Solving Some Mysteries in Power Monitoring of Servers: Take Care of Your Wattmeters!

Mohammed El Mehdi Diouri[1(✉)], Manuel F. Dolz[2], Olivier Glück[1], Laurent Lefèvre[1], Pedro Alonso[3], Sandra Catalán[2], Rafael Mayo[2], and Enrique S. Quintana-Ortí[2]

[1] INRIA Avalon Team, Laboratoire de l'Informatique du Parallélisme, UMR CNRS 5668, ENS Lyon, INRIA, Université Lyon 1, Lyon, France
{mehdi.diouri, olivier.gluck, laurent.lefevre}@ens-lyon.fr

[2] Depto. de Ingeniería y Ciencia de Computadores, Universitat Jaume I, 12071 Castellón, Spain
{dolzm, catalans, mayo, quintana}@uji.es

[3] Depto. de Sistemas Informáticos y Computación, Universitat Politècnica de València, 46022 Valencia, Spain
palonso@dsic.upv.es

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 and energy consumption of these systems is increasingly 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 also provide experimental results for a variety of benchmarks which exercise intensively the main components (CPU, Memory, HDDs, and NICs) of the target platforms to validate the accuracy of the equipment in terms of power dispersion and energy consumption. This paper highlights that external wattmeters do not offer the same measures as internal wattmeters. 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 · Energy and power analysis · Power profiling · Servers

1 Introduction

For decades, the computer science research community exclusively focused on performance, which resulted in highly powerful, but in turn, low 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].

J.-M. Pierson et al. (Eds.): EE-LSDS 2013, LNCS 8046, pp. 3–18, 2013.
DOI: 10.1007/978-3-642-40517-4_1,

Nowadays there exists a strong research effort towards energy-efficient supercomputers. Hardware provides part of the solution by exposing unceasingly 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 [4–6].

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 in terms of Mflops 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 a computing node. Some proposals obtain the consumption of different parts of the node (CPU, memory, disks, fans, ...) but they are circumscribed to either one [8] or a few benchmarks types [9]. Furthermore, the most prevailing infrastructure comprises only an external power meter [1,10]. 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 [11]. Given the foregoing, we contribute in this paper 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.
- 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 [12].
- 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.

The rest of the paper is structured as follows. In the next section, we describe the experiment setup. The energy consumption obtained with the wattmeters using all the benchmarks is analyzed in Sect. 3. In Sect. 4, we discuss in more detail these results by processing samples under a statistical model based on boxplots. Section 5 performs an additional analysis by varying the wattmeters sampling rate. The paper is closed with a section that contains a discussion and a few conclusions.

2 Experimental Setup

This section describes the power measurement devices, the power measurement 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 1 presents in detail the specifications of the wattmeters that we used.

Table 1. 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	1024[f]	1000	28
Accuracy	$< \pm 1\%$	$< \pm 1.5\%$	$\pm 5\%$	$\pm 1\%$	$\pm 1\%$
Interface	RS232	USB	USB	USB	RS232
Price	600 €	200 €	125 €	2700 €	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 consumption of some components of the mainboard (GPUs, NICs, etc.) while 12 V lines measure the power consumption 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 appropriate `pmlib` routines (see Fig. 1).

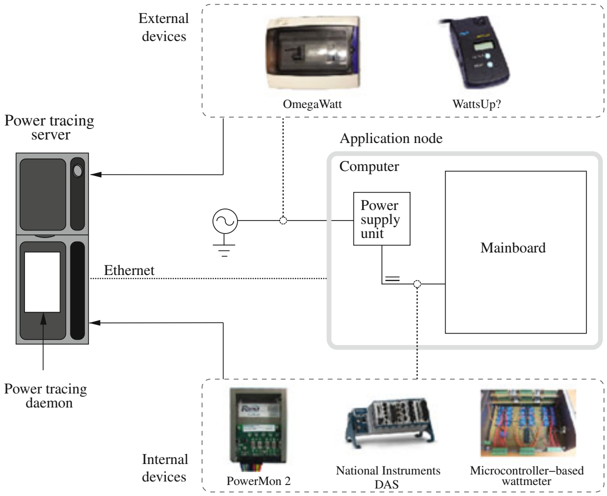

Fig. 1. Single-node application system and sampling points for external and internal wattmeters.

