Sensing Power Consumption of Desktop Computer System Components

Benjamin Schaff
Undergraduate, Electrical and Computer Engineering
University of Maine
Orono, ME
benjamin.schaff@maine.edu

Vincent Weaver
Assistant Professor, Electrical and Computer Engineering
University of Maine
Orono, ME
vincent.weaver@maine.edu

***Abstract*—On modern computing systems power and energy consumption are increasingly important metrics. This is true both in the high end (where supercomputers can draw as much power as small cities) as well as the low end (where cellphone battery life is directly related to CPU power consumption). Despite being important, it is often frustratingly complex to get detailed power readings on modern electronics. Most common computer systems do not include means of making detailed power measurements. Overall system power usage can be easily measured at the wall or power supply, but accurately measuring the detailed power consumption of system components often requires extensive custom instrumentation. We used sensors for conducting detailed power measurement on x86 desktop machines. We investigated USB power measurement using an Adafruit USB Power Gauge and found it useful for characterizing average consumption of common USB devices despite being unsuitable for detailed or high frequency measurement. We intercepted the power supply on an older AMD Phenom system as well as an Intel Haswell-i5 machine and found that we could use benchmarks to correlate power usage patterns to the individual ATX and SATA power lines on both systems. These detailed measurements can be used in the future to provide advanced power-based feedback when optimizing code.**

I. INTRODUCTION

Collecting detailed power measurements about the individual components of a computer system is a task that is both increasingly important and difficult to perform. The overall power consumption of a computer is of interest to users and designers looking to reduce operating cost or improve system performance. In the effort to improve a system's energy efficiency, having access to detailed information about the power usage of individual components would be incredibly valuable. This information would enable detecting the areas that consume the most power as well as allowing for the possibility of code optimization using feedback about power usage.

Despite its importance, the collection of these kinds of detailed measurements on common computer systems remains a difficult task. Most common desktop computers lack the means of measuring and reporting power usage information. Overall system power can be measured at the wall or power cord, but gives little information about specific parts and so is of limited use for optimization efforts. Getting actual detailed power measurements about individual parts usually requires extensive modifications to wiring, much of which is not easily accessible.

The focus of this research was on evaluating and improving methods of sensing the power usage of computer system components. Power measurements were performed on x86 desktop machines and included measurements of common USB devices as well as Advanced Technology eXtended (ATX) and Serial ATA (SATA) power lines while idle and in use. The USB measurements were acquired using an Adafruit USB Power Gauge [4] to log the power of common devices in various states of use. The accuracy and capabilities of the Power Gauge needs to be evaluated before it can be used for measurements for research purposes.

The objective of the ATX and SATA power measurements was to correlate individual power supply lines to specific hardware components in the computer. Knowing which lines power which components is a useful first step for future work analyzing power usage of specific parts of the system.

II.RELATED WORK

Some work has been done by others into investigating the power usage of computer hardware. O'Brien et al. [1] investigated the power consumption and performance characteristics of various USB flash drives and found power usage for reading and writing were not significantly greater than while idle. Mahesri and Vardhan [3] measured the power consumption of a laptop computer on a component level and explored different factors that affect power usage.

Application of power feedback towards real-time optimizations has also been investigated. Lent [2] used a sensor network to measure the power consumption of individual parts of a computer network, and then looked at an algorithm for optimized routing using power sensor feedback.

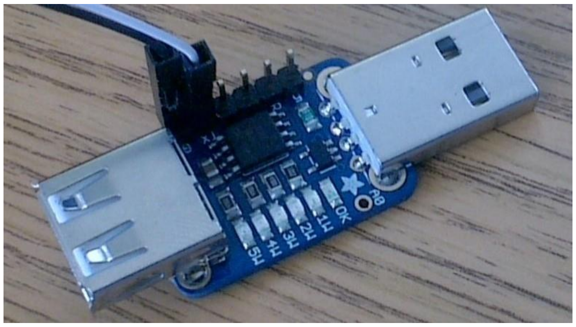

Fig. 1. The Adafruit USB Power Gauge.

