Comput Sci Res Dev (2015) 30:115–124
DOI 10.1007/s00450-014-0266-8

SPECIAL ISSUE PAPER

A power measurement environment for PCIe accelerators

Application to the Intel Xeon Phi

Francisco D. Igual · Luis M. Jara · José I. Gómez-Pérez · Luis Piñuel · Manuel Prieto-Matías

Published online: 5 July 2014

Abstract We describe and validate a complete hardware/software environment for power consumption analysis of PCIe-based accelerators, using the Intel Xeon Phi coprocessor as the target platform. Our environment is flexible and affordable—based on commodity instrumentation—and provides both accuracy and transparency for the user, which enables an easy instrumentation of existing codes from the power consumption perspective. We present empirical power traces for two well known scientific codes (LINPACK and `libflame`) that give insights not only on the benefits of the presented environment, but also on the power profile of the Intel Xeon Phi co-processor under different workloads.

Keywords Measurement environments · Hardware accelerators · Power-efficent HPC · Intel Xeon Phi

1 Introduction

Understanding the behavior of a given architecture not only from the performance perspective, but also considering energy consumption, has become crucial in the road towards the exaflop barrier, even more with the advent of heterogeneous architectures, in which different computing units present dramatically different performance and power profiles. For that to happen, robust measurement environments that provide accurate and detailed power consumption profiles are mandatory.

In this paper, we present and validate a complete hardware/software infrastructure for accurate power measurement of architectures based on PCIe accelerators. We describe the environment from two perspectives: (1) hardware elements and measurement instrumentation deployed in our system, and (2) software stack used to profile real applications using the aforementioned hardware elements. The validation is performed on an Intel Xeon Phi 5110P, comparing it with the power consumption information reported by the Intel SMC tool.

The environment is built around commodity measurement instrumentation (a multi-channel DC power analyzer/datalogger with SCPI—*Standard Commands for Programmable Instruments*—[1] capabilities), and reports isolated information of all power sources for PCIe accelerators, including power lines of the PCIe bus (12V and 3.3V) and external 12V connectors.

From the software perspective, we complement the aforementioned environment by extending the `pmlib` infrastructure [4] to communicate with a central measurement unit via standard SCPI commands. This new functionality allows a transparent replacement of the selected central measurement unit by virtually any other instrumentation equipment, provided it offers SCPI capabilities, which provides flexibility depending on the desired monitoring necessities.

A number of previous works have addressed power consumption analysis or measurement environments for PCIe accelerators, mostly targeting GPUs. [7] proposes a power consumption comparison between CUDA and OpenCL for

F. D. Igual (✉) · L. M. Jara · J. I. Gómez-Pérez · L. Piñuel · M. Prieto-Matías
Depto. Arquitectura de Computadores y Automática,
Universidad Complutense de Madrid, Madrid, Spain
e-mail: figual@ucm.es

L. M. Jara
e-mail: luismigueljara@ucm.es

J. I. Gómez-Pérez
e-mail: jigomez@ucm.es

L. Piñuel
e-mail: lpinuel@ucm.es

M. Prieto-Matías
e-mail: mpmatias@ucm.es

legacy GPU architectures, using a PCIe extender to measure exclusively power drawn through the 12V lines of the PCIe bus, and using datalogging capabilities to trace accelerated codes. On the contrary, [9] completely ignores the power supplied through the PCIe bus and only considers external 12V connectors to profile the energy consumption of the FFT on heterogeneous platforms. Authors in [2] present an accurate power simulator for GPGPU, and support their estimated results with empirical data using a complex ad-hoc power measurement environment that considers all power sources of modern GPUs, including that provided through 12V and 3.3V lines of the PCIe slot. As of today, few works have addressed the Intel Xeon Phi as the target platform for power consumption analysis. [8] models the energy profile of an instrumented beta version of a Xeon Phi card, and validates it using microbenchmarks, using a data acquisition system to collect power samples.

The main advantages of our environment can be summarized as:

Modularity. We base our environment on commodity instrumentation, and virtually all components can be replaced by alternative ones attending considerations such as cost, availability or desired precision and number of power sources to be analyzed.

Precision and simplicity. Attaining a similar precision and sampling frequency using ad-hoc instrumentation than that offered by ready-to-use modern instrumentation would require complex setups.

Cost/Time to setup ratio. We use standard instrumentation material, usually available in general laboratories. This greatly reduces the time to setup a reliable environment, reducing overall acquisition cost.

Flawless integration in existing codes. The extension of `pmlib` to support direct communication via SCPI commands enables a straightforward integration of power profiles into execution traces, being essentially transparent for the developer.

