

# Energy Measurement Tools for Ultrascale Computing: A Survey

*Francisco Almeida<sup>1</sup>, Javier Arteaga<sup>1</sup>, Vicente Blanco<sup>1</sup>, Alberto Cabrera<sup>2</sup>*

© The Authors 2017. This paper is published with open access at SuperFri.org

With energy efficiency one of the main challenges on the way towards ultrascale systems, there is a great need for access to high-quality energy consumption data. Such data would enable researchers and designers to pinpoint energy inefficiencies at all levels of the computing stack, from whole nodes down to critical regions of code. However, measurement capabilities are often missing, and significantly differ between platforms where they exist. A standard is yet to be established. To that end, this paper attempts an extensive survey of energy measurement tools currently available at both the hardware and software level, comparing their features with respect to energy monitoring.

*Keywords: energy measurement, power measurement, data acquisition tools, infrastructure management, ultrascale computing.*

## Introduction

Energy sustainability is a significant concern for high-performance computing, with cost of operation due to power draw being one of the main limiting factors for the design of new systems. This need to improve the energy efficiency of computation is compounded by the growth to ultrascale infrastructure.

To enable energy optimization across the whole stack, it is desirable to gain insight into the consumption of existing systems at all levels possible. Ideally, system designers and operators, as well as application developers, should be able to easily access precise power data ranging from whole systems to individual components inside a computation node. It should also be easily attributable to the code being executed, again ranging from entire processes to parts of specific threads.

However, the power monitoring capabilities of current high-performance computing and ultrascale systems are often more limited. In many cases, only aggregate and approximate data are available. A low level of precision and temporal resolution can be sufficient for administration and maintenance purposes, but many interesting applications, such as application energy efficiency analysis or energy-aware dynamic scheduling, require finer-grained measurements.

The issue of acquisition portability also remains open. Existing built-in component power sensors have to be read using as many different interfaces as vendors are involved. Current data center management standards (such as the Intelligent Platform Management Interface (IPMI) [21] or the Data Center Manageability Interface (DCMI) [20]) cannot leverage most sensors, providing instead low-resolution data from supported motherboards.

For these reasons, we believe there is a justification for standardization of energy data acquisition techniques. Such a unification attempt would benefit from an understanding of the current state of the art. To that end, we will survey a wide range of measurement hardware and software tools, providing a classification and a review of all capabilities identified as relevant to the analysis of energy consumption.

The remainder of this work is structured as follows: section 1 defines the scope of our survey, referencing related work. Section 2 reports available power and energy measurement devices, grouping them by physical location on the compute node. Section 3 presents a taxonomy and

---

<sup>1</sup>Universidad de La Laguna, Santa Cruz de Tenerife, Spain

<sup>2</sup>Instituto Tecnológico y de Energías Renovables, Santa Cruz de Tenerife, Spain

a review of software tools and libraries involved in the measurement process. The final section concludes the paper and outlines future work.

## 1. Related work

This survey covers hardware and software systems reporting direct energy and power measurements, with particular focus on current high-resolution approaches, where consumption data for each component is updated at a high frequency. As such, energy consumption estimation techniques, analytical modelling tools and power-aware management systems fall outside our scope.

Previous surveys have mapped the landscape of energy measurement tools. A broad picture is presented in [6], which defines a taxonomy encompassing methods based on measurement, estimation and analytical modelling. However, coverage of hardware based methods is not comprehensive. The state of the art has also significantly changed since its publication.

In [33], both energy modeling and energy measurement techniques are covered. Here, the main focus is not on data acquisition, and many existing hardware and software tools are not included.

A more measurement-oriented survey of methods is found in [19]. However, at sub-node granularity, only two custom instrumentation systems are presented. It also does not cover later developments in integrated power sensors.

An in-depth exploration on some hardware methods can be found on [16], providing detailed insights into the quality of a few chosen power meters and sensors. It covers a subset of the devices considered in this survey.

## 2. Measurement hardware

Energy consumption data is acquired through hardware sensors. There are a number of implementations, which are typically grouped in three categories by physical location (fig. 1):

- Integrated, where specific hardware components (such as a GPU or CPU) already contain measurement circuitry,
- intra-node, instrumentation devices placed inside the node that can perform probing at component or power lane level, and
- external power meters, measuring total load outside the node's power supply.

