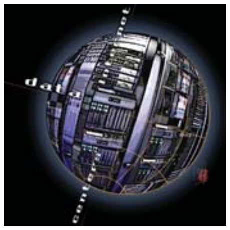
Power and Energy Management for Server Systems



This survey shows that management techniques tailored to different types of servers and their associated workloads can provide substantial energy savings with little or no performance degradation.

Ricardo Bianchini
Rutgers University

Ram Rajamony
IBM Austin Research Lab

Data centers house the server infrastructure that supports most Internet services, such as Web hosting and e-commerce services. These centers typically host clusters of hundreds and sometimes thousands of servers, such as Web and application servers running on off-the-shelf hardware.

Power and energy consumption have become key concerns in data centers. 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. Providing proper cooling becomes ever more challenging as increasing performance demands, decreasing form factors, and tighter packing have been causing higher power densities. High peak power requirements also translate into large and expensive uninterruptible power supplies and backup power generators, which are necessary in case of a power outage. Over the 10-year period of its lifespan, the cost of power and cooling equipment for each rack of a typical data center has been estimated at $52,800.[1]

Each data center's energy consumption dictates its electricity costs. These costs run particularly high for a large or dense server cluster in a heavily air-conditioned room. For example, in a year, a single high-performance 300-watt server will consume 2,628 kWh of energy. Cooling this server will consume an additional 748 kWh. Assuming that electricity costs $0.10 per kWh, the total energy cost for this single server would be $338 per year—excluding energy that air circulation and power delivery subsystems consume. Although a server rarely operates at its maximum power consumption, the cost of electricity remains significant: $22,800 over 10 years for a typical data center rack.[1]

Power equipment, cooling equipment, and electricity together thus represent a significant portion of a data center's cost[1,2]—up to 63 percent of the total cost of ownership of its physical IT infrastructure.[1]

Perhaps more importantly, power and energy management can help protect the environment. Most power-generation technologies—such as nuclear- and coal-based generation—negatively affect the environment. Further, air pollution from diesel generators activated when the electrical grid becomes unstable or unavailable can cause many health problems.

Unfortunately, it is not always ideal or even possible to leverage previous work on the management techniques used for battery-operated devices in the server context. In particular, management techniques for server systems must take into account the high consumption of system components—such as power supplies, disk arrays, and interconnection switches—absent in battery-operated devices. Further, the intensity of busy server loads often makes it infeasible to move components to low-power states by, for example, turning them off.

Realizing the differences between portable and server-class workloads and operating environments, researchers have developed server-specific manage-

Published by the IEEE Computer Society

ment strategies. A few research efforts[3-5] have examined energy management strategies in server clusters. These efforts tackled the high *base* power consumption of traditional server hardware—the power a powered-on but idle system consumes—by dynamically reconfiguring the cluster to operate with fewer nodes under light load. Other efforts[6,7] tackled the high amounts of energy that server CPUs consume. This approach tried to conserve energy by using either dynamic voltage scaling or request batching under light load. Finally, a few efforts[8-12] addressed the energy consumption in the storage subsystem.

Despite these efforts, much must still be done. Our groups are currently addressing two issues in particular. We seek to conserve power and energy in heterogeneous server clusters composed of a combination of traditional and blade servers. We also seek to enforce limits on the power consumed by each server.

MANAGEMENT ISSUES

Power management mechanisms transition hardware components back and forth between high- and low-power states or modes. In high-power mode, components are fully active and operational, while low-power mode functionality depends on the particular component.

Regardless of the specific functionality, changing power modes usually incurs both energy and performance penalties. Thus, management techniques must carefully consider the implications of mode transitions before actually effecting them.

Mechanisms

To illustrate these issues more concretely, we focus on microprocessors and disks, although similar observations can be made regarding other component types.

Microprocessors. Some current microprocessors, such as the Transmeta Crusoe, allow power management by *dynamic voltage scaling*. DVS works because the dynamic power that the microprocessor consumes is a quadratic function of its operating voltage. Thus, reducing the voltage—and, consequently, the frequency—provides substantial savings in power at the cost of slower program execution. The number of low-power modes and transition costs vary widely by microprocessor.

