Peak Power Modeling for Data Center Servers with Switched-Mode Power Supplies

David Meisner
meisner@umich.edu

Thomas F. Wenisch
twenisch@umich.edu

Advanced Computer Architecture Lab
University of Michigan

ABSTRACT

Accurately modeling server power consumption is critical in designing data center power provisioning infrastructure. However, to date, most research proposals have used average CPU utilization to infer the power consumption of clusters, typically averaging over tens of minutes per observation. We demonstrate that average CPU utilization is not sufficient to predict peak power consumption accurately. By characterizing the relationship between server utilization and power supply behavior, we can more accurately model the actual peak power consumption. Finally, we introduce a new operating system metric that can capture the needed information to design for peak power with low overhead.

Categories and Subject Descriptors

C.5.5 [**Computer System Implementation**]: Servers

General Terms

Measurement

1. INTRODUCTION

Power- and energy-related costs make up almost 50% of data center lifetime costs and are increasing [3]. Whereas energy costs have received significant attention lately, the infrastructure investment required to power thousands of servers has received less attention and remains high [7]. Modeling these systems accurately is critical for large-scale evaluation [9, 11]. Designing power infrastructure requires understanding the aggregate *peak power* of multiple servers at the rack, cluster and data center level. Monitoring the power consumption of individual servers can be costly, requiring power meters at each server. Rack-level monitoring can provide more economic monitoring, but masks individual server behavior. As an alternative to direct measurement, prior work has shown that CPU utilization can provide an accurate proxy for average power, as average utilization is roughly proportional to average power [15, 6]. However, estimation approaches that average utilization at a coarse-grain are not sufficient to predict peak power spikes.

Today, it is not unusual for data center operators to collect utilization traces with sampling intervals of tens of minutes to hours; finer-grained sampling is prohibitive for tens of thousands of servers due to storage and processing overheads. For example, for a 1000-node cluster, sampling at the granularity of the OS scheduler (100Hz) would produce 225 GB of data per week.

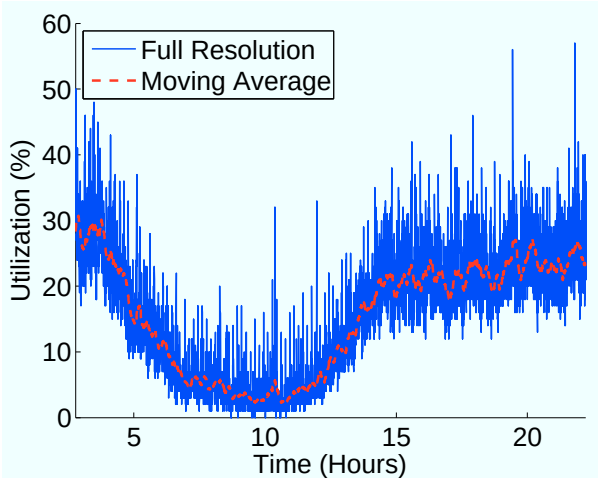


Figure 1: Example data center trace: Average utilization masks important spikes.

Figure 1 shows the utilization of a production data center server running a web 2.0 service. The trace was collected at ten-second granularity ("Full Resolution") and has a wide dynamic range. Unfortunately, most utilization traces are not collected with such fine detail. When a ten-minute average is used instead ("Moving Average"), significant detail is lost. For example, though there appears to be little demand around hour ten when examining the coarse average, the fine-grain trace shows there are still brief spikes that exceed the maximum of the coarse trace.

We show that to determine a server's peak power, it is critical to understand the behavior of server switched-mode power supply units (SMPSUs). These devices are highly efficient, but rely on a switching and charge storage mechanism that introduces RC behavior into the power draw. While SMPSUs are well understood, our contribution is to connect the operating system view of a server to the peak power draw at the power outlet. We construct an analytic model of a server's power draw that can be understood using signal processing techniques.

Finally, we introduce an easily-collected operating system-level metric that can be used determine peak power draw over a time epoch. By leveraging our model, we are able to incorporate the RC behavior of SMPSUs and track peak power with low overhead. This mechanism can enable logging of peak power over time and will facilitate large-scale data center power-provisioning research.

In short, we contribute:

ISLPED'10, August 18–20, 2010, Austin, Texas, USA.