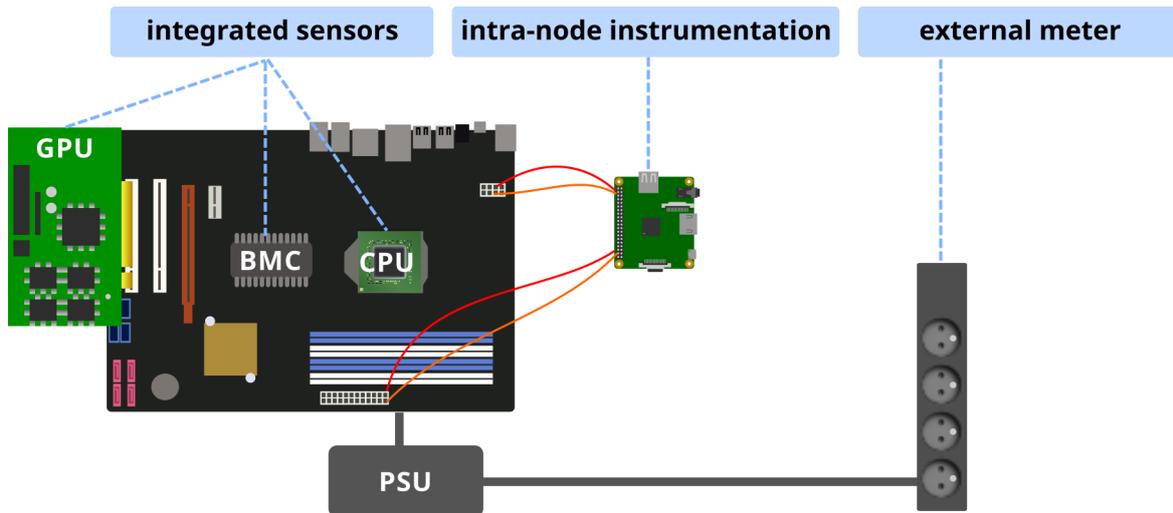
Each of these approaches presents their own tradeoff between precision of measurement, temporal resolution, cost of deployment and intrusiveness. The interfaces involved in data acquisition are also different between them. A comparison of interfaces and sampling frequencies among some of the current systems is provided in tab. 1.

### 2.1. Integrated sensors

#### 2.1.1. CPUs

CPUs typically allow access to a number of performance counters, such as instruction or cache miss counts, that are useful in profiling applications. In modern units, it is also possible to query energy consumption estimates directly provided from the processor.

Intel processors from Sandy Bridge onwards implement the Running Average Power Limit (RAPL) interface [24], which provides running counters of total energy consumed per package



**Figure 1.** Diagram of hardware measurement tools

(and also, on some models, a total for the DRAM). This interface provides updates roughly every millisecond. However, these counters may overflow: it is up to the interface user to take this fact into consideration. This means constantly polling the registers, in turn increasing the load on a CPU core and adds overhead to the measurements.

AMD processors starting from the Bulldozer (family 15h) microarchitecture also export an estimation of average power over a certain interval through the Application Power Management (APM) [1] capability. Unfortunately, the actual implementation for the Bulldozer family has been shown [16] to provide an inaccurate estimate, particularly during processor sleep states.

### 2.1.2. GPUs and accelerator cards

Hybrid systems, containing some sort of accelerator units such as GPUs, have been on the rise in recent years [18]. It is thus also desirable to have these components provide energy consumption data from the hardware level.

Many current GPUs provide support for power limiting, which implies the capability to either measure or perform a meaningful estimate of power usage. Nvidia Tesla and Quadro GPUs (from the Fermi GF11x family onwards) additionally make this data available to the user as instant power draw values with nominal accuracy up to  $\pm 5\%$  through the Nvidia Management Library [34] C API. Update frequency is not documented, and many potential complications, such as significant sensor lag or sampling interval variability, have been experimentally identified [9].

Another accelerator, the Intel Xeon Phi, exposes a greater number of power sensors: connector inputs, voltage regulator outputs, and readings for both instant and averaged power draw of the entire card through its System Management Controller chip [23]. Temporal resolution and precision are of 50ms and 1W, respectively. Two methods are provided to query this data [22]: in-band, which involves the Symmetric Communications Interface and both accelerator and host software support, and out-of-band (without waking up the coprocessor card), via the standard Intelligent Platform Management Bus protocol over the System Management Bus. Multiple software interfaces are provided for both of these methods.

### 2.1.3. Mainboard

The Advanced Configuration and Power Interface (ACPI) open specification [47] defines power management and device configuration interfaces between an operating system and the BIOS or UEFI. Some power-related information is accessible through ACPI, such as supported processor power states and their expected consumption. As for measuring actual power usage, however, the usefulness of ACPI is limited to rough system-wide estimations in battery-powered systems.

