PowerMon: Fine-grained and Integrated Power Monitoring for Commodity Computer Systems

Daniel Bedard, Min Yeol Lim, Robert Fowler, and Allan Porterfield
Renaissance Computing Institute
Chapel Hill, NC 27517
Email: {danb, mylim, rjf, akp}@renci.org

***Abstract*—We have developed PowerMon and PowerMon2, low-cost power monitoring devices that operate inside commodity computer systems, to analyze performance and power consumption tradeoffs in computer applications. Inserted between a system's power supply and motherboard, PowerMon monitors voltage and current on six DC rails and reports measurements at a rate of up to fifty samples per second through a USB interface, allowing monitoring by the target or a separate host. PowerMon2 has a smaller form factor that fits in a standard 3.5" hard drive bay, allowing it to be used in a 1U server chassis. It features a faster measurement rate of up to 1024 Hz on a single channel or 3072 Hz divided among multiple channels. PowerMon2 also adds two measurement channels for additional peripherals, such as disks and graphical processing units.**

The PowerMon devices have been used to resolve and highlight variations in power consumption and energy efficiency at the subsystem level during separate operating phases of scientific performance benchmarks, such as NAS BT and SP.

The device parts cost is low enough to provision an entire cluster for power monitoring and adaptation. Complete schematics, board layouts, and source code for the PowerMon devices are available under a BSD-style open source license at ilab.renci.org/powermon.

I. Introduction

The traditional development paradigm for high performance parallel and distributed computing systems favors increased performance with little regard to other dependent factors. This singular focus on performance has resulted in excessive power consumption, such that the operating cost of a system can surpass the system's initial purchase price within a few years [1]. Consequently, power-aware computing has recently been recognized as a critical research issue in computing systems, and substantial effort is now being committed to measuring and improving power efficiency at the system and cluster levels [2], [3], [4].

In order to study and improve energy efficiency, researchers need to measure computer power consumption in detail. There are many devices on the market designed to measure power consumption of electronics. Popular external power meters, such as the WatchDog.com PowerEgg [5] and WattsUp? Pro [6], allow the measurement of power consumption at the system level at a maximum frequency on the order of twice per second. The limitation of these devices becomes apparent, however, when considering cases such as a quad-core 3-GHz processor that issues three instructions per clock cycle. In this example, a measurement rate of 2 Hz translates to a process resolution of 18 billion instructions. Further, because the DC output of a computer power supply is buffered and filtered for stability, fast variations in the DC load often do not translate to corresponding variations on the AC side of the supply. For these reasons, these legacy power meter devices are not suitable to monitor system power usage with sufficient detail to perform dynamic profiling to make *in situ* decisions for power / performance adaptation during application execution.

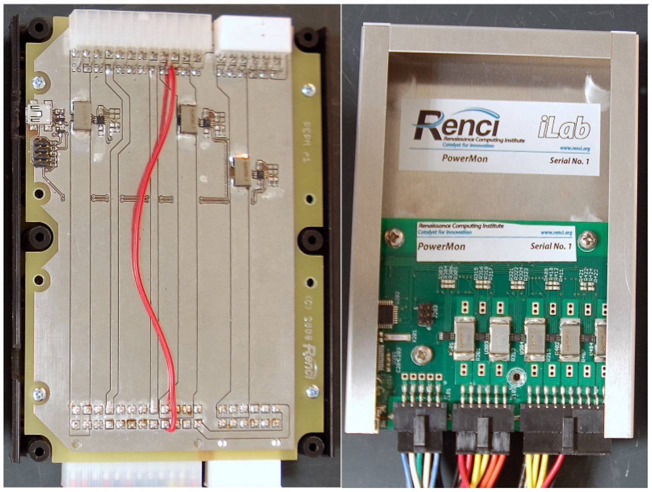


Fig. 1. PowerMon (*left*) and PowerMon2 (*right*)

The approach discussed in [7] addresses some of these problems to implement a power profiling framework for monitoring cluster power consumption by using current sense resistors and multimeters, which are attached to the outputs of a computer power supply and connected to a host computer using RS232. Though this method allows researchers direct access to the DC motherboard inputs of individual target systems, it does not scale well to large clusters, as it requires a nontrivial amount of customization to each machine, and it has relatively high equipment costs. Further, the coarse timing resolution of 4 Hz prevents the researcher from being able to discern some meaningful and interesting power consumption

PowerMon development was supported by DoD and under grant number DE-FC02-06ER25764 from the U.S. Department of Energy PERI Performance Engineering Research Center (PERC-3).

