Assessing Power Monitoring Approaches for Energy and Power Analysis of Computers

Mohammed El Mehdi Diouri, Manuel Francisco Dolz, Olivier Glück, Laurent Lefèvre, Pedro Alonso, Sandra Catalán, Rafael Mayo, Enrique Salvador Quintana-Ortí

Assessing Power Monitoring Approaches for Energy and Power Analysis of Computers

Mohammed El Mehdi Diouri[a], Manuel F. Dolz[b], Olivier Glück[c], Laurent Lefèvre[c], Pedro Alonso[d], Sandra Catalán[e], Rafael Mayo[e], Enrique S. Quintana-Ortí[e]

[a] *Institut supérieur du Génie Appliqué (IGA), 20.300–Casablanca, Morocco*
[b] *Dept. of Informatics, University of Hamburg, 20.146–Hamburg, Germany*
[c] *INRIA Avalon Team, LIP Laboratory, ENS Lyon, Université de Lyon, 69.364–Lyon cedex 07, France*
[d] *Depto. de Sistemas Informáticos y Computación, Universitat Politècnica de València, 46.022–Valencia, Spain*
[e] *Depto. de Ingeniería y Ciencia de Computadores, Universitat Jaume I, 12.071–Castellón, Spain*

Abstract

Large-scale distributed systems (e.g., datacenters, HPC systems, clouds, large-scale networks, etc.) consume and will consume enormous amounts of energy. Therefore, accurately monitoring the power dissipation and energy consumption of these systems is more unavoidable. The main novelty of this contribution is the analysis and evaluation of different external and internal power monitoring devices tested using two different computing systems, a server and a desktop machine. Furthermore, we provide experimental results for a variety of benchmarks which intensively exercise the main components (CPU, Memory, HDDs, and NICs) of the target platforms to validate the accuracy of the equipment in terms of power dissipation and energy consumption. On the other hand, we also evaluate three different power measurement interfaces available on current architecture generations. Thanks to the high sampling rate and to the different measured lines, the internal wattmeters allow an improved visualization of some power fluctuations. However, a high sampling rate is not always necessary to understand the evolution of the power consumption during the execution of a benchmark.

Keywords: Wattmeters, power measurement interfaces, energy and power analysis, power profiling

1. Introduction

For decades, the computer science research community exclusively focused on performance, which resulted in highly powerful, but in turn, low energy-efficient systems with a very high total cost of ownership (TCO) [1]. Yet, in recent years, the HPC community has acknowledged that the energy efficiency of HPC systems is a major concern in designing future Exascale systems [2, 3].

Nowadays there exist intensive efforts that pursue the design of energy-efficient supercomputers. Hardware provides part of the solution by unceasingly exposing more energy-efficient devices which also provide abilities that current operating systems can successfully leverage to save energy [4]. Mechanisms such as Dynamic Voltage Scaling (DVFS) or P-state management have also been used to develop power-aware user-level software [4, 5, 6].

The Green500 list seeks to raise the awareness of power and energy consumption in supercomputing by reporting the power dissipation and energy efficiency of large-scale HPC facilities. Even the Top500 list is currently tracking the power draw by today's most powerful HPC systems, ranking their efficiency

Email addresses: mehdi.diouri@iga-casablanca.ma (Mohammed El Mehdi Diouri), manuel.dolz@informatik.uni-hamburg.de (Manuel F. Dolz), olivier.gluck@ens-lyon.fr (Olivier Glück), laurent.lefevre@ens-lyon.fr (Laurent Lefèvre), palonso@dsic.upv.es (Pedro Alonso), catalans@icc.uji.es (Sandra Catalán), mayo@icc.uji.es (Rafael Mayo), quintana@icc.uji.es (Enrique S. Quintana-Ortí)

in terms of GFLOPS per Watt [7]. The metric used to build the Green500 List is limited by the use of LINPACK benchmark for performance/energy measurement, because this test primarily stresses the CPU component of an HPC system [7]. Clearly, a more elaborate figure is needed to inspect and understand all the power sinks in computing resources of large scale distributed systems [8]. Some proposals obtain the power dissipation of different parts of the node (CPU, memory, disks, fans, ...) but they are circumscribed to either one [9] or a few benchmarks types [10]. Furthermore, the most prevailing infrastructure comprises only an external wattmeter [1, 11]. Another issue arises on how to process the power/energy samples due to the high variability to which they are subject to. Only some contributions deepen to obtain, e.g., a statistical regression on the samples to produce reliable results [12]. Given the foregoing, we contribute in this paper with the following:

- We target five wattmeters (external and internal) to analyze two different systems representative of current general-purpose platforms: a desktop computer and a server node. Our purpose is to evaluate different power measurement sources/wattmeters, regardless of the power nature (average or instantaneous).
- In order to evaluate the precision of the data acquisition devices, we use and deploy five different types of benchmarks which stress different components of the system.
- We use a framework of easy-to-use and scalable tools to analyze the power variability and the energy consumption which comprises, among others, the `pmlib` library to interact with power measurement units [13].
- Results are displayed using *boxplots*. This graphically depicts five-number groups of data that illustrate the variability of the samples which may be highly affected by environmental conditions such as temperature fluctuation.
- We analyze three non-based wattmeter interfaces in two different architectures and compare them with real wattmeters.

While some of these results were already presented in [14], the related work in this field and the analysis of integrated power measurement interfaces and their respective comparison with real wattmeters are new contributions specific to this paper.

The rest of the paper is structured as follows. In the next section we present some related works and discuss the different approaches used there to measure power. Section 3 describes in detail the experiment setup. The energy consumption obtained with the wattmeters using all the benchmarks is analyzed in Section 4. In Section 5, we discuss in more detail these results by processing samples under a statistical model based on boxplots. Section 6 performs an additional analysis by varying the wattmeters sampling rate. Section 7 analyzes three integrate power measurement interfaces and compare them with real wattmeters. The paper is closed with a section that contains a discussion, conclusions and the future work.

2. Related Work

In order to measure the power and energy consumption of computing systems, two methods are generally used: either hardware devices are connected inside or outside each machine, or energy-related counters integrated in some hardware architectures are leveraged.

In [15], the authors monitor the energy consumption of servers thanks to four different power measurement devices:

- PDU[1] Eaton[2] ;

[1]PDU: Power Distribution Unit
[2]Eaton: `http://www.eaton.fr`

- PDU Schleifenbauer[3] ;
- OmegaWatt wattmeter [4] ;
- the Dell supervision card: the Intelligent Platform Management Interface [16] is a set of interface specifications to monitor the energy consumption and the temperature inside the machine using supervision cards proposed by several manufacturers (Dell, Intel, HP and Nec).

To measure the power and energy consumption, other similar devices are also used in the literature:

- WattsUp wattmeter used in [17];
- the IBM PowerExecutive[5] supervision card available in the IBM BladeCenter servers.

These power measurement devices can be mainly characterized according to four criteria: nature, location, accuracy and sampling rate. OmegaWatt, WattsUp and the PDUs measure the active power, that is to say, they calculate an averaged power value over the energy consumption measured in a period of time, the other devices measure instead the instantaneous electric power that is obtained by testing the current and voltage of the machine. The Dell (also used in [18]) and the IBM PowerExecutive supervision cards measure the power dissipation inside the machine (downstream the power supply unit) unlike other devices which sample externally (upstream the power supply unit). This means that the power measurements collected by such supervision cards use direct current (DC) and do not take into account the power dissipated by the power supply unit of the computing system. In contrast, the other three devices measure the overall AC power energizing to the machine, and therefore take into account the power supply unit and all the hardware in the machine. Thus, a disadvantage of IPMI and PowerExecutive is that these devices do not consider the power dissipated by the power supply unit of the server that can represent a significant proportion of the total power of the server (about 20 %).

Although supervision cards offer the advantage of monitoring the dissipated power without having to plug additional equipment, this solution turns out to be inaccurate and not fine-grained enough (as they offer instantaneous power at a sampling rate relatively low). Indeed, the Dell supervision card used in [15] exhibits a measurement uncertainty of ± 7 W which is too large to capture power variations of few watts, while the wattmeters OmegaWatt, WattsUp and the PDU Schleifenbauer present a precision of about ± 0.1 W. The Eaton PDU has an intermediate precision of ± 1 W. In addition, both supervision cards present a sampling rate relatively low (1 measurement every 5 seconds for the Dell supervision card). Although their measurement accuracies are rather high, the other devices do not provide 1 sample per second in the best case. By measuring the power every 3 seconds, the PDUs Schleifenbauer and Eaton are therefore not suitable for applications or tasks that last only a few seconds. In contrast, OmegaWatt and WattsUp provide 1 averaged sample per second thus reflecting the power fluctuations for such period of time.

There are other equipments that operate at higher frequencies (more than 1 S/s). Indeed, internal wattmeters like PowerMon2 [19] are able to measure at very high frequencies the instantaneous internal power dissipation of the different ATX power lines (3.3 V, 5 V and 12 V) that compose the power supply unit of the machine and energize the motherboard and various hardware components. To measure the instantaneous power of an ATX power line, PowerMon2 measures the current value and multiplies it by the voltage which is constant (3 V, 5 V or 12 V). When PowerMon2 is used to obtain the power consumed by a single power line, the sampling rate can reach 3072 S/s.