A more advanced form of power usage monitoring can be performed on some motherboards supporting the Intelligent Platform Management Interface (IPMI) [21] (and thus equipped with a Baseboard Management Controller (BMC) monitoring chip). There exist a number of vendor-specific extensions (as implemented by the Dell Remote Access Controller, HP Integrated Lights-Out or Intel Node Manager technologies) which specifically relate to power usage. A more recent standard, the Data Center Manageability Interface (DCMI) [20], builds on top of IPMI 2.0 and introduces power monitoring sensor requirements. However, both standards are devised for administration rather than research purposes, and offer power sampling rates on the order of seconds in the best case.

Other vendors provide entirely proprietary measurement interfaces. This is the case of the IBM Blue Gene/Q, where every node board is fitted with a FPGA which polls voltage and current of all different power domains every 560ms [55]. This information can then be retrieved through IBM's Environmental Monitoring (EMON) API.

## 2.2. Intra-node instrumentation

For the purposes of dynamic application power profiling, the integrated sensors typically available often do not provide the necessary level of measurement accuracy and subsystem coverage. In these cases, researchers typically employ more sophisticated, and often custom-designed, hardware tooling. For example, the Linux Energy Attribution and Accounting Platform [37] instruments a system's main board to provide power readings using a data acquisition board, which are then exposed to the user via the Linux `/proc` filesystem.

PowerPack [14] was one of the first frameworks aimed towards high-fidelity power-performance profiling. It is comprised of a collection of hardware sensors, meters and data acquisition devices and a software stack providing device drivers and user acquisition interfaces.

The PowerMon line of devices [4], developed by the Renaissance Computing Institute, is inserted between a system's power supply and motherboard, monitoring voltage and current on DC rails to components. Its latest iteration (PowerMon2) can measure up to 8 channels, reaching measurement frequencies up to 1 kSa/s (samples per second) per channel (3 kSa/s aggregate) with low voltage and current measurement error.

A later development in node instrumentation is the PowerInsight [28] from Sandia National Laboratories. Based on the BeagleBone single-board computer plus a custom carrier-board and software layer, PowerInsight allows for instrumentation of up to 15 rails using both special cabling and PCI riser devices fitted with Hall effect sensors. Measurement frequencies are claimed to reach 1 kSa/s counting user-space overhead, down from the 4 kSa/s supported by hardware. Voltage and current accuracies are reported to be higher than those of PowerMon2 [28].

The High Definition Energy Efficiency Monitoring (HDEEM) project [17] implements a similar approach to PowerInsight, claiming a high quality of results due to work on noise filtering

and sensor calibration, potential for high temporal resolution thanks to the speed afforded by the PCIe bus and greater interoperability through integration with the IPMI specification. Hardware cost is also said to be reduced [17], as their hardware implementation builds on top of an already present Baseboard Management Controller chip instead of completely relying on a custom board.

Other devices that may be used for instrumentation include the ARM Energy Probe [3], used with the ARM DS-5 toolchain for energy optimization of software on ARM boards; National Instruments data acquisition equipment [32], which is also available in PCI/PCIe form factors suitable for node instrumentation; and many current measurement integrated circuits based on current-shunt or Hall effect.

### **2.3. External meters**

Measurement from outside the node's power supply is a fairly straightforward method, with little intrusiveness and lower cost than node instrumentation. However, it produces the least useful results: power draw values cannot be attributed to specific components or processes within the node, and the time granularity is typically coarse.

Dedicated power meters available can be inserted between a system and wall outlet. Examples of consumer-grade products are the Kill A Watt [35] and Watt's Up [50] devices, with a time granularity of 1 Sa/s. For the purposes of application power analysis, more high-end devices such as ZES ZIMMER [57] or Yokogawa [54] products can provide far more precise data.

Many power distribution units (PDUs) [10, 40] and power supply units (PSUs) used in data centers also include monitoring capabilities through a variety of interfaces (SNMP, IPMI, Modbus...).

Finally, external custom designs have also been developed for previous work such as PowerScope [13], which used a digital multimeter with a trigger input connected to profiling software, or the Energy Endoscope [43], where an application-specific integrated circuit (ASIC) dedicated to real-time energy monitoring was built.

### **2.4. Assessment**

It is always desirable that power sensors be integrated in hardware components. Some current CPUs and GPUs include reasonably useful measurement capabilities, but there is a room for improvement in accuracy and latency. Other important targets, such as ARM-based processors, as well as subsystems such as disks, memory or network cards, usually offer none of these features. In all cases, energy data is only available at the component level, with no reliable mechanism to attribute energy consumption to a particular software task within a multicore system.