978-1-4244-5853-0/10/$26.00 ©2010 IEEE

Table I
POWERMON AND POWERMON2 CHARACTERISTICS

	PowerMon	PowerMon2
Dimensions (cm)	17.5 x 12.4 x 3.8	15.2 x 10.2 x 2.5
Measurement Channels	6	8
Voltage Accuracy	±0.90%	±0.90%
Current Accuracy	-6.6% / +6.8% (worst-case)	-6.6% / +6.8% (worst-case)
Measurement Frequency	up to ˜50 Hz not configurable	up to 1024 Hz / 3072 Hz (channel / aggregate)
Parts Count	21	26
Total Cost	$102.34	$147.07

features (discussed in Section III-C) that occur during program execution.

PowerMon and PowerMon2, pictured in Fig. 1, are inexpensive devices intended to produce accurate, fine-grained power measurements in commodity computing systems. The devices measure voltage and current on the individual DC power rails between a system's power supply and the motherboard and peripherals. Custom cables are used to accommodate various styles of auxiliary connectors used by different manufacturers.

PowerMon has several advantages over existing measurement tools. It isolates measurements between a machine's individual subsystems. It interfaces directly with most commodity computing systems using the standard ATX pin configuration, and its software interfaces allow simple integration of control and data functions within a researcher's application.

Building on the virtues of PowerMon, PowerMon2 was designed to provide finer measurement granularity—up to 1024 Hz for an individual channel or 3072 Hz shared among multiple channels. It implements timestamping to simplify and improve measurement synchronization with program execution. PowerMon2 fits into a 3.5" drive bay, providing better physical integration into the target system, and it has two additional measurement channels to monitor disks, graphical processing units (GPUs), or other peripherals. PowerMon and PowerMon2 are compared in Table I.

Both PowerMon and PowerMon2 can be used in a "self-monitoring" configuration, in which the target also collects its own power consumption data. This allows researchers to directly correlate power use with specific activities within a software application, but it also introduces some overhead in both the power and processing domains. PowerMon2's timestamping capability provides a facility for synchronization on separate target and host machines, provided the machines' clocks can be synchronized and program execution is carefully scheduled.

PowerMon and PowerMon2 have been used to correlate power consumption measurements to individual motherboard subsystems, to profile computer power supply efficiency, and to profile power consumption of discrete procedures within software applications.

II. PowerMon Design

A. Hardware

PowerMon and PowerMon2 permit the measurement of six and eight individual DC rails, respectively. The multitude of channels allows the measurement of the three or more rails used to power the motherboard (this number varies from manufacturer to manufacturer), in addition to peripherals such as hard drives and GPUs. The PowerMon devices are inserted into the computer power circuit by plugging supply-side power cables into input connectors on the devices and then connecting the outputs to another set of cables supplying the motherboard and peripherals.

USB was chosen as the host interface to PowerMon for its speed and ubiquity. PowerMon uses an Atmel ATtiny45 [8] to act as a bridge between the host system's USB port and the power sensors on the device. For PowerMon2, USB transaction control is offloaded to an FTDI FT232R IC [9], which interfaces with the USART of a separate microcontroller, an Atmel ATmega168 [10]. The use of a separate microcontroller in PowerMon2 allows the tasks of scheduling, formatting, and timestamping measurements to be self-contained within the device, reducing host system overhead and increasing the possible measurement rate. The ATmega168 was chosen for its built-in I^2C and USART modules, which permit interrupt-driven communication with the host and the power sensors, greatly simplifying the host interface and the microcontroller code. An internal 8 MHz RC oscillator controls the microcontroller's program counter, and an inexpensive 32 kHz watch crystal was added to allow precise timestamping of the measurement data.