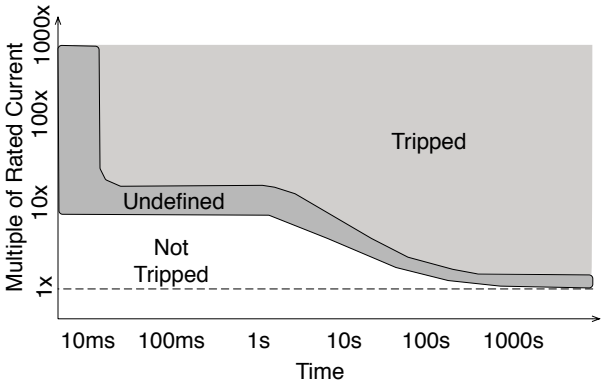


Figure 2: Example PDU Circuit Breaker Curve.

- An illustration of the challenges of collecting utilization at fine granularity and the important differences between peak and average metrics.
- A characterization of server Switched-Mode Power Supply Units and an analytic signal processing model of the relationship between their power draw and server utilization.
- A new operating system-level metric to capture peak power information for server instrumentation.

2. RELATED WORK

There is a significant body of work investigating processor-level power modeling [5, 4]. In particular, [5] shows power consumption of a microprocessor can be predicted accurately using various performance counters. However, for data center design, predicting CPU power alone is insufficient as it only accounts for a fraction of server power (typically 20-30%). Instead, full-system power models that account for non-CPU components (e.g., memory, disk, etc.) are needed.

Multiple studies have demonstrated that full-system average power is approximately linear with respect to CPU utilization [15, 6].

$$P_{\text{Total}} = P_{\text{Dyn}} \cdot U_{\text{Avg}} + P_{\text{Idle}} \quad (1)$$

Particularly when aggregated over a large number of servers, these averages are surprisingly accurate. However, this model provides only *average* power estimates and do not predict peaks.

Switched mode power supply design is well understood [13]: many models are available [16], including many that are based on signal processing [10]. However, the behavior PSUs in running systems, particularly the relationship between CPU utilization and PSU peak power, has not been characterized. We take a full system approach to server power draw, predicting peak power from the logical view of an operating system.

3. DATA CENTER POWER PROVISIONING

Provisioning power infrastructure for data centers is extremely costly; typical installations incur $10-$20 per provisioned watt [3]. A large fraction of this cost is associated with installing power distribution units (PDUs), which provide power to groups of servers. Often, total PDU capacity is overprovisioned [6, 8, 12]. Data center designers typically use conservative estimates for the maximum power draw of servers. However, in aggregate, racks and clusters of servers rarely draw their peak power at the same time [6]. At the PDU level, this conservatism means that PDUs are rarely fully loaded; the provisioned capacity at each PDU is well above its average load.

One method to reduce power infrastructure cost is to *oversubscribe* a data center's power infrastructure with more servers than

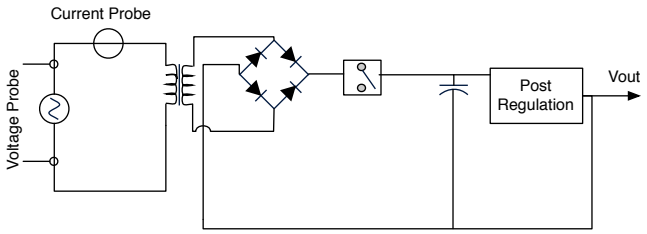


Figure 3: Simplified switched-mode power supply design and instrumentation to measure power.

it can support [6]. Oversubscribing power infrastructure introduces the possibility of exceeding the maximum rated power for a PDU; this scenario can throw a circuit breaker and take a section of the data center offline. Figure 2 depicts a simplified version of a typical PDU circuit breaker curve [1]. Tripping a breaker is not an instantaneous event; the PDU can tolerate brief current overloads. However, since several servers might incur power spikes at the same time, to maintain availability, a design must guarantee that the total power draw at each server remain below a predetermined limit.

Power capping is a data center-level technique to set hard limits on servers' peak power consumption (e.g., using a control loop) [14, 6, 8]. Throttling server power (via frequency/voltage scaling) is used as a safety mechanism to ensure maximum power levels are not exceeded and circuit breakers are not tripped. With this mechanism in place, PDUs and other power provisioning infrastructure can be oversubscribed, reducing the effective capital cost. Since load/power spikes are rare, little performance is lost to throttling. Capital costs can be further reduced by using *Power Routing* [12], which allows load to be shifted among PDUs during imbalances.

