PowerInsight - A Commodity Power Measurement Capability

James H. Laros III #1, Phil Pokorny &2, David DeBonis #3
Sandia National Laboratories and & Penguin Computing
1 jhlaros@sandia.gov, 2 ppokorny@penguincomputing.com, 3 ddeboni@sandia.gov

***Abstract*—The challenge of balancing between power and performance is now well established. While research in this area is well underway, the ability to measure power and energy in situ has remained an obstacle. This problem is magnified in the field of High Performance Computing (HPC). To meet this challenge, a device called PowerInsight has been designed to accomplish component level power and energy instrumentation of commodity hardware. PowerInsight was designed by Penguin Computing, in close cooperation with Sandia National Laboratories, to further power and energy research in HPC and other areas. This paper documents the design and development of PowerInsight, hardware and software. Validation of the functionality of PowerInsight was done during design and development as well as experimentally after integrating the first PowerInsight devices into a commodity cluster. This paper only begins to show the wide range of impact this level of power and energy instrumentation can have on a range of architectural and application research and analysis topics.** [1]

***Keywords*-Power and Energy Measurement, High Performance Computing, Application Energy Profiling**

I. Introduction and Related Work

Previous work at Government Laboratories and Universities motivated and inspired this effort. Our own research[1], [2], [3] exposed the potential of component level power measurement in regards to in situ, large-scale application analysis and generating application profiles to evaluate the effect on application energy efficiency when tuning architectural parameters. While valuable, current and voltage measurements were limited to the CPU and network device and only the CPU data could be used for comparison of dynamic energy use. In addition, while the repeatability of the measurements was trustworthy, the accuracy and frequency of samples did not allow certain types of experimentation to be conducted reliably. Finally, the hardware used for our initial experiments was proprietary[2], limiting research to a specific architecture. An architecture independent solution with the following features was required:

- Component level measurement ability
- Discrete current and voltage samples
- High frequency samples per component
- Out-of-band collection of measurement data
- In-band capability - in parallel with out-of-band
- Ability to instrument at large scale
- Commercially available
- Other management capabilities

In [4], [5] and [6], component measurement capabilities were made possible through a framework called PowerPack. The authors used this framework to analyze power, energy and performance characteristics of applications on commodity hardware. Component level measurements were taken on one node and a technique called node-remapping was used to emulate larger scale runs.

In [7] the authors expanded this capability using laptops and ACPI to gather measurements. While limited in scalability, measurement frequency, and out-of-band capabilities this work illustrates the great potential of component level measurement.

Possibly the closest related work is found in [8] (PowerMon). Similarities include, instrumenting inline between power supply and motherboard, the ability to integrate into commodity nodes and high frequency samples. However, PowerMon requires in-band monitoring. While in-band monitoring is one of our requirements, an out-of-band monitoring capability is higher priority for our needs. In addition, this and other efforts use sense resistors. The loss across the resistor can be managed but a more passive approach is desired. Temperature buildup can also be an issue, especially when instrumenting higher powered components. PowerMon is limited to a 10A circuit, which limits its use on some modern architectures.

No solution was found that would meet our list of requirements. As a result, Sandia National Laboratories (Sandia) partnered with Penguin Computing to design PowerInsight. The Hardware (Section II) and Software (Section III) specifics which have enabled high frequency out-of-band (and in-band) component level current and voltage measurements will be outlined in this paper. Probably most important, Section IV explains the steps taken to date to validate PowerInsight during development and experimentally after integration. Conclusions and Future work will be presented in Section V.