The authors of the paper [20] used three other internal wattmeters that are similar to PowerMon2. Among them, they used a NI9205 card attached to a NIcDAQ-9178 chassis, which can reach 7,000 S/s to measure the power dissipation of a single ATX line. The two other internal wattmeters, DCM and DC2M were constructed by the authors and measure the instantaneous power at 28 S/s and 100 S/s, respectively.

[3] Schleifenbauer: `www.schleifenbauer.eu`

[4] Omegawatt: `http://www.omegawatt.fr`

[5] IBM PowerExecutive: `http://www-03.ibm.com/systems/management/director/about/director52/extensions/powerexec.html`

Table 1: Classification of the different power measurement devices

Category	Power	Location	Accuracy	Sample Rate
PDUs (Eaton and Shleifenbauer)	average	external	intermediate	low
External wattmeters (OmegaWatt and WattsUp)	average	external	high	intermediate
Internal wattmeters (PowerMon2, NI, DCM and DC2M)	instantaneous	internal	high	high
Energy counters (RAPL, NVML)	instantaneous	internal	high	high
Supervision cards (IPMI and IBM PowerExecutive)	instantaneous	internal	low	low

The drawback of these internal wattmeters is that they are not easy to connect and they are only suitable for standard power supply units, that is to say, with power lines complying with the ATX standard. Therefore, their deployment is impossible at a very large-scale.

The authors of [21, 22] use the RAPL[6] energy counters available in the Intel Sandy Bridge microarchitecture. Other analogous energy counters exist on AMD processors and are used, for example, in the BlueWaters supercomputer[7]. Recent NVIDIA GPUs can report power usage via the NVIDIA Management Library (NVML). Indeed, the authors of [23] use NVML to monitor the power and energy consumption of applications. The RAPL energy counters are able to measure the instantaneous power of these Intel processors at a sampling rate equal to 1,000 S/s and with a fairly high accuracy (less than 1 W). They are interesting for monitoring the power consumed by applications executed on large-scale distributed systems, since they do not require any hardware to plug and measure the instantaneous power with a high precision and a high sampling rate. However, they only measure the power consumed by the processors (and not the whole machine). Moreover, they are available only in the Intel Sandy Bridge microarchitecture processors. In addition, it is an intrusive solution since it is often necessary to access the energy counters in order to read the power values, to this end, it is necessary to run a program that collects the power measurements. Thus presents two drawbacks: we must therefore ensure that the user has sufficient rights to run this program; and, the execution of such a program can distort the power measurement.

In summary, we can classify all these power measurement devices into five categories presented in Table 1. External wattmeters and PDUs measure and display the active (or average) power dissipation which is obtained by dividing by a time period T (related to their measurement sampling rate) the energy consumed during that time. The energy consumption is approximated by an integration over a more or less high number of instantaneous power measurements. The internal wattmeters measure the instantaneous power dissipation of the ATX lines used to power the various components of the machine. As these wattmeters use a direct current and the voltage across each ATX line is known (3.3 V, 5 V or 12 V), they only need to measure the current powering each ATX line. The instantaneous power is obtained by multiplying the current by the voltage across the line ATX. Regarding the energy counters (RAPL) and the supervision cards (from Dell and IBM), we have no information about the measurement principle.

Each of these research works leverage a number of wattmeters in order to measure the energy consumed by applications or the energy saved with a proposed solution. However, they do not justify the choice of the wattmeters they use: are the accuracy and the sampling rate of the used wattmeters sufficiently high to consider that the power and energy measurements are reliable enough? Indeed, these research works do not allow us to know whether it is better to use a category of power measurement devices rather than another depending on the specific purpose of the research.

Moreover, studies that try to evaluate energy savings before and after a defined intervention use the same wattmeter. Therefore, the accuracy may lose relevance before and after the intervention. In our paper, we focus more in the appropriateness of the wattmeters chosen with respect to the power nature, the sampling rate and the accuracy that characterize the measurement device. On the one hand, most of studies/publications do not give detailed information/characteristics of the wattmeters used and, on the

[6]RAPL: Running Average Power Limit

[7]BlueWaters project: `https://bluewaters.ncsa.illinois.edu`

other hand, they do not justify the type of wattmeters leveraged (internal, external, integrated, etc.), which is an important fact when accounting the energy consumption of applications or the impact of different energy saving techniques among the different components of the target architecture.

In our study, we will focus on comparing the external and internal wattmeters to determine which are the most appropriate to measure the power and energy consumption of applications. We will also study some non-wattmeters solutions for measuring the power dissipation and compare them to the power measurements obtained by the external and internal wattmeters.

3. Experimental Setup

This section describes the power measurement devices and framework, the target platforms, and the benchmarks used in our evaluation.

Power Measurement Devices. We classify the measurement devices into two main types: external AC meters, which are directly attached to the wires that connect the electric socket to the computer Power Supply Unit (PSU); and the internal DC meters, responsible for measuring the output wires leaving the PSU that energize the components of the mainboard. Table 2 presents in detail the specifications of the wattmeters that we used.

Table 2: Specifications of the wattmeters.

	External AC		Internal DC		
Wattmeter	OmegaWatt	WattsUp	PowerMon2	NI	DCM
Manufactured by	OmegaWatt[a]	WattsUp?[b]	RENCI iLab[c]	National Instruments[d]	Universitat Jaume I
# Channels	6	1	8	32	12
Channel type	Standard power PC cord	Standard power PC cord	All ATX-related lines (3.3 V, 5 V and 12 V)[e]	12 V ATX-related lines	12 V ATX-related lines
Power nature	Average	Average	Instantaneous	Instantaneous	Instantaneous
Microcontroller	-	-	Atmel ATmega16	NI9205 NIcDAQ-9178	*Microchip* PIC 18
Power sensors	-	-	*Analog Devices ADM1191* resistors	*LEM HXS 20-NP* transducers	*LEM HXS 20-NP* transducers
Sampling Rate (S/s) per channel	1	1	1,024[f]	1,000	28
Accuracy	$< \pm 1\%$	$< \pm 1.5\%$	$\pm 5\%$	$\pm 1\%$	$\pm 1\%$
Interface	RS232	USB	USB	USB	RS232
Price	600 €	200 €	125 €	2,700 €	Not commercialized

[a] OmegaWatt: `http://www.omegawatt.fr/`

[b] WattsUp: `https://www.wattsupmeters.com/`

[c] PowerMon2: `http://ilab.renci.org/powermon`

[d] NI: `http://www.ni.com/`

[e] 3.3 V and 5 V lines measure the power dissipation of some components of the mainboard (GPUs, NICs, etc.) while 12 V lines measure the power dissipation of the CPUs and fans.

[f] For PowerMon2, we used only 100 S/s.

Power measurement framework. The `pmlib` software package is developed and maintained by the HPC&A research group of the Universitat Jaume I to investigate power usage of HPC applications. The current implementation of this package provides an interface to utilize all the above-mentioned wattmeters and a number of tracing tools. Power measurement is controlled by the application using a collection of routines that allow the user to query information on the power measurement units, create counters associated to a device where power data is stored, start/continue/terminate power sampling, etc. All this information is managed by the `pmlib` server, which is in charge of acquiring data from the devices and sending back the appropriate answers to the invoking client application via the proper `pmlib` routines (see Figure 1).

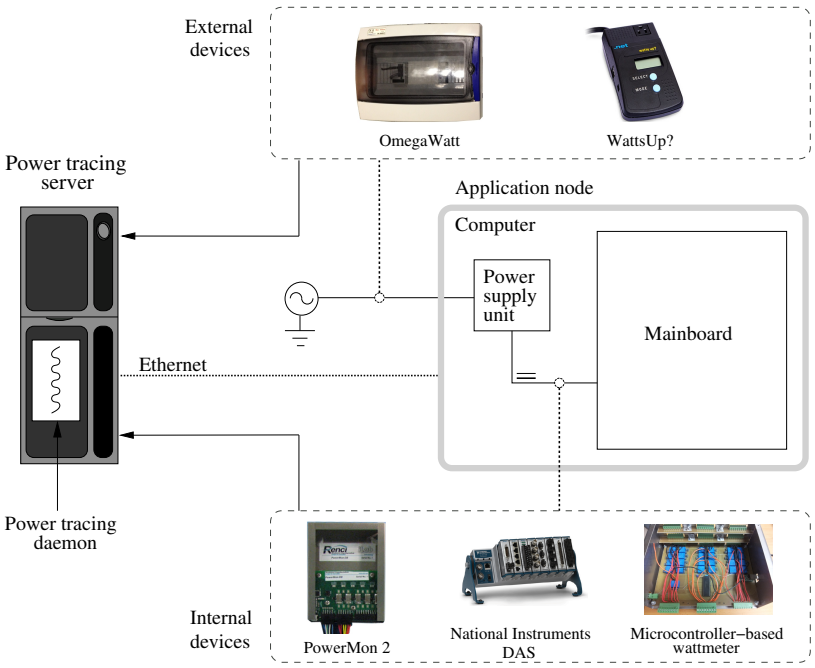

Figure 1: Single-node application system and sampling points for external and internal wattmeters.

Target platforms. The analysis and evaluation has been carried out on two different platforms: a desktop and a server node. The desktop computer is equipped with an Intel Ivy Bridge Core i7-3770K processor (4 cores) running at 3.50 GHz, 16 GB of RAM and a 160 GB Seagate Barracuda 7200.10 HDD. We will denote this machine as INTEL_DESKTOP. The server machine, referred to as AMD_SERVER, integrates 4 AMD Opteron 6172 of 12 cores (total of 48 cores) running at 2.10 GHz, 256 GB of RAM and a 1 TB Seagate Constellation ES HDD.