Voltage and current are detected using an Analog Devices ADM1191 [11] digital power monitor IC on each power rail. Each ADM1191 contains a 12-bit analog-to-digital converter, a current-sense amplifier, and an I^2C transceiver. Connected across a 5-milliohm sense resistor on each power rail, this IC provides a digital measurement of a channel's voltage and the current traversing the resistor. The ADM1191s, numbering six for PowerMon and eight for PowerMon2, communicate as slaves on a common two-wire I^2C bus mastered by the ATmega168. Each chip's address is determined by configuring its two address pins with a combination of resistors.

Particular consideration was made to allow the PowerMon devices to accommodate currents of up to 10 amperes on each trace without generating excessive heat. PowerMon uses 6.35mm wide traces on 2-oz. two-layer FR4 substrate, resulting in a temperature rise of less than 4C while carrying 10 amps. To allow PowerMon2 to fit into a standard 3.5" hard drive bay while adding two measurement channels and controlling costs, a 4-layer design on 1-oz. FR4 was used. External traces are 5.59mm wide and internal traces are 12.45mm wide, for a temperature rise of under 20C while carrying 10 amps. For both devices, the net resistance of each channel, including the sense resistor, is less than 10 milliohms. The low resistance ensures that the voltages delivered to the motherboard and peripherals through the PowerMon devices are within the tolerances listed in the ATX specification [12].

B. Firmware

PowerMon implements a modified version of Till Harbaum's i2c-tiny-usb [13] firmware. The i2c-tiny-usb driver on the host encapsulates I^{2}C transactions inside USB transactions.

The additional latency incurred while decoding multiple USB transactions, and the need for the kernel on the host PC to handle measurement scheduling and USB encoding, prohibited accurate correlation between processor events and measurement results. Consequently, for PowerMon2, the firmware was completely rewritten to maximize measurement throughput while maintaining much of the flexibility, in terms of polling configuration, of direct I^{2}C calls. In PowerMon2, the entire I^{2}C transaction is performed in the hardware of the ATmega168. Device-host interaction consists of a configuration and an output mode. In configuration mode, the active sensors and sample rate are configured using a command set made up of ASCII characters. In output mode, PowerMon2 returns a series of 32-bit words containing sensor data and timestamps. Timestamps are interlaced with the sensor data each complete second during data collection.

PowerMon has a maximum measurement rate of 50 readings per second. PowerMon2 is capable of collecting and transmitting over 3000 readings per second.

C. Software

A sample application was developed for PowerMon using the lm-sensors API and for PowerMon2 using the termio API. The PowerMon application queries each sensor sequentially, while the PowerMon2 application allows the specification of a sensor mask to determine which sensors should be polled, a sample period, and a number of samples to collect. Both applications apply the necessary coefficients to convert sensor readings to voltages and currents and output the results to the console, which can then be easily redirected to a file on disk.

A server application was also developed. The application runs as a daemon process, continuously collecting data from PowerMon or PowerMon2 at the maximum sample rate and serving the latest measurement locally or remotely over TCP. Besides power consumption data, a client can configure the process to integrate power measurements over time to output energy consumption values.

Complete schematics, board layouts, and source code for the PowerMon devices are available under a BSD-style open source license at ilab.renci.org/powermon.

III. Evaluation

The PowerMon devices have been tested successfully on three different machines: a quad-core AMD Phenom machine by HP, a quad-core, dual-socket AMD Opteron machine by Dell, and a whitebox machine with a quad-core Intel Nehalem processor. PowerMon seamlessly tracks power usage on each system's power rails. We performed various tests with the PowerMon devices to evaluate the machines' power consumption characteristics. This section provides a subset of the results obtained on a quad-core Intel Nehalem machine. More detailed information is presented in [14].

A. Power consumption in commodity systems

Fig. 2 shows the power usage measured during 5 different workload tests on the Nehalem machine. Each of the six smaller graphs represents an individual power draw, and the large graph displays total system power consumption. The 3.3V, 5V, and two 12V graphs are labeled according to the stated values from the power supply manufacturer, and they reflect power use values measured directly using PowerMon2. The hard disk's 12V and 5V rails were measured individually using PowerMon2, but in the figure, the power consumption is aggregated for clarity. Power consumption attributed to the system's three case fans and power supply conversion inefficiency was derived by subtracting the aggregate power measured by PowerMon2 from the net system power, which was measured using an external power meter.

The Y-axis of each graph represents power consumption in watts. Each set of workload tests is grouped by color, according to the test's focus: blue for PCI bus activity, red for CPU usage, green for memory accesses, brown for disk I/O, and violet for network throughput.