III.EXPRIMENTAL SETUP

The body of this research is divided into two sections. The first is comprised of the work done categorizing the power consumption of common USB devices and evaluating the usefulness of the Adafruit USB Power Gauge. The second encompasses the measurements performed on the ATX and SATA power lines powering the hard drive disk (HDD) and components on the motherboard.

A. USB Measurements

Power measurements were performed on common USB devices, while in various states of use, using the Adafruit USB Power Gauge. The gauge itself (Figure 1) is a simple device that sits between the USB port and the device being measured. It uses a 0.1 ohm sense resistor in series with the positive power line with an INA169 current shunt monitor to measure the current going to the test device, and an ATtiny85 microcontroller to process the current and voltage data and output it over TTL serial at 9600 baud. A USB-to-serial converter cable was connected to the device and the recorded data was logged using minicom on the host machine.

To verify the results output by the USB Power Gauge, a BK Precision 2831E Bench Multimeter was used to simultaneously record the voltage output at the A0 pin, which is connected directly to the output of the INA169 and gives an output voltage of 1 volt per amp of current going through the device. Measurements from the multimeter were logged using the USB interface and the 2831E Multimeter Software.

Power information was collected for four common kinds of USB devices. The devices included a standard HP Keyboard (model QY776AA), an HP Optical Scroll Mouse (model DC172B), a SanDisk 2GB Cruzer Micro Flash Drive, and a Netgear WNA1000M wireless network adapter.

Approximately 60 seconds of data were recorded for each device while being tested in various states of use with the exception of a few tests, such a the flash drive file transfer, whose duration was determined by other factors. A measurement was made for each while idle, that is, simply connected but not being used, and under what would be considered regular use. For the keyboard and mouse this meant typing and moving the cursor around for a minute. The flash drive was tested by writing a singular 1.5GB file to it, and for the wireless adapter an internet speed test [6] was run. Additionally, the keyboard was tested in an intermediate state where not in use but with the LEDs for capslock and numlock activate, and the wireless adapter was compared connected/disconnected to the network in addition to in use or not.

Both the multimeter and power gauge record at approximately 40 samples per minute, although this is not a fixed rate and neither device’s outputs include useful information about when each sample was taken.

B. ATX and SATA Measurements

The ATX and SATA power measurements were done with the goal of matching power usage in certain system components with individual wires. This is done by inserting a current sensor into the various power lines going from the power supply to the 24-pin connector on the motherboard or the HDD, and then running various software benchmarks designed to stress specific system components. It is then possible to isolate power patterns to find correlation. For example, if a certain power line sees a sudden increase in current during a benchmark that stresses the CPU, then it is very likely that line is feeding at least the CPU socket.

Seven different benchmarks were used to stress hardware. The benchmarks used, and the specific hardware each was targeted towards, were:

- Bzip2 to stress the CPU,
- Equake.spec2k [7] for CPU floating point,
- Iozone [8] for hard disk,
- Matrix_multiply for CPU,
- stream [9] for memory,
- GpuTest [10] for the GPU,
- and scp to send a 1GB file to stress the network.

Each benchmark is intended to put strain primarily on a single part of the system to make the task of isolating power patterns simpler. This was not always the case, however, as certain tasks inevitably require not-insignificant action from other components, such as the network test (scp) requiring first reading the file from the hard disk before it could be sent to the destination machine.

To simplify the process of placing a current sensor, and measuring the current, on each of the power lines, custom made interceptor boards were used. The ATX and SATA boards, shown in Figure 2 and Figure 3, were made to intercept the power lines going to the motherboard and hard drive respectively, and break out individual lines to make it easier to place a current sensor in series with each.

Measurements were performed on two different x86 desktop computers. One was an older machine with an AMD Phenom II CPU, 2GB of DDR3 memory and a GeForce 8400 GPU. The other machine was a more recent system, a Lenovo Think Center with an Intel i5-4570s Haswell CPU, 4GB of DDR3 memory, and an integrated HD Graphics 4600. Both machines had the same model of 500GB hard disk and were running Debian Jessie 8.3. Initial measurements were done on the older machine in case something went wrong with the measurement setup and the system was damaged during testing.