Hardware vendors should aim to provide accurate energy and power data with as fine spatial and temporal granularity as possible. This would enable efficiency gains backed by precise accounting of the power consumption of any specific subsystem down to the process level. Implementation of out-of-band channels would also help minimize the intrusiveness of readings.

In the long term, built-in sensors should phase out any custom-tailored intranode instrumentation on datacenters and HPC deployments. A hardware interface standard for these sensors is also necessary, covering both in-band and out-of-band collection of high-resolution performance data from all components and all external meters present in a computing system. Design insight can be drawn from both vendor-specific and custom implementations seen in this section.

**Table 1.** Comparison of hardware power measurement systems

Hardware	Vendor/group	Acquisition interfaces	Temporal resolution
<i>Integrated sensors</i>			
Intel CPUs (Sandy Bridge+)	Intel	RAPL	1 kSa/s [17]
AMD CPUs (Bulldozer+)	AMD	APM	100 Sa/s [17]
Tesla, Quadro GPUs	Nvidia	NVML	>60 Sa/s [9, 25]
Xeon Phi	Intel	Custom, IPMI+SMBus	20 Sa/s [23]
<i>Node instrumentation</i>			
PowerPack	Virginia Tech	NI DAQs, Watt's Up [14]	Unknown
PowerMon2	RENCI	Serial port	1 kSa/s (per channel) [5]
PowerInsight	Sandia National Labs, Penguin Computing	SPI bus	>1 kSa/s [28]
HDEEM	TU Dresden, Bull	PCIe and IPMI	1 kSa/s planned [17]
<i>External meters</i>			
Watt's Up Pro	Watt's Up Meters	USB, Ethernet (.Net)	1 Sa/s [49]
Schleifenbauer PDU	Schleifenbauer	Network-based (SNMP, custom...)	1 Sa/s [39]
ZES LMG450	ZES ZIMMER	Serial, Parallel	20 Sa/s [56]

### 3. Software tools

Many different software tools can be used for power and energy analysis, from low-level software interfaces to full-fledged analysis frameworks, including profilers, tracing and visualization systems, and estimation and modeling tools.

In this section, we will cover those related to the acquisition of power data from hardware sensors. In particular, a comparison of software acquisition interfaces by supported hardware is provided in tab. 2.

#### 3.1. Power-aware low-level profiling interfaces

The Performance API (PAPI) [8], from the University of Tennessee, is well-established as a library interface for hardware performance counters. In recent years, it has included support for many energy measurement sources, such as RAPL, NVML, Xeon Phi or IBM EMON [29, 51, 52]. This allows for easy extraction of power data for projects and researchers which are already users of PAPI. However, PAPI remains focused on in-band measurement, and data from external meters must be collected with some other tool.

Some profiling frameworks focused on CPU performance counters support CPU-based energy metrics. For example, the `likwid` [46] profiler, as well as the `perf_events` [26] Linux kernel subsystem (a tool designed for profiling CPU performance counters) both directly implement measurement through the RAPL interface.

### 3.2. Power-aware profiling frameworks

Larger profiling frameworks, with tracing and visualization systems, are usually built on top of PAPI and other libraries, using them as data providers for larger analysis and visualization frameworks. Because of this, many of them are capable of working with in-band power consumption measurement to some extent. These include Paraver [36], Vampir [31], HPCView [30], the Tuning and Analysis Utilities (TAU) [42], Open—Speedshop [41]. Scalasca [15] or Periscope [7].

Some attempts exist to offer interoperability between these systems. In particular, the Score-P [27] measurement infrastructure is compatible with Vampir [31] (replacing the older Vampir-Trace open source library), Scalasca, Periscope and TAU tools. Additional support for energy metrics is currently being worked on within the follow-up Score-E [48] project.

Other profiling solutions also rely on specific hardware systems. The Multiple Metrics Modeling Infrastructure (MuMMI) [53] is one such case, building upon PAPI, PowerPack and the Prophecy [45] performance modeling and prediction framework.

### 3.3. Power-specific software interfaces

PowerAPI [38] is a recent attempt from Sandia National Laboratories to standardize access to power measurement data and power control at all levels of a given HPC facility, down to hardware components. It is comprised of a specification defining a model of a computation system, user roles and the reference API. A prototype implementation is provided that already implements native support for some energy data sources including RAPL, Cray products supporting power management, the WattsUp meter and PowerInsight.

The Energy Measurement Library (EML) [12] is an open source C library developed at Universidad de La Laguna providing a simple interface for acquisition of hardware energy consumption data through code instrumentation. The API is designed around the concept of inserting asynchronous instrumentation calls around relevant sections of code. Within these sections, the library polls the underlying interfaces, gathering energy data and managing all needed threads and memory [11]. Hardware support is provided for RAPL, NVML, Xeon Phi and Schleifenbauer PDUs.

