PowerNap: Eliminating Server Idle Power

David Meisner[†] meisner@umich.edu
Brian T. Gold[‡] bgold@cmu.edu
Thomas F. Wenisch[†] twenisch@umich.edu

[†]Advanced Computer Architecture Lab
The University of Michigan

[‡]Computer Architecture Lab
Carnegie Mellon University

Abstract

Data center power consumption is growing to unprecedented levels: the EPA estimates U.S. data centers will consume 100 billion kilowatt hours annually by 2011. Much of this energy is wasted in idle systems: in typical deployments, server utilization is below 30%, but idle servers still consume 60% of their peak power draw. Typical idle periods—though frequent—last seconds or less, confounding simple energy-conservation approaches.

In this paper, we propose PowerNap, *an energy-conservation approach where the entire system transitions rapidly between a high-performance active state and a near-zero-power idle state in response to instantaneous load. Rather than requiring fine-grained power-performance states and complex load-proportional operation from each system component, PowerNap instead calls for minimizing idle power and transition time, which are simpler optimization goals. Based on the PowerNap concept, we develop requirements and outline mechanisms to eliminate idle power waste in enterprise blade servers. Because PowerNap operates in low-efficiency regions of current blade center power supplies, we introduce the* Redundant Array for Inexpensive Load Sharing (RAILS), *a power provisioning approach that provides high conversion efficiency across the entire range of PowerNap's power demands. Using utilization traces collected from enterprise-scale commercial deployments, we demonstrate that, together, PowerNap and RAILS reduce average server power consumption by 74%.*

Categories and Subject Descriptors C.5.5 [*Computer System Implementation*]: Servers

General Terms Design, Measurement

Keywords power management, servers

1. Introduction

Data center power consumption is undergoing alarming growth. By 2011, U.S. data centers will consume 100 billion kWh at a cost of $7.4 billion per year [27]. Unfortunately, much of this energy is wasted by systems that are idle. At idle, current servers still draw about 60% of peak power [1, 6, 13]. In typical data centers, average utilization is only 20-30% [1, 3]. Low utilization is endemic to data center operation: strict service-level- agreements force operators to provision for redundant operation under peak load. Idle-energy waste is compounded by losses in the power delivery and cooling infrastructure, which increase power consumption requirements by 50-100% [18].

Ideally, we would like to simply turn idle systems off. Unfortunately, a large fraction of servers exhibit frequent but brief bursts of activity [2, 3]. Moreover, user demand often varies rapidly and/or unpredictably, making dynamic consolidation and system shutdown difficult. Our analysis shows that server workloads, especially interactive services, exhibit frequent idle periods of less than one second, which cannot be exploited by existing mechanisms.

Concern over idle-energy waste has prompted calls for a fundamental redesign of each computer system component to consume energy in proportion to utilization [1]. Processor dynamic frequency and voltage scaling (DVFS) exemplifies the energy-proportional concept, providing up to cubic energy savings under reduced load. Unfortunately, processors account for an ever-shrinking fraction of total server power, only 25% in current systems [6, 12, 13], and controlling DVFS remains an active research topic [17, 30]. Other subsystems incur many fixed power overheads when active and do not yet offer energy-proportional operation.

We propose an alternative energy-conservation approach, called *PowerNap*, that is attuned to server utilization patterns. With PowerNap, we design the entire system to transition rapidly between a high-performance active state and a minimal-power nap state in response to instantaneous load. Rather than requiring components that provide fine-grain power-performance trade-offs, PowerNap simplifies the system designer's task to focus on two optimization goals: (1) optimizing energy efficiency while napping, and (2) minimizing transition time into and out of the low-power nap state.

Based on the PowerNap concept, we develop requirements and outline mechanisms to eliminate idle power waste in a high-density blade server system. Whereas many mechanisms required by PowerNap can be adapted from mo-

ASPLOS'09 March 7–11, 2009, Washington, DC, USA.

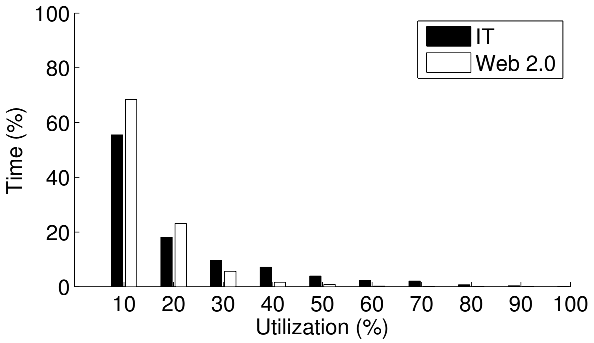


Figure 1: Server Utilization Histogram. Real data centers are under 20% utilized.

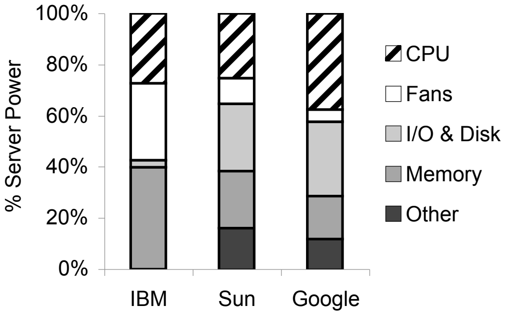


Figure 2: Server Power Breakdown. No single component dominates total system power.

Table 1: Enterprise Data Center Utilization Traces.

Workload	Avg. Utilization	Description
Web 2.0	7.4%	"Web 2.0" application servers
IT	14.2%	Enterprise IT Infrastructure apps

bile and handheld devices, one critical subsystem of current blade chassis falls short of meeting PowerNap's energy-efficiency requirements: the power conversion system. PowerNap reduces total ensemble power consumption when all blades are napping to only 6% of the peak when all are active. Power supplies are notoriously inefficient at low loads, typically providing conversion efficiency below 70% under 20% load [5]. These losses undermines PowerNap's energy efficiency.

Directly improving power supply efficiency implies a substantial cost premium. Instead, we introduce the Redundant Array for Inexpensive Load Sharing (RAILS), a power provisioning approach where power draw is shared over an array of low-capacity power supply units (PSUs) built with commodity components. The key innovation of RAILS is to size individual power modules such that the power delivery solution operates at high efficiency across the entire range of PowerNap's power demands. In addition, RAILS provides N+1 redundancy, graceful compute capacity degradation in the face of multiple power module failures, and reduced component costs relative to conventional enterprise-class power systems. Through modeling and analysis of actual data center workload traces, we demonstrate:

- **Analysis of idle/busy intervals in actual data centers.** We analyze utilization traces from production servers and data centers to determine the distribution of idle and active periods. Though interactive servers are typically over 60% idle, most idle intervals are under one second.
- **Energy-efficiency and response time bounds.** Through queuing analysis, we establish bounds on PowerNap's energy efficiency and response time impact. Using our models, we determine that PowerNap is effective if state transition time is below 10ms, and incurs no overheads below 1ms. Furthermore, we show that PowerNap provides greater energy efficiency and lower response time than solutions based on DVFS.
- **Efficient PowerNap power provisioning with RAILS.** Our analysis of commercial data center workload traces demonstrates that RAILS improves average power conversion efficiency from 68% to 86% in PowerNap-enabled servers.

