# Is State of the Art Good Enough for Power Measurements

David DeBonis
Sandia National Laboratories

July 22, 2013

## 1 Abstract

## 2 Introduction

## 3 Related Work

Previous work which have targeted live data collection rather than simulation have shown that thermal research on advanced aggresively- clock-gated superscalar processors is problematic [3]. In that study the total power is gathered live through inductive clamp with component power estimated based on performance counters which are then correlated using Bayesian similarity matrices. In our work we gather our power values directly without the need for probabolistic methods since we achieve component level granularity directly.

Component level profiling was achieved in another study utilizing ten multimeters per node, each monitoring power over a shunt resistor inserted through ATX extension cables [2]. While their results gathered accurate per node power data, the collection had to be made through individual collection over each node through multiple passes. The application under test also needed to be instrumented directly to control the power gathering. Our work allows us to collect concurrently over all nodes regardless of scale in situ, avoiding the need to manually retool a particular node in the system.

The implementation of a scalable power measurement framework was presented utilizing board level interface exploitations to measure power usage [**?**]. Though a great proof-of-concept and of utility, the accuracy proved rather poor at +/- 2 amperes. The concept of Application Power Signatures was introduced and the first quantitative study of OS noise was performed. Many of the concepts from that work were leveraged in our study.

A low-cost power monitoring device that can operate inside commodity computing systems was introduced [1]. In their study it was noted that the use of AC monitors like PowerEgg and WattsUp were inadequate to capture fast variations in the DC load of the supply. Through the use of sense resistors, individual DC power rails were monitored with high resolution. There is a cost, of computational overhead on the host and temperature rise of up to 20C on the device with a 10 amp load. Our presented work is passive to both the host and power rails through the use of a seperate embedded processor with its own communication infrastructure and inductive based monitoring.

# 4 Methods

## 4.1 System Level Data Collection

The power monitoring system provides centralized logging of distributed clustered node power usage.

A daemon running on the top level node acts as a proxy for receiving power information from individual agents running on each of the PI modules at all times. These agents can individually be configured through peer-to-peer communication between the PI and any node within the system (i.e. login node, compute node, top level node, etc.). The ability to configure an agents sample-rate and individual sensor port state (collecting or not) is exposed through a peer-to-peer communication protocol over TCP/IP posix sockets. Each agent periodically (dependent on its configured sample rate) calls getRawPower to extract sensor port values which are communicated to the proxy daemon. Other topological configurations and communication patterns are available but not presented within this work.

The proxy daemon aggregates all of the agents log messages into a flat file that forms the base data to be analyzed post-process. Fine grained information down to the sample are retained without data thinning to avoid any loss of detail. Timing information (which is regulated via the local cluster NTP) is attained down to microseconds and power values to milliamps, millivolts, and milliwatts are recorded.

A post-processing analysis suite is used to partition the collection based on PI node and sensor port into individual files for further visualization and mining. Plots of the partitioned files are also generated for fast visual analysis of a given collection run. A summary statistical analysis is performed over all of the PI nodes and sensor ports to distille range and average values for amperage, voltage, and wattage along with running time and power consumed. A plot of this course-grained information is generated for visual inspection.

# 5 Experiments

## 5.1 Fidelity

## 5.2 Accuracy

# 6 Conclusions

A key differentiator of PowerInsight is its out-of-band communication, allowing for neither performance or power impact on system components. We have shown that by using a seperate computing device to perform the collection, calculations, and aggregation of power information that the CPU can essentially offload all but what it is interested in (TBD - could mention producer/subscriber model vs. polled). In addition, electically it is outside of the system since power is measured through induced signaling.

# 7 Future Work

# 8 Acknowledgements

I would like to thank James Larros and Kevin Pedretti for their support of this research.

# References

[1] D. Bedard, Min Yeol 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*, pages 479–484, 2010.

[2] Xizhou Feng, Rong Ge, and K.W. Cameron. Power and energy profiling of scientific applications on distributed systems. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, pages 34–34, 2005.

[3] Canturk Isci and Margaret Martonosi. Runtime power monitoring in high-end processors: Methodology and empirical data. In *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 36, pages 93–, Washington, DC, USA, 2003. IEEE Computer Society.