Other microprocessors, such as Intel's Pentium 4, allow power management by halting or deactivation. In contrast to DVS-based microprocessors, these processors cannot perform useful work in the halted or inactive low-power modes. Halting the microprocessor stops it from executing any instructions, therefore reducing the amount of internal activity. Deactivation sends the microprocessor to an even deeper low-power mode, directly addressing its static power requirements. The system must deliver a specific set of signals to reawaken the processor. Again, transition costs vary depending on the microprocessor.

Disks. Current disks also allow power management through deactivation, often exhibiting multiple inactive modes. During accesses, the disk is in active mode and consumes the most power. In idle mode, the disk still spins at its regular speed and can perform accesses without delay. Other low-power modes involve high transition overheads because they require turning the spindle motor off in standby mode and turning off the disk interface in sleep mode. The transition overheads depend on the particular disk.

Energy management techniques for battery-operated devices are not always appropriate for servers.

Battery-operated devices

Based on these mechanisms, researchers have proposed several energy management techniques 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.[13,14]

When the system must deactivate hardware components to save energy, the techniques typically send components to lower power modes after periods of inactivity[15,16] or based on high-level information.[17,18] The length of time a component must be inactive before the techniques perform a mode transition depends on the transition costs.

Server systems

Unfortunately, these techniques are not always appropriate for servers. For example, busy servers often cannot afford to send their hardware components to low-power modes because of the resulting performance degradation. Even in relatively lightly loaded servers, components such as disks must remain inactive for a long time to amortize their mode transition overheads. Servers are rarely idle for such long periods. To make matters worse, servers pose a few new problems:

- provisioned for peak load, server hardware typically exhibits high performance and thus high power consumption;
- popular servers rely on widespread replication

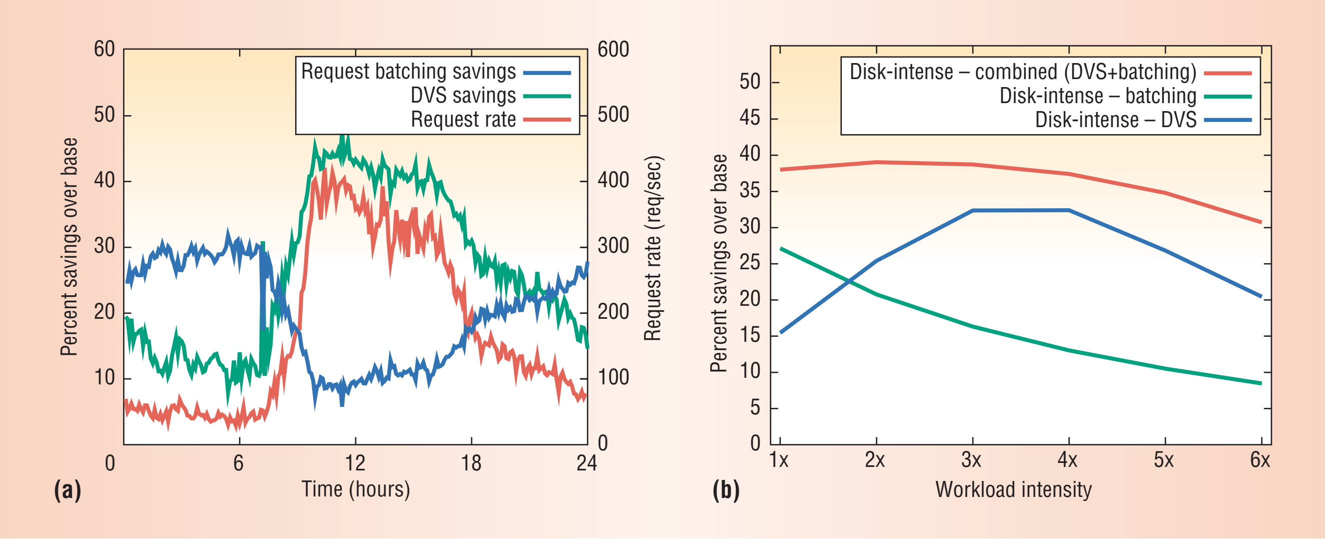


Figure 1. Front-end energy management techniques. (a) Energy savings for the 90th-percentile response time goal of 50 ms for the finance workload, superposed with the request rate; (b) energy savings for the 90th-percentile response time goal of 50 ms for the disk-intense workload, over a range of intensities.

of resources, such as clusters of machines and disk arrays, for high availability and high bandwidth;

- server power supplies typically exhibit high power losses because they must store spare capacity to deal with sudden spikes in load; and
- server systems often involve components, such as large main memories and interconnection switches, for which few management techniques have been proposed.