Target Platforms. The analysis and evaluation made with the wattmeters has been carried out on two different platforms: a desktop platform and a server node. The desktop computer consists of an Intel Ivy Bridge Core i7-3770K equipped with 4 cores running at 3.50 GHz and 16 GB of RAM. 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 and contains 256 GB of RAM.
Benchmarks. To evaluate the energy and power behavior, we run different types of workloads to provoke and encourage the use of specific parts of the platforms. CPU, memory, NICs and HDDs are the main components we stress in our experiments. To achieve this purpose, we selected the following specific benchmarks:

- `idle`: This benchmark employs the `sleep` POSIX routine[1] to suspend processor activity, thus generating idle periods that let the hardware promote the cores to low power consumption states, also known as C-states[2].
- `iperf`[3]: 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 TCP client.
- `hdparm`[4]: This application provides a command line interface to various kernel interfaces supported by the Linux SATA/PATA/SAS *libATA* subsys-

[1] `sleep`: http://linux.die.net/man/3/sleep
[2] Advanced Configuration and Power Interface. Revision 5.0. http://www.acpi.info/
[3] `iperf`: http://iperf.fr
[4] `hdparm`: http://linux.die.net/man/8/hdparm

tem. We use the `-t` option to perform timings on device reads and to stress the HDD.

- `cpuburn`[5]: This benchmark heats up any 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.
- `burnMMX`[6]: 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.

3 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 measured 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 by wattmeters when, respectively, the HDDs and the NICs are 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 (≈ 10 min.). We represent the aggregated energy consumption calculated as the addition of energy measurements in 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 12 V lines.

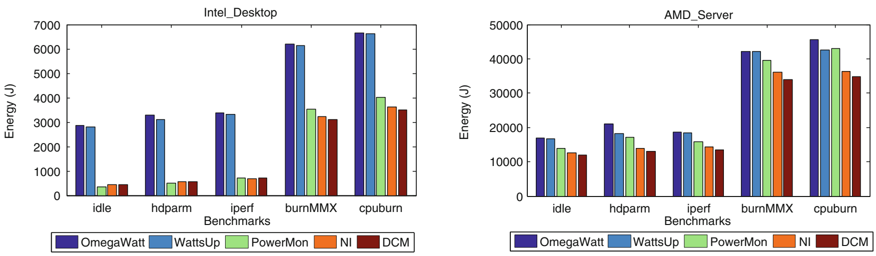

Fig. 2. Extra energy consumption of the benchmarks measured with the wattmeters.

[5] `cpuburn`: http://manpages.ubuntu.com/manpages/precise/man1/cpuburn.1.html

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

Figure 2 shows that the energy consumptions registered with both external wattmeters (OmegaWatt and WattsUp) are very similar. However, 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 respectively sample at 100 S/s and 1,000 S/s, and rapid variations are not captured by low sampling devices.

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

We also studied if the energy fluctuations observed in Fig. 2 were related to the wattmeter calibration. Figure 3 shows the extra energy consumption corresponding to the same experiments reported in Fig. 2, but removing the energy consumption of the `idle` benchmark from the energy consumption of each benchmark. While we could expect small or no differences between the different measurement devices, some variations appear. Specifically, in some cases there are still differences that range from 2.23% to 65.75%.

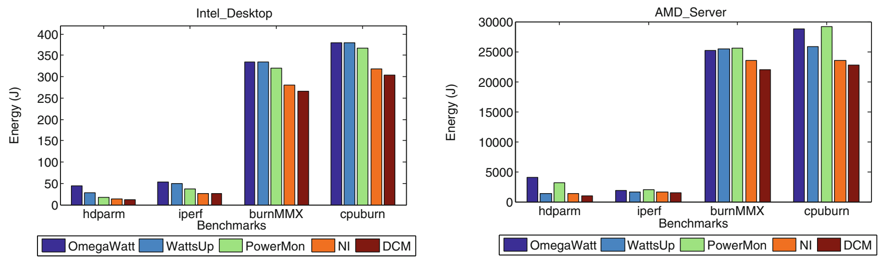

Fig. 3. Extra energy consumption of the benchmarks (i.e. without the `idle` part) measured with the wattmeters.