The Haswell machine used a different, non-ATX, motherboard and power supply that used a fewer power lines and, unfortunately, did not connect directly to the ATX interceptor board being used. To remedy this issue, a regular ATX power supply was installed and an adapter connected the interceptor to the motherboard. This did not fully solve the problem however, as only a few of the 24 lines were actually in use, complicating the task of isolating power patterns. Additionally, the hard drive was powered through a connector on the motherboard instead of directly from the power supply, further complicating the problem.

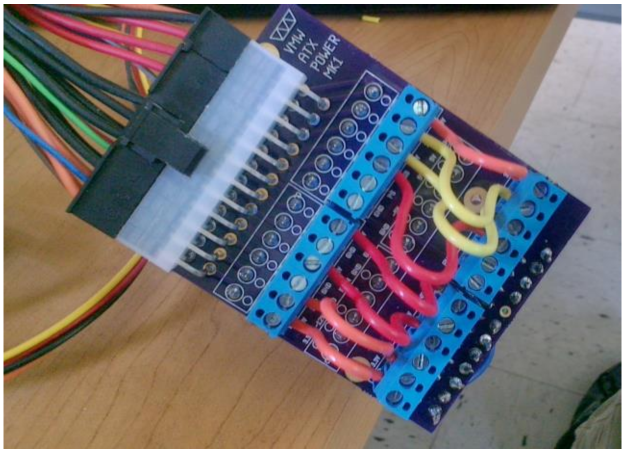

Fig. 2. The ATX inteceptor.

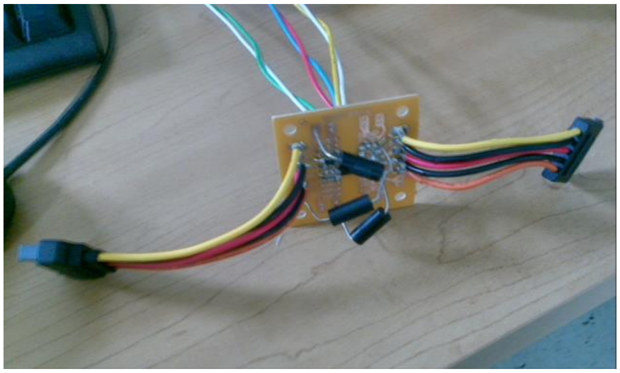

Fig. 3. The SATA interceptor.

The majority of the power measurements were acquired using an INA219 high side current sensor [11] to measure the current on each line broken out by the interceptor boards. The INA219 is capable of measuring up to 3.2A of current with 0.8mA resolution, and data from it can be read using I^2C. The first set of ATX measurements were made using an ACS715 hall effect sensor and a separate ADS1015 ADC module, but the switch to the singular INA219 module was made after the ADC unit was destroyed by an accidental short circuit and a replacement was not immediately available. In past research using the SATA interceptor, sense resistors (visible in Figure 3) soldered in series with the power lines were used for determining the current, but were removed for this work so the same current sensor module could be used for both the ATX and SATA measurements.

A Raspberry Pi was used for logging the measurements generated by the INA219. Values were read from the sensor via I^2C at approximately 1.5kS/s using a python script running on the Pi. The 1.5kS/s limit was imposed by the maximum bitrate of the I^2C protocol without making any changes to the configuration of the Pi. Another python script, running on the test machine, automated the task of running the benchmarks as well as remotely triggering the recording script on the Pi to synchronize it with the benchmarks. 1 second of measurements was recorded before and after each benchmark run to provide an idle reference for comparison, as well as to ensure that any effects of the script itself running on the machine were separated from the actual benchmark results.