All of these techniques require software mechanisms to track and predict peak power, to manage power budgets at each server, circuit, and PDU, while minimizing performance throttling. Though peak power could be tracked with explicit metering and logging, assessing peak power directly from operating system-level metrics can drastically reduce costs. To infer and record peak power from OS level metrics, we must understand the operation of server power supplies and its relationship to utilization.

4. UNDERSTANDING SMPSU BEHAVIOR

In this section, we explore the behavior of SMPSU devices in servers and its connection to OS-observed utilization. To ensure our observations generalize, we study two different kinds of systems: a smaller system with a cheap, commodity PSU ("Commodity") and a larger system with an enterprise class PSU ("Server"). Since SMPSU designs vary, these systems exhibit some differences in behavior; nevertheless, aspects relevant to predicting peak power draw are similar. First, we briefly describe the operation of SMPSU devices; a detailed description of such devices may be found in [13]. Next, we describe our experimental methodology for characterizing these devices and measuring the important behaviors of SMPSUs. Finally, we develop a high-level signal processing model to predict peak power.

4.1 Operation

Modern servers use some form of SMPSU to convert from 120/240V AC to 12V DC power. SMPSUs are far more efficient than, for example, linear regulators, but are also more complicated in their design. While the design and operation of these devices is well understood, our contribution is to understand how the processor's logical view of utilization maps to the physical power draw at an outlet. This connection is important because, as we show, the de-

sign of typical SMPSU devices impacts the manner in which we should model server power.

Figure 3 illustrates the topology of a typical SMPSU. In the first stage, line AC voltage is rectified and passed to a storage element (i.e., a capacitor). The second stage typically includes some form of regulation to maintain a DC voltage. As the demands of the DC devices powered by the SMPSU change, the SMPSU controller adjusts the duty cycle of its switching to transfer more or less charge.

Because of its design, a SMPSU does not draw power continuously. Instead, there are spikes of current during each charging cycle. During these spikes, current is transferred from the high-voltage supply to the SMPSU capacitor. Example measurements of power draw in idle systems are shown in Figure 4. While the basic principle of operation is the same, the Commodity PSU clearly transfers current in more pronounced spikes than the Server. This difference is due to extra switching regulation in the first stage, common in higher-end devices, used to produce a more continuous current draw.

Because of the capacitor used to store and transfer charge, this circuit exhibits RC behavior. We would like to know the effect a typical SMPSU has on the frequency response and phase of power consumption with respect to processor utilization. Such an understanding will allow us to better determine at what granularity we must track utilization.

4.2 Experimental Methodology

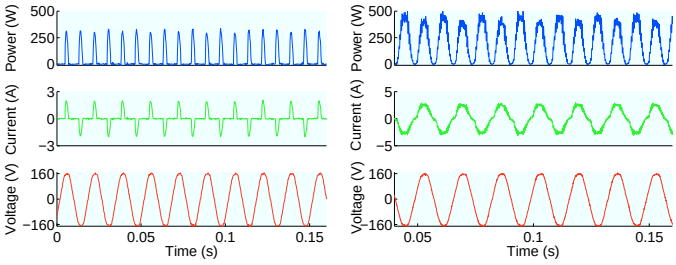
We measure the power consumed by a server at the wall outlet to observe its behavior with respect to utilization. We accomplish this by simultaneously measuring the instantaneous voltage over and current entering the PSU as illustrated in Figure 3. A simple power probe is not sufficient for this measurement because these devices typically report average RMS power, masking the phenomenon we are attempting to observe. We record detailed traces of the instantaneous signals from both these probes.

We measure the two machine configurations described earlier: a smaller, inexpensive system ("Commodity") and an enterprise, dual socket system ("Server"). Comparing these systems allow us to determine if the size or price class of the machines influences their behavior. The measured idle and maximum power consumptions of these machines are provided in Table 1.

To understand how utilization and the PSU interact, we wish to characterize two effects. First, we investigate how the frequency at which utilization varies is reflected in the power draw at the wall outlet. Intuitively, we expect that utilization variations that occur faster than some cut-off frequency will be filtered by the PSU behavior and not be reflected at the outlet; measuring utilization at a granularity finer than this cut-off is not necessary for accurate peak power prediction. The precise cut-off frequency has not previously been characterized. Second, we wish to determine the latency between utilization changes and a corresponding change in the SMPSU power draw; in other words, how rapidly a step function in utilization affects power draw at the PSU.