II. Hardware

PowerInsight was designed to instrument commodity hardware. Version one of PowerInsight was installed at

[1]Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. Document SAND2013-3622C

[2]Research conducted at Sandia National Laboratories for the cited papers was accomplished on the Cray XT architecture

Sandia to instrument a 104 node commodity cluster to enable power and energy research. Each node has a single AMD Fusion A10-5800K processor which contains four Piledriver 3.8Ghz x86 cores and 384 800MHz Radeon accelerator cores. High speed network connectivity is provided using Qlogic Quad Data Rate Infiniband[3]. There are two management Ethernet networks, one for general cluster management such as booting, monitoring and control, and one connecting all PowerInsight devices to the top-level node (see Figure 1). There is one PowerInsight device integrated into each node of the cluster. The entire PowerInsight device fits nicely in a 3.5 inch disk bay in the front of each node. The PowerInsight devices operate in a disk-less mode. The top-level node is used for cluster management (Linux kernel, nfs-root, etc.) as well as a data aggregator for out-of-band data collection from the PowerInsight devices. The PowerInsight hardware is composed of three major components described in the following sections.

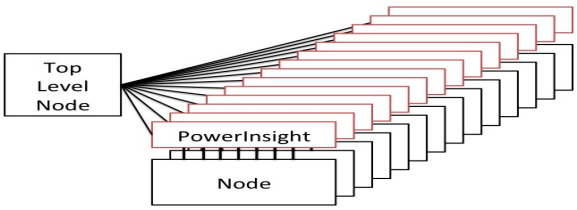

Figure 1. PowerInsight Network Connectivity

A. BeagleBone

The BeagleBone[9] (Figure 2) was selected as the core for PowerInsight. It is small in size but exceeds all connectivity and capability requirements. The processor on the BeagleBone is an ARM®Cortex™A8 with 256 MB of DDR2 Memory. The ARM processor supports hardware floating point, so there is no performance penalty incurred for scaling raw values. Connectivity to the BeagleBone is possible via the onboard 10/100 Ethernet and the USB device. A JTAG interface, and a large number of I/O pins are exposed through the expansion headers. The expansion headers (or cape connectors) are two 46 pin connectors on each side of the BeagleBone (top and bottom in Figure 2).

An embedded controller chip with similar I/O capabilities was considered prior to selecting the BeagleBone, but the advantages of a full Linux OS and functional network stack proved to be invaluable during development and use. Overall, the small size and ease of connectivity and integration provided by the expansion headers (SPI links, UARTs, analog inputs, GPIO) made the BeagleBone an excellent fit

[3]Intel True Scale Quad Data Rate Infiniband

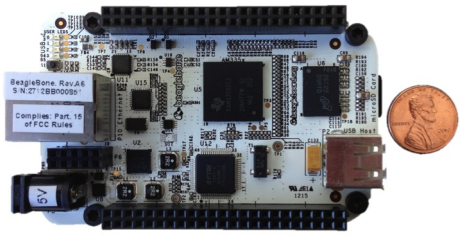

Figure 2. BeagleBone - Size Compared to a U.S. Penny

for the project. Note, some kernel work was required to expose all the available SPI links and chip selects to support the four Analog to Digital Converters (ADCs - one on the BeagleBone and three on the custom cape).

B. Custom Power Cape - Carrier Board

The PowerInsight cape (Figure 3), or carrier board, provides three of the ADCs, the voltage reference and the connectors for the sensor modules in the harnesses. The carrier board provides connections for 15 sensor modules (four pin connectors with white guides depicted in Figure 3 labeled J1-J15). Each connection provides power to the attached sensor module and routes the voltage and current signals to the associated channel on an ADC. The first eight voltage signals are connected to an ADC on the carrier board with a high-precision 4.096V reference voltage. The last seven connections are routed to the ADC built into the BeagleBone. These ADC inputs have a max voltage of 1.8V - a pull-down resistor is used to bring the sensed voltage in range. All the current signals are connected to ADC sensors with V_{cc} as the reference voltage. The ADC chips are Microchip MCP3008[10] and provide a 10-bit result. PowerInsight is powered by standby power through connector J16. This allows PowerInsight to remain active as long as the node is plugged in even if the node is powered down.