`pmlib` [2] is a software package developed at Universitat Jaume I to research the power-performance of parallel scientific code. The framework uses a client/server model for out-of-band power tracing of a target node instrumented with power meter devices. Supported meters include APC PDU units, Watt's Up Pro devices, a data acquisition system from National Instruments, and custom transducer-based designs.

SchedMon [44], from the Signal Processing Group at INESC-ID, reports hardware power consumption by coupling a Linux kernel module for access to hardware counters, a library providing the measurements to userspace, and a reference commandline user interface. It currently obtains power data from the RAPL interface.

**Table 2.** Comparison of low-level software power measurement interfaces

Name	Author	License	Power measurement support
<i>Power-aware profilers and interfaces</i>			
PAPI	University of Tennessee	BSD	Intel RAPL, Intel Xeon Phi, Nvidia NVML, IBM EMON
likwid	Jan Treibig et al. [46]	GPLv3	Intel RAPL
perf_events	Community	GPLv2	Intel RAPL
<i>Power-specific software interfaces</i>			
PowerAPI (prototype)	Sandia National Laboratories	BSD	Intel RAPL, Cray XTPM, PowerInsight, Watt's Up
EML	Universidad de La Laguna	GPLv2	Intel RAPL, Nvidia NVML, Intel Xeon Phi, Schleifenbauer PDU
pmlib	Universitat Jaume I	Unknown	Intel RAPL, APC PDU, Watt's Up, National Instrument DAQs, custom
SchedMon	INESC-ID	MIT	Intel RAPL

### 3.4. Assessment

Both HPC tools and research code benefit from a set of standard capabilities for energy data acquisition. Thus, they should be built upon low-level software abstraction layers which give a simple interface to performance data sources, including energy. In the absence of hardware standards to leverage, it would fall upon this layer to encapsulate the complexity of dealing with different sensor types. Currently, PAPI is in a good position to also fill this role for energy and power, although it is hindered by a lack of out-of-band measurement support.

Further abstraction layers should define standard data trace formats and utilities for easy interoperability between tools. This format must take into account the scalability concerns involved in dealing with high-resolution data in large-scale deployment. In the HPC analysis tooling front, some important standardization work is already underway in the form of projects such as Score-P. These standards should continue to mature and be adopted by more HPC frameworks.

Lastly, many high-level tools could be extended through these libraries to consider available energy data. Datacenter operators and researchers alike should have analysis and visualization tools with the ability to “zoom in” on the energy consumption incurred for a given program, user, or even piece of code, both offline and online, looking at any level of hardware granularity: from a whole system, to a specific core within one processor. Many other practical applications where this data could drive energy efficiency gains can be imagined, such as energy-aware scheduling and load balancing, or compiler-assisted energy optimization of program code.

## Conclusions

This survey has attempted to outline the current state of direct energy and power measurement techniques. In doing so, we aim to help future researchers choose an appropriate solution

for their analysis needs, as well as help guide future developments in hardware and software tools.

As we have shown, there is a great amount of diversity in energy and power measurement approaches, with many coexisting implementations and interfaces. Most of the current power measurement solutions were originally designed for administration and management purposes, typically measuring at the whole node level, and with a temporal resolution of 1 Sa/s. This accuracy can be unsuitable for fine-grained application consumption analysis. However, recent designs from both HPC vendors and research groups have achieved significantly higher temporal resolution (up to the order of 1 kSa/s) and better spatial granularity (with separate per-component measurement channels).

A number of challenges still stand in the way of ubiquitous high-fidelity consumption monitoring. The community depends on hardware vendors to embed accurate, fine-grained sensors into hardware components, relieving the need to design and install custom-tailored instrumentation hardware into their systems.

Additionally, to ensure the portability of instrumentation software, a greater degree of standardization of capabilities and interfaces is desirable. This would entail cooperation between vendors to ensure hardware interoperability of energy data acquisition techniques, much like what has already been done in the data center administration front with the IPMI and DCMI standards.

Another important consideration for future developments is scalability of both the instrumentation and acquisition processes. Instrumentation scalability would benefit from integration of reliable power measurement sensors exposing standard interfaces within hardware components: custom instrumentation systems can incur significant development and installation difficulty. Acquisition scalability will depend on the development of advanced collection and data processing systems, given that handling of fine-grained measurements at large scale will likely prove not to be an easy task in itself.