Although we focus on the evaluation of the power profile of the Xeon Phi, once validated, the infrastructure can be easily adapted or extended to support other types of PCIe-based accelerators (e.g. GPUs or FPGAs), and even hybrid or non-accelerated architectures.

With this seminal paper, we do not intend to expose detailed power profile insights of a given PCIe architecture (e.g. the Intel Xeon Phi), but to use it as a suitable platform to expose the design, implementation, and, more important, validation of a fundamental empirical environment for power measurement of modern parallel architectures, which is the main contribution of our work. We will,s however, give some high-level insights on fundamental properties ofs the Intel Xeon Phi from the energy consumption perspective.

The rest of the paper is structured as follows. Section 2 outlines the main architectural and power-related features of the Intel Xeon Phi co-processor. In Sects. 3 and 4, we describe the main components of the hardware and software power measurement infrastructure, respectively. Section 5 reports the obtained power measurements using our infrastructure and the Intel Xeon Phi as the target architecture. Section 6 presents the final closing remarks and open research lines.

2 Intel Xeon Phi: architectural, electrical and power monitoring features

The Intel Xeon Phi is a many-core co-processor featuring up to 61 × 86 cores in its current generation, and extended vector units (512-bit), with a fully coherent L2 cache. It can support up to 8 memory controllers, each one with two GDDR5 channels. The co-processor is connected to the host through a PCIe Gen2 bus.

The reason behind the selection of the Intel Xeon Phi as the target platform for the evaluation of our environment is three-fold: first, the Intel Xeon Phi provides power consumption information via the Intel SMC [3] (System Management Controller), that gathers power samples from on-board power sensors via I2C, and reports isolated measurements for PCIe and external 12V connectors, that will allow us to validate the measurements gathered by our environment. This level of detail is not exposed by other power monitoring software (e.g. NVML from NVIDIA). Second, unlike GPUs, the Xeon Phi can be run as a completely standalone computing system. This allows us to run our experiments using exclusively the resources of the co-processor, which is useful for our validation purposes. Third, as of today, there are few works thoroughly analyzing the power consumption profile of the Intel Xeon Phi under scientific HPC applications. Thus, our environment will be of wide appeal to those interested in the power behavior of the co-processor.

Focusing on electrical specifications and power consumption, the Intel Xeon Phi 5110P board is powered through two different channels: the PCIe slot (3.3V and 12V lines, sourcing up to 75W), and two 12V external connectors (sourcing up to 225W), for a peak power draw of 300W. In practice, the card can dissipate up to 245W according to the Intel documentation.

3 Measurement environment: hardware side

3.1 Environment setup: general overview

Figure 1 shows a schematic description of the environment (left), and its actual physical setup (right) on a real server featuring an Intel Xeon Phi. Conceptually, the environment