IV. RESULTS

A. USB Results

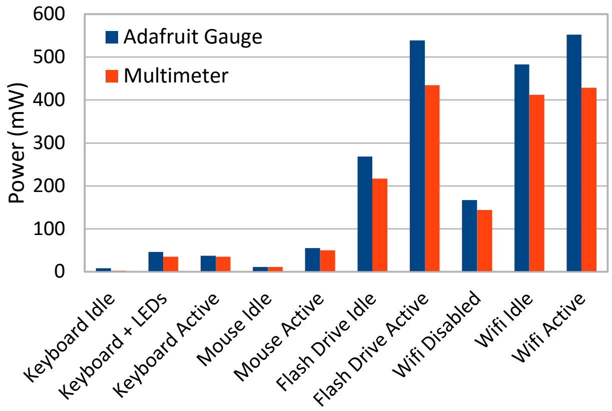

Fig 4. Average USB device power usage.

TABLE I. USB DEVICE POWER MEASUREMENTS

Device Tested	**Time (s)**	**Power Gauge**		**Multimeter**	
		V (V)	***P (mW)***	***V (V)***	***P (mW)***
Keyboard - Idle	60	5.80	8.03	5.0	2.16
Keyboard – Idle with LEDs	60	5.80	46.39	5.0	34.82
Keyboard - Active, LEDs	60	5.80	37.03	5.0	35.14
Mouse - Idle	60	5.76	10.78	5.0	11.36
Mouse - Active	60	5.75	54.86	5.0	49.91
Flash drive - Idle	90	5.95	268.13	5.0	214.21
Flash drive - Active	150	5.95	539.42	5.0	434.56
Wifi Adapter - Disconnected	60	5.93	166.73	5.0	143.88
Wifi Adapter - Connected + Idle	60	5.92	483.43	5.0	412.13
Wifi Adapter – Connected + Active	50	5.94	552.49	5.0	428.58

Current and voltage readings from both the power gauge and multimeter are used determine the power used by each USB device during the tests. Table 1 and Figure 4 show the average power values reported by both measurement devices for each test.

The results show some unexpected power usage patterns for the tested devices. There is very little change in power consumption for the keyboard between idle with LEDs on and active with LEDs on; over 95% of the power going to the keyboard was just for two small indicator lights; actual use has little effect. Similarly, the largest change in the average power use for the wireless network adapter is just for connecting to a network, sending or receiving data under normal conditions does not cause any significantly higher current draw. Also of interest is that the flash drive used nearly a quarter watt of power to do effectively nothing. Whether this is due to the large indicator LED or another cause is currently unknown.

Overall, the power gauge reported values show similar behavior to those of the multimeter. However, they are consistently higher, anywhere from 5% to 30% more, with the exception of the idle keyboard measurements, which is nearly four times larger (likely a result of the values being less than the minimum resolution of the ADC on the ATtiny85). The cause of the overall offset is almost certainly an issue with the internal calibration on the gauge, as the reported line voltages are anywhere from 0.75V to 0.95V higher than the actual 5.0V of the power line.

The variance of the power gauge readings is much larger than the multimeter, as shown in Figure 5. This may be a result of the power gauge being able to pick up the high frequency digital activity where the multimeter's larger nature would filter it out, or it may just be noise inside the gauge or on the power line. It is nearly impossible to tell which it could be due to the lack of detail from the low sample rate. The effect of this wide variation is that the measurements taken using the power gauge only have useful accuracy when sufficient data points have been collected that an average value can be used.

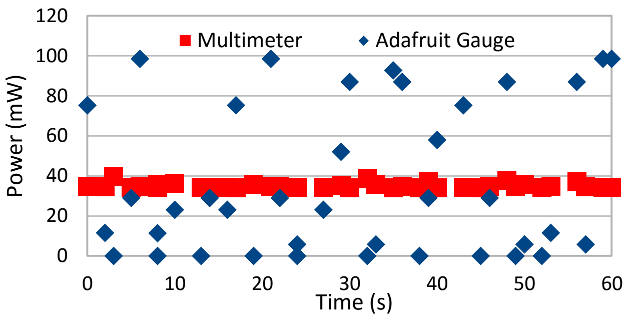

Fig. 5. Keyboard active power usage over time.