*This work was supported by the Spanish Ministry of Education and Science through the TIN2011-24598 project, Spanish CAPAP-H4 network, and NESUS IC1315 COST Action.*

*This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.*

## References

1. Advanced Micro Devices. AMD BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 00h-0Fh Processors, January 2013.
2. Pedro Alonso, Rosa M Badia, Jesus Labarta, Maria Barreda, Manuel F Dolz, Rafael Mayo, Enrique S Quintana-Orti, and Ruymán Reyes. Tools for power-energy modelling and analysis of parallel scientific applications. In *Parallel Processing (ICPP), 2012 41st International Conference on*, pages 420–429. IEEE, 2012. DOI: 10.1109/icpp.2012.57.
3. ARM Limited. ARM Energy Probe. <http://ds.arm.com/ds-5/optimize/arm-energy-probe/>.
4. D. Bedard, Min Yeol Lim, R. Fowler, and A. Porterfield. Powermon: Fine-grained and integrated power monitoring for commodity computer systems. In *IEEE South-*

- eastCon 2010 (SoutheastCon), Proceedings of the*, pages 479–484, March 2010. DOI: 10.1109/secon.2010.5453824.
5. Daniel Bedard, R Fowler, M Linn, and Allan Porterfield. Powermon 2: Fine-grained, integrated power measurement. *Renaissance Computing Institute, Tech. Rep. TR-09-04*, 2009.
  6. Shajulin Benedict. Energy-aware performance analysis methodologies for HPC architectures—an exploratory study. *Journal of Network and Computer Applications*, 35(6):1709 – 1719, 2012.
  7. Shajulin Benedict, Ventsislav Petkov, and Michael Gerndt. Periscope: An online-based distributed performance analysis tool. In *Tools for High Performance Computing 2009*, pages 1–16. Springer, 2010. DOI: 10.1007/978-3-642-11261-4\_1.
  8. S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci. A portable programming interface for performance evaluation on modern processors. *Int. J. High Perform. Comput. Appl.*, 14(3):189–204, August 2000.
  9. Martin Burtscher, Ivan Zecena, and Ziliang Zong. Measuring GPU power with the K20 built-in sensor. In *Proceedings of Workshop on General Purpose Processing Using GPUs, GPGPU-7*, pages 28:28–28:36, New York, NY, USA, 2014. ACM. DOI: 10.1145/2576779.2576783.
  10. APC by Schneider Electric. Metered-by-outlet rack pdus, March 2013.
  11. Alberto Cabrera, Francisco Almeida, Javier Arteaga, and Vicente Blanco. Measuring energy consumption using EML (Energy Measurement Library). *Computer Science-Research and Development*, pages 1–9, 2014.
  12. Alberto Cabrera, Francisco Almeida, and Vicente Blanco. Eml, an energy measurement library. *31st International Symposium on Computer Performance, Modeling, Measurements and Evaluation 2013: Student Poster Abstracts*, 2013.
  13. Jason Flinn and Mahadev Satyanarayanan. Powerscope: A tool for profiling the energy usage of mobile applications. In *WMCSA*, pages 2–10. IEEE Computer Society, 1999. DOI: 10.1109/mcsa.1999.749272.
  14. Rong Ge, Xizhou Feng, Shuaiwen Song, Hung-Ching Chang, Dong Li, and K.W. Cameron. Powerpack: Energy profiling and analysis of high-performance systems and applications. *Parallel and Distributed Systems, IEEE Transactions on*, 21(5):658–671, May 2010. DOI: 10.1109/TPDS.2009.76.
  15. Markus Geimer, Felix Wolf, Brian JN Wylie, Erika Ábrahám, Daniel Becker, and Bernd Mohr. The scalasca performance toolset architecture. *Concurrency and Computation: Practice and Experience*, 22(6):702–719, 2010. DOI: 10.1002/cpe.1556.
  16. D. Hackenberg, T. Ilsche, R. Schone, D. Molka, M. Schmidt, and W.E. Nagel. Power measurement techniques on standard compute nodes: A quantitative comparison. In *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*, pages 194–204, April 2013. DOI: 10.1109/ISPASS.2013.6557170.
  17. Daniel Hackenberg, Thomas Ilsche, Joseph Schuchart, Robert chöne, Wolfgang E. Nagel, Marc Simon, and iannis Georgiou. Hdeem: High definition energy efficiency monitoring. In *Proceedings of the 2Nd International Workshop on Energy Efficient supercomputing, E2SC ’14*, pages 1–10, Piscataway, NJ, USA, 2014. IEEE Press.