2. Understanding Server Utilization

It has been well-established in the research literature that the average server utilization of data centers is low, often below 30% [2, 3, 6]. In facilities that provide interactive services (e.g., transaction processing, file servers, Web 2.0), average utilization is often even worse, sometimes as low as 10% [3]. Figure 1 depicts a histogram of utilization for two production workloads from enterprise-scale commercial deployments. Table 1 describes the workloads running on these servers. We derive this data from utilization traces collected over many days, aggregated over more than 120 severs (production utilization traces were provided courtesy of HP Labs). The most striking feature of this data is that the servers spend the vast majority of time under 10% utilization.

Data center utilization is unlikely to increase for two reasons. First, data center operators must provision for peak rather than average load. For interactive services, peak utilization often exceeds average utilization by more than a factor of three [3]. Second, to provide redundancy in the event of failures, operators usually deploy more systems than are actually needed. Though server consolidation can improve average utilization, performance isolation, redundancy, and service robustness concerns often preclude consolidation of mission-critical services.

Low utilization creates an energy efficiency challenge because conventional servers are notoriously inefficient at low loads. Although power-saving features like clock gating and

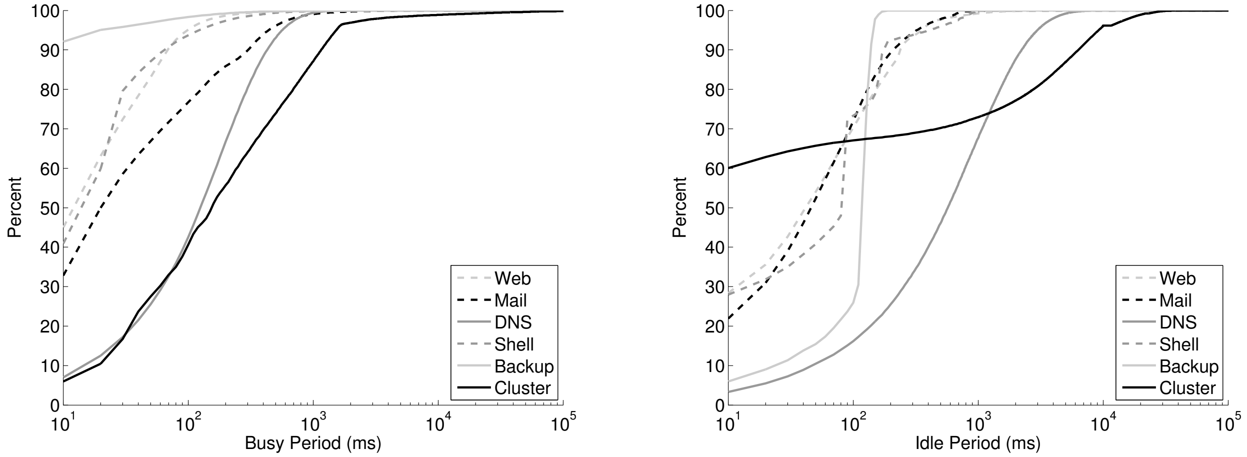


Figure 3: Busy and Idle Period Cumulative Distributions.

Table 2: Fine-Grain Utilization Traces.

Workload	Utilization	Avg. Interval		Description
		Busy	Idle	
Web	26.5%	38 ms	106 ms	Department web server
Mail	55.0%	115 ms	94 ms	Department POP and SMTP servers
DNS	17.4%	194 ms	923 ms	Department DNS and DHCP server
Shell	32.0%	51 ms	108 ms	Interactive shell and IMAP support
Backup	22.2%	31 ms	108 ms	Continuous incremental backup server
Cluster	64.3%	3.25 s	1.8 s	600-node scientific computing cluster

dynamic voltage and frequency scaling (DVFS) nearly eliminate processor power consumption in idle systems, present-day servers still dissipate about 60% as much power when idle as when fully loaded [4, 6, 13]. Processors often account for only a quarter of system power; main memory and cooling fans contribute larger fractions [14]. Figure 2 reproduces typical server power breakdowns for the IBM p670 [14], Sun UltraSparc T2000 [12], and a generic server specified by Google [6], respectively.

2.1 Frequent Brief Utilization

Clearly, eliminating server idle power waste is critical to improving data center energy efficiency. Engineers have been successful in reducing idle power in mobile platforms, such as cell phones and laptops. However, servers pose a fundamentally different challenge than these platforms. The key observation underlying our work is that, although servers have low utilization, their activity occurs in frequent, brief bursts. As a result, they appear to be under a constant, light load.

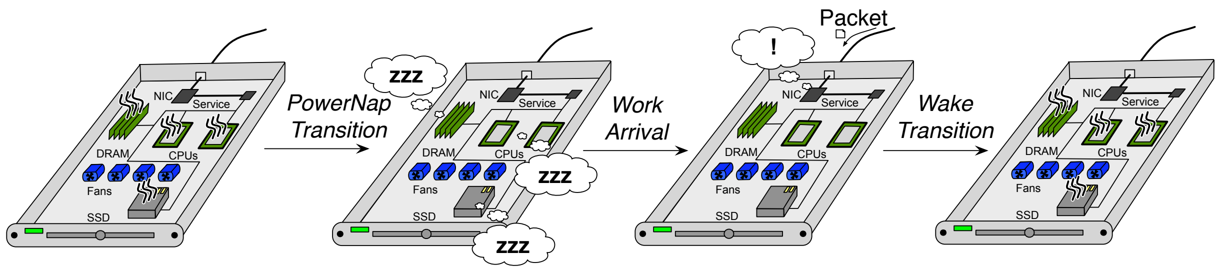
To investigate the time scale of servers' idle and busy periods, we have instrumented a series of interactive and batch processing servers to collect utilization traces at 10ms granularity. To our knowledge, our study is the first to report server utilization data measured at such fine granularity. We classify an interval as busy or idle based on how the OS scheduler accounted the period in its utilization tracking. The traces were collected over a period of a week from seven departmental IT servers and a scientific computing cluster comprising over 600 servers. We present the mean idle and busy period lengths, average utilization, and a brief description of each trace in Table 2.

Figure 3 shows the cumulative distribution for the busy and idle period lengths in each trace. The key result of our traces is that the vast majority of idle periods are shorter than 1s, with mean lengths in the 100's of milliseconds. Busy periods are even shorter, typically only 10's of milliseconds.

2.2 Existing Energy-Conservation Techniques

The rapid transitions and brief intervals of server activity make it difficult to conserve idle power with existing approaches. The recent trend towards server consolidation [20] is partly motivated by the high energy cost of idle systems. By moving services to virtual machines, several services can be time-multiplexed on a single physical server, increasing average utilization. Consolidation allows the total number of physical servers to be reduced, thereby reducing idle inefficiency. However, server consolidation, by itself, does not close the gap between peak and average utilization. Data centers still require sufficient capacity for peak demand, which inevitably leaves some servers idle in the average case. Furthermore, consolidation does not save energy automatically; system administrators must actively consolidate services and remove unneeded systems.

