A Fine-grained Component-level Power Measurement Method

Zehan Cui[1,2], Yan Zhu[1,2], Yungang Bao[1], Mingyu Chen[1]
[1]State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences
[2]Graduate School of Chinese Academy of Sciences
Beijing, China
{cuizehan, zhuyan, baoyg, cmy}@ict.ac.cn

Abstract—**The ever growing energy consumption of computer systems have become a more and more serious problem in the past few years. Power profiling is a fundamental way for us to better understand where, when and how energy is consumed. This paper presents a direct measurement method to measure the power of main computer components with fine time granularity. To achieve this goal, only small amount of extra hardware are employed. An approach to synchronize power dissipation with program phases has also been proposed in this paper. Based on the preliminary version of our tools, we measure the power of CPU, memory and disk when running SPEC CPU2006 benchmarks, and prove that measurement with fine time granularity is essential. The phenomenon we observe from memory power may be served as a guide for memory management or architecture design towards energy efficiency.**

Keywords- power profiling; direct measurement; component level; fine time granularity

I. INTRODUCTION

With increasing performance, the power dissipation of computers has also increased a lot and becomes a more and more serious problem. For example, the average power dissipation of individual server has increased from 50 watts in 2000 to 250 watts in 2008 [24]; it is believed that energy cost will exceed the server cost if this trend does not change.

Power profiling is the basis of power management. Firstly, profiling can help us find and understand the problem of energy consumption. Secondly, the profiling results can be directly used by the operation system to schedule its workload towards energy efficiency. Then, it can help application developer to choose between alternative designs in the energy-performance trade-off space [21]. Finally, it is necessary for evaluating the effectiveness of low-power strategy.

The power profiling at component-level, for example, separating the power dissipation by different components, can be useful in helping us better understand where energy is consumed and thus reveals opportunities for power and energy saving either through software scheduling or hardware innovation. For example, if a component has not been used over time, it could be changed to low power state.

There are two approaches for component level power profiling: modeling and measuring. The modeling approach builds energy model for each component, usually based on hardware counters [13] [17] or resource utilization [21] [16]. But this approach has the accuracy issue. The measuring approach can achieve a better accuracy. Several measuring methods have been proposed, however, neither of these methods can directly measures the power of all main components, especially memory, NIC and video card [22] [18] [20] [15].

However, measuring power at component-level alone is insufficient to better understand the interaction between application performance and power dissipation. The profiling should further be of fine time granularity. Profiling with time distribution has the potential to synchronize power dissipation with program phases to identify how and when energy is consumed. Experiments show that power dissipation varies along the execution of program, and coarse-grained profiling may miss the fluctuation of power and result in imprecision. Fine-grained profiling also provides opportunities for fast scheduling.

In this paper, we propose an approach which is easy to implement for directly measuring power of main components with fine time granularity. According to the ways of power supply, different methods are used to measure power of main components: for disk, the current of ATX wires that directly supply power for disk is measured; for memory, NIC and video card, wrapper cards are employed to gather the current from multiple pins for measurement; for CPU, the dedicate ATX wires to supply power for CPU are measured to obtain CPU power. Based on the proposed scheme, our preliminary version of the profiling tool can measure the power of CPU, memory and disk every 20 microseconds using DMM (Digit MultiMeter).

This paper has made the following four contributions. First, we propose an approach to directly measure the power of main components with fine time granularity, so that distributions of each component's power can be obtained. Second, we present a method to synchronize the measured power data with application code to track the power of program phases. Then, we have experimentally proved that power fluctuation is very rapid and measurement with fine time granularity is necessary. Finally, we give some advice based on our observation of the characteristics of memory power dissipation obtained by our tool.

The remainder of this paper is organized as follows: a detailed description of our measurement methodology is presented in Section II. Then, Section III gives some results of our experiments. The related works are examined in Section IV. Finally, we give a conclusion to this paper and talk out the future work in Section V.