Given the characteristics of servers and their workloads, power and energy management requires new ideas. Fortunately, we can exploit some of these same characteristics to manage power and energy differently. In particular, researchers have proposed techniques, such as multispeed disks and coordinated DVS for server clusters, that exploit the wide variations in the load offered to servers. These load variations and the replication of resources have motivated proposals to concentrate load onto a subset of resources so that the system can turn off other resources. The frequency of server requests has 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.

PRIOR RESEARCH

Previous work has focused almost exclusively on local and cluster-wide energy management techniques. Each server independently implements local techniques, while 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 and database servers.

Local techniques

Researchers have designed local management techniques for front-end and storage servers.

Front-end servers. The techniques for reducing energy consumption in Web servers employ either DVS, request batching, or both mechanisms.[6]

The first technique extends task-based DVS policies[19] 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. This technique uses the network interface processor to accumulate incoming requests in memory, while the server's host processor remains in a low-power state, such as deep sleep. The host processor awakens when an accumulated request has been pending for longer than a batching timeout. Request batching conserves the most energy for low load intensities.

The third technique uses both DVS and request batching to reduce processor energy usage over a wide range of load intensities. When no requests are pending, this technique places the processor in deep-sleep mode. When the processor activates, it begins operating at the lowest possible frequency, and the DVS technique takes over.

All these techniques trade off system responsiveness to save energy. However, they employ the mechanisms in a feedback-driven control framework to conserve energy while maintaining a specified quality-of-service level, as defined by a percentile-level response time.

To evaluate the techniques, researchers have used a validated Web server simulator and three day-long static Web workloads from real Web server systems: the Nagano Olympics 98 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 50 ms, the DVS

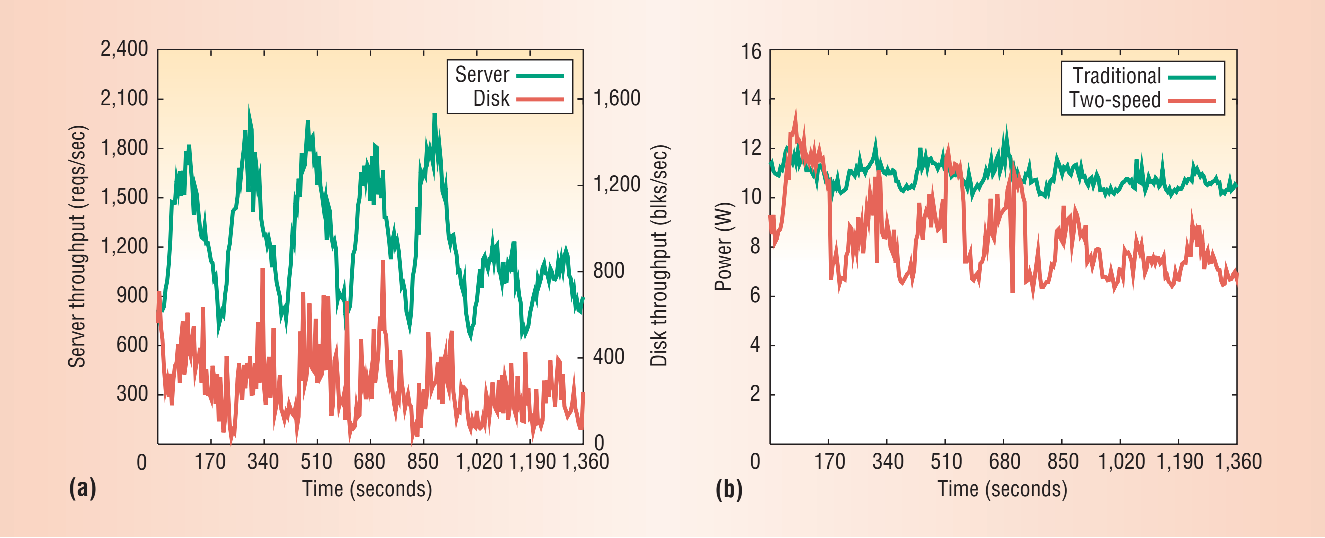


Figure 2. Disk energy management techniques. (a) Server and disk throughputs show server-load peaks and valleys alternating with lighter weekend loads, while burstier disk loads follow a similar trend; (b) power consumption of traditional and two-speed disks.

technique saves from 8.7 to 38 percent of the CPU energy the base system uses, while the request batching technique saves from 3.1 to 27 percent. 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 percent of the CPU energy. Figure 1 shows the impact of these techniques on the finance and disk-intense workloads.[6]

Storage servers. Some studies have investigated using multispeed disks for servers.[8,10] This work shows that dynamically adjusting speeds according to the load imposed on the disk can accrue significant energy savings.