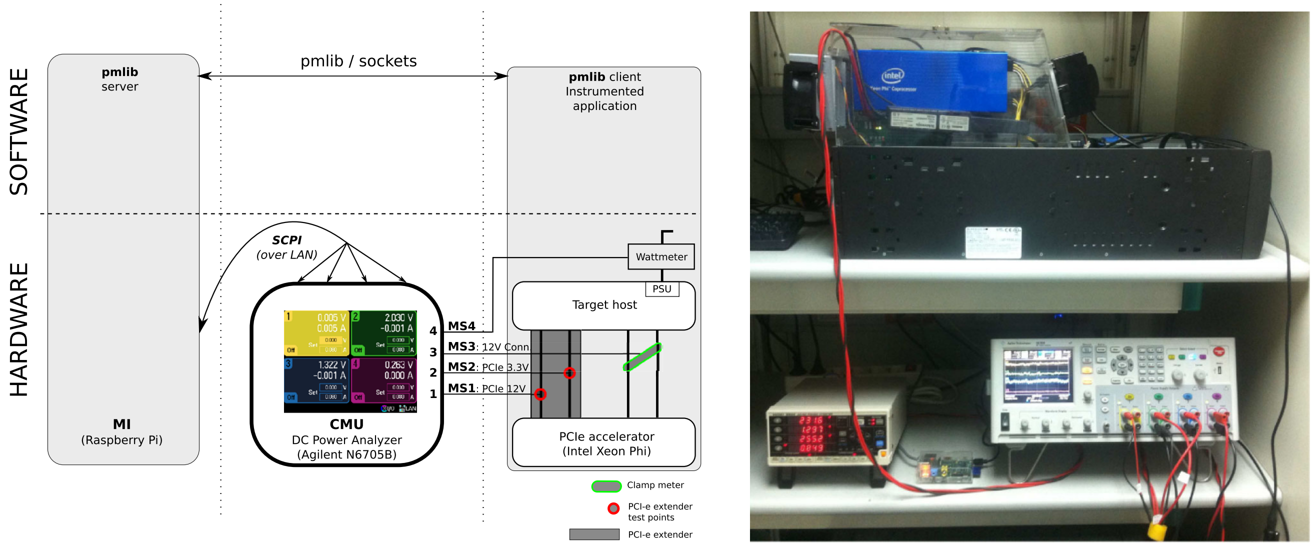


Fig. 1 *Left* Schematic overview of the HW/SW power measurement environment. *Right* Power measurement environment: N6705B power analyzer (*bottom right*), Raspberry Pi (*bottom middle*) and HIOKI Wattmeter (*bottom left*). On top, target host with the Intel Xeon Phi attached to the PCIe extender and clamp meter measuring 12V PCIe external lines

Table 1 Summary of power gathering devices (MS#) connected to the central DC power analyzer (CMU)

Channel	Source	Model	Connected to	Value	Range	Target
N6705B						
1	PCIe extender	FURAXA PCIeEXT16HOT	12V PCIe line	DC current	1 V/A	PCIe
2	PCIe extender	FURAXA PCIeEXT16HOT	3.3V PCIe line	DC current	1 V/A	
3	Clamp meter	Tektronix A622	12V ext. connectors	DC current	100 m V/A	
4	Wattmeter	HIOKI HiTester 3334	AC Power Supply	AC power	±1 V f.s.	Node

is built around a *Central Measurement Unit* (CMU in the following), that typically features a number of input channels, each one connected to a different *Measurement Source* (MS). Besides, there is a *Measurement Interface* (MI), that acts as an interface between the target architecture to be analyzed and the analysis environment.

The environment is intended to be fully modular; thus, it is possible to choose different specific instrument models for each component, depending on availability, desired precision, or number of sources to be analyzed. For our specific setup, we have chosen the following main components to illustrate the potential of such an environment:

CMU: Multi-channel precision DC power analyzer with SCPI capabilities *(Agilent N6705B)*.
MS1 and MS2: PCI-Express extender with independent current reporting capabilities for 12V and 3.3V PCIe power supply lines *(FURAXA PCIeEXT16HOT)*.
MS3: Clamp meter connected to the external 12V connectors of the PCIe card *(Tektronix A622)*.
MS4: (Optional) AC/DC Wattmeter for node AC power analysis *(Hioki AC/DC HiTester 3334)*.
MI: Trace server *(built on a Raspberry Pi)*.

3.1.1 Central measurement unit (CMU)

The environment is built on top of a commodity DC power analyzer with datalogging capabilities, that gathers power readings from each of the desired sources to be analyzed. We have selected an Agilent N6705B DC power analyzer, but a simpler equipment (e.g. a multi-channel oscilloscope) would suffice, provided it features:

1. Enough *channels* to fit the needs of the target platform to be measured. Armed with the appropriate measurement sources, the N6705B gathers power readings from: (1) 12V connectors to the card; (2) PCIe 12V line; (3) PCI 3.3V line; (4) overall node power readings. Table 1 provides an overview of the devices connected to each channel of the power analyzer providing power measurements (MS1 to MS4).

2. Enough *sampling frequency* to fit the application necessities. Our CMU features a peak sampling frequency of 50 kSA/s (samples per second) when polling from the four available channels simultaneously. The election of other measurement devices will ultimately depend on the desired sampling frequency, according to the target application to be monitored.
3. *SCPI capabilities*, that allow an easy and portable mechanism to control and interact with the CMU, including power measurement gathering from remote software components through different communication interfaces. In our case, the interaction with the instrument is performed via Ethernet using the corresponding SCPI commands from the MI.

3.1.2 PCIe internal power lines (MS1 and MS2)

Modern PCIe cards use the 3.3V, 3.3VAUX, and 12V power lines from the PCIe bus as power sources. To isolate and analyze the current through each of these lines, it is necessary to use a device that reports isolated and accurate current measurements for each line.

The PCIeEXT16HOT device from FURAXA is a PCIe Gen3 compliant bus extender that exposes isolated test points reporting instantaneous current through each one of the three power lines in the connector. This information is reported in terms of 1 V/A, so the determination of the instantaneous power on each of the lines is straightforward. The Intel Xeon Phi only uses the 3.3V and 12V lines of the PCIe bus, so our environment gathers measurements of these two sources through two of the available channels in the CMU.

3.1.3 External 12V connectors (MS3)

Clamp meters are a convenient solution for measuring AC/DC current using Hall-effect technology; in our case, they provide an easy, accurate and portable solution to measure current on the 12V lines that supply power to the PCIe card. The Tektronix A622 clamp measures current in ranges form a few milliamperes to over 100A at 100 kHz. In our case, the clamp embraces both 12V power lines simultaneously, and is connected to a second channel of the N6705B analyzer to report real current on the external lines.