II. Methodology

A. Overview

There are various ways of power supply for main components in a computer machine. Basically, apart from the displays, the computer system is powered up by an ATX power supply which performs the transformation from commercial AC power to computer required DC power [2]. The motherboard is responsible for delivering the DC power to each component except for disks which directly uses the ATX power. Since different voltages are required by various components while ATX only supplies several fixed value, the motherboard has to employ additional regulation circuits to regulate the ATX voltages to that needed by different components. Unlike disk which gets its power from ATX connectors, other components get their power from connectors with motherboard: for IC (Integrated Circuit) chips like CPU, the connectors are the pads or lands on which they are mounted; for PCB (Printed Circuit Board) modules like memory, NIC and video card, the connectors are the corresponding slots in which they are plugged, such as DIMM (Dual-Inline Memory Module) slot, PCI slot or PCI-E slot.

The main components of computer are then divided into three categories according to the difference of power supply:

- from wires – disk;
- from slots – memory, NIC, video card;
- from lands – CPU.

Power of each category is measured using a different method.

Our methods employ a little amount of extra hardware to effectively perform fine-grained power measurement. However, the best way in future is to integrate the hardware into motherboard to easily monitor the power dissipation of every component in a computer.

B. Disk

The disk is either powered by the ATX peripheral power connector or the serial ATA connector. Fig. 1 shows the pin-side view of the two connectors. The ATX peripheral power connector is attached with 4 wires: two COMs, one +12VDC and one +5VDC. The ATX serial ATA connector is attached with 5 wires: two COMs, one +12VDC, one +5VDC and one +3.3VDC. The +3.3VDC wire of serial ATA connector is usually useless and removed by the manufacture of ATX power supply, which is the case of our experiment platform.

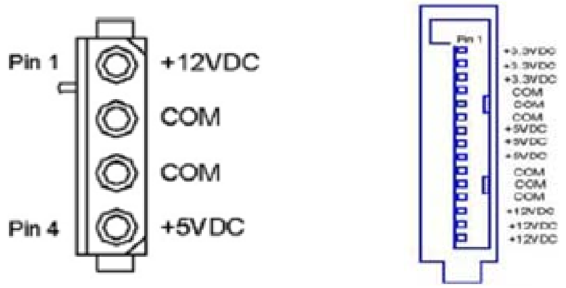

Figure 1. Peripheral power connector and serial ATA connector.

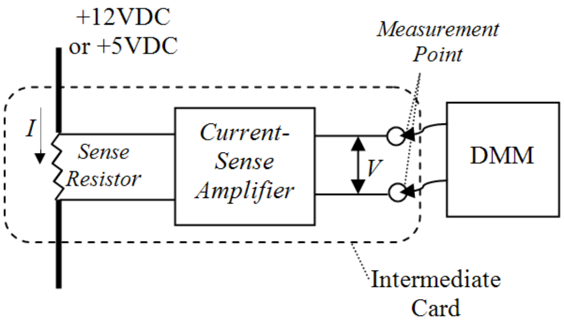

Figure 2. Implement of the current measurement module.

It’s very easy to get the power of disk by measuring the current of +12VDC and +5VDC wires. Fig. 2 shows the current measurement module. An intermediate card is made so that the probes of DMM can be easily attached to the measurement point. When the current flows through the sense resister, a small voltage drop generates; the current-sense amplifier then amplifies the voltage drop for 50 times [7] and drives it to the measurement point; the DMM attaches probes to the measurement point, measures the voltage every 20 microseconds and transfers the measured data to another computer through its LAN interface.

The current of +12VDC and +5VDC wires can be calculated using this formula:

$$I = \frac{V / Gain}{R_{sense}} = \frac{V / 50}{0.02} = V(A)\,, \qquad (1)$$

where I is the current of the wires in ampere, V is the voltage of the measurement point in volt, $Gain$ is the amplification factor of current-sense amplifier which is 50, and R_{sense} is the resistance of the sense resistor which is 0.02ohm. The power of disk, P_{disk}, can be calculated as follows:

$$P_{disk} = 12V * I_{12V} + 5V * I_{5V}\,, \qquad (2)$$

where I_{12V} and I_{5V} are the current of +12VDC and +5VDC wires. We directly use 12V and 5V to multiply the current instead of measuring the corresponding voltage at the same time. Experiment has been done and proved that the diversity of voltages across different workloads is less than 2%, so the direct use of 12V and 5V is reasonable for the accuracy. The energy consumption of disk over a period of time, E_{disk}, can then be calculated using this:

$$E_{disk} = \sum P_{disk} * T\,, \qquad (3)$$

where T is the interval between two samples of DMM.