4 Power Consumption Analysis

In this section, we analyze the power measurements made by the different considered wattmeters. Through this analysis, we first seek to 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 consumption internally or externally with a variable measurement frequency.

At this point, we consider the following scenarios: the target machine is `idle` (Fig. 4), one core of the machine runs `hdparm` (Fig. 5), and all cores of the machine run `cpuburn` (Fig. 6). We execute each benchmark on the two different target machines, and measure the power consumption during 60 seconds using the external wattmeters OMEGAWATT and WATTSUP, and the internal wattmeters POWERMON2, NI and DCM.

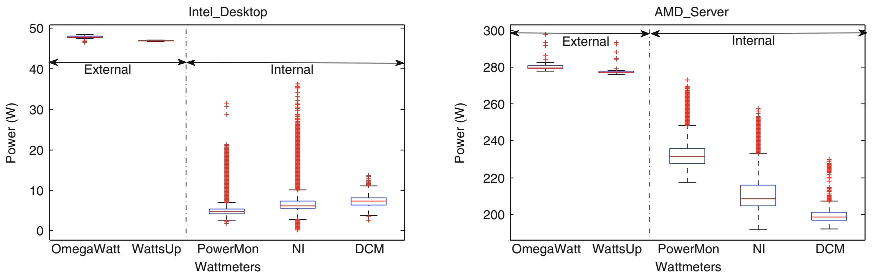

Fig. 4. Dispersion of the power consumption measurements for benchmark `idle`.

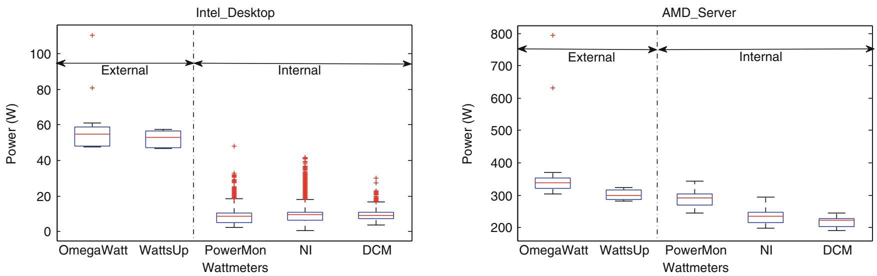

Fig. 5. Dispersion of the power consumption measurements for benchmark `hdparm`.

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

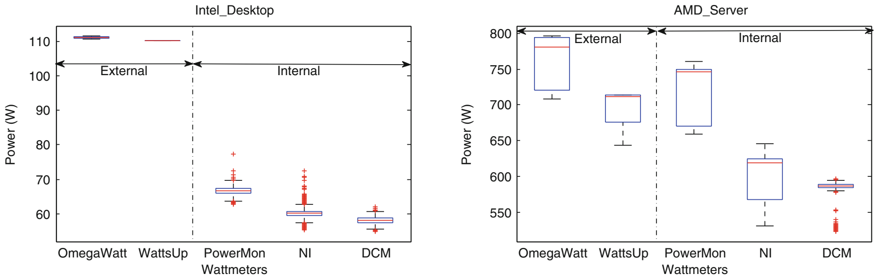

Fig. 6. Dispersion of the power consumption measurements for benchmark `cpuburn`.

(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 external wattmeters.

Alike in the previous experiment, with energy measurements, 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 wattmeters. As concerns AMD_Server, the differences are more visible: neither the external nor the internal power measurements are concordant, 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. Furthermore, compared with WattsUp, OmegaWatt provides higher power measurements, 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, especially when applied to measure too high power consumptions (more than 300 W).

What is even more mysterious is that when benchmark `cpuburn` is run on AMD_Server, PowerMon2 registers higher power measurements compared to WattsUp, 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 suspect that PowerMon2 is less accurate when used to measure too high power consumptions (more than 500 W).