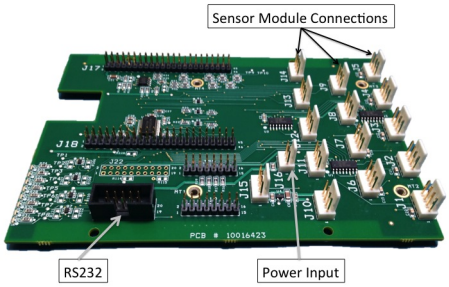

Figure 3. PowerInsight Cape - Carrier Board

C. Harnesses - Sensor Modules

The Harnesses (two examples pictured in Figure 4), which contain the sensor modules, are integrated into in-line between the power supply and the standard motherboard connectors (depicted in Figure 5).

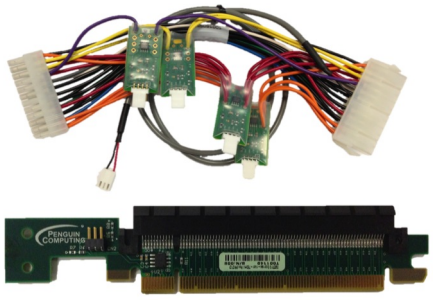

Figure 4. PowerInsight Harness - Sensor Modules

The sensor module is a small PCB with an Allegro Microsystems ACS713[11] Hall effect current sensor and voltage divider. A Hall effect sensor was selected for its low impact to the power rail being measured. The ACS713 can be inserted in-line on up to a 30A circuit without significant power loss or voltage drop. The ACS713 can also be placed on the high-side of the voltage rail regardless of the voltage due to the inherent magnetic isolation of the sensor. The magnetic isolation also allows use on negative voltage rails.

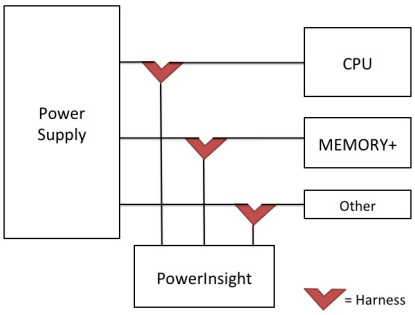

Figure 5. PowerInsight Harness/Sensor Module Integration into Motherboard

The output of the sensor is proportional to the measured current plus an offset. Both the slope and the offset are proportional to the supplied V_{cc} power pin. By using the V_{cc} as the reference for the analog inputs, the digital result automatically takes this scaling into account.

The carrier board uses two different voltage converters. One device uses a precision 4.096V reference and one, built into the BeagleBone module, uses a 1.8V reference from the BeagleBone power supply chip. The different reference voltages are accounted for in the design of the sensor module

Rail	R1	R2	R_{th}	V_{Ratio}
12V	40.2k	13.3k	9.994k	4.023
5V	16.5k	24.9k	9.924k	1.663
3.3V	11.0k	110.0k	10.00k	1.100

R_{th} = (R1*R2)(R1+R2) V_{th} = Voltage * R2/(R1+R2)

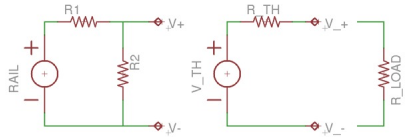

Figure 6. Voltage Divider Design and Resistor Selection

and carrier board (Figure 6). The resistors are selected so that the typical rail voltage is divided to a V_{th} of 3.0V for measurement by the 4.096V referenced converter. This provides a margin for voltage variations and maximizes the number of significant bits in the digital result. Resistor values are chosen such that the R_{th} equivalent resistance is 10kΩ. A single loading resistor, (R_{load} on the carrier board) is used on the inputs to the 1.8V referenced converter to further divide the sensor voltage by a 2:1 ratio resulting in a target voltage of 1.5V. In this way, any sensor board can be plugged into any channel and the sensing voltages will be in range for the voltage converter used on that channel.