C. Memory, NIC, video card

These components are plugged into specific slots on the motherboard, which take responsibility for both signal transmission and power supply. Power is supplied from the motherboard to the components through multiple pins of the slots. Some slots supply only one voltage like DIMM slot, while some supply two voltages like PCI and PCI-E slots.

To the best of our knowledge, there have been no ways to directly measure the current flowing through the slots. We propose a method to gather the flowing current of multi pins across the slots, make it measurable, and not influence the signal transmission. In the following description, we will take memory for example. A wrapper card is made to assist measurement, which is transparent to the computer system: it directly transmits the signals from motherboard to memory, and gives special treatment for power supply to measure the current. The wrapper card we use is slot-based and can easily be implemented to a new platform, since the slots are usually standardized.

Fig. 3 shows how the wrapper card works. The wrapper card itself has a DIMM slot, into which memory is plugged. The wrapper card together with memory is then plugged into the DIMM slot of motherboard. Firstly, we find out the pin definition of DIMM slot from specification [5], according to which we divide the pins into power pins and signal pins. Secondly, on the motherboard side of the wrapper card, the power pins all connect to a copper foil which gathers the current of multi power pins together, while the signal pins directly connect to corresponding pins of memory. All power pins are of the same voltage - 1.8VDC for DDR2 and 1.5VDC for DDR3, so we can connect them together. A copper foil is a large area of copper in PCB, which is widely used for power supply in PCB design. Thirdly, after the current is gathered in the copper foil, it is driven out the wrapper card through copper wire to the intermediate card described in Fig. 2, where the current is measured. Finally, the current flows back to the wrapper card into another copper foil, where it is driven to the multi power pins of memory.

Fig. 4 is a photo of the wrapper card for memory. The actually needed hardware for power measurement, which is that in the red frame, is much smaller than that shown in the photo, since large area of the PCB is for other purpose - HMTT (Hybrid Memory Trace Tracker) [12].

The wrapper card we add will not influence the normal operation of memory. The signals can be directly transmitted between memory and motherboard. Though the power supply flows through some detour, the resistance of copper foils and wires are nearly zero, and the only resistance worth concerning is that of the sense resistor. According to [6], the maximum current of memory is 1.12A; this will introduce 0.0224V voltage drop across a 0.02ohm sense resistor, which still meets the +/-0.1V requirement of power supply.

The power of memory, P_{memory}, can be calculated as this:

$$P_{memory} = V_{memory} * I_{memory}, \qquad (4)$$

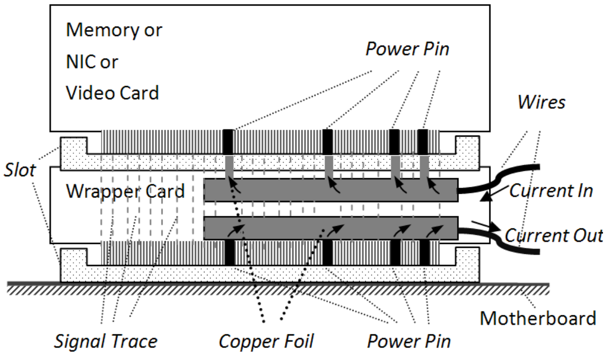

Figure 3. Framework of the wrapper card.

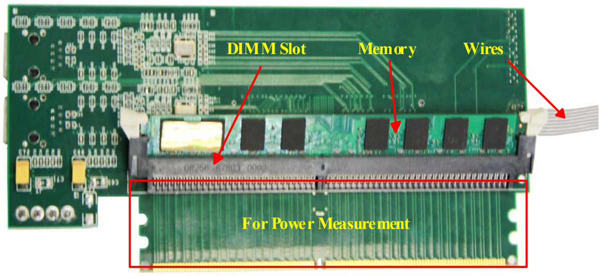

Figure 4. Photo of the wrapper card for memory

where V_{memory} is the operating voltage of memory - 1.8V for DDR2 and 1.5V for DDR3, and I_{memory} is the current of memory calculated using (1).