Although support for sleep states is widespread in handheld, laptop and desktop machines, these states are rarely used in current server systems. Unfortunately, the high restart latency typical of current sleep states renders them unaccept-



Server operates at full performance to finish existing work

System components nap while server is idle

The NIC detects the arrival of work

Server returns to full performance to finish work as quickly as possible

Figure 4: PowerNap.

able for interactive services; current laptops and desktops require several seconds to suspend using operating system interfaces (e.g., ACPI). Moreover, unlike consumer devices, servers cannot rely on the user to transition between power states; they must have an autonomous mechanism that manages state transitions.

Recent server processors include CPU throttling solutions (e.g. Intel Speedstep, AMD Cool'n'Quiet) to reduce the large overhead of light loads. These processors use DVFS to reduce their operating frequency linearly while gaining cubic power savings. DVFS relies on operating system support to tune processor frequency to instantaneous load. In Linux, the kernel continues lowering frequency until it observes ∼20% idle time. Improving DVFS control algorithms remains an active research area [17,30]. Nonetheless, DVFS can be highly effective in reducing CPU power. However, as Figure 2 shows, CPUs account for a small portion of total system power.

Energy proportional computing [6] seeks to extend the success of DVFS to the entire system. In this scheme, each system component is redesigned to consume energy in proportion to utilization. In an energy-proportional system, explicit power management is unnecessary, as power consumption varies naturally with utilization. However, as many components incur fixed power overheads when active (e.g., clock power on synchronous memory busses, leakage power in CPUs, etc.) designing energy-proportional subsystems remains a research challenge.

Energy-proportional operation can be approximated with non-energy-proportional systems through dynamic virtual machine consolidation over a large server ensemble [25]. However, such approaches do not address the performance isolation concerns of dynamic consolidation and operate at coarse time scales (minutes). Hence, they cannot exploit the brief idle periods found in servers.

3. PowerNap

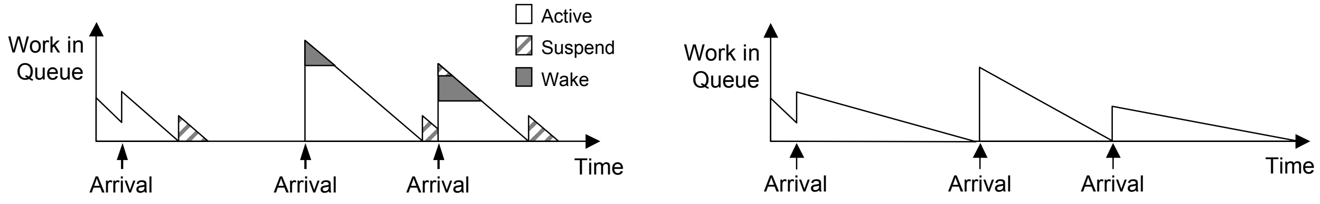
Although servers spend most of their time idle, conventional energy-conservation techniques are unable to exploit these brief idle periods. Hence, we propose an approach to power management that enables the entire system to transition rapidly into and out of a low-power state where all activity is suspended until new work arrives. We call our approach *PowerNap*.

Figure 4 illustrates the PowerNap concept. Each time the server exhausts all pending work, it transitions to the nap state. In this state, nearly all system components enter sleep modes, which are already available in many components (see Section 4). While in the nap state, power consumption is low, but no processing can occur. System components that signal the arrival of new work, expiration of a software timer, or environmental changes, remain partially powered. When new work arrives, the system wakes and transitions back to the active state. When the work is complete, the system returns to the nap state.

PowerNap is simpler than many other energy conservation schemes because it requires system components to support only two operating modes: an active mode that provides maximum performance and a nap mode that minimizes power draw. For many devices, providing a low-power nap mode is far easier than providing multiple active modes that trade performance for power savings. Any level of activity often implies fixed power overheads (e.g., bus clock switching, power distribution losses, leakage power, mechanical components, etc.) We outline mechanisms required to implement PowerNap in Section 4.

3.1 PowerNap Performance and Power Model

To assess PowerNap's potential, we develop a queuing model that relates its key performance measures—energy savings and response time penalty—to workload parameters and PowerNap implementation characteristics. We contrast PowerNap with a model of the upper-bound energy-savings possible with DVFS. The goal of our model is threefold: (1) to gain insight into PowerNap behavior, (2) to derive requirements for PowerNap implementations, and (3) to contrast PowerNap and DVFS.



(a) PowerNap

(b) DVFS

Figure 5: PowerNap and DVFS Analytic Models.

We model both PowerNap and DVFS under the assumption that each seeks to minimize the energy required to serve the offered load. Hence, both schemes provide identical throughput (matching the offered load) but differ in response time and energy consumption.

PowerNap Model. We model PowerNap as an M/G/1 queuing system with arrival rate λ, and a generalized service time distribution with known first and second moments $E[S]$ and $E[S^2]$. Figure 5(a) shows the work in the queue for three job arrivals. Note that, in this context, work also includes time spent in the wake and suspend states. Average server utilization is given by $\rho = \lambda E[S]$. To model the effects of PowerNap suspend and wake transitions, we extend the conventional M/G/1 model with an exceptional first service time [29]. We assume PowerNap transitions are symmetric with latency T_t. Service of the first job in each busy period is delayed by an initial setup time I. The setup time includes the wake transition and may include the remaining portion of a suspend transition as shown for the rightmost arrival in Figure 5(a). Hence, for an arrival x time units from the start of the preceding idle period, the initial setup time is given by:

$$I = \begin{cases} 2T_t - x & \text{if } 0 \le x < T_t \\ T_t & \text{if } x \ge T_t \end{cases}$$

The first and second moments $E[I]$ and $E[I^2]$ are:

$$\begin{aligned} E[I] &= \int_0^{\infty} I\lambda e^{-\lambda x}dx = 2T_t + \frac{1}{\lambda}e^{-\lambda T_t} - \frac{1}{\lambda} \\ E[I^2] &= \int_0^{\infty} I^2\lambda e^{-\lambda x}dx \\ &= 4T_t^2 - 2T_t^2 e^{-\lambda T_t} - \\ &\quad \left(\frac{4T_t}{\lambda} + \frac{2}{\lambda^2}\right)\left[1 - (1+\lambda T_t)e^{-\lambda T_t}\right] \end{aligned}$$

We compute average power as

$$P_{avg} = P_{nap} \cdot F_{nap} + P_{max}(1 - F_{nap}),$$

where the fraction of time spent napping F_{nap} is given by the ratio of the expected length of each nap period $E[N]$ to the expected busy-idle cycle length $E[C]$:

$$\begin{aligned} F_{nap} &= \frac{\int_0^{T_t}(0)\lambda e^{-\lambda t}dt + \int_{T_t}^{\infty}(t - T_t)\lambda e^{-\lambda t}dt}{\frac{E[S]+E[I]}{1-\lambda E[S]} + \frac{1}{\lambda}} \\ &= \frac{e^{-\lambda T_t}\left(1 - \lambda E[S]\right)}{1 + \lambda E[I]} \end{aligned}$$