One group introduced performance and power models for multispeed disks, proposed a policy based on disk response time to transition speeds dynamically, and described multiple implementation issues.[10] Using simulation and synthetic workloads, these researchers showed that multispeed disks can provide energy savings of up to 60 percent.

Another study investigated four disk energy management techniques, including combining laptop and SCSI disks, and using simple two-speed disks.[8] Using a real kernel-level implementation and real Web and proxy workloads, this study shows that the combination of laptop and SCSI disks can reduce energy consumption by up to 41 percent, but only for overprovisioned servers. Using emulation with the same workloads, this study also shows that two-speed disks, running at 15,000 and 10,000 rpm, can reduce energy consumption by about 20 percent for properly provisioned servers with a range of hardware and software parameters.

Figure 2a shows some of these results, specifically 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 weekend loads. The disk loads follow the same trend, but are more bursty.

Figure 2b depicts disk power consumption for a server with a traditional high-performance disk and one with a two-speed disk. These results show that the traditional disk consumes 14.8 kJ of disk energy on this workload. The two-speed results show that the disk switches to 15,000 rpm only three times during the whole experiment. The two-speed disk consumes 11.6 kJ of disk energy, a savings of 22 percent.

For disk-array-based servers, some researchers[20] have 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, another group[21] focused on 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. A more elegant energy-aware storage cache replacement policy[12] uses dynamically adjusted memory partitions for caching data from different disks. None of these approaches involve data movement, however, which can provide further energy savings.

Two other techniques *do* apply data movement. Researchers proposed using the *massive array of idle disks*[9] to replace old tape backup archives that have hundreds or thousands of tapes. Because only a small part of the archive would be active at one time, MAID can copy the accessed data to a set of *cache disks* and spin down all other disks. All accesses to the archive then check the cache disks first. The system uses an LRU policy to implement cache disk replacements. If it is clean, MAID can simply discard replaced data, while it must write dirty replaced data back to the corresponding non-cache disk.

Concentrating load on certain resources increases their power consumption by only a fraction of the fixed power consumed by simply having the resource online.

Researchers proposed the *popular data concentration* technique[11] specifically to manage energy for disk-array-based servers. Inspired by the heavily skewed file access frequencies of several server workload types, in which the server frequently accesses a few popular files while only rarely accessing most others, PDC concentrates the most popular disk data by migrating it to a subset of the disks. This concentration skews the disk load toward this subset, while other disks become idle longer and more often. The system can then send these other disks into low-power modes to conserve energy. More specifically, PDC seeks to lay out data across the disk array so that the first disk stores the most popular disk data, the second stores the set of second most popular data, and so on, with the last disk storing the least popular data. Because data popularity can change over time, the system might need to apply PDC periodically during the server's lifetime.

MAID and PDC come from the same general observation: Concentrating load on certain resources, thereby increasing their utilization, increases their power consumption by only a fraction of the fixed power consumed by simply having the resource online. Further, both techniques sacrifice the access time of certain files in favor of energy conservation. However, unlike PDC, which relies on file popularity and migration to conserve energy, MAID relies on temporal locality and copying.

One quantitative comparison of MAID and PDC[11] examines their application to a file server with an array of conventional or two-speed disks, for a wide range of workload and server parameters. In this study, the simulation results for arrays of conventional disks show that MAID and PDC can only conserve energy when server load is extremely low. When using two-speed disks, both MAID and PDC can conserve as much as 30 to 40 percent of disk energy, with only a small fraction of delayed requests.

Overall, the comparison found that PDC is more consistent and robust than MAID, whose behavior depends greatly on the number of cache disks. Further, PDC achieves these properties without the overhead of extra disks. However, PDC energy savings degrade substantially for long migration intervals.