18. Nicole Hemsoth. Are Supercomputing's Elite Turning Backs on Accelerators? <http://www.hpcwire.com/2014/06/26/accelerators-hold/>, June 2014.
19. Chung-Hsing Hsu and S.W. Poole. Power measurement for high performance computing: State of the art. In *Green Computing Conference and Workshops (IGCC), 2011 International*, pages 1–6, July 2011. DOI: 10.1109/IGCC.2011.6008596.
20. Intel Corporation. Data Center Manageability Interface Specification, August 2011.
21. Intel Corporation. Intelligent Platform Management Interface Spec, October 2013.
22. Intel Corporation. Determining the Idle Power of an Intel<sup>®</sup> Xeon Phi<sup>™</sup> Coprocessor. <https://software.intel.com/en-us/articles/determining-the-idle-power-of-an-intel-xeon-phi-coprocessor>, June 2014.
23. Intel Corporation. Intel<sup>®</sup> Xeon Phi<sup>™</sup> Coprocessor Datasheet. <http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-coprocessor-datasheet.html>, April 2014.
24. Intel Corporation. *Intel<sup>®</sup> 64 and IA-32 Architectures Software Developer's Manual*. Number 253669-053US. January 2015.
25. Kiran Kasichayanula, Dan Terpstra, Piotr Luszczek, Stan Tomov, Shirley Moore, and Gregory D Peterson. Power aware computing on gpus. In *Application Accelerators in High Performance Computing (SAAHPC), 2012 Symposium on*, pages 64–73. IEEE, 2012. DOI: 10.1109/saahpc.2012.26.
26. kernel.org. Perf Wiki. [https://perf.wiki.kernel.org/index.php?title=Main\\_Page&oldid=3491](https://perf.wiki.kernel.org/index.php?title=Main_Page&oldid=3491), 2014.
27. Andreas Knüpfer, Christian Rössel, Dieter an Mey, Scott Biersdorff, Kai Diethelm, Dominic Eschweiler, Markus Geimer, Michael Gerndt, Daniel Lorenz, Allen Malony, et al. Score-p: A joint performance measurement run-time infrastructure for periscope, scalasca, tau, and vampir. In *Tools for High Performance Computing 2011*, pages 79–91. Springer, 2012. DOI: 10.1007/978-3-642-31476-6\_7.
28. James H Laros, Phil Pokorny, and David DeBonis. Powerinsight-a commodity power measurement capability. In *Green Computing Conference (IGCC), 2013 International*, pages 1–6. IEEE, 2013. DOI: 10.1109/igcc.2013.6604485.
29. Heike McCraw, James Ralph, Anthony Danalis, and Jack Dongarra. Power monitoring with papi for extreme scale architectures and dataflow-based programming models. 2014. DOI: 10.1109/cluster.2014.6968672.
30. John Mellor-Crummey, Robert J Fowler, Gabriel Marin, and Nathan Tallent. Hpcview: A tool for top-down analysis of node performance. *The Journal of Supercomputing*, 23(1):81–104, 2002. DOI: 10.1023/a:1015789220266.
31. Wolfgang E Nagel, Alfred Arnold, Michael Weber, Hans-Christian Hoppe, and Karl Solchenbach. Vampir: Visualization and analysis of mpi resources. 1996.
32. National Instruments Corporation. What Is Data Acquisition? <http://www.ni.com/data-acquisition/what-is/>.
33. Adel Nouredine, Romain Rouvoy, and Lionel Seinturier. A review of energy measurement approaches. *ACM SIGOPS Operating Systems Review*, 47(3):42–49, 2013. DOI: 10.1145/2553070.2553077.