B. *ATX and SATA Results*

Current measurements gathered for from power lines on each machine are used to generate plots of the average power usage during benchmarks and when idle. Figure 6 and Figure 8 show the idle power provided by each line of the Phenom and Haswell machines respectively. Figure 7 and Figure 9 show by how much the power on each line increased during individual benchmark runs. A value of zero means there was no change from the idle readings and a negative value means a decrease from normal usage for that line during the benchmark run. The results for the Haswell machine have fewer power lines because only a small number of lines actually connected to the motherboard.

The results show some surprisingly distinct patterns caused by the benchmarks, especially for the Phenom machine. Figure 7 shows large power spikes during the GpuTest benchmark that are exclusively on the 12V lines, strongly suggesting majority of the power to the PCIe slot is provided by those lines. Power on the 5V lines shows significant increases when the CPU and main memory are stressed. Smaller, less significant, increases on the 5V lines occur during the scp test of the network, although this could be a result of CPU or RAM utilized in sending the file

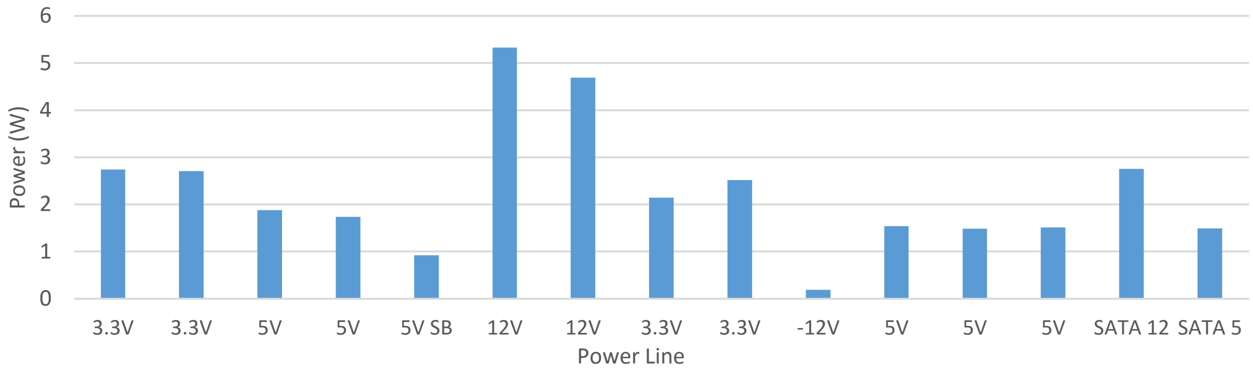

Fig. 6. Idle power usgae of Phenom machine.

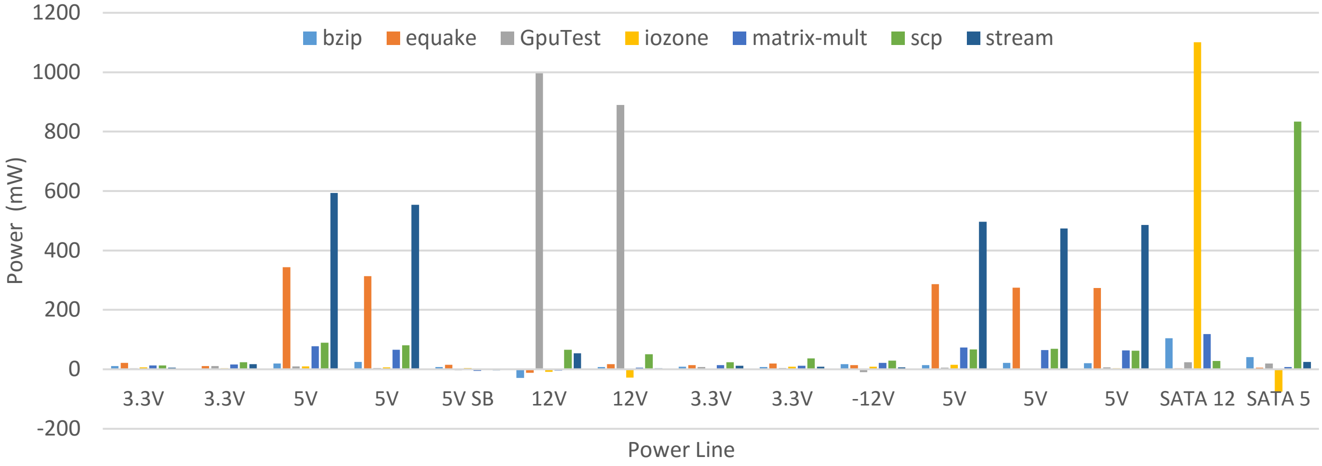

Fig. 7. Power increase from idle of Phenom machine.