1) PCI Bus Activity: To measure the effect of PCI bus activity, we compared the power utilization at idle state with a VGA card installed to the utilization without the card. The net system difference is about 33W, mostly due to power draw changes in the 3.3V and two 12V power rails. The added load also increases power overhead due to the power supply and case fans.

2) CPU Activity: The second test evaluates the power impact of CPU-intensive workloads using a *while* loop. Multiple processor cores are activated simultaneously by using processor affinity to assign additional loops to idle cores. The greatest variation is seen in the second 12V power rail. Notably, the difference between executing 1 and 4 loops is more than 12W, but there is only about a 4W difference between executing 5 and 8 loops. This feature occurs because while the first 4 loops are executed on distinct physical cores, Hyperthreading technology actually emulates eight cores by having each of four physical cores run two threads simultaneously. The CPU power consumption of running two threads on 2 different physical cores is significantly greater than the power consumed when running them on a single core. Again, consumption due to the power supply and fans increases slightly as CPU load increases.

3) Memory Access: To evaluate the effects of memory-intensive workloads, we used the STREAM benchmark [15], which measures an effective memory bandwidth, on one or more cores. As the number of threads running the STREAM benchmark increases, the power consumption measured on the 5V and two 12V rails increases significantly. Though the STREAM benchmark is not CPU-intensive, it does increase CPU utilization, which the previous tests indicate should largely account for the increased consumption in the second 12V rail. However, since the 5V and first 12V rails demonstrated an increased load during the STREAM benchmark but not during the CPU utilization tests, we have determined that