Benchmarks. To evaluate the energy and power behavior, we run different types of workloads to intensively exercise the use of specific parts of the platforms. CPU, memory, NICs and HDDs are the main components that we stressed in our experiments. To achieve this purpose, we selected the following specific benchmarks:

- `idle`: This benchmark employs the `sleep` POSIX routine[8] to suspend processor activity, thus generating idle periods that let the hardware promote the cores to low-power states, also known as C-states[9].
- `iperf`[10]: This tool performs network throughput measurements. It can test either TCP or UDP throughput. To perform an `iperf` test, we set both a server and a client. Since the package features a large number of options, we only measure this tool running it as a TCP client.
- `hdparm`[11]: This application provides a command line interface to various kernel interfaces supported by the Linux SATA/PATA/SAS *libATA* subsystem. We set the `-t` option to perform timings on device reads and to stress the HDD.
- `cpuburn`[12]: This benchmark heats up a CPU to the maximum operating temperature that is achievable using ordinary software. We map `cpuburn` processes into specific cores in order to measure the power when different number of cores are used.

[8] `sleep`: `http://linux.die.net/man/3/sleep`
[9] Advanced Configuration and Power Interface. Revision 5.0. `http://www.acpi.info/`
[10] `iperf`: `http://iperf.fr`
[11] `hdparm`: `http://linux.die.net/man/8/hdparm`
[12] `cpuburn`: `http://manpages.ubuntu.com/manpages/precise/man1/cpuburn.1.html`

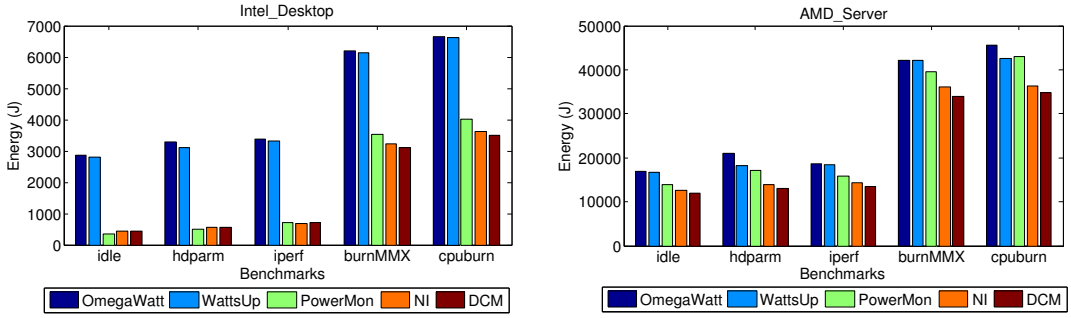

Figure 2: Extra energy consumption of the benchmarks measured with the wattmeters.

- `burnMMX`[13]: This program, included in the `cpuburn` package, specifically stresses the cache and memory interfaces. `burnMMX` processes are mapped into specific cores in order to measure the power when different number of cores are used.

4. Energy Consumption Analysis

In this section we analyze the variability and accuracy of the external and internal wattmeters using all benchmarks introduced in the previous section on the two selected machines: Intel_Desktop and AMD_Server.

Figure 2 presents the energy consumption attained from the execution of the benchmarks, during 60 seconds each, on both platforms. Bars labeled as `idle` represent the energy consumption when leaving the platform doing nothing for a full minute. Bars `hdparm` and `iperf` report the energy registered when the HDDs and the NICs are, respectively, stressed. For the `burnMMX` and `cpuburn` benchmarks we heat all the cores by mapping one process per core. Specifically, we use the `taskset` command to bind processes to the cores of the machines. It is important to note that, before taking measures with these two benchmarks, we warm up the machines by running the corresponding tests up to the maximum temperature ($\approx$ 10 min.). We represent the aggregated energy consumption calculated as the addition of energy measurements for all the 12 V lines. Even though PowerMon2 can also measure the 3.3 V and 5 V lines, we rather prefer to account for the 12 V lines only, to provide a fair comparison with other internal wattmeters that are only able to measure a more limited number of lines.

Figure 2 shows that the energy consumptions registered with both external wattmeters (OmegaWatt and WattsUp) are very similar except for `hdparm` on Intel_Desktop and on AMD_Server and for `cpuburn` on AMD_Server. Such inaccuracies may come from the power variation occurring in these three scenarios. The analysis of the power profiles in Section 6.1 will allow us to confirm this power variation.

We observe a different scenario for the internal wattmeters. Indeed, in the light of the energy measured by these devices, it is easy to observe that the values provided by PowerMon2 are almost always higher than those registered by NI and DCM. These variations are mainly due to the use of different components to measure voltage/current of the internal wires. The differences between internal wattmeters, sometimes significant, are also due to the large amount of samples per second taken from the lines. DCM works at 28 samples per second (S/s), but PowerMon2 and NI sample at 100 S/s and 1,000 S/s, respectively, thus rapid power fluctuations are not reflected by low sampling devices.

In contrast to the external wattmeters, the internal devices measure the instantaneous power dissipation downstream of the PSU. Thus, the internal measurements do not account for the PSU and other components like HDDs and/or GPUs. This explains why the energy consumption registered by the internal wattmeters is lower than that obtained with the external wattmeters, as it is easily observed in the graph for all the benchmarks regardless of the target machine. Providing an internal measurement in addition to an external

[13] `burnMMX`: `http://pl.digipedia.org/man/doc/view/burnMMX.1`

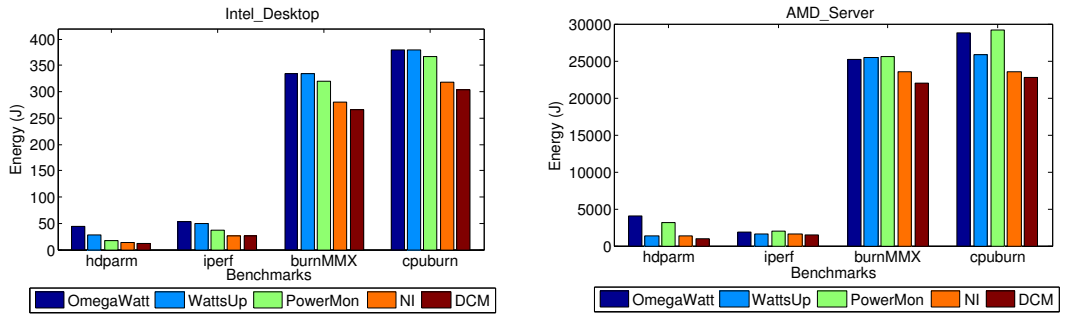

Figure 3: Extra energy consumption of the benchmarks (i.e. without the `idle` part) measured with the wattmeters.

measurement can inform us about the inefficiency of the PSU, which may be different from a node to another. This difference between external and internal wattmeters seems to be relatively less significant for AMD_Server than in the Intel_Desktop. This is mainly due to the fact that AMD_Server is composed of 48 cores and several fans whose power dissipation is included in the internal power measurements and represents a significant part of the total power of the machine.

Figure 3 shows the extra energy consumption corresponding to the same experiments reported in Figure 2, but removing the energy consumption of the `idle` benchmark from the energy consumption of each benchmark. By removing the `idle` part from the energy consumption, the purpose of Figure 3 is to check whether the difference between the wattmeters observed in Figure 2 in terms of energy consumption is linked to a problem of calibration experienced by the used wattmeters. While we could expect small or no differences between the different measurement devices, Figure 3 shows that some differences remain in terms of extra energy consumption.

5. Power Consumption Analysis

In this section, we analyze the power measurements made by the wattmeters. Throughout this analysis, we first show the variability of the power measurements for different workloads running on the two different target platforms, Intel_Desktop and AMD_Server. By comparing distinct wattmeters and different machines, we intend to identify the impact of measuring the power dissipation internally or externally with a variable measurement frequency.

At this point, we consider the following scenarios: the target machine is `idle` (Figure 4), one core of the machine runs `hdparm` (Figure 5), and all cores of the machine run `cpuburn` (Figure 6). We execute each benchmark on the two different target machines, and measure the power during 60 seconds using the external wattmeters OmegaWatt and WattsUp, and the internal wattmeters PowerMon2, NI and DCM.

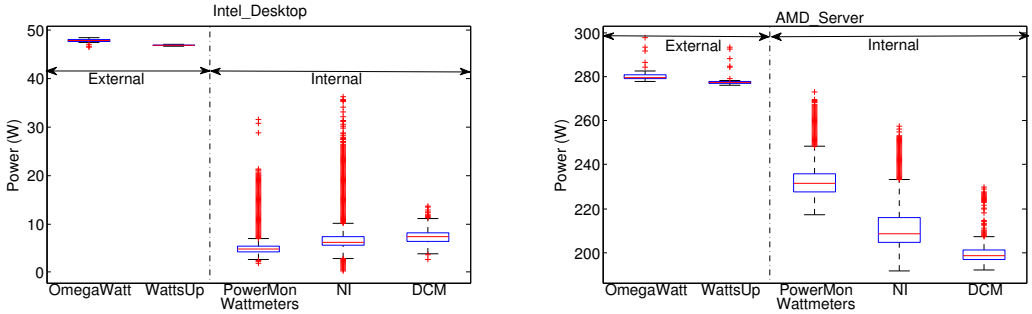

Figure 4: Dispersion of the power dissipation measurements for benchmark `idle`.

