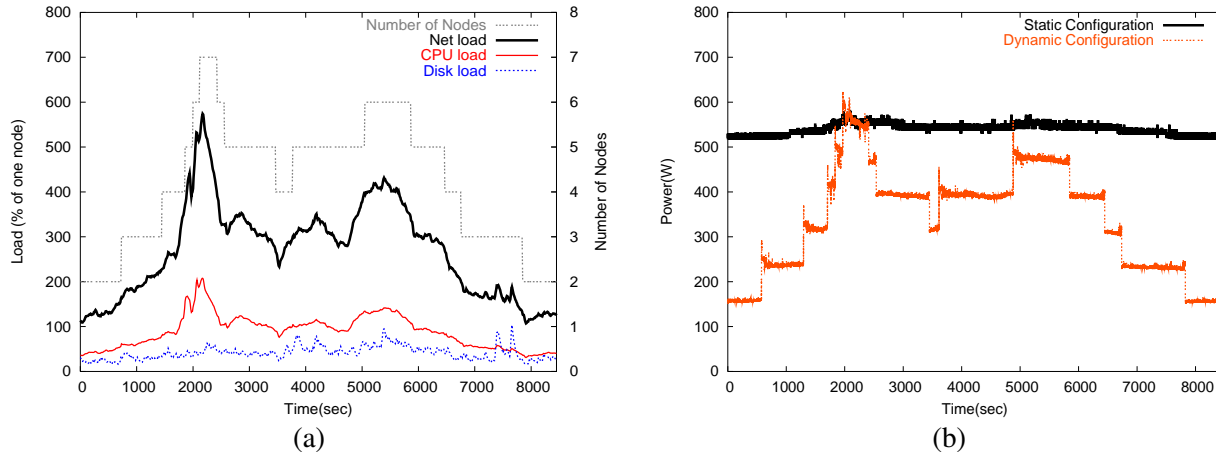


Figure 3: (a) Cluster evolution and per-resource offered loads. (b) Power under static and dynamic configurations.

accelerated) 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 figure shows that for this workload the network interface is the bottleneck resource throughout the whole 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 3(b) presents the power consumption of the whole cluster for two versions of the same experiment, again 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. The figure shows that reconfiguration reduces power consumption significantly for most of the experiment. As a result, the dynamic system saves 38% in energy. Similar results were accrued for the clustered cycle servers.

Rajamani and Lefurgy [20] have studied how to improve the energy saving potential of the cluster re-configuration technique by using spare servers and history information about peak server loads. They also modeled the key system and workload parameters that influence the cluster reconfiguration technique.

Elnozahy *et al.* [8] evaluated different combinations of cluster reconfiguration and dynamic voltage scaling for clusters in which the base power is 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 load it is receiving. In CVS, nodes coordinate their voltage and frequency settings in order to optimize the overall energy consumption. Their simulation results showed that the choice between CVS and reconfiguration depends on workload, while their combination is always the best approach.

Hot server clusters: Throttling and load distribution. Weissel and Bellosa [22] have proposed the throttling of processes to keep CPU temperatures within pre-established limits in server clusters. They infer the energy consumed by each process (on behalf of each client) using the CPU performance counters. Temperature management is done indirectly by limiting the rate of energy consumption. At periodic intervals, the energy consumption over the past interval is compared to a desired consumption level. If the CPU consumed more energy than permitted, halt cycles are introduced to temporarily place it in a low-power state. Weissel and Bellosa implemented their throttling technique in the Linux kernel. Their results for a server cluster with one Web, one factorization, and one database server demonstrate that they can manage temperature accu-

rately. The results also show that they can schedule the client requests according to pre-established energy allotments, when the system starts throttling processes.

At a higher level, researchers from Hewlett-Packard Labs have been considering thermal management of entire data centers [17]. Their temperature modeling work shows that hot spots can develop at certain parts of a center, even when the cooling infrastructure has been properly designed. To counter these hot spots and other cooling problems, they discuss several temperature-aware load distribution policies. One of the policies adjusts the load distribution to racks, according to the difference in temperature between the racks on the same row of the center. The other policy moves load away from the regions of the center that are directly affected by a failed air conditioner. In both cases, the temperature profile of the data center was improved substantially.

4 Current and Future Work

4.1 Peak Power Management

Most of the work we described in section 3 has addressed energy management, i.e., it does not reduce the maximum power that can be consumed by the server system. However, dynamic power management can limit the over-provisioning of the cooling infrastructure to the maximum possible power consumption. In particular, one would like to provide the best possible performance under a fixed and smaller power budget.

Researchers at the IBM Austin Research Laboratory are working on improving power management in a number of areas. Research into memory systems employing lower power states and enforceable caps on power focuses on accurate memory power budgeting, effective delivery, and potentially enhanced performance even under constrained power budgets. A “power shifting” project aims to reduce the system power budget without degrading performance by dynamically re-distributing the budget between active and inactive components. For example, when running a server workload that is processor-intensive but not memory-intensive, power can be increased to the processor, borrowing from the memory’s budget, giving better performance for that workload. As part of this project, they are continuing their research into lightweight mechanisms to control power and performance of different system components, automatic workload characterization techniques, and the necessary algorithms to allocate power among components.