The method to measure the power of NIC and video card is similar to that of memory. The difference is that, there is one supply voltage for memory, but are two voltages for NIC and video card, which usually use PCI or PCI-E slots. PCI slots use +5VDC and +3.3VDC as their main power supply [9], while PCI-E slots use +12VDC and +3.3VDC [8]. Since there are two voltages, another copy of the two copper foils and wires should be added into the wrapper card to measure both the current simultaneously. The power of NIC and video card can then be calculated using the following formulas according to their slot types:

$$P_{PCI} = 5V * I_{5V} + 3.3V * I_{3.3V}, \qquad (5)$$

$$P_{PCI-E} = 12V * I_{12V} + 3.3V * I_{3.3V}, \qquad (6)$$

where P_{PCI} is the power of NIC or video card using PCI slot, $P_{PCI\text{-}E}$ is the power of NIC or video card using PCI-E slot, and I_{12V}, I_{5V}, $I_{3.3V}$ are the current of corresponding voltages calculated using (1).

The energy consumption of memory, NIC and video card over time can be calculated using (3).

D. CPU

The CPU is usually mounted on the motherboard and gets its power supply through multi lands. The dynamic voltage that CPU required is regulated from the ATX wires by regulation circuits [11]. It is difficult to directly measure the output current of regulation circuits, which are integrated in the motherboard. It is also impractical to make a wrapper card for CPU as that for memory, for both electrical and mechanical reasons. An approach that might work is to find out which ATX wires are used by the voltage regulator of CPU. However, the wires that supply power for CPU may also supply power for other components, if so, isolating the CPU power may be difficult and this approach will turn out to be infeasible.

Fortunately, there does exist a dedicated +12VDC power connector which only supplies power for CPU according to these specifications [11] [2]. Ge et al. [20] and Chen et al. [15] have also experimentally proved this. The power of CPU, P_{CPU}, can be calculated by directly measuring the current of this connector using method illustrated in Fig. 2:

$$P_{CPU} = 12V * I_{12V}, \tag{7}$$

where I_{12V} is the current of the +12VDC power connector calculated using (1). The energy consumption of CPU over time can be calculated using (3).

It is worth mentioning that, unlike disk and memory, the measured power for CPU includes the power dissipation of voltage regulator.

E. *Synchronization*

It will be more helpful to understand when, where and how the energy is consumed if the collected power data can be synchronized with the application code.

To further synchronize the collected data with concerned code segments, a time-based method is introduced. Once the system time of the beginning of measurement and every concerned point are recorded during execution of application, it will be easy to find the related power data since the data itself contain time information. Instrumentation [14] can be employed to do such recording work.

III. Experiments

A. *Experimental Setup*

The experimental setup consists of the experiment platform, the measuring instrument and the related software.

We use a PC running Linux as our experiment platform. It has one Intel Core2 Duo E4500 processor running at 2.20 GHz with 64-byte cache line, one 2GB memory operating at DDR2-400, one 640GB SATA disk, one Gigabit NIC and one video card. In our preliminary version of power profiling tool, the additional hardware we have made to assist power measurement consist of one intermediate card with 8 units illustrated in Fig. 2 and one wrapper card for DDR2 memory shown in Fig. 3 and Fig. 4. It is worth mentioning that, the height of the wrapper card limits that only one can be plugged into the motherboard though there are another three vacant DIMM slots, so the system memory is limited to 2GB. This problem will be fixed for our newly designed wrapper card with the extra hardware for HMTT removed, resulting in a much smaller package.

Currently, two Agilent 34411A DMMs [1] are used as our main measuring instruments. The DMM can offer voltage measurement at the speed of 50,000 samples per second. The DMM also has LAN interfaces, through which it can be controlled remotely and transfer measured data out. Our test has validated that the LAN interface can work well with no data lost at the highest measuring rate. Each DMM only has one channel, so the number of measurement points that can be measured simultaneously is limited to two. Basically, CPU and memory need one measurement point each, while disk, NIC and video card need two measurement points each. This means that when measuring the power of disk, there is no channel to measure the power of memory or CPU. This limitation will also be derestricted in our newly designed measurement system by using ADC and FPGA as the main measuring instruments.