To observe the effect of the frequency of utilization variation, we use a synthetic workload we refer to as SQUARE. This workload produces a square wave in system utilization by switching cores between a matrix multiplication kernel designed to maximize CPU power draw and an idle mode where the processors enter a power-save mode. The duty cycle of the workload is fixed at 50%, producing an average utilization of 50%. We vary the frequency of the square wave and observe the response at the PSU.

To characterize the latency between a utilization change and the PSU response, we idle the system and wait until the PSU behavior reaches steady state. We then trigger execution of the matrix multiply kernel on all cores. We refer to this synthetic workload as STEP. Because CPU utilization is not directly observable externally, we send a signal (using general purpose I/O that is significantly faster than the expected SMPSU response) immediately before the transition to initiate timing at our oscilloscope.



(a) **Commodity** (b) **Server**

Figure 4: Systems at idle: switch-mode power supplies draw power in discrete spikes.

	RMS Power (W)		Dyn. Range (max/min)
	Idle	Max	
Commodity	57 W	188 W	3.3
Server	212 W	355 W	1.7

Table 1: Systems Under Test.

4.3 Measured Behavior

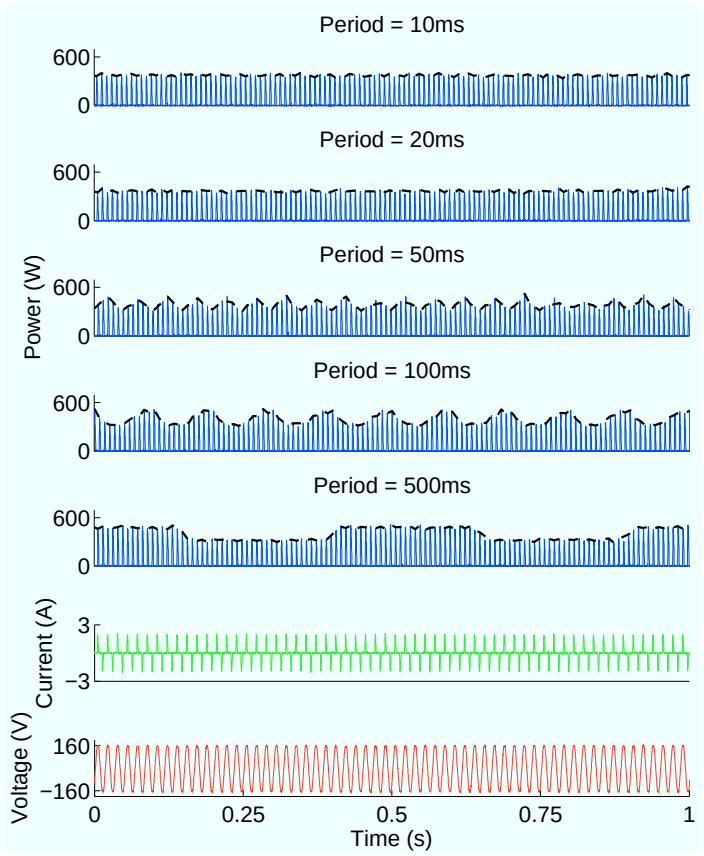
We now present results for frequency and phase delay behavior.

Frequency Response. To understand the relationship between the frequency of utilization and power, we ran the SQUARE benchmark on both test systems with varying frequencies. Figure 5 shows the observed instantaneous power at each frequency on both systems. We used 100 Hz as the maximum frequency we investigate because we found that the Linux kernel could not reliably schedule faster than this frequency (in general this will depend on the OS kernel configuration). The current draw and voltage of an idle system are provided for reference. The dotted line ("Envelope"), connects the peaks of the power waveform and functions as an envelope detector. The varying utilization modulates the instantaneous power waveform of a system at idle; the envelope detector reveals the modulated signal.