Cluster-wide techniques

Researchers have proposed a few cluster-wide energy and thermal management techniques for server clusters.

Front-end server clusters. Two groups[3,4] have concurrently proposed similar strategies for managing energy in the context of front-end server clusters. The *load concentration* (LC) technique dynamically distributes the load offered to a server cluster under light load, so that it can idle some hardware resources and put them in low-power modes.[4] Under heavy load, the system should reactivate resources and redistribute the load to eliminate performance degradation. Because Web server disk data is replicated at all nodes, and the base power of traditional server hardware—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.

Figure 3a shows the behavior of an LC-based cluster of seven Web servers running a real but accelerated Web trace. The figure plots the evolution of the cluster configuration and offered loads on each resource, in seconds. The load on each resource is the percentage of the same resource's nominal throughput in one node. For this workload, the network interface is the performance bottleneck for the experiment's full 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 rapid succession. The traffic then subsides until another period of high traffic occurs, followed by a substantial decline in traffic.

Figure 3b shows the power consumption of the entire cluster for two versions of the same experiment, again as a function of time. The lower curve, dynamic configuration, represents the system's power-aware version, in which the cluster configuration dynamically adapts to respond to variations in resource demand. The higher curve, static configuration, represents a static cluster configuration fixed at seven nodes. Reconfiguration reduced power consumption significantly for most of this experiment. As a result, the dynamic system has a 38 percent energy savings.

Other researchers[5] 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. They also modeled the key system and workload parameters that influence the cluster reconfiguration technique.

Researchers have also focused on evaluating different combinations of cluster reconfiguration and dynamic voltage scaling for clusters with relatively low base power.[7] In this context, they proposed two techniques: *independent voltage scaling* and *coor-*

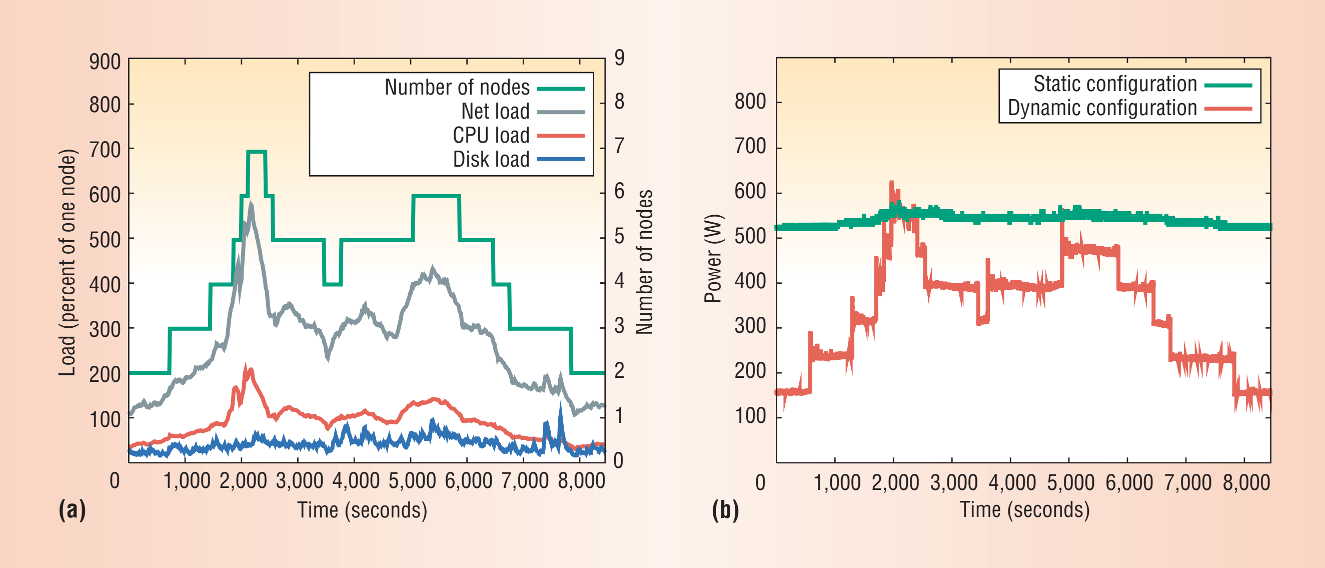


Figure 3. Load concentration energy management technique. (a) System behavior for a seven-node cluster, showing cluster configuration and per-resource offered loads; (b) power consumption for the entire cluster under static and dynamic configurations.

dinated voltage scaling. 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 to optimize overall energy consumption. Simulation results show that the choice between CVS and reconfiguration depends on workload, while combining the techniques provides the best overall approach.