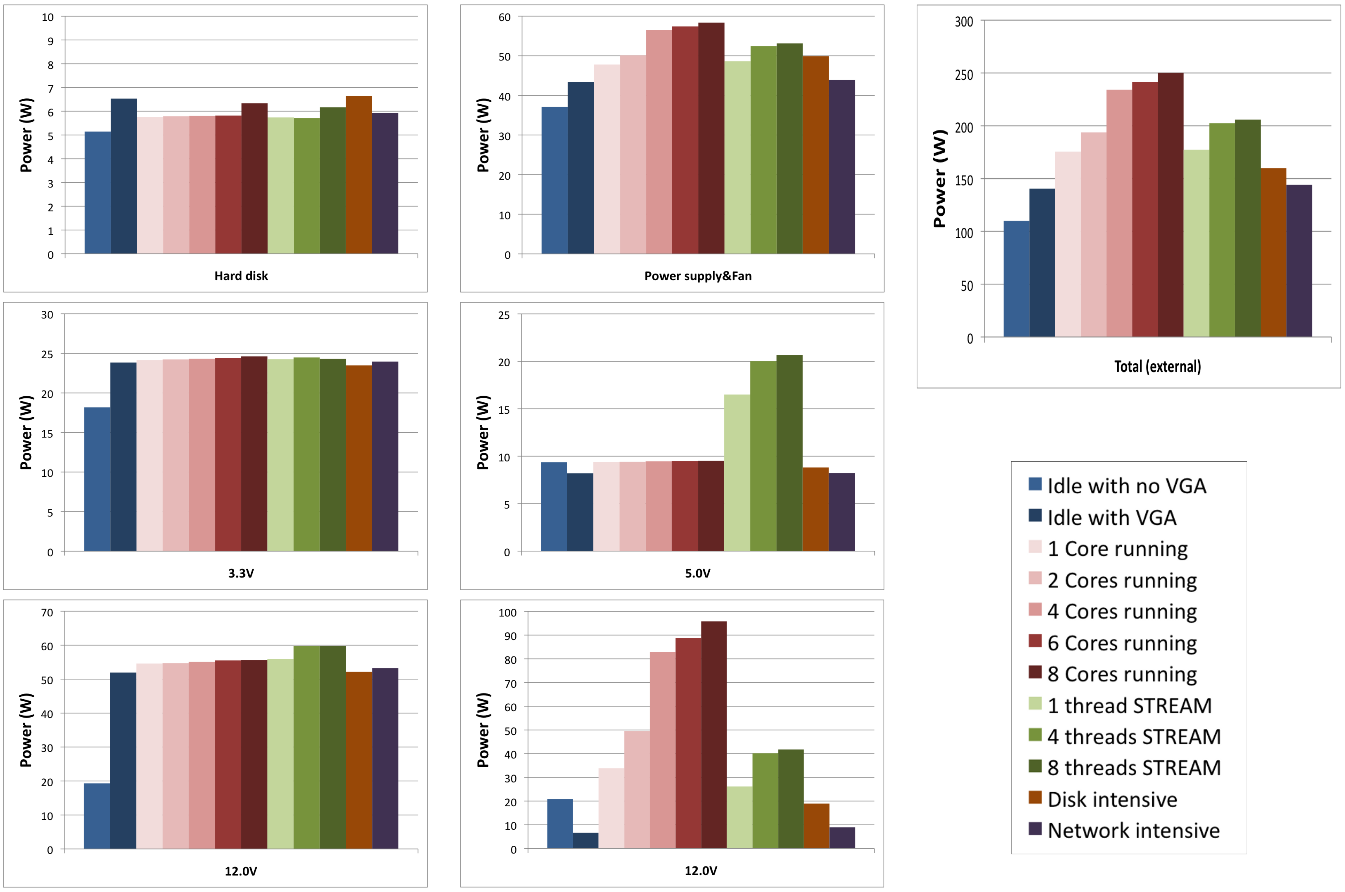


Fig. 2. Power usage by different power rails on a quad-core Intel Nehalem machine

those rails correlate with memory-related power consumption on this system.

4) Disk I/O: For disk-intensive workloads, we used the file system benchmark called Bonnie [16], which determines the read and write performance of hard disks and file systems. In our evaluations, running the benchmark results in no significant difference in power consumption. This occurs because a buffer allows the disk to process reads and writes in batch, minimizing the total number of required disk accesses. The hard disk power ranges between 5.1W and 6.6W during our tests.

5) Network Throughput: Finally, we attempted to isolate the Network Interface Chip (NIC) power by running network-intensive workloads. The workloads are generated by the iPerf benchmark [17], which measures TCP bandwidth. However, the test resulted in no fluctuation in power consumption on any of the rails. This observation confirms that the power consumed by a NIC is the same whether it is maintaining an idle connection or actively carrying data.

In addition to the subsystem workload tests, we measured the power impact of the Nehalem processor's Turbo Boost technology [18], which modulates CPU core frequency based on the number of active cores and workloads. Table II compares CPU power consumption between turbo mode and normal mode. CPU power is measured on the second 12V rail, as increasing CPU activity does not significantly change power draw on the other rails, per Fig. 2. In turbo mode, the CPU consumes about 9-10% more power than in normal mode, but as described in [18], the performance gain is only about 5% as the CPU frequency is increased from 2933 Mhz to 3067 Mhz. Consequently, this test reveals that power-critical applications may benefit by turning turbo mode off.

Table II
CPU POWER CONSUMPTION BY INTEL TURBO BOOST TECHNOLOGY

	Normal power (W)	Turbo mode power (W)
1 loop	34.40	38.26
2 loops	50.43	55.81
3 loops	67.16	73.80
4 loops	83.24	91.38

B. *Power supply efficiency*

One topic motivating the development of PowerMon is the characterization of power supply efficiency. Computer power supplies convert an incoming AC voltage to multiple DC voltages used in a system. However, due to component losses and the cost and complexity of building a more efficient supply, this conversion does not occur with 100% efficiency. Computer power supply efficiency, determined by dividing a supply's output power by its input power, varies based on the load driven by the supply [19]. Further, individual power

Table III
POWER SUPPLY EFFICIENCY BY CPU LOADS ON A QUAD-CORE INTEL NEHALEM MACHINE

	Input (external) power (W)	Output (internal) power (W)	Power supply overhead (W)	Power supply efficiency (%)
Idle	142.79	97.96	44.83	68.60
1 loop	173.99	125.19	48.80	71.95
2 loops	189.54	138.85	50.69	73.26
3 loops	206.49	154.93	51.56	75.03
4 loops	224.16	168.20	55.96	75.04
5 loops	229.80	172.44	57.36	75.04
6 loops	234.97	176.87	58.10	75.27
7 loops	240.17	183.12	57.05	76.25
8 loops	245.12	185.91	59.21	75.84

supplies vary significantly in efficiency characteristics, and the load-variant efficiency of a power supply can hide features in system power consumption. For example, when measuring CPU power utilization, the difference between two chips can be masked by differences in the target systems' power supplies. Therefore, characterizing power supply efficiency is essential to understanding a system's overall power behavior.