D. Additional Capabilities

The PowerInsight device was designed with additional connectivity to maximize potential uses of the device now and in the future. The USB port from the BeagleBone is connected to the Motherboard and allows console connectivity from the node to the BeagleBone. This interface has proven to be useful during development for troubleshooting the Linux boot process for example. Ethernet over USB is also supported and provides a network link between the node and the BeagleBone. We are currently exploring using the USB interface from the node operating system to directly (in-line) obtain measurement information from PowerInsight.

The carrier board provides a connector and transceivers for an RS-232 serial port to the motherboard (black ten pin connector box depicted in Figure 3). Full modem control signals are connected to GPIO pins in addition to transmit and receive data signals. This enables console access to the Motherboard from PowerInsight and a simple signaling mechanism. This feature could be used to provide terminal server capability or other remote serial access capabilities.

There are LED's and connectors on the carrier to provide signaling, feedback and control. Chassis connectors enable remote power on/off and reset capabilities. This enables PowerInsight remote device control of the node rather than the limited power on/off/cycle typically provided by smart Power Distribution Units.

In combination, these capabilities allow PowerInsight to provide much of the functionality of a dedicated system management controller to extend system capabilities beyond what is typically found on commodity clusters.

III. Software

A. Node Level Data Collection

During initial design and testing, a simple but functional utility was developed to demonstrate how to collect readings from the ADC hardware and scale them to report values in milliamps (mA), millivolts (mV) and milliwatts (mW). The utility, getRawPower, interfaces with the Linux kernel SPI drivers via a user space *ioctl()* interface. First, the SPI device is opened and initialized by setting bits per word and max clock settings. Raw data is collected by issuing an SPI call to trigger a transfer. Each raw data sample from the ADC consists of a start byte, a channel select byte and a dummy byte. The ADC performs a conversion as the bits are clocked (in/out). The *ioctl()* call returns the three bytes clocked in during the transfer and the ten bit reading is extracted from the second and third bytes. The BeagleBone ADC driver provides results through a set of files created in *sysfs*. Each time a read call is made, a new reading is returned as decimal ASCII text.

Scaling is performed on the raw values to produce current (in mA), voltage (in mV) and power (in mW). The mapping of channels to the voltage rail being measured is currently hard coded. We plan to implement a configuration file to control mapping of channels to voltage rails which will allow the scaling equations to be "tweaked" based on calibration factors.

Currently, getRawPower accepts a list of arguments on the command line specifying which channels (ports) to read. A value for each channel requested is produced, allowing values from one or a list of channels to be returned with a single call. We have also modified getRawPower to continuously return values, as fast as possible, at greater than 1KHz (from user space). This sampling rate is limited by user-space overhead. The sensors should be capable of four times this sample rate from kernel space, likely limited by the SPI bus. The output, raw and scaled values, is in formatted text. This output is later processed by a range of post processing scripts developed along with the system level data collection software.

B. System Level Data Collection

Since all nodes in the cluster are instrumented with PowerInsight, a centralized data collection mechanism was implemented to aggregate the potentially large amount of data produced. Figure 1 depicts the simple logical network hierarchy representing connectivity of each PowerInsight device (integrated into each node of the cluster) with the top-level node. A number of daemon processes, scaled automatically, on the top-level node act as proxies for receipt of power data from individual agents running on each PowerInsight device. Agents can be individually configured through peer-to-peer communication from any node with network connectivity by using a control program. Sample rate and sensor port state (collecting or not) are two of the configuration parameters currently exposed. The peer-to-peer communication runs over TCP/IP, which is sufficiently scalable for the numbers of clients currently configured. In our current implementation, each agent periodically (depending on its last instructed sample rate) calls getRawPower to extract sensor port values. These values are communicated to the proxy daemons running on the top-level node. Note, proxy daemons can be run on any node with network connectivity to the PowerInsight devices.

The proxy daemon aggregates all data originating from the agents and outputs a formatted flat file. This flat file is used as input for all post-processing analysis. Fine grained information down to the individual sample is retained. All data includes a timestamp (microsecond precision). It is important that time is synchronized among the PowerInsight devices and the cluster nodes so data can be correlated with job start and completion times.