Figures 4, 5 and 6 present boxplots showing the distribution of the power dissipation measurements from this experiment. Each boxplot graphically depicts groups of numerical data using a five-number summary:

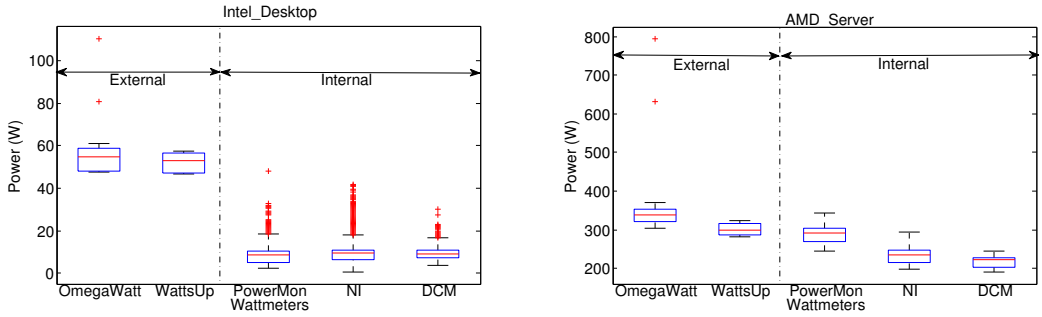

Figure 5: Dispersion of the power dissipation measurements for benchmark `hdparm`.

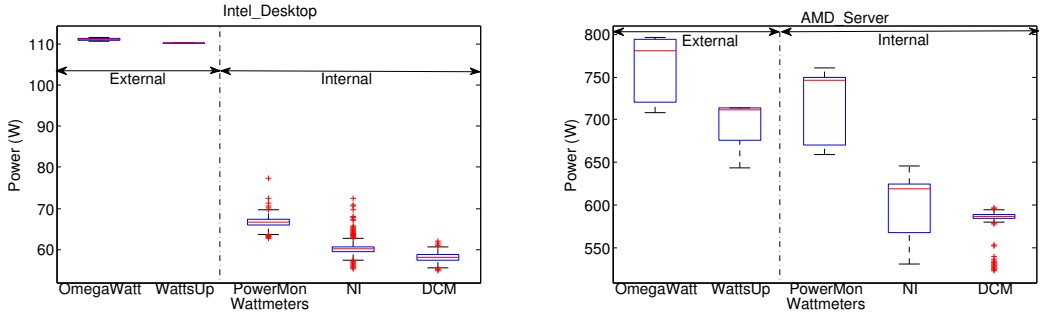

Figure 6: Dispersion of the power dissipation measurements for benchmark `cpuburn`.

the smallest observation (sample minimum), lower quartile (Q1), median (Q2), upper quartile (Q3), and largest observation (sample maximum). The plots also indicate which measurements, if any, should be considered outliers. Since internal measurements do not take into account the inefficiencies of the PSU and even some components, the corresponding boxplots are, in most cases, below those corresponding to the external wattmeters.

As in the previous experiment, the boxplots for INTEL_DESKTOP show that the variability of the power measurements obtained from external devices are similar, independently of the benchmark considered. This also holds for the internal power measurements, except for `cpuburn` where POWERMON2 provides slightly higher power measurements compared with the other internal wattmeters. For AMD_SERVER, the differences are more visible: neither the external nor the internal power measurements are concordant among themselves, particularly for the `cpuburn` benchmark. Indeed, for all the benchmarks running on AMD_SERVER, power measurements from POWERMON2 are always higher than those provided by the other internal wattmeters. For the external wattmeters, OMEGAWATT provides higher power measurements than WATTSUP, especially for the `cpuburn` and `hdparm` benchmarks. These differences may be due to the fact that these wattmeters are made of different components offering different degrees of accuracy, also their different nature (averaged for external wattmeters and instantaneous for internal wattmeters).

What is even more mysterious is that when benchmark `cpuburn` is run on AMD_SERVER, POWERMON2 registers higher power measurements than some external wattmeters, even when POWERMON2 power measurements do not take into account the PSU inefficiencies and some internal components. This also holds for some power samples when the `hdparm` runs on AMD_SERVER. We relate this behavior to the accuracy of POWERMON2, which is the least accurate internal wattmeter among the selected set of internal wattmeters.

Furthermore, in Figure 4 we notice that the results captured with the internal devices are more dispersed and generate many outliers than those obtained with the external wattmeters. This is certainly due to the high sampling frequency of internal wattmeters that allow to measure some power fluctuations that the external wattmeters do not reflect, i.e, averaged measurements of the external wattmeters hide the major part of these outliers compared with instantaneous measurements of internal wattmeters. We notice from Figure 5 that, contrary to WATTSUP, OMEGAWATT registers some strange outliers while measuring the

`hdparm` benchmark. This can be due to the high variability of the power dissipation during the execution of the `hdparm` benchmark causing that OmegaWatt is occasionally not accurate. Moreover, Figure 6 shows that when `cpuburn` runs on AMD_Server, the power measurements are highly dispersed: a variability of almost 100 W for OmegaWatt, PowerMon2 and NI; and about 70 W for WattsUp and for DCM. This high power variability for the `cpuburn` on AMD_Server suggests that this benchmark is fluctuating too much. By analyzing the power profiles in Section 6, we expect to confirm the fluctuation in `hdparm` and `cpuburn` when they are executed on AMD_Server.

6. Power Profile Analysis

In this section we extend the behavior analysis of the set of benchmarks on the target machines. In particular, we show power profiles, depict the impact of reducing the sample rate of wattmeters and, finally, analyze the power transferred by each power line using PowerMon2. Analyzing the power profiles of an application is a step beyond studying its energy consumption and power dissipation.

6.1. Analysis of the power profiles

Let us start by considering Figure 7 and 8 which, respectively, represent the profiles on Intel_Desktop and AMD_Server of 20 seconds of the execution of benchmarks `idle`, `hdparm`, and `cpuburn` obtained from both external and internal wattmeters. The plot in the left-hand side of Figure 7 depicts the power trace when OmegaWatt and PowerMon2 (sampling only the 12 V lines) are simultaneously connected to Intel_Desktop; the plot in the right shows analogous information obtained with WattsUp and NI. The aim of this comparison is to inspect the same scenario with two different power measurement device configurations. The results show that the power profile from the external wattmeters are nearly the same; however, for the internal devices some variations exist. The first slight drop of power that PowerMon2 provides when compared with the NI is remarkable; this observation was already made in the experiments of Sections 4 and 5. Apart from that, we observe much more noise with NI; nevertheless this behavior is due to the high sampling rate of this device. These comparisons demonstrate that it is easy to observe how these two scenarios, using different wattmeters, provide reliable power profiles. Our measurements could be displaced along time due to the high frequency. However, by plotting power profiles from internal wattmeters, we are interested in analyzing the internal behavior of the applications: for example, to detect some special power increases that could be filtered by external wattmeters.

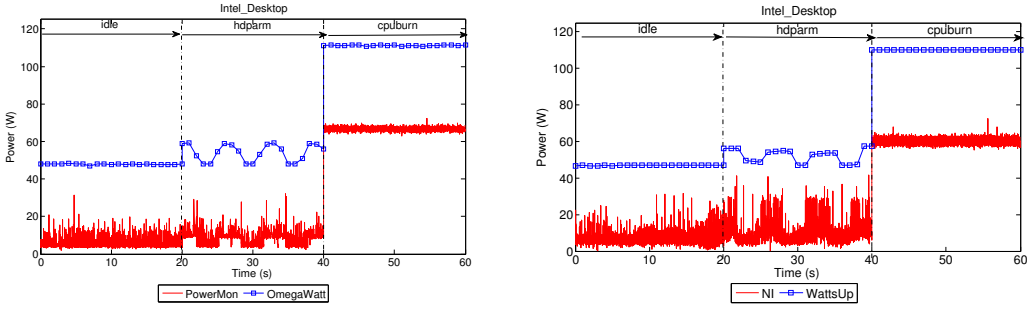

Figure 7: Power profiles obtained when running benchmarks `idle`, `hdparm` and `cpuburn` benchmarks on Intel_Desktop.

Focusing next in Figure 8, we observe how the external wattmeters OmegaWatt and WattsUp provide different power profiles for `hdparm` and, more acutely, `cpuburn`. These differences mainly come from the specifics of the devices and from potential environment changes (e.g., room temperature [24]), as these experiments could not be performed simultaneously. (We were not able to connect two external or internal wattmeters at the same time.) In [25], Patterson shows that without air cooling, increasing the room temperature from 20^o to 30^o results in an increase of a datacenter energy consumption evaluated at 2.5 %.