3.1.4 Overall node power consumption (MS4)

The HIOKI HiTester 3334 AC/DC power tester is a single-phase wattmeter able to report measurements of voltage, current, active power, apparent power and power factor, among others. These values can be displayed on the meter screen or queried via SCPI through RS-232, with a relatively poor resolution (200 ms). To overcome this limitation, the HIOKI 3334 provides an analog output (referred as *waveform output*) that converts instantaneous AC voltage, current and/or power waveforms to ±1 V f.s. (*full scale*) at the output, with a conversion interval of approximately 13 μsec. In our case, the HIOKI wattmeter measures AC power consumption at the entrance of the node power supply; these data are gathered by the fourth channel of the CMU in order to accurately estimate the instantaneous power consumption of the complete node. While we have added this component to report overall node power consumption, we consider it as an additional piece of hardware for our environment, not strictly necessary to measure the power dissipated by the PCIe card.

3.1.5 Measurement interface (MI)

The trace server is an interface between the CMU and the target system to be measured. On one side, it continuously queries the CMU for power measurements via standard SCPI commands (as explained in Sect. 4); on the other, it accepts remote connections from the target host/application querying for those power measurements and serves them under request.

For the sake of power saving, we have employed a Raspberry Pi board as a trace server. While it shows enough processing power to handle both connections, it can be easily replaced by a more powerful machine if necessary. Even the target host can act as a trace server itself, communicating directly with the measurement unit; however, this introduces a non-negligible overhead that would disturb the overall measured system, both for performance and for energy measurements.

3.2 Thermal considerations

The Intel Xeon Phi 5110P features a passive cooling system, and thus requires strict cooling guidelines and airflow restrictions [3]. The elevation introduced when using the PCIe extender naturally breaks the optimal chassis airflow, leading to undesired thermal issues. Our environment requires a careful redesign of the airflow, using an ad-hoc annex to the chassis as seen in Fig. 1 (right) to keep the board correctly refrigerated. In our tests, we have observed no dramatic differences in temperature between the original and the modified chassis, and a completely stable execution environment. Alternative Xeon Phi models featuring active cooling would not require such a modification.

4 Measurement environment: software side

4.1 General overview and goals

The goal of the proposed environment is not only to obtain accurate and detailed power consumption measurements of

PCIe devices, but to be able to instrument existing codes on a target platform, obtaining power profiles of them as a whole, or focusing on specific code sections. This integration of power measurements needs to be essentially transparent for the programmer, including synchronization, data transfer and accuracy management. We describe next the main software components used to provide such functionality using the aforementioned hardware framework.

4.2 The pmlib framework

pmlib [4] is a software infrastructure that eases the process of measuring energy consumption and instrumenting applications running on different target architectures. In previous works, the developers of the library have mainly used pmlib combined with ad-hoc or simple commercial AC meters (e.g. Watts-up) in order to estimate the energy consumption of a number of target platforms. pmlib is a client-server framework in which the server module directly communicates with the power analysis instrumentation, and the client module (that can be embedded on the target code though a provided API) queries for gathered power measurements, synchronizing the obtained data with the execution of the target program. pmlib is fully modular and extensible; our effort extends the framework to interact with commodity instrumentation via standard SCPI commands, as described next.

4.3 The SCPI standard: SCPI support in pmlib

SCPI is a standard that allows remote controlling of test and measurement devices, including multimeters, power analyzers and oscilloscopes. SCPI does not define a physical layer, so it can be used in conjunction with virtually any communication interface (e.g. GPIB, USB, Ethernet—as in our case—or RS-232). SCPI provides standard commands to control virtually every single functionality of the target instrument. In our case, the usage of SCPI provides full flexibility to the environment, as the MI (and thus the pmlib server) will be able to communicate with a number of different CMU models, provided they support the standard (as many measurement instrumentation do nowadays).

We have revamped the pmlib server running on the MI to directly interact with the CMU using standard SCPI commands. For that to happen, the pmlib server running on the MI establishes a connection with the remote CMU when it is started, and continuously gathers measurement samples from it via SCPI.

The sampling procedure is divided into two stages: *sampling* (triggering a measure and storing it in the CMU internal buffers), and *fetching* (retrieving obtained data to the MI). Usually, the fetching stage takes the major part of the overall process. However, it is possible to force the CMU to transfer data using a predefined binary format instead of the default ASCII format, which increases performance up to an order of magnitude. Figure 2 illustrates the effective sampling frequency (time from source to requester) attained using binary or ASCII transfers, for an increasing size of samples requested per trigger. The effective sampling frequency using binary transfers varies between 15 kSA/s and 20 kSA/s, depending on the number of channels queried. A similar experiment using ASCII transfers yields a performance between 0.5 kSA/s and 1.5 kSA/s, which demonstrates the importance of using a correct fetching mechanism.