The software contains the program to fetch data from DMMs and several benchmarks. Prime95 [3] and STREAM [10] are used to separately stress CPU and memory, through which the maximum power they dissipate can be measured and compared with datasheets. SPEC CPU2006 benchmarks and micro-benchmarks written by ourselves are also used to perform measurement.

B. *Validity*

1) *Disk*

The disk is directly powered through ATX wires; what we measure is just the current flowing through those wires. So the validity is obvious.

2) *Memory*

STREAM benchmark which can maximize performance of memory is used to make the memory dissipate maximum power. The maximum power we measured is 2.029W, while the number from datasheet is 2.016W [6].

3) *CPU*

Prime95 has a feature called "Torture Test" that allows maximum stress testing on the CPU, which will make the CPU dissipate the most power. The maximum power we measured is 64.272W, which is very close to the CPU's thermal design power – 65W [4].

C. *Result of SPEC CPU2006*

Since the available DMM channels are limited to two as described above, the power of CPU, memory and disk are

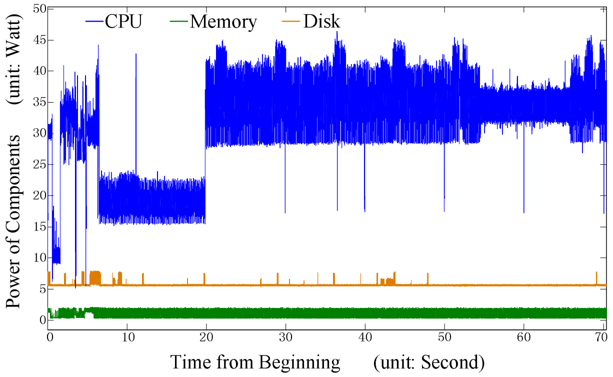

Figure 5. Power of different components when running 401.bzip2.

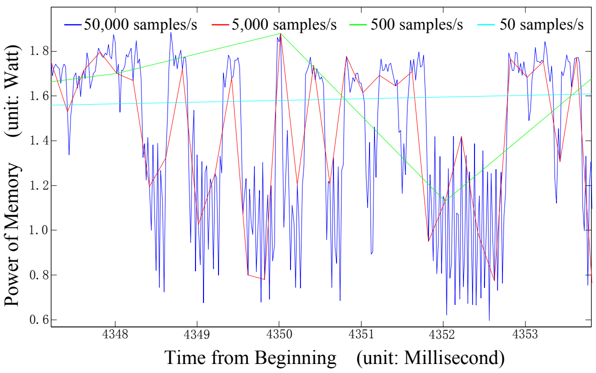

Figure 6. Measurement with different speeds.

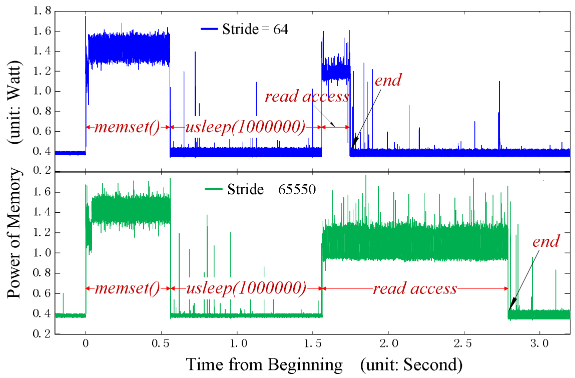

Figure 7. Power of memory with different reading strides.

obtained by repeatedly running the program and separately measuring the power of one component each time. Fig. 5 illustrates the power of main components for the initial 70 seconds when running 401.bzip2 from SPEC CPU2006 benchmarks with reference input set. The power of CPU is dominated and fluctuates wildly; while the power of disk stays stable, which is seldom accessed by the benchmark actually. Fig. 5 shows that the power of 2GB memory can be neglected when comparing with CPU. However, in today's server, one CPU may connect with more than a hundred GB of memory, and the power of memory will certainly be a big problem [24].