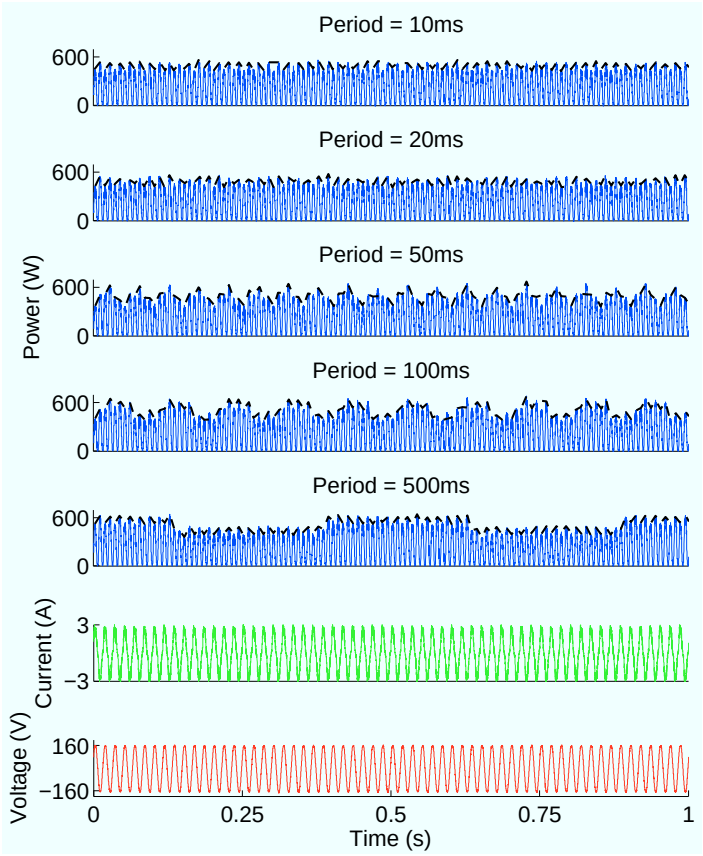
The results in Figure 5 show that the frequency of modulation has a strong influence on the observed power waveform. As long as the utilization of the CPU is modulated slowly, the envelope of power draw roughly resembles a square wave, matching the CPU behavior. However as the frequency is increased, the power draw becomes more uniform.

We draw several conclusions from this result. First, the SMPSU effectively acts as a low-pass filter with respect to utilization. We construct a model for this behavior in Section 4.4. Second, to faithfully model the peak power of an SMPSU, it is necessary to monitor utilization at fine granularity (near the kernel scheduling interval for many systems). Averages that use coarser windows lose information. However, monitoring utilization at a time-scale finer than 50 Hz is unnecessary: the variations in the 50 Hz (20 ms period) and 100 Hz (10 ms period) waveforms are filtered.

To give a better sense of the filtering in the SMPSU system, we construct a Bode-style plot of the systems in Figure 7. The figure illustrates the attenuation of the modulating signals. We show the Commodity and Server frequency responses compared to a idealized first-order RC low-pass filter ("Ideal"). We find that these systems are closely approximated by a filter with a frequency cutoff of 30Hz.