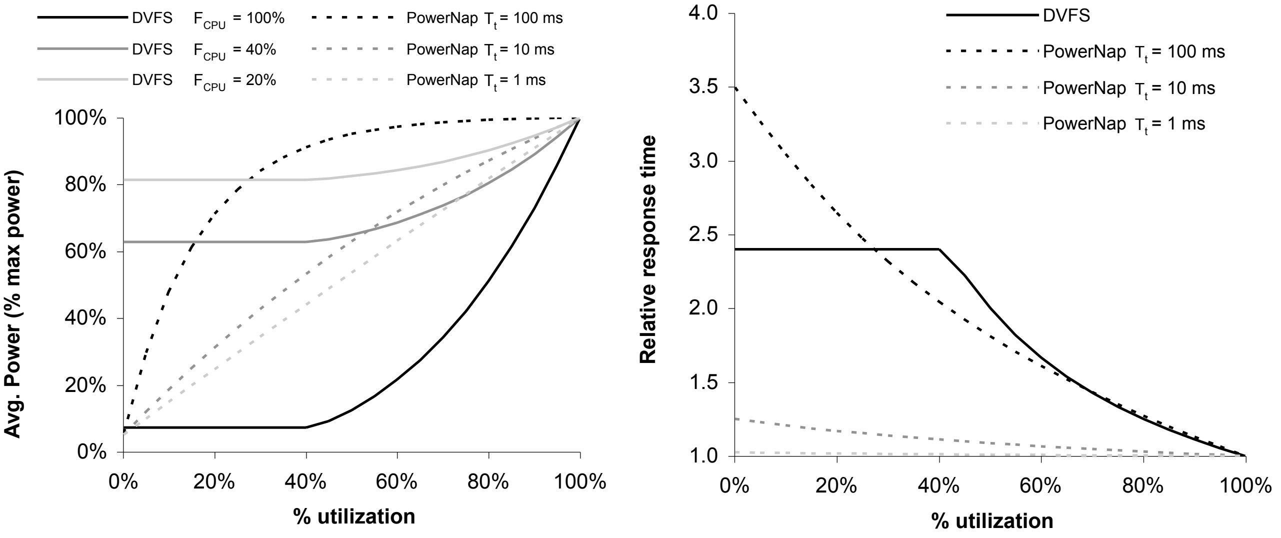
The response time for an M/G/1 server with exceptional first service is due to Welch [29]:

$$E[R] = \tfrac{\lambda E[S^2]}{2(1-\lambda E[S])} + \tfrac{2E[I]+\lambda E[I^2]}{2(1+\lambda E[I])} + E[S]$$

Note that the first term of $E[R]$ is the Pollaczek-Khinchin formula for the expected queuing delay in a standard M/G/1 queue, the second term is additional residual delay caused by the initial setup time I, and the final term is the expected service time $E[S]$. The second term vanishes when $T_t = 0$.

DVFS model. Rather than model a real DVFS frequency control algorithm, we instead model the upper bound of energy savings possible with DVFS. For each job arrival, we scale instantaneous frequency f to stretch the job to fill any idle time until the next job arrival, as illustrated in Figure 5(b), which gives $E[f] = f_{max}\rho$. This scheme maximizes power savings, but cannot be implemented in practice because it requires knowledge of future arrival times. We base power savings estimates on the theoretical formulation of processor dynamic power consumption $P_{CPU} = \frac{1}{2}CV^2Af$. We assume C and A are fixed, and choose the optimal f for each job within the range $f_{min} < f < f_{max}$. We impose a lower bound $f_{min} = f_{max}/2.4$ to prevent response time from growing asymptotically when utilization is low. We chose a factor of 2.4 between f_{min} and f_{max} based on the frequency range provided by a 2.4 GHz AMD Athlon. We assume voltage scales linearly with frequency (i.e., $V = V_{max}(f/f_{max})$), which is optimistic with respect to current DVFS implementations. Finally, as DVFS only reduces the CPU's contribution to system power, we include a parameter F_{CPU} to control the fraction of total system power affected by DVFS. Under these assumptions, average power P_{avg} is given by:

$$P_{avg} = P_{max}(1 - F_{CPU}(\tfrac{E[f]}{f_{max}})^3)$$



(a) Power Scaling

(b) Response Time Scaling

Figure 6: PowerNap and DVFS Power and Response Time Scaling.

Response time is given by:

$$E[R] = E\left[\frac{R_{base}}{f}\right]$$

where R_{base} is the response time without DVFS.

3.2 Analysis

Power Savings. Figure 6(a) shows the average power (as a fraction of peak) required under PowerNap and DVFS as a function of utilization. For DVFS, we show power savings for three values of F_{CPU}. F_{CPU} = 100% represents the upper bound if DVFS were applicable to all system power. $20\% < F_{CPU} < 40\%$ bound the typical range in current servers. For PowerNap, we construct the graphs with $E[s] = 38ms$ and $E[s^2] = 3.7E[s]$, which are both estimated from the observed busy period distribution in our Web trace. We assume P_{nap} is 5% of P_{max}. We vary λ to adjust utilization, and present results for three values of T_t: 1ms, 10ms, and 100ms. We expect 10ms to be a conservative estimate for achievable PowerNap transition time. For transition times below 1ms, transition time becomes negligible and the power savings from PowerNap varies linearly with utilization for all workloads. We discuss transition times further in Section 4.

When F_{CPU} is high, DVFS clearly outperforms PowerNap, as it provides cubic power savings while PowerNap's savings are at best linear in utilization. However, for realistic values of F_{CPU} and transition times in our expected range ($T_t \leq 10ms$), PowerNap's savings rapidly overtake DVFS. As transition time increases, the break-even point between DVFS and PowerNap shifts towards lower utilization. Even for a transition time of 100 ms, PowerNap can provide substantial energy savings when utilization is below 20%.

Table 3: Per-Workload Energy Savings.

Workload	PowerNap Energy Savings	DVFS Energy Savings
Web	59%	23%
Mail	35%	21%
DNS	77%	23%
Shell	55%	23%
Backup	61%	23%
Cluster	34%	18%

Response time. In Figure 6(b), we compare the response time impact of DVFS and PowerNap. The vertical axis shows response time normalized to a system without power management (i.e., that always operates at f_{max}). For DVFS, response time grows rapidly when the gap between job arrivals is large, and reaches the f_{min} floor below 40% utilization. DVFS response time penalty is independent of F_{CPU}, and is bounded at 2.4 by the ratio of f_{max}/f_{min}. For PowerNap, the response time penalty is negligible if T_t is small relative to average service time E[S], which we expect to be the common case (i.e., most jobs last longer than 10ms). However, if T_t is significant relative to E[S], the PowerNap response time penalty grows as utilization shrinks. When utilization is high, the server is rarely idle and few jobs are delayed by transitions. As utilization drops, the additional delay seen by each job converges to T_t (i.e., every job must wait for wake-up).