D. *Time granularity*

More frequently we measure the power, more details we can get. Fig. 6 illustrates a fraction of memory's power when running 429.mcf of SPEC CPU2006 at different measuring speeds. With 50,000 samples per second, the climb up and down of memory's power appears clearly; while at a hundred times slower speed, the result doesn't show any details. Considering the computing speed of up to GHz, millions of instructions and memory operations have been executed in a millisecond, so we believe power measurement of fine time granularity is necessary.

E. *A case study : Memory access pattern vs. Power*

A special study to memory's power has been performed by running a simple micro-benchmark: 512MB of memory are allocated using *malloc()* and all bytes are first set to zero using *memset()*, then the memory are read continuously and iteratively for 8 million times with different strides. The hardware-prefetch function of CPU is turned off and strides greater than 64 bytes are used to make every accessed data not cached. Each read operation returns a cache line of 64 bytes and the total amount of data read from memory to CPU is 512MB.

Fig. 7 illustrates the memory's power of two different access patterns: the upper part is for the stride of 64 bytes and the lower part is for the stride of 65550 bytes. The first peak periods of the both curves represent *memset()*, while the second peak periods correspond to the read access. The read access at the stride of 65550 bytes has much lower bandwidth, which takes 6.5 times longer to access the same amount of data, but the power is slightly lower; this leads to the total energy consumption is 5.9 times compared to 64 bytes stride. This is mainly because read at the stride of 65550 bytes is more like a random access from the view of memory - the row buffers of which rarely hit and every read should first active the entire row, read 64 bytes out and then precharge the row back. On the contrary, read at the stride of 64 bytes will substantially hit in the row buffer and save a lot of active and precharge operations, resulting in faster execution and lower energy consumption.

It is worth noting that though *memset()* also access 512MB data by writing zero to them all, when compared to read access with 64 bytes stride, the execution time is 3 times longer, power is slightly higher and energy consumption is 3.6 times more. This is mainly caused by two reasons. Firstly, *malloc()* doesn't actually allocate the physical memory. When *memset()* accesses the data for the first time, a lot of page faults happen and physical memory are allocated at this time. The additional system call routines increase the total number of memory access and lengthen the execution time. Secondly, one CPU write operation actually consists of three strides – read data from memory to CPU, modify the data and write it back to memory. There are two memory accesses which will certainly increase the execution time and energy consumption.

Memory management towards energy efficiency may be advised to carefully organize data allocation in main memory and adjust the access pattern to save energy consumption while maintain performance. From the view of architecture, how to reduce the number of memory accesses which are caused by write operations of CPU may worth working on.

IV. RELATED WORK

To profile power dissipation of all components, an approach is to build models for each component and calculate the power based on information such as resource utilization, hardware counters and performance state [13] [17] [21] [16]. The accuracy of modeling is the key for the profiling. However, experiment shows that CPU dissipates different power even at the same utilization [15], which means CPU utilization is not a good indicator of power. Besides the modeling approach, hardware measuring method has also been proposed. In [22], components are divided into directly measurable ones and indirectly measurable ones.

Power of indirectly measurable components is calculated by subtracting the system power when the components are in different power state. In [18] [20] [15], similar methods are used to measure the current of all ATX wires and figure out which wires supply power for separate component. The correspondences they finally get are different, which may due to the different voltage regulation strategy used in different motherboard. To use this method, the deduction of the correspondence has to be well verified; otherwise the correctness of measurement cannot be guaranteed. Additionally, in [22] [15], only power of particular states are measured to get a static breakdown of components, whereas power fluctuation over time is ignored. In [23], power measurement hardware is integrated in an embedded platform to enable monitoring the energy used by each hardware resource; whereas, it is impractical for us to do this in computer system, except for the manufacturer of motherboard. PowerScope [19] maps energy consumption to program structure, in much the same way that CPU profilers map processor cycles to specific processes and procedures.