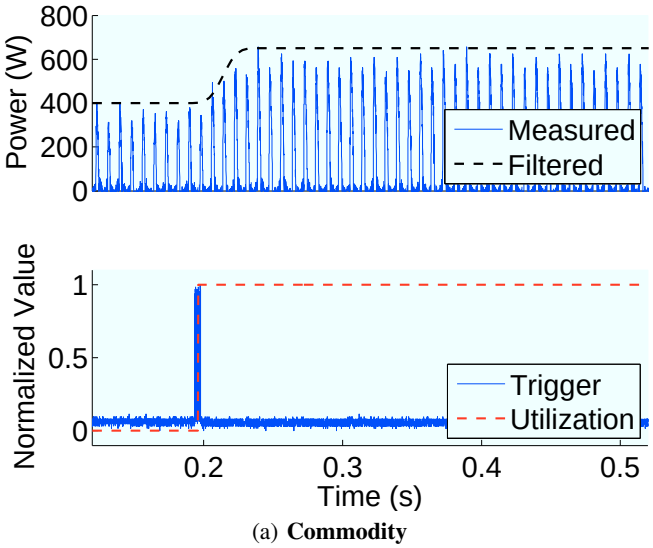


(a) **Commodity**

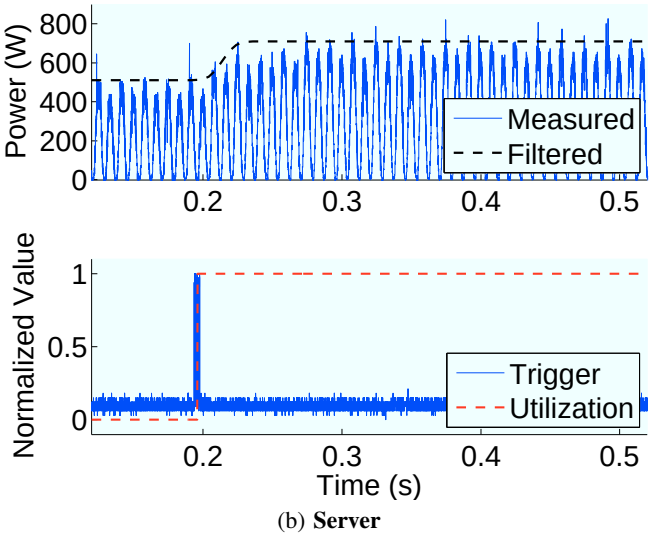


(b) **Server**

Figure 5: Effect of modulation frequency: All examples have the same *average* utilization, but exhibit different *peak* power.



(a) **Commodity**



(b) **Server**

Figure 6: Delay of a step function in utilization.

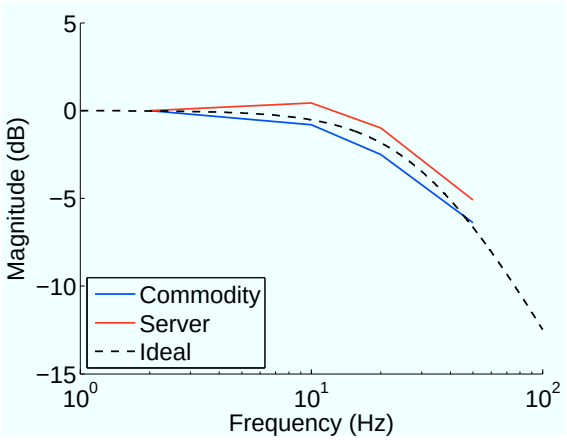


Figure 7: Frequency response.

Phase Delay. Next, we investigate the phase delay of SMPSU power load using the STEP workload. The step function response of both test machines is provided in Figure 6. There is a delay in the instantaneous power response, which rises as one would expect of a step function with RC filtering. We report the I/O signal indicating the utilization transition ("Trigger"), as well as the implied utilization waveform ("Utilization"). Finally, we show a filtered ("Filtered") step function that fits the observed rising waveform. This signal is produced from a first order RC filter with a frequency cutoff of 30 Hz.

4.4 Model

The goal of our investigation has been to model the peak power draw of SMPSU for server systems. Accordingly, we now construct an analytic model of the PSU behavior using signal processing. We start with the observation that the PSU is effectively exhibiting *amplitude modulation* (AM). An idle system demonstrates the carrier signal: the periodic spikes in current consumption. This signal is then modulated by changes in utilization. The block system diagram for our model is illustrated in Figure 8.

We can describe the observed power draw of an SMPSU as a signal:

$$p_{\text{wall}}(t) = p_{\text{dyn}} \cdot c(t) \cdot (h(t) * x(t)) + p_{\text{idle}} \quad (2)$$

Where $p_{\text{wall}}(t)$ is the observed time varying power consumption, $c(t)$ is the SMPSU current carrier waveform, $h(t)$ is the transfer function for a low-pass filter and $x(t)$ is the *instantaneous* fractional utilization. Note that $x(t)$ can only take on one of 2^N values, where N is the number of cores in the system. Finally, p_{dyn} and p_{idle} are constants that are the same as in Equation 1 and are provided for our system in Table 1. While we use a relatively simple power model, if more sophisticated models are needed (e.g., to model the use of low-power modes), they can easily fit within this framework; we leave such extensions to future work.

We have observed that a first order low-pass filter is quite accurate; therefore, the transfer function is:

$$h(t) = \frac{1}{\tau} e^{-t/\tau} \quad (3)$$

Where τ is the time constant and is approximately 33 ms for our system (alternatively, the cutoff frequency is 30 Hz).

5. SERVER PEAK POWER ACCOUNTING

A key result of our measurement study is that utilization must be monitored at a granularity below 30 Hz to predict peak power.