Hot server clusters. Some researchers have proposed throttling processes to keep CPU temperatures within preestablished limits in server clusters.[22] They used the CPU performance counters to infer the energy that each process consumes on behalf of each client. Their system manages temperature indirectly by limiting the rate of energy consumption. At periodic intervals, the system compares the energy consumption over the past interval to a desired consumption level. If the CPU consumes more energy than permitted, the system introduces halt cycles to place the CPU temporarily in a low-power state.

Implementing this throttling technique in the Linux kernel for a server cluster with one Web, one factorization, and one database server demonstrated accurate temperature management. The results also show that this approach can schedule the client requests according to preestablished 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.[23] Their temperature modeling work shows that hot spots can develop at certain parts of a data center, even when the cooling infrastructure has been properly designed. To counter these hot spots and other cooling problems, they have explored several temperature-aware load-distribution policies. One policy adjusts the load distribution to racks, according to temperature differences between racks on the same row. Another policy moves load away from the data center regions directly affected by a failed air conditioner. In both cases, the data center's temperature profile improved substantially.

CURRENT RESEARCH

Our groups are currently investigating approaches to limit power consumption and to exploit cluster heterogeneity.

Peak power management

Although most of the prior work in this area has addressed energy management, dynamic power management can limit overprovisioning of the cooling infrastructure. In particular, data center managers may want to provide the best possible performance under a fixed and smaller power budget. To meet this need, the IBM Austin Research Laboratory is working to improve power management in memory systems that employ lower power states and enforceable caps on power while providing 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 redistributing the budget between active and inactive components. For example, when running a server workload that is processor-intensive but not memory-intensive, the system can increase the power to the processor by borrowing from the memory's budget, which gives better performance for that workload. As part of this project, researchers are exploring lightweight mechanisms to control the power and performance of different system components, automatic workload characterization techniques, and the necessary algorithms for allocating power among components.

Exploiting heterogeneity

Previous work on server cluster energy management focused solely on homogeneous systems. However, real-life clusters are almost invariably heterogeneous in terms of their hardware components' performance, capacity, and power consumption. The reason is that data center managers replace and upgrade hardware components with more powerful ones, as cost-to-performance ratios for off-the-shelf components keep falling. In essence, the server cluster is only homogeneous when first installed, if ever. Further, blade servers are starting to make their way into existing large server clusters, but they typically do not replace all the traditional servers at once. This replacement will more likely occur in multiple stages, producing clusters that will, at least temporarily, include nodes with widely varying characteristics.

Blade servers are starting to make their way into existing large server clusters, but they typically do not replace all the traditional servers at once.

Heterogeneity raises the issue of how to distribute the clients' requests to the different cluster nodes for best performance. Further, heterogeneity must be considered when conserving energy through cluster reconfiguration,[3,4] which raises 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 cluster configuration and request distribution to optimize power, energy, throughput, latency, or some combination of these metrics. The system manager can define the particular optimization function. For example, the manager can select 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. Highly heterogeneous nodes make designing such a server a nontrivial task that becomes even more complex when nodes communicate: for example, when node subsets have different functionalities, such as multitier e-commerce servers, or when nodes cooperate to share resources such as CPUs, main memory caches, or disk storage.

FUTURE CHALLENGES

Several challenges and avenues for research in power and energy management for servers remain.

Modeling and prediction

As server hardware and software become more power- and energy-efficient, management techniques will need to more carefully evaluate or predict the effect of their potential actions. This essentially means that analytically modeling power and energy consumption will become even more important than it is now. Unfortunately, modeling the power and energy that complex server systems consume presents a difficult challenge.

Modeling power is potentially simpler, provided that we understand the details of the hardware components' power behavior. In contrast, modeling energy is more difficult because 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.