Furthermore, in Fig. 4 we notice that, compared to the external measurements, the results captured with the internal devices are more dispersed and generate many outliers. This is certainly due to the high sampling frequency of internal wattmeters that allow to measure some power fluctuations that the external wattmeters cannot capture. We notice from Fig. 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 consumption during the execution of the `hdparm` benchmark provoking that OmegaWatt is occasionally not accurate. Moreover, Fig. 6 shows that when `cpuburn` runs on AMD_Server the power measurements are highly dispersed: a variability of almost 100 W for OmegaWatt, PowerMon2, 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 Sect. 5, we expect to confirm the fluctuation in `hdparm` and `cpuburn` when they are executed on AMD_Server.

5 Power Profile Analysis

In this section we continue 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 and power consumption.

5.1 Analysis of the Power Profiles

Let us start by considering Figs. 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 Fig. 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 the same information but replacing the wattmeters by WattsUp and NI. The aim of this comparison is to inspect the same scenario with two different power measurement devices 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

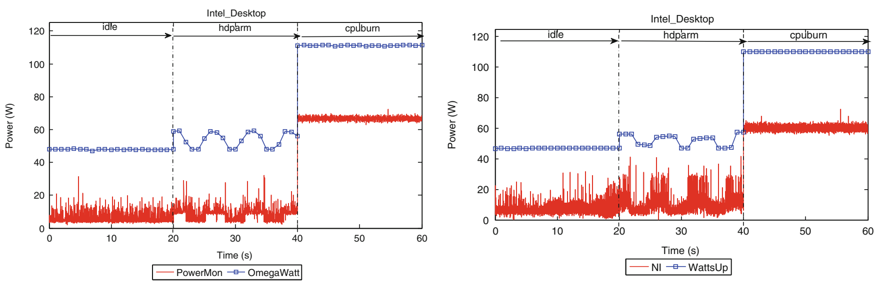

Fig. 7. Power profiles obtained when running benchmarks `idle`, `hdparm` and `cpuburn` benchmarks on Intel_Desktop.

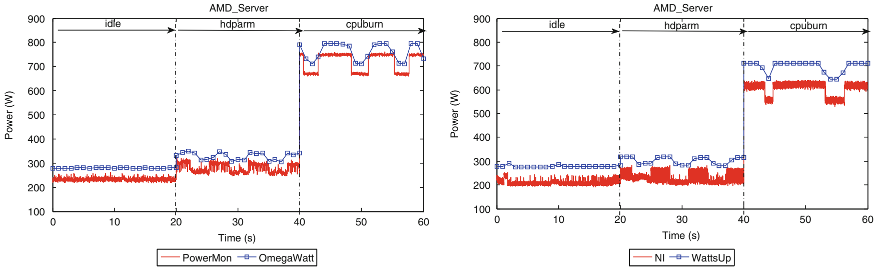

Fig. 8. Power profiles obtained when running benchmarks `idle`, `hdparm` and `cpuburn` benchmarks on AMD_SERVER.

with the NI; this observation was already made in the experiments of Sects. 3 and 4. 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.

Drawing attention to Fig. 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), as these experiments could not be performed simultaneously. (We were not able to connect two external or internal wattmeters at the same time.) 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 kind 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.

5.2 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 consumption with NI and a frequency of 1,000 S/s during 30 seconds for `hdparm` and `cpuburn` on INTEL_DESKTOP and AMD_SERVER. Then, we reduce the sample rate by

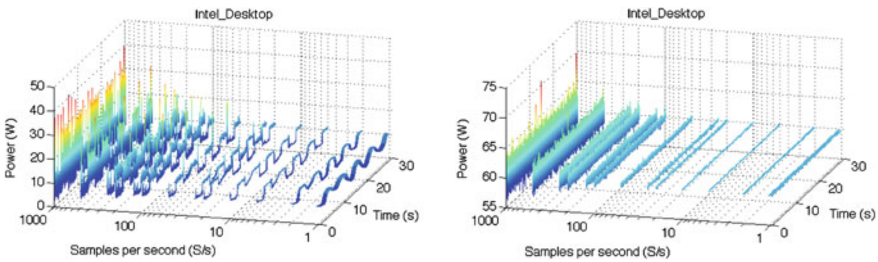

Fig. 9. Power profiles obtained with NI for `hdparm` (left side) and `cpuburn` (right side) running on INTEL_DESKTOP with a configurable sample rate.