A post-processing analysis suite is used to partition the formatted flat file data based on PowerInsight device (which corresponds to an individual compute node) and sensor port into individual files for further visualization and data mining. Plots of the partitioned files are automatically generated for fast visual analysis. Statistical analysis is also performed on the data.

IV. PowerInsight Validation

The design and subsequent implementation of PowerInsight has been validated both during initial development and after integration into the cluster at Sandia National Laboratories.

A. Design Validation

Initial validation of the PowerInsight carriers was done using precision voltmeters and ammeters to validate the measurements made by PowerInsight. A set of power resistors were used to provide a static load on each rail for measurement. Demonstrated load current measurement accuracy for the entire system averages 1.8%. The Hall effect sensor is responsible for 1.5% of the error. Demonstrated rail voltage measurement accuracy averages within 0.3%. Future refinements in the conversion algorithm may reduce the error further. The measured error was consistent with the data-sheet[10]. The ADCs on the carrier board were tested, using a potentiometer to produce a stable voltage signal, and proved extremely stable, +/-1 count change over time in the digital result. The ADC on the BeagleBone module has proven less accurate with significant noise in the digital result. Note, the BeagleBone ADC is not used in our current implementation but might be applied in later

designs if appropriate. A *full load* test was performed at 15 Amps to confirm that there is no significant power or voltage drop across the current sensor. The sensor board handled the load for several hours with no significant heat buildup.

Currently, scaling equations based on the data-sheet values and precision resistor values are being used. Future work will characterize a large sample of parts to confirm the distribution of key sensor parameters. Additionally, the software will be enhanced to allow calibration values to be modified to trim each sense and channel.

B. Experimental Validation

In addition to the validation processes conducted during the design and prototype phases of developing PowerInsight, experiments were conducted after installation to validate the devices were installed and working properly and to gain confidence in the data reported. The first experiment compared the results from an identical run on 89 nodes of the cluster (the nodes not included were being used for other experiments). Single node High Performance Linpack (HPL) was run on all cores of all nodes (single execution per node) using identical input parameters. Note, in each case, HPL was run only on the general purpose (x86) cores of the AMD processor. Data was collected during this experiment for all nodes at a one sample per second rate. The raw data was then processed and the average power (energy divided by time) over the duration of the run was used to determine if individual PowerInsight devices were reporting similar values. Figure 7 plots the average power in watts for each node used in the experiment.

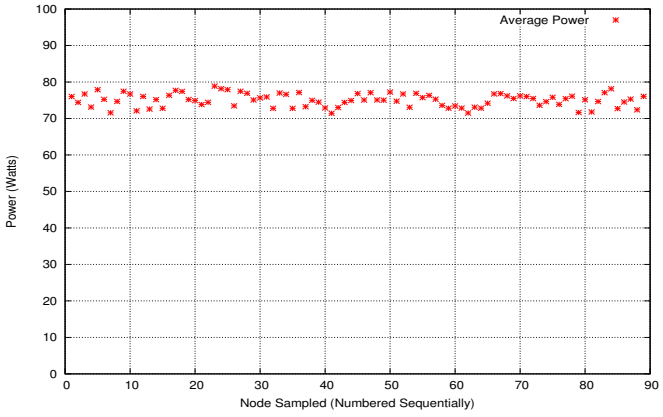

Figure 7. Average Power Per Node

As can be seen in Figure 7, the average power of each individual node during the HPL run was very consistent, varying by only 2.45%[4]. Individual processors of the same type while expected to draw similar amounts of power for a fixed workload, do vary on a per processor basis due to manufacturing processes and individual processor temperature. The AMD A10-5800K processor is nominally a 100W CPU. A measured average power between 70W and 80W is as expected[12].

