
Francisco Almeida\textsuperscript{1}, Javier Arteaga\textsuperscript{1}, Vicente Blanco\textsuperscript{1}, Alberto Cabrera\textsuperscript{2}

© 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

\textsuperscript{1}Universidad de La Laguna, Santa Cruz de Tenerife, Spain

\textsuperscript{2}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

CPU 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.
(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

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

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 \texttt{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 Prophesy [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.

\texttt{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

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

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


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-


Supercomputing Frontiers and Innovations


Received March 1, 2015.