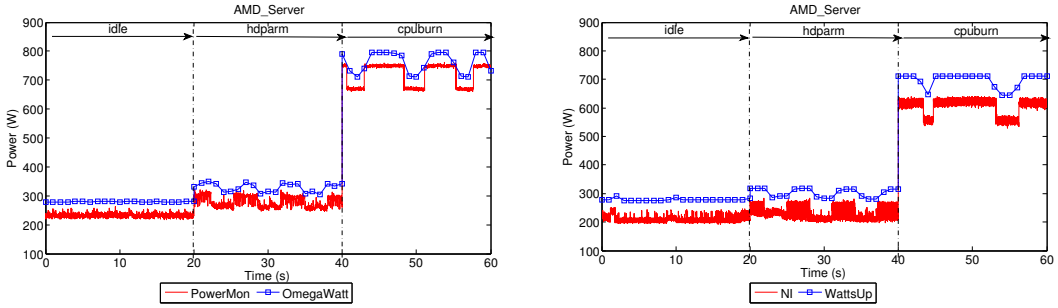

Figure 8: Power profiles obtained when running benchmarks `idle`, `hdparm` and `cpuburn` benchmarks on AMD_SERVER.

The same situation occurs for the internal wattmeters with the POWERMON2 profile being highly displaced from that obtained from NI.

We also highlight the spikes and drops observed in the power profile when running the `cpuburn` benchmark on AMD_SERVER. Remember that this platform contains 48 cores which we fully populated with this type of processes. We relate this behavior to the BIOS-mainboard settings and fans, that are constantly on and off in order to cool the machine and maintain the platform's temperature at a constant level when a threshold is reached.

6.2. Impact of the sample rate: the NI *case*

In this section, we investigate whether a very high sample rate (more than 100 S/s) produces power fluctuations that are not observed with a low sample rate (1 S/s). For this purpose, we measure the power dissipation with NI at 1,000 S/s during 30 seconds for `hdparm` and `cpuburn` on INTEL_DESKTOP and AMD_SERVER. Then, we reduce the sample rate by applying the formula:

$$S_j^r = r \frac{\sum_{i=1+1000(j-1)/r}^{\frac{1000j}{r}} (S_i^{1000})}{R}, \tag{1}$$

where r is the reduced sample rate, R is the original sample rate, S_i^{1000} is the i^{th} sample taken with 1,000 S/s and S_j^r is the j^{th} sample taken with r as sample rate.

Figure 9 shows the power profiles obtained with NI by reducing the sample rate from 1,000 S/s to 1 S/s for `hdparm` (left side) and `cpuburn` (right side), respectively, on INTEL_DESKTOP. From this figure, we notice that the noise induced by the high sample rate (more than 200 S/s) masks the spikes and drops of `hdparm` making them harder to perceive. However, a reduced sample rate (less than 50 S/s) hides some interesting power fluctuations, like the high spikes that we can observe just before each drop when the sample rate is 50 S/s. With respect to the `cpuburn` benchmark, we can observe that the power fluctuates between 57 W and 63 W when the sample rate is 1,000 S/s. The shape of the power profile becomes thinner when reducing the sampling rate. Below 50 S/s, we notice that the power profile is a constant line devoid of noise. Thus, for INTEL_DESKTOP, we may need a medium sample rate (between 50 S/s and 20 S/s) for a benchmark like `hdparm`, while a low sample rate (1 S/s) is enough to observe the power profile for a benchmark like `cpuburn`.

The same experiment was repeated in AMD_SERVER and results are given in Figure 10. In this case, we notice a behavior for `hdparm` similar to that observed on INTEL_DESKTOP. On the other hand, the behavior is different for AMD_SERVER for the `cpuburn` benchmark. Indeed, contrary to what we observed for `cpuburn` running on INTEL_DESKTOP, a medium sample rate (between 50 S/s and 200 S/s) helps to understand better the spikes and drops that appear when `cpuburn` is executed on this platform. With 1 S/s, we still perceive the spikes and drops, which confirms that they correspond to real power fluctuations on AMD_SERVER and are not simply due to the noise that may be generated by a high sampling rate.

In summary, measuring at a very high sample rate (500 S/s) is not always necessary for power profiling applications and may even cause some noise that masks the general shape of the power profile. The best sampling rate is not always the highest one, but the one that best enables to understand fluctuations. For

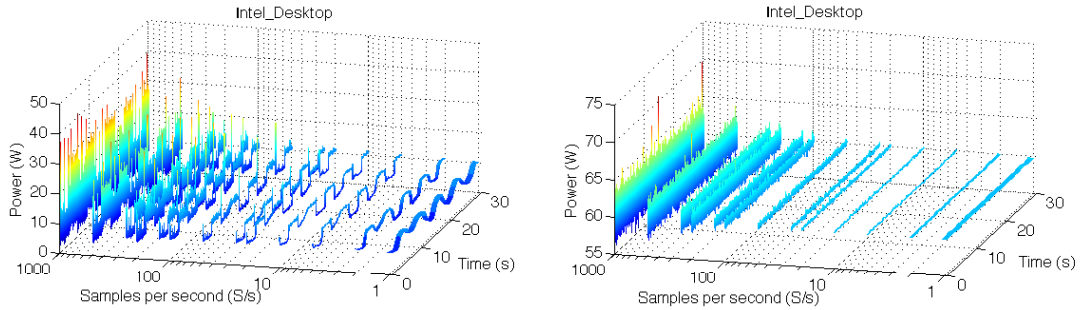

Figure 9: Power profiles obtained with NI for `hdparm` (left plot) and `cpuburn` (right plot) running on INTEL_DESKTOP with a configurable sample rate.

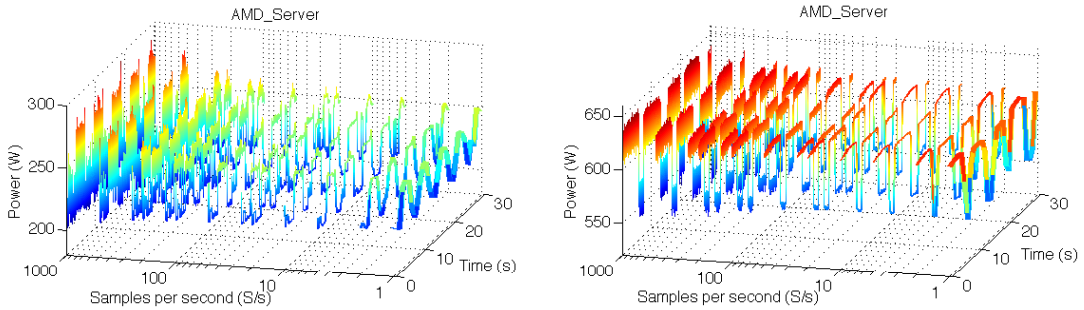

Figure 10: Power profiles obtained with NI for `hdparm` (left plot) and `cpuburn` (right plot) running on AMD_SERVER with a configurable sample rate.

this reason, determining the best sampling rate depends not only on the used measurement device but also on the considered application and the hardware platform. One of our future work is to help the users to choose the right sampling rate depending on these latter parameters.

6.3. Internal channel analysis: the POWERMON2 *case*

In this section we provide an specific analysis of the power dissipation profiles drawn by the `idle`, `hdparm`, and `cpuburn` benchmarks when POWERMON2 is used to measure, independently, the 3.3 V, 5 V and 12 V lines on INTEL_DESKTOP and AMD_SERVER. We also show how, by using an internal wattmeter like POWERMON2, it is possible to distinguish the power source line. The sum of all these lines is included in the results as well.

We depict the behavior of INTEL_DESKTOP in Figure 11. In this platform, depending on the benchmark, the different lines draw different activity. For `idle`, the 3.3 V and 5 V lines transport more power than the 12 V lines but power remains constant in all of them. The situation varies for `hdparm`: the 5 V lines fluctuate in conjunction with the 12 V socket-related lines; meanwhile the 3.3 V lines and 12 V mainboard lines show a fairly plain profile. The situation becomes even more interesting for the `cpuburn` benchmark. In this case we observe how the 5 V and 12 V socket-related lines initially increase their power, which could be expected since the `cpuburn` processes highly stress the cores and, therefore the power dissipated by the cores. It is also important to note that the mainboard 3.3 V and 12 V lines feature a very plain profile.

The same experiment was repeated on AMD_SERVER and results are shown in Figure 12. This platform exhibits a quite different range of lines that POWERMON2 is able to analyze. In this case we analyze the 3.3 V, 5 V and 12 V for mainboard and the 12 V socket-related lines. As shown in Figure 12, each one of the 4 sockets of AMD_SERVER is not fed by one specific 12 V line. Indeed, while varying the number of sockets during the `cpuburn` benchmark, the power measured by the each of the four 12 V lines changes. The `idle` benchmark provides a flat profile for all the lines, nevertheless; the 12 V constantly transport more power

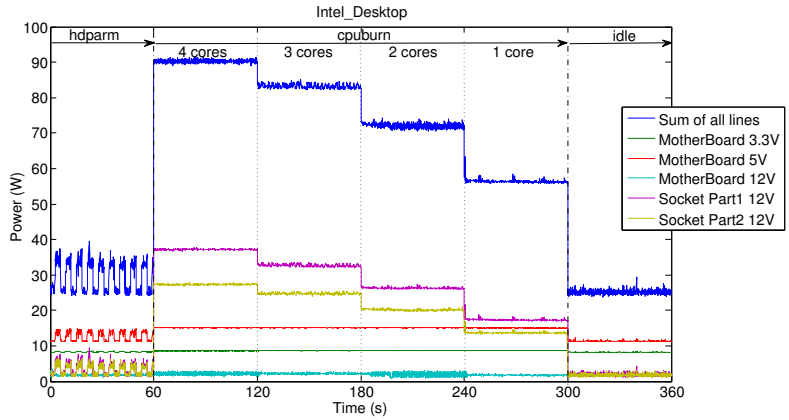

Figure 11: Power dissipation profile provided by PowerMon2 wattmeter when running different benchmarks on Intel_Desktop.