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 is S_j^r 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 sample 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 200 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 Fig. 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 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 which 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 even may cause some noise that

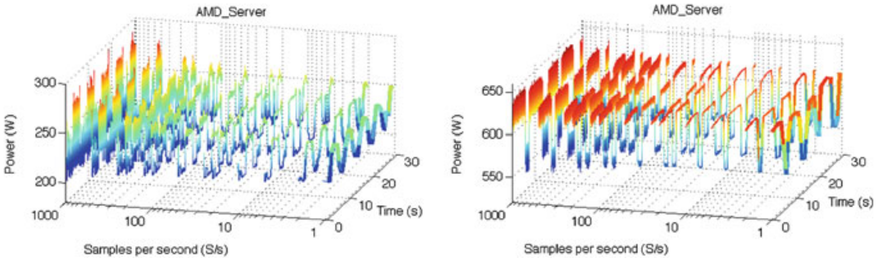

Fig. 10. Power profiles obtained with NI for `hdparm` (left side) and `cpuburn` (right side) running on AMD_SERVER with a configurable sample rate.

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.

5.3 Internal Channel Analysis: The PowerMon2 Case

In this section we provide an specific analysis of the power consumption 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 Fig. 11. In this platform, depending on the benchmark, the different lines draw different powers. For `idle`, the 3.3 V and 5 V lines transport more power than the 12 V lines and 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

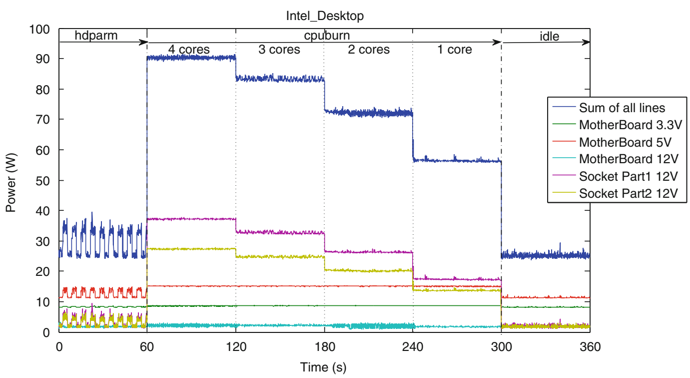

Fig. 11. Power consumption profile provided by PowerMon2 wattmeter when running different benchmarks on INTEL_DESKTOP.

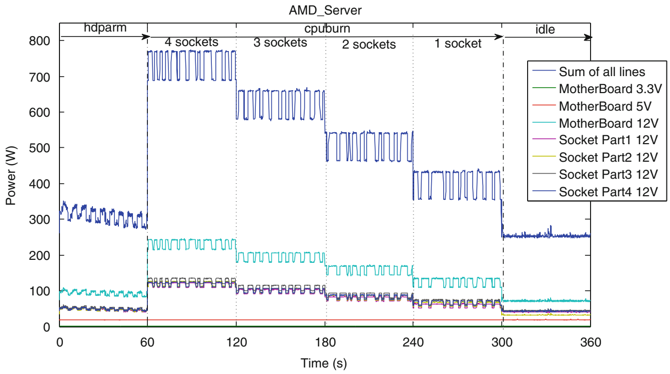

Fig. 12. Power consumption profile provided by PowerMon2 wattmeter when running different benchmarks on AMD_Server.

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 Fig. 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 Fig. 12, each one of the 4 sockets of AMD_Server is not fed with one specific 12 V socket-related line. Indeed, while varying the number of sockets during the cpuburn benchmark, the power measured by the each of the four 12 V socket-related lines changes. The idle benchmark provides a flat profile for all the lines, nevertheless; the 12 V constantly transport more power 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 socket-related 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.

6 Discussion and Conclusions

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 and energy consumption.