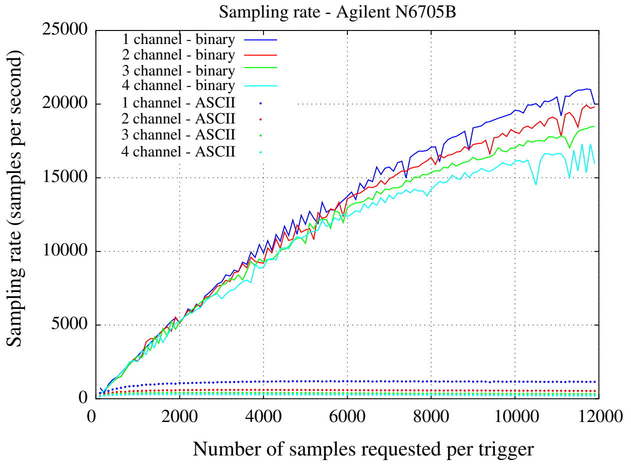


Fig. 2 Number of read samples per second using the SCPI standard and binary or ASCII data fetch

Besides the communication with the CMU, the second function of the MI running the pmlib server is to provide fetched measurements to the target application upon request. The application to be profiled on the target architecture (in our case, the Intel Xeon Phi) establishes a connection to the MI and starts, stops, pauses and resumes power measurement to profile sections of interest in the code. This process is transparent and non-intrusive by using the pmlib C API; results can be retrieved as raw text, CSV format, or in a format compliant with extended performance tracing mechanisms.

5 Environment validation on the Intel Xeon Phi

5.1 Environment validation using datalogging

In order to validate the measurement environment from the hardware point of view, we will exclusively consider the capabilities of the CMU, comparing the obtained results with those provided by the Intel SMC. Thus, for this first step, we do not integrate the measurement management into pmlib yet, but use exclusively the datalogging capabilities of the analyzer.

The validation test consists on the execution of the LINPACK [6] test (a *de-facto* standard for evaluating performance and energy consumption on HPC architectures) running in native mode, that is, using exclusively the Intel Xeon Phi for the complete test with no intervention from the host. At the starting point of the test, we synchronize the execution of a procedure that gathers power consumption measurements from Intel SMC, and simultaneously triggers a datalogging procedure on the `CMU` using the four available channels. Ideally, this would result on synchronized traces of instantaneous power readings using valid measures (those provided by Intel SMC) and actual measures (those provided by our environment).

The plots in Fig. 3 provide a comparative report of the aforementioned traces, dividing the study in terms of each one of the measurement sources. "Peaks" in the power trace indicate actual LINPACK executions for increasing problem sizes. Zones with less power activity correspond to problem preparation, that is a purely sequential section of code. More precisely, Fig. 3a, b compares the power consumption rates provided by both mechanisms considering the PCIe bus; Fig. 3c reports similar results for the analysis of the 12V external connectors. Fig. 3d joins both measurements, and provides a complete power measurement comparative analysis of the overall consumption of the card. In addition, the plot includes the overall power draw from the node power supply offered by `MS4`. In all plots, we have added the theoretical maximum power draw of the corresponding source [PCIe power lines (Fig. 3a, b), 12V connectors (Fig. 3c) and Intel Xeon Phi peak power draw (Fig. 3d)]. In the test, we have used a sampling period of 50 ms for the Intel SMC measurements and 1 ms for the measurements taken by the `CMU`; execution time of the test is limited to 200 seconds, for the sake of clarity. We use 240 threads (the maximum available) for the execution of the LINPACK test.

Comparing both power traces, and considering the Intel SMC measurements as a valid reference, it is possible to conclude that the environment provides accurate measurements. For the integrated trace including all power sources of the PCIe card (Fig. 3d) we have observed a deviation in energy consumption of less than 1 % comparing both mechanisms for the LINPACK test, which is acceptable. The gathered results for the 12V external lines mimics that of the Intel SMC measurements in all cases, with the clear advantage of a dramatic increase in the sampling frequency. There is a non-negligible deviation in the PCIe power lines analysis, see Fig. 3b; from our observations, the Intel SMC mechanism does not provide full report on the PCIe power consumption, but only that related to the 12V line. Isolating exclusively this line (Fig. 3a) in our environment once more mimics the reference behavior. While the power provided by the 3.3V line is relatively small and constant (in our observations, around 4.5W independently from the load of the co-processor), this observation is key to actually trust on the Intel SMC measurements provided by Intel. For the sake of illustrating the validity of the environment, the overall Intel Xeon Phi power consumption in Fig. 3d does not include the 3.3V line on the PCIe measurement by our environment.