Possibly more important for experimentation is the repeatability of results on the same node. If the measurements produced are repeatable they can be used for delta comparisons. To determine if the PowerInsight devices produced repeatable results, HPL was again run on 90 nodes 10 executions per node. As in the first experiment, single node HPL runs were used to ensure the same work was done on each node and all four of the general purpose cores were used. All data of HPL runs that passed residual (897 of 900 executions) was used including runs that would typically be considered outliers. The raw data was processed to produce energy values for each of the ten runs on every node. The CV was then calculated, per node, and plotted (Figure 8 bottom). Runtime for each execution per node was also captured and the CV was calculated and plotted (Figure 8 middle). Note, the energy and run-time variation is very low for the majority of the nodes. Nodes that have a larger energy variation map directly to larger variations in run-time. In fact, examining the raw data in detail reveals that this is typically caused by one or two runs per node on the outliers that have a very different, usually longer, run-time than the norm. As would be expected, the energy value varies as the run-time varies. If the energy is normalized to the run-time (divide the energy value by the run-time for each execution) and the CV is calculated the variation becomes very small from execution to execution on a per node basis (Figure 8 top). The CV on 84 out of 90 nodes is under 1%. The greatest normalized value, including outliers, was 2.28%. If anomalous executions are excluded variation from run to run is consistently far less than 1%.

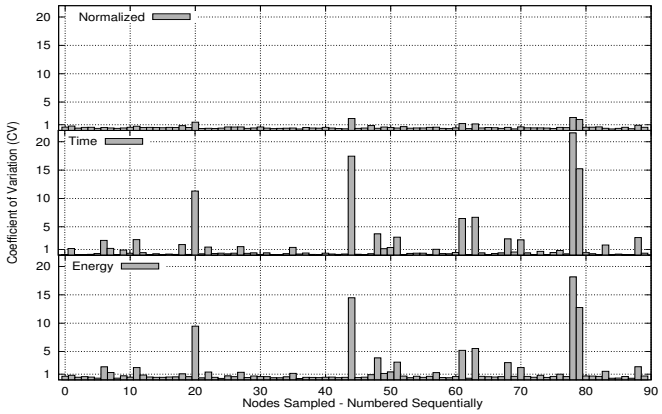

Figure 8. Variation of Energy Per Node

The results of this experiment validated that the PowerInsight device produced repeatable measurements over many runs on the same node. Conducting experiments where differences between baseline power or energy measurements

[4]calculated coefficient of variation

and measurements taken after varying the parameter under investigation can be expected to produce reliable results.

V. Conclusions and Future Work

The primary goal of this paper was to describe a new commodity power and energy measurement device, PowerInsight. Figure 9 shows an *Application Profile*[5] obtained using PowerInsight. This Application Profile shows one of the Mantevo[13], [14] proxy applications developed at Sandia, MiniFE, executing first on only the x86 cores of the AMD Fusion processor followed by the same problem executing only on the accelerator cores of the processor.

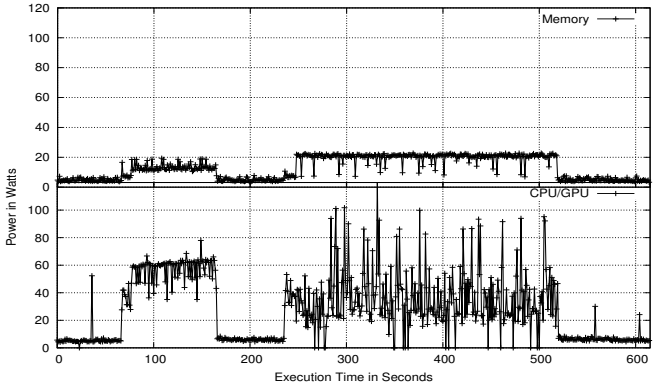

Figure 9. MiniFE - Executed on CPU only followed by GPU only

The lower graph shows the processor profile while the upper graph shows the memory profile. It is this fine grained measurement ability that will allow us to analyze our applications in detail with the goal of understanding where energy is used and how to optimize our applications for both performance and energy. Future work will include analyzing a wide range of component energy use now enabled by PowerInsight. We have recently added the capability of instrumenting accelerators, examples include PCI devices like Nvidia and ATI GP-GPUs and Intel Phi. Simply inserting a harness inline with the power pig-tail to these devices won't tell the complete story. A PCI riser device (see Figure 4) was designed to measure the substantial power supplied by the PCI bus for these types of devices. As we have described, while the primary purpose of developing PowerInsight was to measure power and energy at the component level, from the beginning it was designed for expanded capabilities. We expect PowerInsight will enable a wide range of research that was not previously possible.