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 consumption captured by PowerMon2 are often higher than the ones provided by NI and DCM, 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 overhead and even some internal components. This result tends to show that PowerMon2 is less accurate when used to measure too high power consumptions (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. This paper pointed out that, contrary to the external measurements, the internal values are more dispersed and generate many outliers samples. This is certainly due to the high frequency of internal wattmeters. However the question that remains is whether this is because a very high frequency allows to measure some power fluctuations that the external wattmeters cannot provide or, instead, it is because it generates too much noise.

Unlike the power profiles obtained with the external wattmeters, the ones registered by the internal devices show some differences and a noise more visible with the NI. This behavior can be explained by the very high sampling rate of 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 and potential environment changes like the room temperature. 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 provoke some noise that masks 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 lines (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 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 consumption during the execution of a benchmark. A key to achieve accurate and reliable measurements requires a calibration with oscilloscope and previous 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 as these equipments are not easy to connect and the price per node is expensive (e.g., 2,700 € for NI). In this study, we focused on power measurement devices but other ways to measure the power consumption exist, like the IPMI (Intelligent Platform Management Interface) and RAPL (Running Average Power Limit) counters available on INTEL_DESKTOP. These internal sensors are promising techniques to measure power consumption for large scale distributed systems since they do not require an extra device. However, they are still not adapted since the power measurements they provide are not yet very accurate and their sample rate can be relatively low; moreover, they may introduce noise. From this paper, some open questions remain unsolved. Thus, as a follow-up to this work, we will further investigate to get the answers to some of the open questions that we expressed in the paper.

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 project CICYT TIN2011-23283 and FEDER.

References

1. Hsu, C.H., Feng, W.C., Archuleta, J.S.: Towards efficient supercomputing: a quest for the right metric. In: Proceedings of the High Performance Power-Aware Computing, Workshop (2005)
2. Dongarra, J., et al.: The international ExaScale software project roadmap. Int. J. High Perform. Comput. Appl. **25**(1), 3–60 (2011)
3. Feng, W., Feng, X., Ge, R.: Green supercomputing comes of age. IT Prof. **10**(1), 17–23 (2008)
4. Laros III, J.H., Pedretti, K.T., Kelly, S.M., Shu, W., Vaughan, C.T.: Energy based performance tuning for large scale high performance computing systems. In: Proceedings of the Symposium on High Performance Computing, HPC '12, San Diego, CA, USA, pp. 6:1–6:10 (2012)
5. Alonso, P., Dolz, M.F., Igual, F.D., Mayo, R., Quintana-Ortí, E.S.: DVFS-control techniques for dense linear algebra operations on multi-core processors. Comput. Sci. - R&D **27**(4), 289–298 (2012)
6. Alonso, P., Dolz, M.F., Mayo, R., Quintana-Ortí, E.S.: Energy-efficient execution of dense linear algebra algorithms on multi-core processors. In: Cluster Computing, May 2012
7. Feng, W., Cameron, K.: The green500 list: encouraging sustainable supercomputing. Computer **40**(12), 50–55 (2007)

8. Ltaief, H., Luszczek, P., Dongarra, J.: Profiling high performance dense linear algebra algorithms on multicore architectures for power and energy efficiency. Comput. Sci. **27**(4), 277–287 (2012)
9. Subramaniam, B., Feng, W.: 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
10. Ge, R., Feng, X., Song, S., Chang, H.C., Li, D., Cameron, K.W.: Powerpack: energy profiling and analysis of high-performance systems and applications. IEEE Trans. Parallel Distrib. Syst. **21**(5), 658–671 (2010)
11. Subramaniam, B., Feng, W.: Statistical power and performance modeling for optimizing the energy efficiency of scientific computing. In: Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications, GREENCOM, Washington, DC, USA, pp. 139–146. IEEE Computer Society (2010)
12. Alonso, P., Badia, R.M., Labarta, J., Barreda, M., Dolz, M.F., Mayo, R., Quintana-Ortí, E.S., Reyes, R.: Tools for power-energy modelling and analysis of parallel scientific applications. In: 41st International Conference on Parallel Processing - ICPP, pp. 420–429 (2012)