Per-Workload Energy Savings. Finally, we report the energy savings under simulated PowerNap and DVFS schemes for our workload traces. Because these traces only contain busy and idle periods, and not individual job arrivals, we cannot estimate response time impact. For each workload, we perform a trace-based simulation that assumes busy periods will start at the same time, independent of the current PowerNap state (i.e., new work still arrives during wake or suspend transitions). We assume a PowerNap transition time of 10ms and nap power at 5% of active power, which we believe to be conservative estimates (see Section 4). For DVFS, we assume $F_{CPU} = 25\%$. Table 3 shows the results of these simulations. All workloads except Mail and Cluster hit the DVFS frequency floor, and, hence, achieve a 23% energy savings. In all cases, PowerNap achieves greater energy savings. Additionally, we extracted the average arrival rate (assuming a Poisson arrival process) and compared the results in Table 3 with the M/G/1 model of F_{nap} derived above. We found that for these traces, the analytic model was within 2% of our simulated results in all cases. When arrivals are more deterministic (e.g., Backup) than the exponential we assume, the model slightly overestimates PowerNap savings. For more variable arrival processes (e.g., Shell), the model underestimates the energy savings.

3.3 Implementation Requirements

Based on the results of our analytic model, we identify two key PowerNap implementation requirements:

Fast transitions. Our model demonstrates that transition speed is the dominant factor in determining both the power savings potential and response time impact of PowerNap. Our results show that transition time must be less than one tenth of average busy period length. Although a 10ms transition speed is sufficient to obtain significant savings, 1ms transitions are necessary for PowerNap's overheads to become negligible. To achieve these transition periods, a PowerNap implementation must preserve volatile system state (e.g., memory) while napping—mass storage devices transfer rates are insufficient to transfer multiple GB of memory state in milliseconds.

Minimizing power draw in nap state. Given the low utilization in most enterprise deployments, servers will spend a majority of time in the nap state, making PowerNap's power requirements the key factor affecting average system power. Hence, it is critical to minimize the power draw of napping system components. As a result of eliminating idle power, PowerNap drastically increases the range between the minimum and maximum power demands on a blade chassis. Existing blade-chassis power-conversion systems are inefficient in the common case, where all blades are napping. Hence, to maximize PowerNap potential, we must re-architect the blade chassis power subsystem to increase its efficiency at low loads.

Although PowerNap requires system-wide modifications, it demands only two states from each subsystem: active and nap states. Hence, implementing PowerNap is substantially simpler than developing energy-proportional components. Because no computation occurs while napping, many fixed power draws, such as clocks and leakage power, can be conserved.

Table 4: Component Power Consumption.

Component	Power			Transition	Sources
	Active	Idle	Nap		
CPU chip	80-150W	12-20W	3.4W	30 μs	[10] [9]
DRAM DIMM	3.5-5W	1.8-2.5W	0.2W	$< 1\mu$s	[16] [8]
NIC	0.7W	0.3W	0.3W	no trans.	[24]
SSD	1W	0.4W	0.4W	no trans.	[22]
Fan	10-15W	1-3W	-	independent	[15]
PSU	50-60W	25-35W	0.5W	300 μs	[19]
Typical Blade	450W	270W	10.4W	300 μs	

4. PowerNap Mechanisms

We outline the design of a PowerNap-enabled blade server system and enumerate required implementation mechanisms. PowerNap requires nap support in all hardware subsystems that have non-negligible idle power draws, and software/firmware support to identify and maximize idle periods and manage state transitions.

4.1 Hardware Mechanisms

Most of the hardware mechanisms required by PowerNap already exist in components designed for mobile devices. However, few of these mechanisms are exploited in existing servers, and some are omitted in current-generation server-class components. For each hardware subsystem, we identify existing mechanisms or outline requirements for new mechanisms necessary to implement PowerNap. Furthermore, we provide estimates of power dissipation while napping and transition speed. We summarize these estimates, along with our sources, in Table 4. Our estimates for a ”Typical Blade” are based on HP's c-series half-height blade designs; our PowerNap power estimate assumes a two-CPU system with eight DRAM DIMMs.

Processor: ACPI S3 “Sleep” state. The ACPI standard defines the S3 “Sleep” state for processors that is intended to allow low-latency transitions. Although the ACPI standard does not specify power or performance requirements, some implementations of S3 are ideal for PowerNap. For example, in Intel's mobile processor line, S3 preserves last-level cache state and consumes only 3.4W [10]. These processors require approximately 30 μs for PLL stabilization to transition from sleep back to active execution [9].

If S3 is unavailable, clock gating can provide substantial energy savings. For example, Intel's Xeon 5400-series power requirements drop from 80W to 16W upon executing a halt instruction [11]. From this state, resuming execution requires only nanosecond-scale delays.

DRAM: Self-refresh. DRAM is typically the second-most power-hungry system component when active. However, several recent DRAM specifications feature an operating mode, called self-refresh, where the DRAM is isolated from the memory controller and autonomously refreshes DRAM content. In this mode, the memory bus clock and PLLs are disabled, as are most of the DRAM interface circuitry. Self-refresh saves more than an order of magnitude of power. For example, a 2GB SODIMM (designed for laptops) with a peak power draw above 5W uses only 202mW of power during self- refresh [16]. Transitions into and out of self-refresh can be completed in less than a microsecond [8].

Mass Storage: Solid State Disks. Solid state disks draw negligible power when idle, and, hence, do not need to transition to a sleep state for PowerNap. A recent 64GB Samsung SSD consumes only 0.32W while idle [22].

Network Interface: Wake-on-LAN. The key responsibility PowerNap demands of the network interface card (NIC) is to wake the system upon arrival of a packet. Existing NICs already provide support for Wake-on-LAN to perform this function. Current implementations of Wake-on-LAN provide a mode to wake on any physical activity. This mode forms a basis for PowerNap support. Current NICs consume only 400mW while in this mode [24].

Environmental Monitoring & Service Processors: PowerNap transition management. Servers typically include additional circuitry for environmental monitoring, remote management (e.g., remote power on), power capping, power regulation, and other functionality. These components typically manage ACPI state transitions and would coordinate PowerNap transitions. A typical service processor draws less than 10mW when idle.

Fans: Variable Speed Operation. Fans are a dominant power consumer in many recent servers. Modern servers employ variable-speed fans where cooling capacity is constantly tuned based on observed temperature or power draw. Fan power requirements typically grow cubically with average power. Thus, PowerNap's average power savings yield massive reductions in fan power requirements. In most blade designs, cooling systems are centralized in the blade chassis, amortizing their energy cost over many blades. Because thermal conduction progresses at drastically different timescales than PowerNap's transition frequency, chassis-level fan control is independent of PowerNap state (i.e., fans may continue operating during nap and may spin down during active operation depending on temperature conditions).

Power Provisioning: RAILS. PowerNap fundamentally alters the range of currents over which a blade chassis must efficiently supply power. In Section 5, we explain why conventional power delivery schemes are unable to provide efficient AC to DC conversion over this range, and present RAILS, our power conversion solution.

4.2 Software Mechanisms

For schemes like PowerNap, the periodic timer interrupt used by legacy OS kernels to track the passage of time and implement software timers poses a challenge. As the timer interrupt is triggered every 1ms, conventional OS time keeping precludes the use of PowerNap. The periodic clock tick also poses a challenge for idle-power conservation on laptops and for virtualization platforms that consolidate hundreds of OS images on a single hardware platform. Hence, the Linux kernel has recently been enhanced to support "tickless" operation, where the periodic timer interrupt is eschewed in favor of hardware timers for scheduling and time keeping [23]. PowerNap depends on a kernel that provides tickless operation.