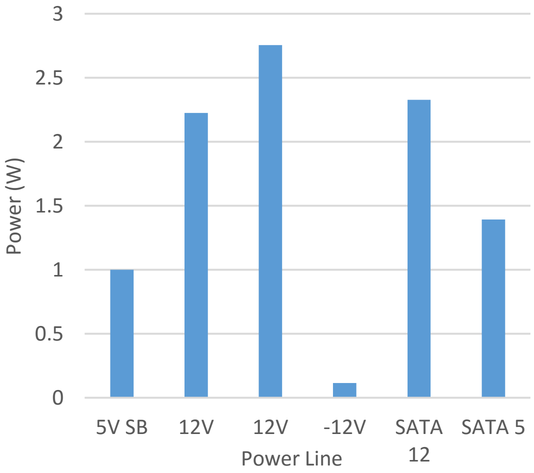

Fig. 8. Idle power of Haswell machine.

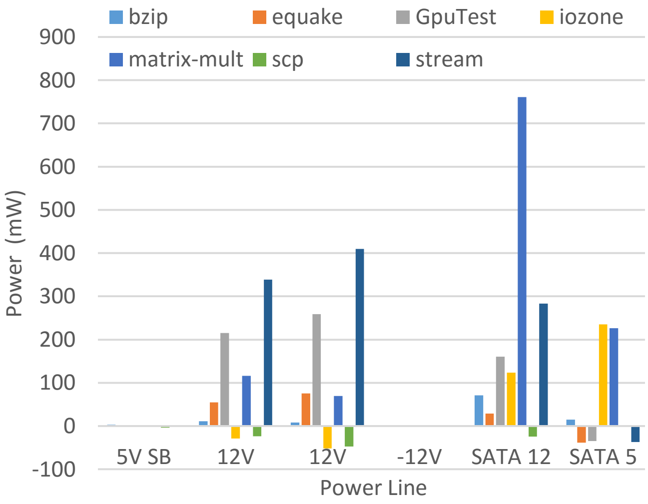

Fig. 9. Power increase from idle of Haswell machine.

and not any significant link to the networking hardware itself. The high degree of similarity between the results of all the 5V lines suggests that they are likely all directly connected.

The 3.3V, -12V and 5V standby (5V SB) lines show almost zero change during any of the benchmarks. This suggests that these lines do not provide any significant power to components stressed by any of these benchmarks. That does not, however, mean they do not provide significant power to other parts of the system, just that those parts are in constant use while the system is running. The 3.3V lines provide about a third of the total idle power to the Phenom machine, and the 5V SB lines for both machines provide about a watt each. The -12V line does not provide much power to either machine, although it is enough to be a distinctly non-zero value and not just noise in the sensors.

Some very unexpected results came from the SATA power measurements. The large increase to the SATA 12V line of the Phenom machine caused by Iozone was expected, as that benchmark is intended to stress the HDD. Such a large spike on the SATA 5V caused by scp, however, was not anticipated, even with the inevitable disk activity needed to send a file. Running the Iozone benchmark had another unexpected result: certain periodic disk activity performed by the system while idle was absent while running the benchmark and actually caused a decrease in power on the 5V SATA line.

It is also particularly strange that there is such dramatic difference between the SATA results of the two machines. Both systems have the same model of HDD and are running the same software, so it is unexpected to see such vastly different results.