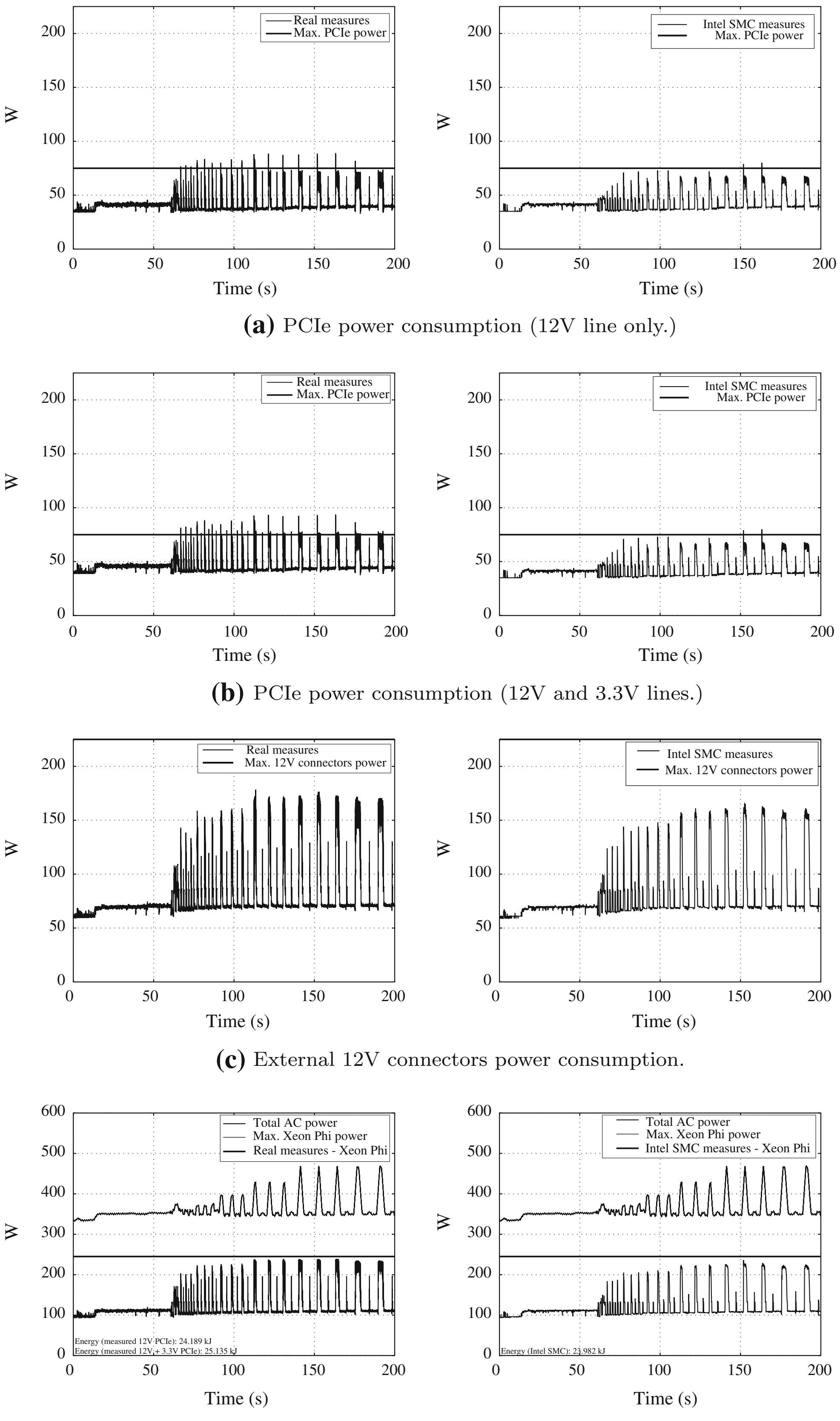
There are a number of additional advantages in using an environment like the one presented:

- The sampling frequency attained by Intel SMC is, in most cases, not acceptable. For the Intel Xeon Phi, Intel SMC updates power information every 50 ms; in our case, datalogging using our selected `CMU` attains a maximum sampling frequency of 50 kSA/s.
- By comparing the attained AC node power and DC PCIe power results (Fig. 3d), it is clear that overall AC power measuring hides many details of the internal PCIe consumption, e.g. on fast power variations. This effect would be even more evident with hybrid executions (in which power contributions of different components would be completely hidden), or in platforms with multiple accelerators. In those cases, the use of independent power monitoring systems becomes mandatory.
- Accurately measuring the power consumption of the Intel Xeon Phi in *idle* state is only possible using an external measurement mechanism. In our experiments, we have observed severe influence of the Intel SMC measurement execution on the power status of the accelerator, increasing its consumption even though it runs on the host side.In an additional experiment (not shown for space reasons) gathering measurements from both sources during 100 seconds, the idle Intel Xeon Phi dissipates an average power of 94.05W when querying via Intel SMC, and 41.66W when using our external environment.