PowerNap's effectiveness increases with longer idle periods and less frequent state transitions. Some existing hardware devices (e.g., legacy keyboard controllers) require polling to detect input events. Current operating systems often perform maintenance tasks (e.g., flushing disk buffers, zeroing memory) when the OS detects significant idle periods. These maintenance tasks may interact poorly with PowerNap and can induce additional state transitions. However, efforts are already underway (e.g., as described in [23]) to redesign device drivers and improve background task scheduling.

5. RAILS

AC to DC conversion losses in computer systems have recently become a major concern, leading to a variety of research proposals [7, 15], product announcements (e.g., HP's Blade System c7000), and standardization efforts [5] to improve power supply efficiency. The concern is particularly acute in data centers, where each watt wasted in the power delivery infrastructure implies even more loss in cooling. Because PowerNap's power draw is substantially lower than the idle power in conventional servers, PowerNap demands conversion efficiency over a wide power range, from as few as 300W to as much as 7.2kW in a fully-populated enclosure.

In this section, we discuss why existing power solutions are inadequate for PowerNap and present RAILS, our power solution. RAILS provides high conversion efficiency across PowerNap's power demand spectrum, provides N+1 redundancy, allows for graceful degradation of compute capacity when PSUs fail, and minimizes costs by using commodity PSUs in an efficient arrangement.

5.1 Power Supply Unit background

Poor Efficiency at Low Loads. Although manufacturers often report only a single efficiency value, most PSUs do not have a constant efficiency across electrical load. A recent survey of server and desktop PSUs reported their efficiency across loads [5]. Figure 7 reproduces the range of efficiencies reported in that study. Though PSUs are often over 90%

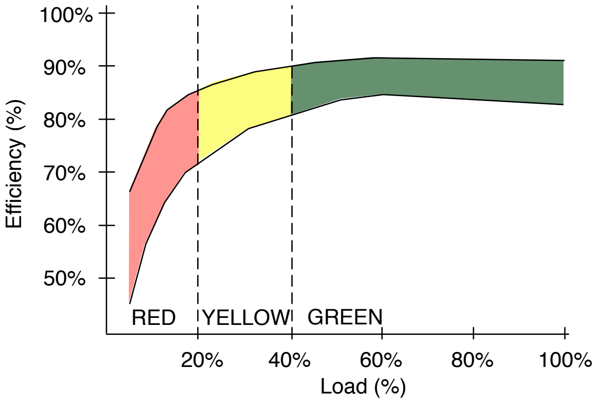


Figure 7: Power Supply Efficiency.

efficient at their optimal operating point (usually near 75% load), efficiency drops off rapidly below 40% load, sometimes dipping below 50% (i.e., >2W in for 1W out). We divide the operating efficiency of power supplies into three zones based on electrical load. Above 40% load, the PSUs operate in the "green" zone, where their efficiency is at or above 80%. In the 20-40% "yellow" zone, PSU efficiency begins to drop, but typically exceeds 70%. However, in the "red" zone below 20%, efficiency drops off precipitously.

Two factors cause servers to frequently operate in the "yellow" or "red" efficiency zones. First, servers are highly configurable, which leads to a large range of power requirements. The same server model might be sold with only one or as many as 20 disks installed, and the amount of installed DRAM might vary by a factor of 10. Furthermore, peripherals may be added after the system is assembled. To simplify ordering, upgrades, testing, and safety certification, manufacturers typically install a power supply rated to exceed the power requirements of the most extreme configuration. Second, servers are often configured with 2N redundant power supplies (i.e., twice as many as are required for a worst-case configuration). The redundant supplies typically share the electrical load to minimize PSU temperature and to ensure current flow remains uninterrupted if a PSU fails. However, the EPRI study [5] concluded that this load-sharing arrangement often shifts PSUs from "yellow"-zone to "red"-zone operation.

Recent Efficiency Improvements. A variety of recent initiatives seek to improve server power efficiency:

- **80+ certification.** The EPA Energy Star program has defined the "80+" certification standard [26] to incentivize PSU manufacturers to improve efficiency at low loads. The 80+ incentive program is primarily targeted at the low-peak-power desktop PSU market. 80+ supplies require considerably higher design complexity than conventional PSUs, which may pose a barrier to widespread adoption in the reliability-conscious server PSU market. Furthermore, despite their name, the 80+ specification does not require energy efficiency above 80% across all loads, rather, only within the typical operating range of conventional systems. This specified efficiency range is not wide enough for PowerNap.

- **Single voltage supplies.** Unlike desktop machines, which require five different DC output voltages to support legacy components, server PSUs typically provide only a single DC output voltage, simplifying their design and improving reliability and efficiency [7]. Although PowerNap benefits from this feature, a single output voltage does not directly address inefficiency at low loads.

- **DC distribution.** Recent research [7] has called for distributing DC power among data center racks, eliminating AC-to-DC conversion efficiency concerns at the blade enclosure level. However, the efficiency advantages of DC distribution are unclear [21] and deploying DC power will require multi-industry coordination.

- **Dynamic load-sharing.** Blade enclosures create a further opportunity to improve efficiency through dynamic load-sharing. HP's Dynamic Power Saver [15] feature in the HP Blade Center c7000 employs up to six high-efficiency 2.2kW PSUs in a single enclosure, and dynamically varies the number of PSUs that are engaged, ensuring that all active supplies operate in their "green" zone while maintaining redundancy. Although HP's solution is ideal for the idle and peak power range of the c-class blades, it requires expensive PSUs and provides insufficient granularity for PowerNap.

While all these solutions improve efficiency for their target markets, none achieve all our goals of efficiency for PowerNap, redundancy, and low cost.

5.2 RAILS Design

We introduce a new power delivery solution tuned for PowerNap: the Redundant Array for Inexpensive Load Sharing (RAILS). The central idea of our scheme is to load-share over multiple inexpensive, small PSUs to provide the efficiency and reliability of larger, more expensive units. Through intelligent sizing and load-sharing, we ensure that active PSUs operate in their efficiency sweet spots. Our scheme provides 80+ efficiency and enterprise-class redundancy with commodity components.

RAILS targets three key objectives: (1) efficiency across the entire PowerNap dynamic power range; (2) N+1 reliability and graceful degradation of compute capacity under multiple PSU failure; and (3) minimal cost.

Figure 8 illustrates RAILS. As in conventional blade enclosures, power is provided by multiple PSUs connected in

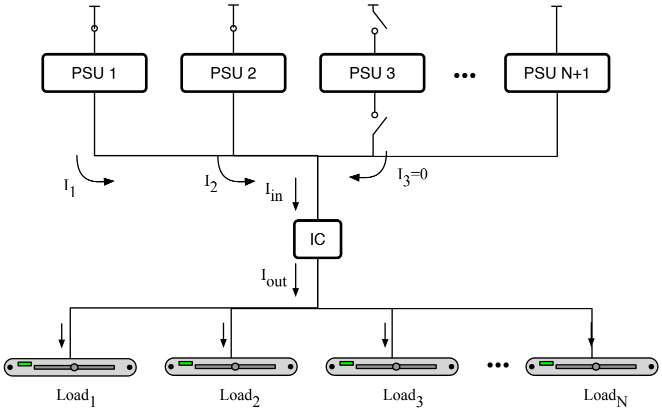


Figure 8: RAILS PSU Design.