PowerMon allows researchers to determine power supply efficiency by directly measuring the output power of a supply. An external power meter measures input power to the supply, and efficiency is calculated by dividing the output power by the supply's input power.

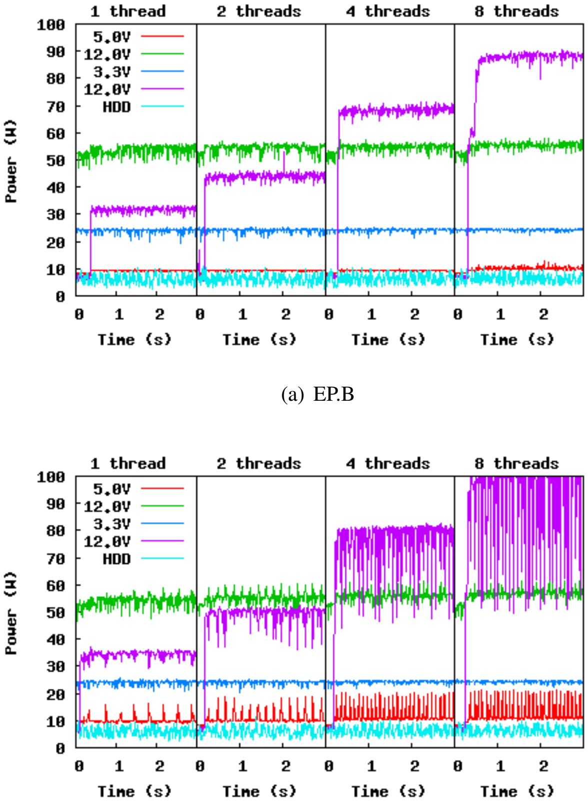
We evaluated the efficiency of a 700W power supply in a quad-core Intel Nehalem machine with Hyperthreading technology (8 effective cores). A WattsUp? Pro external power meter was used to measure the input power to the supply, and PowerMon2 was used to measure the supply's output power. Efficiency was calculated over varying CPU loads by executing CPU-intensive *while* loops on one or more cores, transitioning cores between low-power and active states.

In Table III, we show the power supply efficiency measured at different CPU loads. While output from the supply increases 103W, from 142W to 245W, power supply overhead increases only 15W, from 44W to 59W. Thus, power supply efficiency actually improves by about 7% with an increased load.

The study [19] shows that power supply efficiency increases and then decreases as the power load grows. Our results show a similar trend: efficiency increases at low power loads and stops increasing at higher loads. In our evaluation, however, the tested system operates within a much narrower range of power provisioning than the 700W power supply can actually achieve. We were only able to test a limited range of power consumption behavior. We were not able to increase the load enough to observe a decrease in the supply's efficiency.

C. Power profiling of multithreaded applications

Using the PowerMon devices, we profiled power consumption while running a portion of the NAS Parallel Benchmark (OpenMP version 3.3 CLASS=B). We ran the benchmark on a quad-core Intel Nehalem machine, varying the number of running threads, with each thread assigned to a specific CPU using CPU affinity. The power consumption of each rail was measured at 512 Hz.



(a) EP.B

(b) BT.B

Fig. 3. Power profiling graphs while running EP and BT in NPB OMP benchmark suite. The number of threads is scaled up to 8. Each thread is assigned to a different CPU core while running.

Fig. 3 shows the power profiling results of the NPB OMP EP and BT benchmarks. For each benchmark, results are displayed from test iterations executing on 1, 2, 4, and 8 active threads. Each subfigure focuses on the power usage for the first few seconds of the test in order to clearly show power behavior patterns. The EP benchmark (Fig. 3(a)) consumes a consistent amount of power for its entire execution because its workload is perfectly balanced, and each thread executes a CPU-intensive job. In the case of BT (Fig. 3(b)), however, the power load is not consistent over the test run because the workloads are more memory-intensive, and load distribution is slightly less balanced. The figure also shows power usage patterns that are repeated over time on the 5V and two 12V rails. We have found that the number of cycles in the power data exactly equals the number of process iterations executed by the benchmark. In other words, the fluctuations visible in the power consumption data correspond directly to individual subroutine iterations in the benchmark program. The accuracy and timing resolution of the PowerMon devices allows us to relate power consumption to source code at a level previously impractical.