5.2 Experimental results using `pmlib`

Although the LINPACK test is a reference for performance and power characterization of HPC facilities, the distribution for the Intel Xeon Phi is not open source, and it is difficult (if not impossible) to instrument the code to demonstrate the capabilities of our `pmlib` port. We will carry out our experiments by porting and instrumenting the `libflame` library [10] to the Xeon Phi.

`libflame` is a complete dense linear algebra library providing much of the functionality of LAPACK. To target multi-/many-core architectures, the library integrates SuperMatrix [5], a runtime task scheduler mechanism that exploits task parallelism managing data dependencies between tasks, and mapping them to available computing resources (e.g. cores) as data dependencies are satisfied. The user can easily modify the number of working threads consuming tasks, and the granularity of those tasks (that is, dimension of the matrices sub-blocks, *nb*). In our case, the port of the library



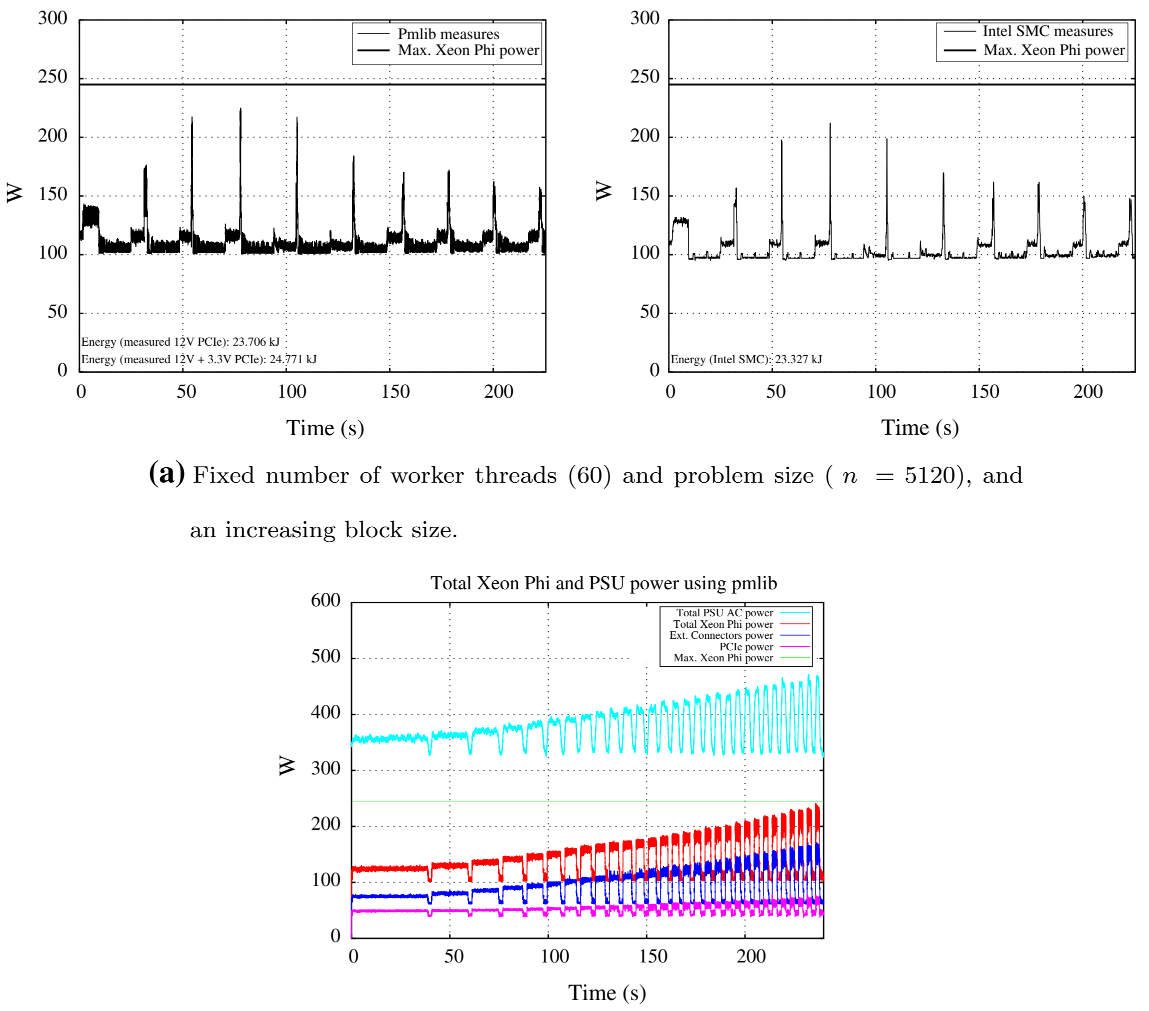
(a) PCIe power consumption (12V line only.)