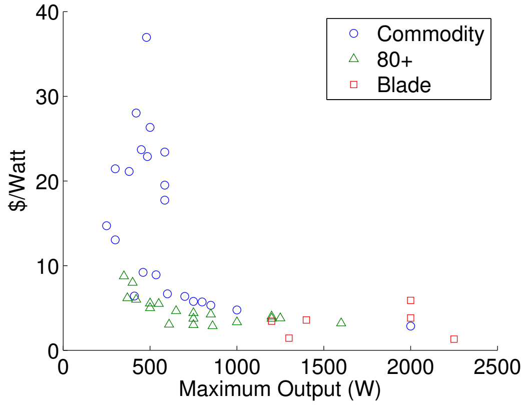


Figure 9: Power Supply Pricing.

parallel. A conventional load-sharing control circuit continuously monitors and controls the PSUs to ensure load is divided evenly among them. As in Dynamic Smart Power [15], RAILS disables and electrically isolates PSUs that are not necessary to supply the load. However, our key departure from prior designs is in the granularity of the individual PSUs. We select PSUs from the economic sweet spot of the high-sales-volume market for low-wattage commodity supplies.

We choose a power supply granularity to satisfy two criteria: (1) A single supply must be operating in its "green" zone when all blades are napping. This criterion establishes an upper bound on the PSU capacity based on the minimum chassis power draw when all blades are napping. (2) Subject to this bound, we size PSUs to match the incremental power draw of activating a blade. Thus, as each blade awakens, one additional PSU is brought on line. Because of intelligent sizing, each of these PSUs will operate in their optimal efficiency region. Whereas current blade servers use multi-kilowatt PSUs, a typical RAILS PSU might supply 500W.

RAILS meets its cost goals by incorporating high-volume commodity components. Although the form-factor of commodity PSUs may prove awkward for rack-mount blade enclosures, precluding the use of off-the-shelf PSUs, the power density of high-sales-volume PSUs differs little from high-end server supplies. Hence, with appropriate mechanical modifications, it is possible to pack RAILS PSUs in roughly the same physical volume as conventional blade enclosure power systems.

RAILS meets its reliability goals by providing fine-grain degradation of the system's peak power capacity as PSUs fail. In any N+1 design, the first PSU failure does not affect compute capacity. However, in conventional blade enclosures, a subsequent failure may force shutdown of several (possibly all) blades. Multiple-failure tolerance typically requires 2N redundancy, which is expensive. In contrast, in RAILS, where PSU capacity is matched to the active power draw of a single blade, the second and subsequent failures each require the shutdown of only one blade.

5.3 Evaluation

We evaluate the power efficiency and cost of PowerNap with four power supply designs, commodity supplies ("Commodity"), high-efficiency 80+ supplies ("80+"), dynamic load sharing ("Dynamic"), and RAILS ("RAILS"). We evaluate all four designs in the context of a PowerNap-enabled blade system similar to HP's Blade Center c7000. We assume a fully populated chassis with 16 half-height blades. Each blade consumes 450W at peak, 270W at idle without PowerNap, and 10.4W in PowerNap (see Table 4). We assume the blade enclosure draws 270W (we neglect any variation in chassis power as a function of the number of active blades). The non-RAILS systems employ 4 2250W PSUs (sufficient to provide N+1 redundancy). The RAILS design uses 17 500W PSUs. We assume the average efficiency characteristic from Figure 7 for commodity PSUs.

Cost. Server components are sold in relatively low volumes compared to desktop or embedded products, and thus, command premium prices. Some Internet companies (e.g., Google), have eschewed enterprise servers and instead assemble systems from commodity components to avoid these premiums. PSUs present another opportunity to capitalize on low-cost commodity components. Because desktop ATX PSUs are sold in massive volumes, their constituent components are cheap. A moderately-sized supply can be obtained at extremely low cost. Figure 9 shows a survey of PSU prices in Watts per dollar for a wide range of PSUs across market segments. Price per Watt increases rapidly with power delivery capacity. This rise can be attributed to the proportional increase in required size for power components such as inductors and capacitors. Also, the price of discrete power components grows with size and maximum current rating. Presently, the market sweet spot is around 500W supplies. Both 80+ and blade server PSUs are substantially more ex-

Table 5: Relative PSU Density.

	microATX	ATX	Custom Blade
Density (Normalized W/vol.)	675.5	1000	1187

pensive than commodity parts. Because RAILS uses commodity PSUs with small maximum outputs, it takes advantage of PSU market economics, making RAILS far cheaper than proprietary blade PSUs.

Power Density. In data centers, rack space is at a premium, and, hence, the physical volume occupied by a blade enclosure is a key concern. RAILS drastically increases the number of distinct PSUs in the enclosure, but each PSU is individually smaller. To confirm the feasibility of RAILS, we have compared the highest power density available in commodity PSUs, which conform to one of several standard form-factors, with that of PSUs designed for blade centers, which may have arbitrary dimensions. Table 5 compares the power density of two commodity form factors with the power density of HP's c7000 PSUs. We report density in terms of Watts per unit volume normalized to the volume of one ATX power supply. The highly-compact microATX form factor exhibits the worst power density—these units have been optimized for small dimensions but are employed in small form-factor devices that do not require high peak power. Though they are not designed for density, commodity ATX supplies are only 16% less dense than enterprise-class supplies. Furthermore, as RAILS requires only a single output voltage, eliminating the need for many of a standard ATX PSU's components, we conclude that RAILS PSUs fit within blade enclosure volumetric constraints.

Power Savings and Energy Efficiency. To evaluate each power system, we calculate expected power draw and conversion efficiency across blade ensemble utilizations. As noted in Section 2, low average utilization manifests as brief bursts of activity where a subset of blades draw near-peak power. The efficiency of each power delivery solution depends on how long blades are active and how many are simultaneously active. For each utilization, we construct a probability mass function for the number of simultaneously active blades, assuming utilization across blades is uncorrelated. Hence, the number of active blades follows a binomial distribution. From the distribution of active blades, we compute an expected power draw and determine conversion losses from the power supply's efficiency- versus-load curve. We obtain efficiency curves from the Energy Star Bronze 80+ specification [26] for 80+ PSUs and [5] for commodity PSUs.

Figure 10 compares the relative efficiency of PowerNap under each power delivery solution. Using commodity ("Commodity") or high efficiency ("80+") PSUs results in the lowest efficiency, as PowerNap's low power draw will operate these power supplies in the "Red" zone. RAILS ("RAILS") and Dynamic Load-Sharing ("Dynamic") both improve PSU performance because they increase average PSU load. RAILS outperforms all of the other options because its fine-grain sizing best matches PowerNap's requirements.

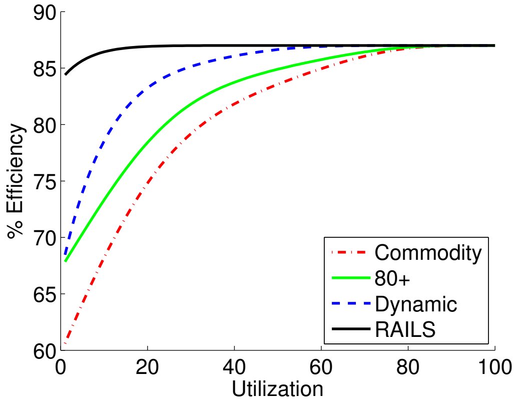


Figure 10: Power Delivery Solution Comparison.