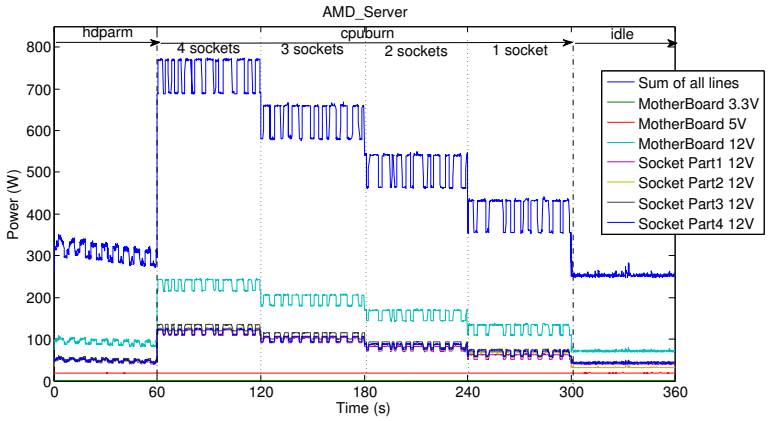

Figure 12: Power dissipation profile provided by PowerMon2 wattmeter when running different benchmarks on AMD_Server.

than the 3.3 V and 5 V lines. For `hdparm` this situation changes: the 12 V lines start by drawing the natural spikes of this benchmark, specifically the 12 V mainboard line almost consumes twice the power of the 12 V socket-related lines. For the execution of the `cpuburn`, the situation is repeated: the 12 V mainboard line doubles the power drawn by the 12 V lines and starts dropping down when less sockets are working. For 3.3 V and 5 V lines is interesting to point out how plain the profile is.

By measuring independently the different power lines of the motherboard, we determine from which internal component the power dissipation is coming from. For example, in the case of a compute-bound application, the lines that come from the CPU (12V lines) will dissipate more power compared to memory-bound or HDD-bound applications. Thanks to such internal channel analysis, we can determine the specific power dissipation of each internal component (CPU, memory, HDD, etc.) for a given application.

7. Analysis of Integrated Power Measurement Interfaces

Recent computer hardware generations include support for measuring energy and power dissipation internally. This allows fine-grain power analysis without requiring external measurement devices. In this section, we point out the evaluation of three non-wattmeter based power measurement interfaces: *i)* Intel RAPL, *ii)* NVIDIA Management Library and *iii)* Intelligent Platform Management Interface.

i) Intel RAPL: The Intel Sandy Bridge architecture includes the new Running Average Power Limit (RAPL) interface [26] that allows to account for the power consumed by the chip at different levels under

a specific design. In this sense internal circuitry is able to estimate current power based on a model driven by hardware counters, temperature and leakage models. The results of this model are available via a model specific register (MSR) and provide an update frequency in the order of milliseconds. An evaluation by Intel [27] demonstrates that the results fairly match the actual usage of the processor, so their consideration in our evaluation is especially important.

Specifically, the RAPL interface reports several power/energy readings. Power Package 0 (PKG0) accounts for the total power consumed by the whole socket. Inside, two other measurements can be accessed. Power Plane 0 (PP0) accounts the power used by all the cores/caches, and Power Plane 1 (PP1) reports the power of the on-board GPU, in case of desktop Sandy Bridge processors, or the DRAM memory, for server Sandy Bridge EP processors. Furthermore, an uncore component can be derived from the available readings taking into account that $PKG0 = PP0 + PP1 + UNC$ [28]; hence it is straightforward to obtain the power dissipation due to internal chip circuitry (QPI network, DDR channels, shared levels of cache, etc.) by subtracting both power planes to the package power, $UNC = PKG0 - (PP0 + PP1)$.

To acquire data from this interface we developed a new module for `pmlib` that retrieves these values via MSR at 100 S/s. This frequency can be set by the user. In a study conducted in [29] we assessed that an increase in the sampling frequency directly affects the performance and power dissipation of the platform due to the overhead caused by the daemon responsible for collecting power samples running on the target machine. Therefore, the sampling frequency needs to be carefully configured.

ii) NVIDIA Management Library: Recent NVIDIA GPUs are able to report power usage via the NVIDIA Management Library (NVML) [30]. Routine `nvmlDeviceGetPowerUsage()` retrieves the current power of the GPUs. In the tested platform, a NVIDIA Tesla Kepler K20, the power readings are accurate to within ±5 W of current power draw. To read these measurements we developed a new NVML module for `pmlib` that samples GPU power at 1 S/s (this is the maximum sampling frequency this interface provides). Specifically the NVIDIA Tesla Kepler K20 allows power management features, among them one can obtain the current power, and specifications for drawn, default and constraint power limits.

iii) Intelligent Platform Management Interface (IPMI): is an industry-standard protocol for monitoring the status of computers and perform basic operations on them. These operations are possible thanks to the Baseboard Management Controller (BMC) that directly interfaces through an Ethernet port or serial links. Most of its functionality is intended to monitor critical elements, like fan speed, CPU temperature or power dissipation. The command `impitool` can be used to read the sensors of the machines, either local or remotely. For the experiments we developed an IPMI module for `pmlib` that allows to read power from IPMI-capable platforms. The frequency in this case is set automatically by the response of the `impitool` command and is about 0.5 S/s.

To perform the study of the RAPL interface we use the Intel_Desktop platform and test with the aforementioned benchmarks. For the case of NVML and IPMI interfaces we use two new platforms, since the previous configurations are not compatible with these new measurement sources. An Intel Ivy Bridge Core i7-3930K equipped with 6 cores running at 3.20 GHz, 24 GB of RAM, and a 120 GB Seagate Barracuda 7200.7 HDD, also equipped with a NVIDIA Tesla Kepler K20 serves to analyze both RAPL and NVML counters. We denote this platform as Intel_Desktop_2. On the other hand, we consider a system with four 16-core AMD 6276 Opteron processors (64 cores in total), 64 GB of RAM and a 500 GB Seagate Constellation.2 HDD to test the IPMI interface. This platform is denoted as AMD_Server_2.

7.1. Analysis of the RAPL *interface*

Let us commence by analyzing the behavior of the RAPL interface on the Intel_Desktop platform. Figure 13 depicts the power drawn by the set of readings of RAPL simultaneously with the external WattsUp wattmeter during the execution of the `hdparm`, `cpuburn` and `idle` benchmarks, respectively. As shown there the readings related to the cores, i.e., PKG0 and therefore PP0, are affected by the execution of the benchmarks. We consider this a normal behavior since none of them is making use of the on-board GPU. Also the uncore component shows a fairly plain profile, which implies that the leakages inside the socket are constant.

At this point, a different comparison can be done with the external WattsUp wattmeter. It is clear that the gap between the PKG0 and WattsUp profiles is due to the different nature (internal and external) of

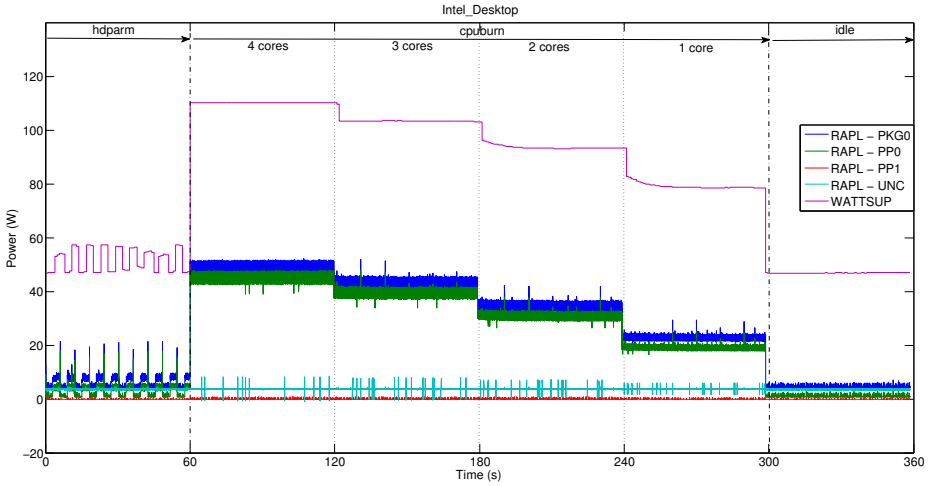

Figure 13: Power dissipation profile provided by RAPL and WattsUp when running different benchmarks on Intel_Desktop.

the power measurement sources. Specifically, the difference between these two components for `hdparm` and `idle` benchmarks is less than that for `cpuburn` when all the cores are used. This difference is caused by the fact that other external components are being used simultaneously when the cores are stressed by the `cpuburn` benchmark. Therefore, we relate this extra dissipation to the fans, spinning at full speed to cool down the socket.

In general, readings from RAPL interface correlate very well with the external power measurements and reveal the benefits of this approach at large-scale as in many cases it avoids the purchase of expensive wattmeters. The main drawback is that they can only be accessed via MSR, so a daemon must run in the same machine, therefore, introducing an overhead. Moreover, the small set of RAPL interface is limited in functionality in that they represent metrics that are of socket scope. Thus, individual cores cannot be measured. New architectures, like POWER7 from IBM, permit similar features at core-level. Furthermore, thanks to a specific hardware, the counters can be externally accessed with no visible impact on the performance [31].