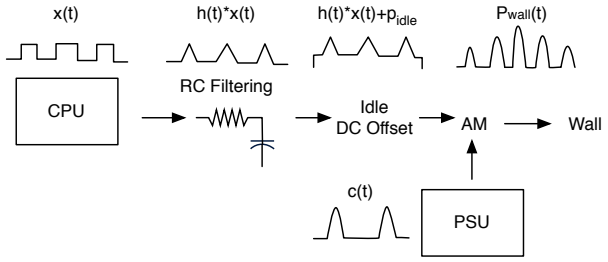


Figure 8: Simplified Server SMPSU Model.

However, finer-grained variation is filtered by the RC behavior of the power supply and need not be monitored. With our new understanding of the operation of SMPSUs and their relationship with server utilization, we construct a low-overhead method to infer peak power from utilization in the operating system kernel. We then validate our model using real machines and show we can predict the peak power trace with an error below 20%.

5.1 A Compact Metric for Peak Power

We propose a new operating system-level metric to track spikes in peak power. We use the model presented in Section 4.4 to filter fine-grain utilization signals (idle/busy transitions) collected in the OS scheduler to determine and record maximum power draw over a time epoch T. To obtain accurate estimates of power draw spikes, we must know the RC response of a particular PSU, which may vary among PSUs. (However, the fact that the two PSUs we study, which differ drastically in their design, exhibit similar RC response provides some evidence that other PSUs will have similar behavior). Over the course of each epoch, we evaluate the estimated power at each sampling interval and retain the peak value.

Most current releases of the Linux kernel are tickless [2]; that is, they operate without a periodic timer interrupt, and can have a variable scheduling interval. Variations in the length of idle/busy periods complicate construction of the input utilization signal to our filter-based model; we must correct for these variations prior to calculation. Note that these corrections do not lose information, as idle/busy transitions cannot occur without invoking the scheduler. We detect scheduler transitions at each core and compute utilization in sampling intervals of 4 ms each.

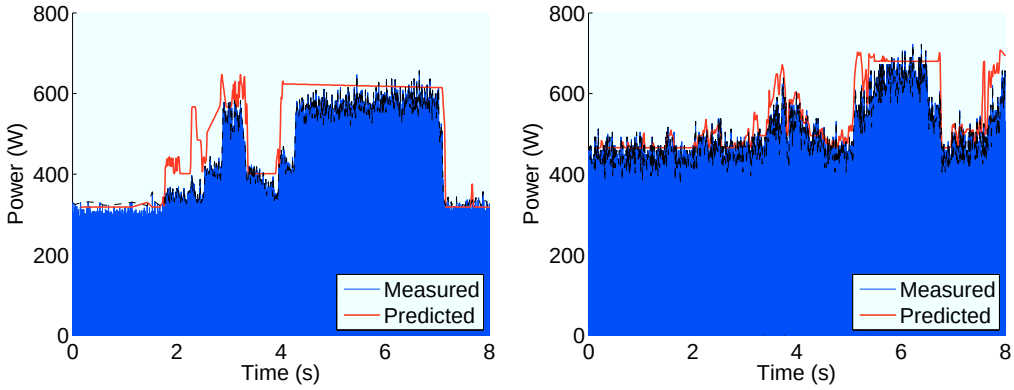
To construct an estimator for peak power, we transform the utilization signal using an in-kernel finite impulse response (FIR) filter of the form:

$$y[n] = \sum_{i=0}^{N} b_i x[n-i] \quad (4)$$

This processing allows us to model the RC behavior of the PSU. Since our tracking and processing takes place in the scheduling subsystem of the OS kernel, it must be light-weight and use fixed point arithmetic [17]. We have found that a 10^{th} order FIR filter captures the behavior well. This filter can compute our metric easily, it requires only the last 10 utilization observations and 10 multiply-accumulate operations per update.

5.2 Validation

We validate our models against the power consumption of the two server configurations presented in Table 1. Two representative traces of measured power are presented in Figure 9. These traces were collected while the system executed a parallel compile of the Linux kernel, a workload that produces a chaotic, bursty utilization



(a) **Commodity** (b) **Server**

Figure 9: Predicted peak power closely follows measured value.

pattern. The instantaneous power ("Measured") is measured the same way as described in Section 4.2.