Service-level information

Previous work on server energy management did not consider exploiting service-level information, such as request priorities, to increase savings. Request priorities can help keep server resources in low-power mode longer and more often. For example, blocking low-priority requests to a CPU or a disk in low-power mode until a timeout expires can amortize the component's activation costs over multiple low-priority requests. In effect, this strategy trades off higher nonpremium-request response times for lower energy. We can accept this tradeoff because the wide-area network's latency overwhelms relatively short delays at the servers, and nonpremium requests typically do not provide response time guarantees to clients. The challenge then involves determining the range of priority distributions for which service-level information provides gains and quantifying these gains.

Application servers

As far as we know, no previous work has considered how to conserve the energy that application servers consume. These servers differ markedly from Web and storage servers. In particular, application servers are often written in Java, use CPU and memory intensively, and store soft state that is typically not replicated.

Intensive CPU and memory use means that power management mechanisms may have unacceptable performance and energy overheads under moderate and high loads, whereas nonreplication of soft state means that reconfiguring clusters by turning application servers on and off would require state migration. The challenge then involves developing energy conservation techniques for these servers that can correctly trade off 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, which often have extremely large main memories to optimize performance. Processors access these memories frequently because hardware caches are usually much smaller than the working sets of real servers. Direct-memory-access engines also access these memories during network and disk operations.

Researchers at IBM are seeking ways to limit peak power by focusing on memory energy consumption. The next challenge will be properly laying out data across the memory banks and chips so that the server can use low-power states more extensively. Some researchers have suggested a few other potential avenues.[24]

Interconnects and interfaces

Generally, previous research on high-bandwidth network interfaces and cluster interconnects[25] did not consider servers and their communication patterns. Nevertheless, a high-performance switch can consume significant power. Our measurements in the Rutgers DARK Lab show that a 32-port Gigabit Ethernet switch consumes more than 700W when completely idle. A complete understanding of server cluster power and energy consumption clearly requires addressing these components. Unfortunately, because the literature often does not describe the internal architecture of these interconnects, the task of accurately modeling them is extremely complex.

Temperature issues

Given the high power consumption and thermal dissipation of large clusters of densely packed servers, designing equipment-room cooling and ventilation systems to avoid overheating and subsequent hardware reliability problems is crucial. Even with properly designed cooling and ventilation, the system may need to monitor temperatures and shift load around to achieve the most even temperature distribution. During thermal emergencies, such as a partial cooling failure, more sophisticated policies could be useful. For example, for performance reasons it may be important to keep using the equipment affected by the emergency. Obviously, we can only schedule as much load on the equipment as the remaining cooling can withstand. To successfully manage such situations, we must map and understand the thermal behavior of different components and system layouts, and the air flow in server enclosures and data centers. In addition, we must work to achieve accurate temperature monitoring, and then tie all this into a systemwide load-balancing framework. Researchers have made little progress in these directions so far.

Although researchers have made important strides in server power and energy management, they must still address several open issues. For this technology to achieve its full economic and environmental benefits, researchers must make more progress in areas such as peak power management, thermal monitoring and control, modeling and prediction, and energy conservation for application servers, main memories, and interconnection infrastructure. ■

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, 2003.
2. M. Hopkins, "The Onsite Energy Generation Option," *The Data Center J.*; http://datacenterjournal.com/News/Article.asp?article_id=66, Feb. 2004.
3. J. Chase et al., "Managing Energy and Server Resources in Hosting Centers," *Proc. 18th Symp. Operating Systems Principles*, ACM Press, 2001, pp. 103-116.
4. E. Pinheiro et al., "Dynamic Cluster Reconfiguration for Power and Performance," L. Benini, M. Kandemir, and J. Ramanujam, eds., *Compilers and Operating Systems for Low Power*, Kluwer, 2003, pp. 75-94.
5. K. Rajamani and C. Lefurgy, "On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters," *Proc. IEEE Int'l Symp. Performance Analysis of Systems and Software*, IEEE CS Press, 2003, pp. 111-122.
6. E.N. 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.
7. E.N. Elnozahy, "M. Kistler, and R. Rajamony, "Energy-Efficient Server Clusters," *Proc. 2nd Workshop on Power-Aware Computing Systems*, Springer, 2002, pp. 179-196.
8. E.V. Carrera, E. Pinheiro, and R. Bianchini, "Conserving Disk Energy in Network Servers," *Proc. 17th Int'l Conf. Supercomputing*, ACM Press, 2003, pp. 86-97.