7.2. Analysis of the NVML *interface*

In this section we perform an analysis that combine the use of the NVML interface with RAPL and the external wattmeter WattsUp. The use of the Intel_Desktop_2 platform with GPU is necessary in this case to access the power readings that the NVML interface offers. For the same reason, apart from `hdparm`, `cpuburn` and `idle`, we also include the `dgemm` benchmark to make use of the GPU.

Figure 14 shows the behavior of the execution of these four benchmarks using the aforementioned power measurement mechanisms. Generally, the same assumptions for the RAPL measurements can be done in this case. However, it is interesting to fix our attention in the computation of the `dgemm` benchmark. This benchmark computes the matrix-matrix product using the `cublasDgemm` kernel from NVIDIA CUBLAS 5.0 with polling and blocking synchronization modes. We denote them as `dgemm-P` and `dgemm-B`, respectively. For the polling mode we first invoke `cudaSetDeviceFlags` with `cudaDeviceScheduleAuto` parameter set. Because of the intensive use of the computational units of the GPU, the power drawn by the device for `dgemm-P` (NVML profile) is increased by an important factor. However, the PKG0/PP0 power profiles from RAPL are also increased, as the calling thread remains in a busy-wait (polling) until the kernel finishes its execution.

On the other hand, the blocking mode in `dgemm-B` is set by means of `cudaDeviceBlockingSync` parameter in the aforementioned routine. In this case, there is a similar increase of the power drawn by the GPU (NVML) but, the calling thread, running on a CPU core, remains blocked on a synchronization point until

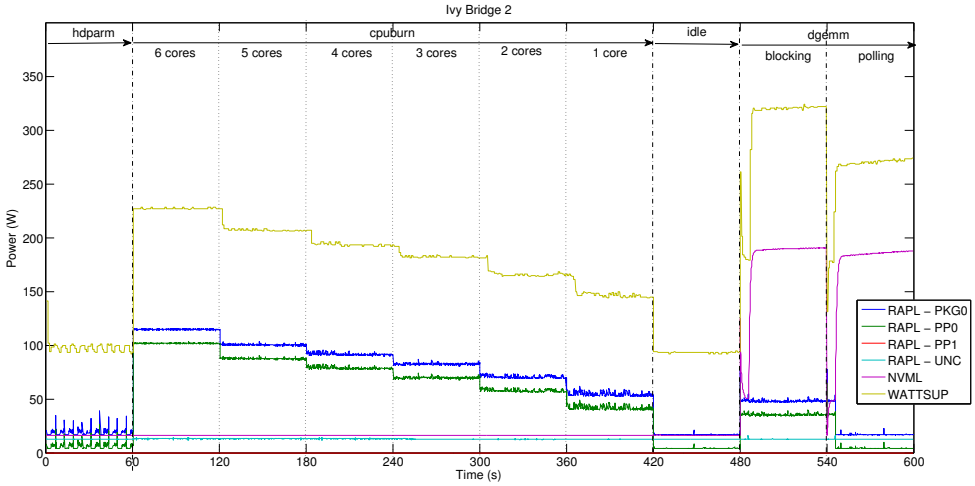

Figure 14: Power dissipation profile provided by RAPL, NVML and WattsUp when running different benchmarks on Intel_Desktop_2.

the GPU completes its work. The blocking mode is interesting for GPU intensive applications since it avoids potential energy-wasting states.

Finally, the comparison with the external WattsUp wattmeter shows behaviors as similar to those observed in the platform Intel_Desktop. Besides differences between `dgemm-P` and `dgemm-B` are also visible and useful to observe the use of techniques that exploit low power states.

7.3. Analysis of the IPMI *interface*

In this section we analyze the power measurements obtained with the IPMI interface and the WattsUp wattmeter. Figure 15 depicts the execution of the benchmarks `hdparm`, `cpuburn` and `idle`, respectively. In general, the power drawn by the IPMI interface is, on average, lower than that offered by WattsUp. The reason is that IPMI measures the power dissipation of the whole motherboard, in contrast to WattsUp, that considers the full machine taking into account, e.g., the power leakage due to the PSU as well.

Shapes of both power profiles are correlated but the differences between them vary depending on the benchmark. The reason is that IPMI considers a subset of the dissipation sources for the WattsUp, hence the last one observes higher power figures. The sampling frequency is also interesting to analyze. IPMI offers, on average, 0.5 S/s; i.e, one measurement every 2 seconds, vs. 1 S/s offered by the WattsUp. Although both sampling frequencies are not too high, the accuracy of the IPMI interface is relatively low for this particular case when compared with the WattsUpwattmeter. For IPMI we are not able to observe the noise that is captured by WattsUp for the `hdparm` and `idle` benchmarks. Therefore, IPMI measurements are appropriate for monitoring purposes and accessible from the outside, nevertheless not suitable for users requiring high accuracy and sampling frequency.

8. Discussion, Conclusions and Future Work

In this paper, we analyze and evaluate different external and internal power monitoring devices tested using two different computing systems, a server (AMD_Server) and a desktop machine (Intel_Desktop), offering a complete comparison in terms of power dissipation and energy consumption. We also evaluate three power measurement interfaces using the aforementioned platforms and two new sever machines (Intel_Desktop_2 and AMD_Server_2).

First of all, we show that, unlike external wattmeters, internal wattmeters do no register neither an equal energy consumption nor a similar power variability. Results show indeed that the energy and the power dissipation captured by PowerMon2 are often higher than the ones provided by NI and DCM,

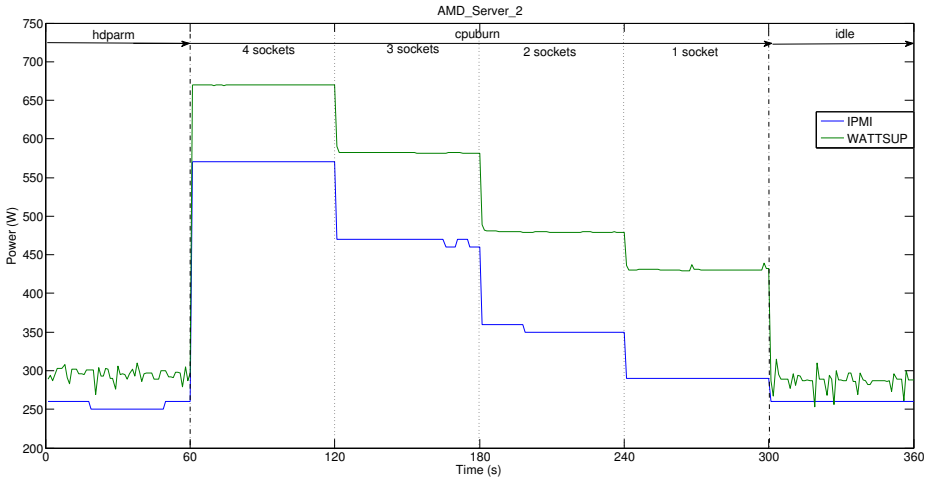

Figure 15: Power dissipation profile provided by IPMI and WattsUp when running different benchmarks on AMD_Server_2.

especially for AMD_Server. These results can be explained by the fact that these wattmeters use different components to measure the power energizing the internal wires. As expected, we show that the energy consumption measured by the internal wattmeters is always lower than those measured by external ones. However, that was not always the case with the power measurements. As a matter of fact, when the `cpuburn` benchmark runs on AMD_Server, PowerMon2 registers higher power measurements compared to WattsUp, even if PowerMon2 does not take into account the PSU leakage overhead and even some internal components. This result tends to show that PowerMon2 is less accurate when used to measure too high power values (more than 500 W).

Contrary to what one could expect, we pointed out that the extra energy consumption (i.e. mean-idle) of a given benchmark running on a machine is not equal for all the wattmeters. Indeed, it signals a difference of more than 50 % for benchmarks like `hdparm` and for low consuming ones line `iperf`. This generates doubts about the accuracy of the wattmeters. In this paper it is pointed out that the internal values are dispersed and generate many outlier samples. This is certainly due to the instantaneous power measurements registered by the internal wattmeters. Another important reason is the high frequency sampling of the internal wattmeters. However, the question that remains is whether this is because a very high frequency that allows to measure power fluctuations that the external wattmeters cannot capture or, instead, it is because they generate too much noise.

The power profiles obtained by the internal devices (instantaneous measurements) capture significant fluctuations. This behavior can be explained due to the very high sampling rate of instantaneous values that the internal wattmeters provide, especially the NI. Furthermore, external wattmeters on the one hand and internal wattmeters on the other provide different power profiles and make them displaced for `hdparm` and, more specifically, `cpuburn`. These differences mainly come from differences built into the devices, sensors, measurements nature and potential environment changes, like the room temperature [24]. Contrary to Intel_Desktop, the power profile on AMD_Server permits to observe clear spikes and drops when running the `cpuburn` benchmark. This behavior is related to the numerous fans that are alternatively turned on and off in order to cool the server machine.