Acknowledgments

The authors would like to recognize the significant contributions of Paul West in developing the getRawPower utility during the initial design and validation process. We would also like to recognize members of the Advanced Architecture Test Bed project at Sandia including James Ang, Sue Kelly, Simon Hammond, Bob Ballance, James Brandt, Ann Gentile, Victor Kuhns, Jason Repik, and Charlene Arias, for their contributions to the project. This work was funded by the Advanced Simulation and Computing (ASC) program of the National Nuclear Security Administration (NNSA).

[5]a term we used in [1] to describe the power and energy fingerprint of an application

References

[1] J. H. Laros III, K. T. Pedretti, S. M. Kelly, J. P. Vandyke, K. B. Ferreira, C. T. Vaughan, and M. Swan, "Topics on measuring real power usage on high performance computing platforms," in *IEEE Cluster 2009, International Conference on Cluster Computing*. Sandia National Laboratories, September 2009.

[2] J. H. Laros III, K. T. Pedretti, S. M. Kelly, W. Shu, and C. T. Vaughan, "Energy based performance tuning for large scale high performance computing systems," in *HPCS 2012, 20th High Performance Computing Symposium*. Sandia National Laboratories, March 2012.

[3] J. H. Laros III, K. T. Pedretti, S. M. Kelly, W. Shu, K. Ferreira, J. Van Dyke, and C. T. Vaughan, *Energy-Efficient High Performance Computing - Measurement and Tuning*. Springer, ISBN 978-1-4471-4492-2, 2012.

[4] X. Feng, R. Ge, and K. W. Cameron, "Power and Energy Profiling on Scientific Applications on Distributed Systems," in *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2005.

[5] K. Cameron, R. Ge, and X. Feng, "High-performance, power-aware distributed computing for scientific applications," *Computer*, vol. 38, no. 11, pp. 40 – 47, nov. 2005.

[6] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. Cameron, "PowerPack: Energy Profiling and Analysis of High-Performance Systems and Applications," *Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 658–671, 2010.

[7] R. Ge, X. Feng, and K. W. Cameron, "Performance-Constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage, and Analysis (SC)*. ACM/IEEE, 2005.

[8] D. Bedard, M. Y. Lim, R. Fowler, and A. Porterfield, "Powermon: Fine-grained and integrated power monitoring for commodity computer systems," in *IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the*. IEEE, 2010.

[9] BeagleBone. BeagleBone.org. [Online]. Available: http://beagleboard.org/bone

[10] MCP3008. Microchip.com. [Online]. Available: http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010530

[11] ACS713. Allegromicro.com. [Online]. Available: http://www.allegromicro.com/Products/Current-Sensor-ICs/Zero-To-Fifty-Amp-Integrated-Conductor-Sensor-ICs/ACS713.aspx

[12] AMD, "ACP The Truth About Power Consumption Starts Here," Applied Micro Devices, Tech. Rep. 43761C, 2009. [Online]. Available: http://www.amd.com/us/Documents/43761C_ACP_WP_EE.pdf

[13] Mantevo. Sandia National Laboratories. [Online]. Available: http://www.mantevo.org/

[14] M. A. Heroux, D. D. Doerfler, P. S. Crozier, J. M. Willenbring, H. C. Edwards, A. Williams, M. Rajan, E. R. Keiter, H. K. Thornquist, and R. W. Numrich, Tech. Rep.