9. D. Colarelli and D. Grunwald, “Massive Arrays of Idle Disks for Storage Archives,” *Proc. 2002 Conf. High-Performance Networking and Computing*, IEEE CS Press, 2002, p. 47.
10. S. Gurumurthi et al., “DRPM: Dynamic Speed Control for Power Management in Server Class Disks,” *Proc. Int’l Symp. Computer Architecture*, IEEE CS Press, 2003, pp. 169-179.
11. E. Pinheiro and R. Bianchini, “Energy Conservation Techniques for Disk Array-Based Servers,” *Proc. 18th Int’l Conf. Supercomputing*, ACM Press, 2004, pp. 68-78.
12. Q. Zhu, A. Shankar, and Y. Zhou, “PB-LRU: A Self-Tuning Power Aware Storage Cache Replacement Algorithm for Conserving Disk Energy,” *Proc. 18th Int’l Conf. Supercomputing*, ACM Press, 2004, pp. 79-88.
13. C-H. Hsu and U. Kremer, “The Design, Implementation, and Evaluation of a Compiler Algorithm for CPU Energy Reduction,” *Proc. ACM Sigplan Conf. Programming Languages, Design, and Implementation*, ACM Press, 2003 pp. 38-48.
14. M. Weiser et al., “Scheduling for Reduced CPU Energy,” *Proc. 1st Symp. Operating System Design and Implementation*, Usenix Assoc., 1994, pp. 13-23.
15. F. Douglis, P. Krishnan, and B. Marsh, “Thwarting the Power-Hungry Disk,” *Proc. 1994 Winter Usenix Conf.*, Usenix Assoc., 1994, pp. 292-306.
16. K. Li et al., “A Quantitative Analysis of Disk Drive Power Management in Portable Computers,” *Proc. 1994 Winter Usenix Conf.*, Usenix Assoc., 1994, pp. 279-291.
17. T. Heath et al., “Application Transformations for Energy and Performance-Aware Device Management,” *Proc. 11th Int’l Conf. Parallel Architectures and Compilation Techniques*, IEEE CS Press, 2002, pp. 121-130.
18. A.Weissel, B. Buetel, and F. Bellosa, “Cooperative I/O—A Novel I/O Semantics for Energy-Aware Applications,” *Proc. 5th Symp. Operating Systems Design and Implementation*, ACM Press, pp. 117-129.
19. K. Flautner, S. Reinhardt, and T. Mudge, “Automatic Performance Setting for Dynamic Voltage Scaling,” *Proc. 7th ACM Int’l Conf. Mobile Computing and Networking*, ACM Press, 2001, pp. 260-271.
20. S. Gurumurthi et al., “Interplay of Energy and Performance for Disk Arrays Running Transaction Processing Workloads,” *Proc. Int’l Symp. Performance Analysis of Systems and Software*, IEEE CS Press, 2003,pp. 123-132.
21. Q. Zhu et al., “Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management,” *Proc. 10th Int’l Symp. High-Performance Computer Architecture*, IEEE CS Press, 2004, pp. 118-129.
22. A. Weissel and F. Bellosa, “Dynamic Thermal Management for Distributed Systems,” *Proc. 1st Workshop Temperature-Aware Computer Systems*; www.cs.virginia.edu/~skadron/tacs/weiss.pdf.
23. J. Moore et al., “Going Beyond CPUs: The Potential of Temperature-Aware Solutions for the Data Center,” *Proc. 1st Workshop Temperature-Aware Computer Systems*; www.cs.virginia.edu/~skadron/tacs/rang.pdf.
24. C. Lefurgy et al., “Energy Management for Commercial Servers,” *Computer*, Dec. 2003, pp. 39-47.
25. E.J. Kim et al., “Energy Optimization Techniques in Cluster Interconnects,” *Proc. Int’l Symp. Low-Power Electronics and Design*, ACM Press, 2003, pp. 459-464.

***Ricardo Bianchini** is an associate professor at Rutgers University. His research interests include the performance, availability, and power consumption of server systems. He received a PhD in computer science from the University of Rochester. Contact him at ricardob@cs.rutgers.edu.*

***Ram Rajamony** is a research staff member and manages the Novel Systems Architecture group at IBM Austin Research Lab. His research interests include computer architecture, operating systems, and energy-aware computing. He received a PhD in computer science from Rice University. Contact him at rajamony@us.ibm.com.*