The distinct power patterns are much more visible for the old Phenom system than the Haswell one. The primary cause of this is the reduced number of power lines utilized by the non-ATX motherboard. With only a few 12V lines providing nearly all the power to the system instead of the full 24-pin ATX connector, power usage from multiple components inside the system becomes consolidated. Patterns that are easily visible on the Phenom system are superimposed upon one another on the Haswell machine, and far more difficult to separate, making the task of correlating individual power lines to specific hardware components a challenge.

V. CONCLIUSION AND FUTURE WORK

This research was done with the goal of serving as groundwork for future power measurement efforts and analyzing the usefulness of measurement methods.

The results from the Adafruit USB Power Gauge show its potential uses in further research. While not very accurate in high frequency applications, and lacking the resolution to make fine measurements at very low currents, its results do follow actual power patterns over the long run, and could be used in applications such as monitoring the long term energy usage of USB devices for use in energy-sensitive or battery operated systems, or systems such as the Raspberry Pi that are powered entirely via USB.

Because of its open source nature, making modifications to the power gauge is also easily possible, both to its programming and hardware. Changes could be made that might improve the resolution of measurements, or the rate samples are gathered and reported. Some quick changes to the code could be to disable onboard LED display (unused for data logging), remove superfluous characters from the serial printout or even send the raw ADC values without performing lengthy conversion calculations, all functions that slow down the microcontroller and consequently the sample rate. Also, by using an external analog reference voltage instead of the internal AREF, much higher resolutions could be obtained, at the cost of a smaller full scale range and lower maximum value.

The ATX and SATA measurements show that there are in fact noticeable correlations between certain components and power lines, which will be useful for future work looking to gauge the power usage of those specific components. The only power lines that showed any changes in activity during benchmarks were the 5V and 12V on the older system and just the 12V on the other, non-ATX, system, so similar studies should focus on those. The 3.3V line of the SATA power cable was also unused by the hard drive during testing.

This work only covers measurements of the 24-pin ATX connector and SATA power lines, but a similar setup could be used for measuring other parts of the system. Parts like PCIe cards and DRAM (with appropriate extender/risers to intercept power lines) or the ATX P4 connector that provides auxiliary 12V power to the CPU.

It would also be useful to get ATX measurements of a different modern system that uses a standard ATX power supply and motherboard, so results are more directly comparable to the older Phenom system.

Obtaining data on hardware component utilization from software utilities while running the benchmarks would help in differentiating power patterns found in the measurement results by providing a more detailed understanding of how each benchmark stresses different hardware.

These power measurement methods can also be used in designing systems or applications capable of utilizing real time power feedback to optimizing energy efficiency.

ACKNOWLEDGMENT

A thanks to Professor Weaver for advising this work and designing ATX interceptor board used. This research was sponsored and made possible by the University of Maine NSF REU-sensors program. This work was supported by the National Science Foundation under Grant No. SSI-1450122.

REFERENCES

[1] K. O'Brien, D. C. Salyers, A. D. Striegel and C. Poellabauer, "Power and performance characteristics of USB flash drives," *World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a*, Newport Beach, CA, 2008, pp. 1-4.

[2] R. Lent, "A sensor network to profile the electrical power consumption of computer networks," *2010 IEEE Globecom Workshops*, Miami, FL, 2010, pp. 1433-1437.

[3] A. Mahersi and V. Vardhan. "Power Consumption of a Modern Laptop." In Proceedings of the 4th International Workshop on Power-Aware Computer Systems, 2004, pp 165-180.

[4] Adafruit Industries. USB Power Gauge. *http://www.adafruit.com/.*

[5] BK Precision. 2831E Multimeter. *http://www.bkprecision.com/.*

[6] Ookla. Speedtest. *http://www.speedtest.net.*

[7] Standard Performance Evaluation Corporation. Equake. *http://www.spec.org/cpu2000/.*

[8] IOzone Filesystem Benchmark. IOzone. *http://www.iozone.org/*

[9] McCalpin, John D., 1995: "Memory Bandwidth and Machine Balance in Current High Performance Computers", IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter, December 1995.

[10] Adafruit Industries. INA219 High Side DC Current Sensor Breakout *https://www.adafruit.com/product/904*