34. NVIDIA Corporation. NVML API Reference Guide, March 2014.
35. P3 International. Kill A Watt product page. <http://www.p3international.com/products/p4400.html>.
36. Vincent Pillet, Jesús Labarta, Toni Cortes, and Sergi Girona. Paraver: A tool to visualize and analyze parallel code. In *Proceedings of WoTUG-18: Transputer and occam Developments*, volume 44, pages 17–31. mar, 1995.
37. Sebastian Ryffel. Lea<sup>2</sup>p: The linux energy attribution and accounting platform. Master’s thesis, Swiss Federal Institute of Technology, 2009.
38. Sandia National Laboratories. High performance computing power application programming interface (api) specification. <http://powerapi.sandia.gov/>, 2014.
39. Schleifenbauer. Schleifenbauer operation manual. [http://schleifenbauer.eu/dynamisch/bibliothek/16\\_0\\_NL\\_usermanual\\_v2.1.pdf](http://schleifenbauer.eu/dynamisch/bibliothek/16_0_NL_usermanual_v2.1.pdf).
40. Alain Schuermans. Schleifenbauer products by, March 2012.
41. Martin Schulz, Jim Galarowicz, Don Maghrak, William Hachfeld, David Montoya, and Scott Cranford. Open— speedshop: An open source infrastructure for parallel performance analysis. *Scientific Programming*, 16(2-3):105–121, 2008. DOI: 10.1155/2008/713705.
42. Sameer S Shende and Allen D Malony. The tau parallel performance system. *International Journal of High Performance Computing Applications*, 20(2):287–311, 2006. DOI: 10.1177/1094342006064482.
43. Thanos Stathopoulos, Dustin McIntire, and William J. Kaiser. The energy endoscope: Real-time detailed energy accounting for wireless sensor nodes. In *IPSN*, pages 383–394. IEEE Computer Society, 2008. DOI: 10.1109/ipsn.2008.36.
44. Luís Taniça, Aleksandar Ilic, Pedro Tomás, and Leonel Sousa. Schedmon: A performance and energy monitoring tool for modern multi-cores. In *Euro-Par 2014: Parallel Processing Workshops*, pages 230–241. Springer, 2014. DOI: 10.1007/978-3-319-14313-2\_20.
45. Valerie E Taylor, Xingfu Wu, Jonathan Geisler, Xin Li, Zhiling n, Rick Stevens, Mark Hereld, and Ivan R Judson. Prophecy: An infrastructure for analyzing and modeling the performance parallel and distributed applications. In *High-Performance Distributed Computing, 2000. Proceedings. The nth International Symposium on*, pages 302–303. IEEE, 2000. DOI: 10.1109/hpdc.2000.868668.
46. J. Treibig, G. Hager, and G. Wellein. Likwid: A lightweight performance-oriented tool suite for x86 multicore environments. In *Proceedings of PSTI2010, the First International Workshop on Parallel Software Tools and Tool Infrastructures*, San Diego CA, 2010.
47. Unified EFI, Inc. Advanced Configuration and Power Interface Specification, July 2014.
48. Virtual Institute High Productivity Supercomputing. Score-E. <http://www.vi-hps.org/projects/score-e/>, 2013.
49. Watt’s Up Meters. Operators Manual. [https://www.wattsupmeters.com/secure/downloads/manual\\_rev\\_9\\_corded0812.pdf](https://www.wattsupmeters.com/secure/downloads/manual_rev_9_corded0812.pdf).
50. Watt’s Up Meters. Watt’s Up product page. <https://www.wattsupmeters.com/>.
51. V. Weaver, M. Johnson, K. Kasichayanula, J. Ralph, P. Luszczek, D. Terpstra, and S. Moore. Measuring energy and power with papi. In *International Workshop on Power-Aware Systems and Architectures*, Pittsburgh, PA, September 2012. DOI: 10.1109/icppw.2012.39.

52. Vincent M Weaver, Daniel Terpstra, Heike McCraw, Matt Johnson, Kiran Kasichayanula, James Ralph, John Nelson, Philip Mucci, Tushar Mohan, and Shirley Moore. Papi 5: Measuring power, energy, and the cloud. In *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*, pages 124–125. IEEE, 2013. DOI: 10.1109/ispas.2013.6557155.
53. Xingfu Wu, Charles Lively, Valerie Taylor, Hung-Ching Chang, Chun-Yi Su, Kirk Cameron, Shirley Moore, Daniel Terpstra, and Vince Weaver. Mummi: multiple metrics modeling infrastructure. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference on*, pages 289–295. IEEE, 2013. DOI: 10.1109/snpsd.2013.73.
54. Yokogawa Electric Corporation. Power Analyzers. <http://tmi.yokogawa.com/products/digital-power-analyzers/>.
55. K. Yoshii, K. Iskra, R. Gupta, P. Beckman, V. Vishwanath, Chenjie Yu, and S. Coghlan. Evaluating power-monitoring capabilities on ibm blue gene/p and blue gene/q. In *Cluster Computing (CLUSTER), 2012 IEEE International Conference on*, pages 36–44, September 2012. DOI: 10.1109/CLUSTER.2012.62.
56. ZES ZIMMER. Brochure LMG450. [http://www.zes.com/en/content/download/286/2473/file/lmg450\\_prospekt\\_1002\\_e.pdf](http://www.zes.com/en/content/download/286/2473/file/lmg450_prospekt_1002_e.pdf).
57. ZES ZIMMER Electronic Systems GmbH. Products: Precision Power Analyzer. <http://www.zes.com/en/Products/Precision-Power-Analyzer>.

*Received March 1, 2015.*