4.2 Exploiting Heterogeneity

The previous work on 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 [3, 19], 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 server cluster 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).

4.3 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.

Exploiting service-level information. The previous work on energy management for servers has not considered exploiting service-level information, such as request priorities, to increase savings. 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. The challenge then is to determine the range of priority distributions for which service-level information provides gains and quantifying these gains.

Energy conservation for application servers. As far as we know, no previous work has considered how to conserve the energy consumed by application servers. These servers are interesting in that they differ markedly from Web and storage servers. In particular, application servers (1) are often written in Java, (2) use CPU and memory intensively, and (3) store soft state that is typically not replicated. Characteristic (2) means that power management mechanisms may have unacceptable performance and energy overheads under moderate and high loads, whereas characteristic (3) means that cluster reconfiguration by turning application servers on/off would require state migration. The challenge then is to develop energy conservation techniques for these servers that can correctly tradeoff energy savings and performance overheads.

Main memory. Some research has already been done on memory energy conservation, but no previous work has evaluated techniques tailored specifically to servers. 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. Current work at IBM is considering memory energy consumption to limit peak power. The next challenge is in properly laying data out across the memory banks and/or chips, so that the low-power states can be used more extensively. Researchers have suggested a few other potential avenues in [15].

Network interfaces and cluster interconnects. As far as we know, high-bandwidth network interfaces

and cluster interconnects have received little attention so far in the literature. Unfortunately, the few papers on this topic (e.g. [6]) did not consider servers and their communication patterns. 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.

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 necessary to monitor temperatures in different parts of the system and shift load around to achieve the most even temperature distribution. During thermal emergencies, e.g. a partial cooling failure, more sophisticated policies may be useful. For example, it may be important to keep using the equipment affected by the emergency for performance reasons. Obviously, we can only schedule as much load on the equipment as the remaining cooling can withstand. The challenge here is to map and understand the thermal behavior of different components and system layouts, the air flow in server enclosures and data centers, achieve accurate temperature monitoring, and tie it all into a system-wide load balancing framework. As we mentioned in section 3, very little research has been done on these issues.

5 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] APC – American Power Conversion. Determining Total Cost of Ownership for Data Center and Network Room Infrastructure. ftp://www.apcmedia.com/salestools/CMRP-5T9PQG_R2_EN.pdf, December 2003.
- [2] 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.
- [3] J. Chase, D. Anderson, P. Thacker, 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.
- [4] D. Colarelli and D. Grunwald. Massive Arrays of Idle Disks for Storage Archives. In *Proceedings of SC'2002*, November 2002.
- [5] Fred Douglass, P. Krishnan, and Brian Marsh. Thwarting the Power-Hungry Disk. In *Proceedings of the 1994 Winter USENIX Conference*, 1994.
- [6] E. J. Kim et al. Energy Optimization Techniques in Cluster Interconnects. In *Proceedings of the International Symposium on Low-Power Electronics and Design*, August 2003.
- [7] E. N. 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.
- [8] 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.

- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] M. Hopkins. The Onsite Energy Generation Option. The Data Center Journal. http://datacenterjournal.com/News/Article.asp?article_id=66, February 2004.
- [14] 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.
- [15] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller. Energy Management for Commercial Servers. *IEEE Computer*, 36(12), December 2003.
- [16] 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.
- [17] J. Moore, R. Sharma, R. Shih, J. Chase, C. Patel, and P. Ranganathan. Going Beyond CPUs: The Potential of Temperature-Aware Solutions for the Data Center. In *Proceedings of the 1st Workshop on Temperature-Aware Computer Systems*, June 2004.
- [18] E. Pinheiro and R. Bianchini. Energy Conservation Techniques for Disk Array-Based Servers. In *Proceedings of the 18th International Conference on Supercomputing (ICS'04)*, June 2004.
- [19] 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, August 2003. Earlier version published as “Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems” in the Proceedings of the Workshop on Compilers and Operating Systems for Low Power, September 2001.
- [20] 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.
- [21] 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.
- [22] A. Weissel and F. Bellosa. Dynamic Thermal Management for Distributed Systems. In *Proceedings of the 1st Workshop on Temperature-Aware Computer Systems*, June 2004.
- [23] 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.
- [24] Q. Zhu, F. M. David, Y. Zhou, C. F. Devaraj, P. Cao, and Z. Li. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. In *Proceedings of the 10th International Symposium on High-Performance Computer Architecture*, February 2004.
- [25] Q. Zhu, A. Shankar, and Y. Zhou. PB-LRU: A Self-Tuning Power Aware Storage Cache Replacement Algorithm for Conserving Disk Energy. In *Proceedings of the 18th International Conference on Supercomputing*, June 2004.