While plotting the power profiles with a varying sample rate, we highlighted that measuring at a very high sample rate (500 S/s) is not always necessary for profiling power dissipated by the applications, and may even provide too much samples, e.g. outliers, that mask the general shape of the power trace. The appropriate sample rate is not always the highest possible but the one that best enables to understand the power fluctuations. Also, an internal wattmeter like PowerMon2 offers power profiles showing the different line voltages (3.3 V, 5 V and 12 V) in a separated way. This allows to detect where the power fluctuations come from. One may hope that each line is related to one specific component, which was not the case neither with Intel_Desktop nor AMD_Server.

In this study, we also focused on other approaches to measure the power dissipation: RAPL counters available on Sandy Bridge microarchitectures, the NVML to measure the power of GPUs, and the IPMI which provides energy measures for the whole motherboard. Contrary to the internal wattmeters, these internal interfaces allow to measure power dissipation for large-scale distributed systems since they do not require an extra device. However, since they measure instantaneous power dissipation, they need to operate at a high sampling rate in order to capture power fluctuations. However, as the sampling rate is increased, the power measurement mechanism becomes in some cases too intrusive, since the program needs to check more frequently the energy counters through the RAPL and the NVML interfaces which, in the end, compromise the performance of the applications and the measures themselves.

In sum, thanks to the high sample rate and to the different measured lines, the internal measurement devices allow to better visualize some power fluctuations that the external wattmeters are not able to capture. However, a high sample rate is not always necessary to understand the evolution of the power dissipation during the execution of a benchmark. A key to achieve accurate and reliable measurements requires a calibration with oscilloscope and initial tests which ensure their quality; however, this operation is not always easy to perform. Moreover, monitoring very large-scale distributed systems (like Exascale supercomputers) with internal wattmeters is not practicable, especially because these equipments are not easy to connect and the price per node is expensive (e.g., 2,700€ for NI). For these reasons, other sources like IPMI, RAPL or NVML are appealing alternatives for monitoring issues.

As a future work, we plan to extend this study with a wide range of wattmeters (internal and external) and machines at different levels, also we plan to perform a separated study for the PSUs, which are an important source of leakages and deserves to be studied for a better understanding of the power dissipation and energy consumption of current machines. With this study we will try to answer questions that remained open after this work.

Acknowledgments

This research was supported by the European COST Actions IC804 ("Energy efficiency in large scale distributed systems") and IC805 ("Complex HPC systems"). Authors from Universitat Jaume I were also supported by the project CICYT TIN2011-23283 and FEDER.

References

[1] C. Hsu, W. Feng, and J. S. Archuleta. Towards efficient supercomputing: A quest for the right metric. In *Proceedings of the High Performance Power-Aware Computing Workshop*, 2005.

[2] J. Dongarra et al. The international ExaScale software project roadmap. *Int. J. of High Performance Computing & Applications*, 25(1), 2011.

[3] W. Feng, X. Feng, and R. Ge. Green supercomputing comes of age. *IT Professional*, 10(1):17 –23, 2008.

[4] 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 *Proceedings of the 2012 Symposium on High Performance Computing*, HPC '12, pages 6:1–6:10, San Diego, CA, USA, 2012.

[5] P. Alonso, M. F. Dolz, F. D. Igual, R. Mayo and E. S. Quintana-Ortí. DVFS-control techniques for dense linear algebra operations on multi-core processors. *Computer Science - R&D*, 27(4):289–298, 2012.

[6] P. Alonso and M. F. Dolz and R. Mayo and E S. Quintana-Ortí. Energy-efficient execution of dense linear algebra algorithms on multi-core processors. *Cluster Computing*, May 2012.

[7] W. Feng and K. Cameron. The green500 list: Encouraging sustainable supercomputing. *Computer*, 40(12):50–55, December 2007.

[8] M. E. M. Diouri, O. Glück, L. Lefèvre, and J.-C. Mignot. Your Cluster is not Power Homogeneous: Take Care when Designing Green Schedulers! In *4^{th} IEEE International Green Computing Conference, Arlington, USA*, June 2013.

[9] H. Ltaief, P. Luszczek, and J. Dongarra. Profiling high performance dense linear algebra algorithms on multicore architectures for power and energy efficiency. *Comput. Sci.*, 27(4):277–287, November 2012.

[10] B. Subramaniam and W. Feng. The Green Index: A Metric for Evaluating System-Wide Energy Efficiency in HPC Systems. In *8th IEEE Workshop on High-Performance, Power-Aware Computing*, Shanghai, China, May 2012.

[11] R. Ge, X. Feng, S. Song, H.C. Chang, D. Li, and K.W. Cameron. Powerpack: Energy profiling and analysis of high-performance systems and applications. *IEEE Trans. Parallel and Distributed Systems*, 21(5):658–671, 2010.

[12] B. Subramaniam and W. Feng. Statistical power and performance modeling for optimizing the energy efficiency of scientific computing. In *Proc. of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications*, GREENCOM, pages 139–146, Washington, DC, USA, 2010. IEEE Computer Society.

[13] P. Alonso, R. M. Badia, J. Labarta, M. Barreda, M. F. Dolz, R. Mayo, E. S. Quintana-Ortí, and R. Reyes. Tools for power-energy modelling and analysis of parallel scientific applications. In *41st Int. Conf. on Parallel Processing – ICPP*, pages 420–429, 2012.

[14] M. E. M. Diouri, M. F. Dolz, O. Glück, L. Lefèvre, P. Alonso, S. Catalán, R. Mayo, and E. S. Quintana-Ortí. Solving some mysteries in power monitoring of servers: take care of your wattmeters! In *Proc. Energy Efficiency in Large Scale Distributed Systems conference, EE-LSDS 2013*, volume 8046 of *Lecture Notes in Computer Science (LNCS)*, pages 1–16, 2013.

[15] J. Carpentier, J.-P. Gelas, L. Lefèvre, M. Morel, O. Mornard, and J.-P. Laisné. Compatibleone: Designing an energy efficient open source cloud broker. In *2012 Second International Conference on Cloud and Green Computing, CGC 2012, Xiangtan, Hunan, China, November 1-3, 2012*, pages 199–205, 2012.

[16] Intel, Hewlett-Packard, NEC, and Dell. Ipmi specification, v2.0/1.5, rev. 4: Addendum, June 2009.

[17] M. Castillo, M. F. Dolz, J. C. Fernández, R. Mayo, E. S. Quintana-Ortí, and V. Roca. Evaluation of the energy performance of dense linear algebra kernels on multi-core and many-core processors. In *25th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2011, Anchorage, Alaska, USA, 16-20 May 2011 - Workshop Proceedings*, pages 846–853, 2011.

[18] R. A. Giri and A. Vanchi. Increasing data center efficiency with server power measurements. Technical report, Intel Information Technology, June 2010.

[19] D. Bedard, M. Y. Lim, R. Fowler, and A. Porterfield. PowerMon: Fine-grained and integrated power monitoring for commodity computer systems. In *Proceedings Southeastcon 2010*, Charlotte, NC, March 2010. IEEE.

[20] S. Barrachina, M. Barreda, S. Catalán, M. F. Dolz, G. Fabregrat, R. Mayo, and E. S. Quintana-Ortí. An integrated framework for power-performance analysis of parallel scientific workloads. In *3rd Int. Conf. on Smart Grids, Green Communications and IT Energy-aware Technologies*, 2013.

[21] M. Hähnel, B. Döbel, M. Völp, and H. Härtig. Measuring energy consumption for short code paths using rapl. *SIGMETRICS Performance Evaluation Review*, 40(3):13–17, 2012.

[22] E. Rotem, A. Naveh, A. Ananthakrishnan, E. Weissmann, and D. Rajwan. Power-management architecture of the intel microarchitecture code-named sandy bridge. *IEEE Micro*, 32(2):20–27, 2012.

[23] K. K. Kasichayanula. Power Aware Computing on GPUs. Master's thesis, University of Tennessee, 2012.

[24] Qinghui Tang, Sandeep K. S. Gupta, and Georgios Varsamopoulos. Thermal-aware task scheduling for data centers through minimizing heat recirculation. In *Proceedings of the 2007 IEEE International Conference on Cluster Computing, 17-20 September 2007, Austin, Texas, USA (CLUSTER 2007)*, pages 129–138. IEEE, 2007.

[25] M.K. Patterson. The effect of data center temperature on energy efficiency. In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2008. ITHERM 2008. 11th Intersociety Conference on*, pages 1167–1174, May 2008.

[26] Intel. *Intel Architecture Software Developer's Manual, Volume 3: System Programming Guide*, 2009.

[27] E. Rotem, A. Naveh, A. Ananthakrishnan, E. Weissmann, and D. Rajwan. Power-management architecture of the intel microarchitecture code-named sandy bridge. *IEEE Micro*, 32(2):20–27, March 2012.

[28] J. Demmel and A. Gearhart. Instrumenting linear algebra energy consumption via on-chip energy counters. Technical Report UCB/EECS-2012-168, EECS Department, University of California, Berkeley, Jun 2012.

[29] M. Barreda, S. Catalán, M. F. Dolz, R. Mayo, and E. S. Quintana-Ortí. Automatic detection of power bottlenecks in parallel scientific applications. *Computer Science - Research and Development*, 2013.

[30] NVIDIA. *NVML Reference Manual*, 2013.

[31] L. Brochard, R. Panda, and S. Vemuganti. Optimizing performance and energy of hpc applications on power7. *Computer Science - Research and Development*, 25:135–140, 2010.