6. Conclusion

We presented *PowerNap*, a method for eliminating idle power in servers by quickly transitioning in and out of an ultra-low power state. We have constructed an analytic model to demonstrate that, for typical server workloads, PowerNap far exceeds DVFS's power savings potential with better response time. Because of PowerNap's unique power requirements, we introduced RAILS, a novel power delivery system that improves power conversion efficiency, provides graceful degradation in the event of PSU failures, and reduces costs.

To conclude, we present a projection of the effectiveness of PowerNap with RAILS in real commercial deployments. We construct our projections using the commercial high-density server utilization traces described in Table 1. Table 6 presents the power requirements, energy-conversion efficiency and total power costs for three server configurations: an unmodified, modern blade center such as the HP c7000; a PowerNap-enabled system with large, conventional PSUs ("PowerNap"); and PowerNap with RAILS. The power costs include the estimated purchase price of the power delivery system (conventional high-wattage PSUs or RAILS), 3-year power costs assuming California's commercial rate of 11.15 cents/kWh [28], and a cooling burden of 0.5W per 1W of IT equipment [18].

PowerNap yields a striking reduction in average power relative to Blade of nearly 70% for Web 2.0 servers. Improving the power system with RAILS shaves another 26%. Our total power cost estimates demonstrate the true value of PowerNap with RAILS: our solution provides power cost reductions of nearly 80% for Web 2.0 servers and 70% for Enterprise IT.

Table 6: Power and Cost Comparison.

	Web 2.0			Enterprise		
	Power	Efficiency	Power costs	Power	Efficiency	Power costs
Blade	6.4 kW	87%	$29k	6.6 kW	87%	$30k
PowerNap	1.9 kW	67%	$10k	2.6 kW	70%	$13k
PowerNap with RAILS	1.4 kW	86%	$6k	2.0 kW	86%	$9k

Acknowledgements

The authors would like to thank Partha Ranganathan and HP Labs for the real- world data center utilization traces, Andrew Caird and the staff at the Michigan Academic Computer Center for assistance in collecting the Cluster utilization trace, Laura Falk for assistance in collecting the departmental server utilization traces, Mor Harchol-Balter for her input on our queuing models, and the anonymous reviewers for their feedback. This work was supported by an equipment grant from Intel, and NSF grant CCF-0811320.

References

[1] L. Barroso and U. Hölzle, “The case for energy-proportional computing,” *IEEE Computer*, Jan 2007.

[2] C. Bash and G. Forman, “Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center,” in *Proc. of the 2007 USENIX Annual Technical Conference*, Jan 2007.

[3] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, and R. Rajamony, “The case for power management in web servers,” *Power Aware Computing*, Jan 2002.

[4] J. Chase, D. Anderson, P. Thakar, and A. Vahdat, “Managing energy and server resources in hosting centers,” in *Proc. of the 18th ACM Symposium on Operating Systems Principles*, Jan 2001.

[5] ECOS and EPR, “Efficient power supplies for data center,” ECOS and EPR, Tech. Rep., Feb. 2008.

[6] X. Fan, W.-D. Weber, and L. A. Barroso, “Power provisioning for a warehouse-sized computer,” in *Proc. of the 34th Annual International Symposium on Computer Architecture*, 2007.

[7] U. Hölzle and B. Weihl, “PSU white paper,” Google, Tech. Rep., Sep 2006.

[8] Hynix, “Hynix-DDR2-1Gb,” Aug 2008.

[9] Intel, “Intel Pentium M processor with 2-MB L2 cache and 533-MHz front side bus,” Jul 2005.

[10] Intel, “Intel Pentium dual-core mobile processor,” Jun 2007.

[11] Intel, “Quad-core Intel Xeon processor 5400 series,” Apr 2008.

[12] J. Laudon, “UltraSPARC T1: A 32-threaded CMP for servers,” Invited talk, Apr 2006.

[13] C. Lefurgy, X. Wang, and M. Ware, “Server-level power control,” in *Proc. of the IEEE International Conference on Autonomic Computing*, Jan 2007.

[14] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller, “Energy management for commercial servers,” *IEEE Computer*, vol. 36, no. 12, 2003.

[15] K. Leigh and P. Ranganathan, “Blades as a general-purpose infrastructure for future system architectures: Challenges and solutions,” HP Labs, Tech. Rep. HPL-2006-182, Jan 2007.

[16] Micron, “DDR2 SDRAM SODIMM,” Jul 2004.

[17] A. Miyoshi, C. Lefurgy, E. V. Hensbergen, R. Rajamony, and R. Rajkumar, “Critical power slope: understanding the runtime effects of frequency scaling,” in *Proc. of the 16th International Conference on Supercomputing*, Jan 2002.

[18] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, “Making scheduling ‘cool’: Temperature-aware workload placement in data centers,” in *Proc. of the 2005 USENIX Annual Technical Conference*, Jan 2005.

[19] National Semiconductor, “Introduction to power supplies,” National Semiconductor, Tech. Rep. AN-556, 2002.

[20] P. Padala, X. Zhu, Z. Wanf, S. Singhal, and K. Shin, “Performance evaluation of virtualization technologies for server consolidation,” HP Labs, Tech. Rep. HPL-2007-59, 2007.

[21] N. Rasmussen, “AC vs. DC power distribution for data centers,” American Power Conversion, Tech. Rep. #63, 2007.

[22] Samsung, “SSD SATA 3.0Gbps 2.5 data sheet,” Mar 2008.

[23] S. Siddha, V. Pallipadi, and A. V. D. Ven, “Getting maximum mileage out of tickless,” in *Proc. of the 2007 Linux Symposium*, 2007.

[24] SMSC, “LAN9420/LAN9420i single-chip ethernet controller with HP Auto-MDIX support and PCI interface,” 2008.

[25] N. Tolia, Z. Wang, M. Marwah, C. Bash, P. Ranganathan, and X. Zhu, “Delivering energy proportionality with non energy-proportional systems – optimizing the ensemble,” in *Proc. of the 1st Workshop on Power Aware Computing and Systems (HotPower ’08)*, Dec 2008.

[26] U.S. EPA, “Energy Star computer specification v. 4.0,” U.S. Environmental Protection Agency, Tech. Rep., July 2007.

[27] U.S. EPA, “Report to congress on server and data center energy efficiency,” U.S. Environmental Protection Agency, Tech. Rep., Aug. 2007.

[28] U.S. Official Information Administration, “Average retail price of electricity to ultimate customers by end-use sector, by state,” Jul 2008.

[29] P. D. Welch, “On a generalized M/G/1 queuing process in which the first customer of each busy period receives exceptional service,” *Operations Research*, vol. 12, pp. 736–752, 1964.

[30] Q. Wu, P. Juang, M. Martonosi, L. Peh, and D. Clark, “Formal control techniques for power-performance management,” *IEEE Micro*, no. 5, Jan. 2005.