Power profiling with PowerMon and PowerMon2 provides several types of valuable information. The CPU- and memoryintensity of different workloads can be distinguished and characterized. Researchers can observe detailed patterns of power consumption.. An ongoing research topic is whether the runtime can take advantage of this information. Potentially, with accurate measurements at a high enough sampling rate, timing information can be used in conjunction with power saving techniques, such as the Dynamic Voltage and Frequency System (DVFS), in order to balance power and performance at runtime. In this case, accurate and fine-grained monitoring may enable online power / performance decisions for a variety of dynamic workloads, allowing computer systems to reduce power consumption without significantly altering performance.

IV. Conclusion

PowerMon and PowerMon2 implement current and voltage sensing at a level of detail previously unobtainable with standard external power monitors or with special-purpose systems developed for prior research efforts. PowerMon2 senses eight independent 10A DC rails in the form factor of a 3.5” hard disk drive, and the total cost is less than $150 each when fabricated in quantities of 5 or more. The high level of integration, low cost, and measurement capability of PowerMon2 make it a natural fit for power-aware experimentation in cluster environments.

The devices have been demonstrated as useful tools for understanding workstation power usage. They have been used to correlate power supply voltage rails with individual subsystems and to profile power supply efficiency with varying workloads. The PowerMon devices have enabled researchers to determine the a software application’s impact—down to the level of individual subroutines—on CPU, memory, hard disk, or peripheral bus power consumption. The fine-grained power measurement capabilities of PowerMon and PowerMon2 provide new opportunities to study and develop new power-aware computing paradigms and applications.

References

[1] K. G. Brill, “The invisible crisis in the data center: The economic meltdown of moore’s law,” tech. rep., Uptime Institute, 2007.

[2] C. Calwell, “80 plus: A strategy for reducing the inherent environmental impacts of computers,” *IEEE International Symposium on Electronics and the Environment*, p. 151, 2005.

[3] C. Isci and M. Martonosi, “Runtime power monitoring in high-end processors: Methodology and empirical data,” in *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, IEEE Computer Society Washington, DC, USA, 2003.

[4] C.-H. Lien, “Estimation by software for the power consumption of streaming-media servers,” *IEEE transactions on instrumentation and measurement*, vol. 56, no. 5, p. 1859, 2007.

[5] “Power egg.” http://www.itwatchdogs.com.

[6] “Watts up.” http://www.wattsupmeters.com.

[7] X. Feng, R. Ge, and K. W. Cameron, “Power and energy profiling of scientific applications on distributed systems,” in *IPDPS ’05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS’05) - Papers*, (Washington, DC, USA), p. 34, IEEE Computer Society, 2005.

[8] “Atmel attiny45 datasheet.” http://www.atmel.com.

[9] “Ftdi ft232r datasheet.” http://www.ftdichip.com.

[10] “Atmel atmega168 datasheet.” http://www.atmel.com.

[11] “Analog devices adm1191 datasheet.” http://www.analog.com.

[12] “Atx specification, version 2.2.” http://www.formfactors.org.

[13] T. Harbaum, “i2c-tiny-usb.” http://www.harbaum.org.

[14] D. Bedard, M. Y. Lim, R. Fowler, and A. Porterfield, “Powermon2: Fine-grained, integrated power measurement,” technical report, Renaissance Computing Institute, 2009.

[15] J. D. McCalpin, “Stream: Sustainable memory bandwidth in high performance computers.” http://www.cs.virginia.edu/stream/.

[16] “The disk and file system performance measurement tool.” http://www.coker.com.au/bonnie++/.

[17] “The TCP/UDP bandwidth measurement tool.” http://dast.nlanr.net/Projects/Iperf/.

[18] Intel, “Intel turbo boost technology in intel core microarchitecture (nehalem) based processors,” 2008.

[19] J. Gerow, “Power supply efficiency.” http://www.motherboards.org/articles/guides/1487_7.html.