V. Conclusion and Future Work

In this paper, an approach to directly measure power of main computer components with fine time granularity is proposed. The measurement method, the assistant hardware and ways to synchronize the power data with code segments are described in detail. Experiments have been done to validate our approach. The power of CPU, memory and disk when running SPEC CPU2006 benchmarks are measured and demonstrate that high speed measuring is essential. Besides, the phenomenon we observed from the power of memory may be helpful for memory management and architecture design towards energy efficiency.

In the future, we will first eliminate the limits of our preliminary version, such as memory capacity and number of simultaneously measurable channels. Then the measurement system will be implemented on mainstream servers to perform power measurement under real workloads. Power management related researches will also be carried on.

References

[1] Agilent 34410A and 34411A Multimeters - Product Overview, 2007

[2] ATX12V Power Supply Design Guide version 2.2, 2005

[3] Great Internet Mersenne Prime Search - GIMPS, http://www.mersenne.org/.

[4] Intel Core2 Extreme Processor X6800 and Intel Core2 Duo Desktop Processor E6000 and E4000 Sequences Datasheet, 2007

[5] JEDEC Standard No.21C, 4.20.13 - 240-Pin PC2-5300/PC2-6400 DDR2 SDRAM Unbuffered DIMM Design Specification Revision 2.0, 2006

[6] KVR800D2N6/2G Memory Module Specification, 2007

[7] MAXIM Single/Dual/Quad, High-Side Current-Sense Amplifiers with Internal Gain - MAX4376/MAX4377/MAX4378, 2011

[8] PCI Express Card Electromechanical Specification Revision 2.0, 2007

[9] PCI Local Bus Specification Revision 3.0, 2004

[10] STREAM: Sustainable Memory Bandwidth in High Performance Computers, http://www.cs.virginia.edu/stream/.

[11] Voltage Regulator-Down (VRD) 11.0 Processor Power Delivery Design Guidelines For Desktop LGA775 Socket 2006

[12] Y. Bao, M. Chen, Y. Ruan, L. Liu, J. Fan, Q. Yuan, B. Song and J. Xu, "HMTT: a platform independent full-system memory trace monitoring system", In Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pp.229-240, 2008

[13] F. Bellosa, "The benefits of event: driven energy accounting in power-sensitive systems", In 9th ACM SIGOPS European Workshop, pp.37-42, 2000

[14] B. Buck and J. K. Hollingsworth, "An API for runtime code patching." International Journal of High Performance Computing Applications 14(4): pp.317, 2000

[15] H. Chen, S. Wang and W. Shi, "Where Does the Power Go in a Computer System: Experimental Analysis and Implications." MIST-TR-2010-004, Wayne State University, Detroit, MI, 2010

[16] T. Do, S. Rawshdeh and W. Shi, "pTop: A Process-level Power Profiling Tool", In Proceedings of the 2nd Workshop on Power Aware Computing and Systems (HotPower'09), 2009

[17] D. Economou, S. Rivoire, C. Kozyrakis and P. Ranganathan, "Full-system power analysis and modeling for server environments", In Workshop on Modeling, Benchmarking, and Simulation, 2006

[18] X. Feng, R. Ge and K. W. Cameron, "Power and energy profiling of scientific applications on distributed systems." Proc. 19th IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS), 2005

[19] J. Flinn and M. Satyanarayanan, "PowerScope: A tool for profiling the energy usage of mobile applications", In Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, 1999

[20] R. Ge, X. Feng, S. Song, H. C. Chang, D. Li and K. W. Cameron, "PowerPack: Energy profiling and analysis of high-performance systems and applications." IEEE Transactions on Parallel and Distributed Systems: pp.658-671, 2009

[21] A. Kansal and F. Zhao, "Fine-grained energy profiling for power-aware application design." ACM SIGMETRICS Performance Evaluation Review 36(2): pp.26-31, 2008

[22] A. Mahesri and V. Vardhan, "Power consumption breakdown on a modern laptop", In Proc. of the 4th Power-Aware Computer Systems, pp.165-180, 2005

[23] D. McIntire, T. Stathopoulos and W. Kaiser, "etop: sensor network application energy profiling on the leap2 platform", In IPSN, pp.576-577, 2007

[24] D. Minas and B. Ellison, "The Problem of Power Consumption in Servers." Hillsboro, Intel Press, 2009