We overlay our predicted power ("Predicted"), which tracks peak power well, but can still overshoot occasionally. Fortunately, this model tends to be conservative, and overestimates power more than it underestimates. Hence, it will provide conservative estimates in, for example, studies of power budgeting/capping. In this example, the Commodity and Server machines exhibit a normalized root mean square deviation (NRMSD) of 14% and 19% respectively.

5.3 Future Work

Our model was constructed with an approach that was intentionally principled (modeling the PSU as a linear system) and simple (using low-order FIR filter), which allowed us to easily modify the kernel to track and update an online model. More sophisticated models may be able to achieve greater accuracy, but will likely come at the expensive of kernel overhead. For example, our model tends to overestimate power as utilization increases, but not show a similar behavior as utilization drops; this phenomenon may suggest a model that incorporates asymmetric rise and fall times. Furthermore, compared to our SQUARE and STEP workload, a validation of a kernel compilation exercises more sub-systems of a server (particularly I/O). The simple linear model we assume for power may be improved by incorporating more sophisticated metrics for memory and disk usage. We leave such improvements to future work.

6. CONCLUSIONS

We have demonstrated the challenge of modeling the peak power consumption of servers using CPU utilization. Our study characterizes a previously-ignored relationship between OS-level utilization and the behavior of SMPSUs in modern servers. By measuring real server PSUs, we have demonstrated that utilization must be monitored at a granularity of 33 ms or below to predict peak power. We introduce an OS-level solution, based on a light-weight signal-processing-inspired model of the RC behavior of PSUs, and demonstrate that peak power can be approximated to within a NRMSD of 20%.

7. ACKNOWLEDGEMENTS

The authors would like to thank Heath Hoffman for discussions on the design of commercial SMPSUs and the anonymous reviewers for their feedback. This work was supported by grants from Intel, Google, and NSF grant CCF-0811320.

8. REFERENCES

[1] "Typical circuit breaker trip curve," Tyco Thermal Controls, 2003.

[2] *Getting maximum mileage out of tickless*, 2007.

[3] L. Barroso and U. Hölzle, *The Datacenter as a Computer*. Morgan Claypool, 2009.

[4] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," *SIGARCH Comput. Archit. News*, 2000.

[5] G. Contreras and M. Martonosi, "Power prediction for intel xscale®processors using performance monitoring unit events," in *ISLPED '05: Proceedings of the 2005 international symposium on Low power electronics and design*, 2005.

[6] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ISCA '07: International symposium on Computer architecture*, 2007.

[7] J. Hamilton, "Internet-scale service infrastructure efficiency," Keynote at the International Symposium on Computer Architecture (ISCA), 2009.

[8] C. Lefurgy, X. Wang, and M. Ware, "Server-level power control," in *ICAC '07: Proceedings of the Fourth International Conference on Autonomic Computing*. Washington, DC, USA: IEEE Computer Society, 2007, p. 4.

[9] D. Meisner and T. F. Wenisch, "Stochastic queuing simulation for data center workloads," *EXERT: Exascale Evaluation and Research Techniques*, 2010.

[10] R. D. Middlebrook, *Proceedings of the IEEE*, vol. 76, no. 4, 1988.

[11] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. VanGilder, "Understanding and absracting total data center power," in *WEED '09: Workshop on Energy Efficienct Design*, 2009.

[12] S. Pelley, D. Meisner, P. Zandevakili, T. F. Wenisch, and J. Underwood, "Power routing: Dynamic power provisioning in the data center," *ASPLOS: Architectural support for programming languages and operating systems*, 2010.

[13] A. Pressman, *Switching Power Supply Design*. McGraw-Hill, Inc., 1998.

[14] P. Ranganathan, P. Leech, D. Irwin, and J. Chase, "Ensemble-level power management for dense blade servers," *SIGARCH Comput. Archit. News*, vol. 34, no. 2, 2006.

[15] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A comparison of high-level full-system power models," *HotPower '08: Workshop on Power Aware Computing and Systems*, 2008.

[16] V. Vorperian, "Simplified analysis of PWM converters using model of PWM switch. Continuous conduction mode," *IEEE Transactions on Aerospace Electronic Systems*, vol. 26, 1990.

[17] R. Yates, "Practical considerations in fixed-point fir filter implementations," Digital Signal Labs, Technical Reference, 2007.