(b) PCIe power consumption (12V and 3.3V lines.)

(c) External 12V connectors power consumption.

(d) Intel Xeon Phi and Server power consumption.

Fig. 3 Comparison between datalogging on the `CMU` and Intel SMC measurements for LINPACK in the Intel Xeon Phi



(a) Fixed number of worker threads (60) and problem size (n = 5120), and an increasing block size.

(b) Increasing number of worker threads and a fixed problem size (n = 10240) and block size (nb = 1024).

Fig. 4 Intel Xeon Phi power profiles running different experiments for parallel DGEMM using SuperMatrix

and runtime runs natively on the Intel Xeon Phi, deploying as many worker threads as cores are in the accelerator using OpenMP; in order to exploit the multi-threading capabilities of the Intel Xeon Phi cores (4 hardware threads per core), tasks consist of calls to the corresponding multi-threaded MKL kernels, using nested parallelism and 4 threads for the task execution. MKL threads are mapped to the same core as that running the associated SuperMatrix worker thread.

We have conducted two different experiments running SuperMatrix on the Intel Xeon Phi, both performing a parallel double precision matrix-matrix multiplication (DGEMM) for square matrices on the co-processor.

5.2.1 Validation of the `pmlib` *environment*

The goal of the first experiment is to validate our `pmlib` integration with the measurement environment, providing a comparison with Intel SMC measurements. For this experiment, we fix the problem size ($n = 5{,}120$) and number of worker threads (60) in SuperMatrix for DGEMM, and use increasing task dimensions (starting at $nb = 128$ up to 1,280, in steps of 128). We have instrumented the `libflame` code in order to invoke `pmlib` API calls to start and finalize power counters within the test driver, and additionally, to gather measurements from Intel SMC. A comparison between both power traces is reported in Fig. 4a. In this case, the trace for the empirically measured values includes 12V and 3.3V lines from the PCIe slot, which SMC does not report. From the accuracy perspective, as with datalogging (Sect. 5.1), the environment provides power readings close to those from Intel SMC (with a mean deviation of roughly 1 % in overall energy consumption when considering exclusively the 12V PCIe lane), which clearly validates the use of `pmlib` to trace existing codes. Note that, for this experiment, the plot is reporting around 3500 power samples for SMC readings (around 15 SA/s), and roughly 5 million samples using our environment (around 22 kSA/s).

5.2.2 Impact of core utilization on the Intel Xeon Phi

The second experiment uses a fixed problem and block size ($n = 10{,}240$, $nb = 1{,}024$) and gradually increases the number of worker threads (that is, the effective number of cores running on the Intel Xeon Phi) from 2 to 60 in steps of 2. Attained power consumption results are shown in Fig. 4b, breaking down the overall power consumption of the accelerator into 12V external connectors and complete power through the PCIe lines. In addition, it reports the overall power consumption of the node gathered from the HIOKI wattmeter.Note the impact of increasing the number of used cores on the overall node power consumption in general, and on each one of the lines supplying power to the Intel Xeon Phi in particular; in fact, the card reaches its peak dissipation close to the peak reported by the manufacturer (around 95 % of the nominal 245W) only when the 60 available cores are running at full load.

A similar effect regarding core usage could be observed in the first experiment: as the task dimension increases, MKL performs better on each core, and core utilization increases, till reaching a peak at $nb = 512$ (fourth power bump in the plot); from that point on, the potential parallelism decreases (there are less potentially ready tasks), and the amount of cores effectively performing useful work decreases, as does power consumption. This behavior is common in many task scheduling runtime systems considering performance, and, from our observations, energy curves follow the same pattern.

6 Conclusions and perspectives

We have introduced a modular hardware/software power measurement environment for PCIe-based accelerators that leverages standard measurement instrumentation and transparent power tracing capabilities from existing accelerated codes. We have validated the environment on an Intel Xeon Phi, comparing it with available mechanisms for power measurement. Our environment is robust, modular and transparent for the developer, attaining sampling ratios that cannot be achieved with existing measurement mechanisms (e.g. Intel SMC).

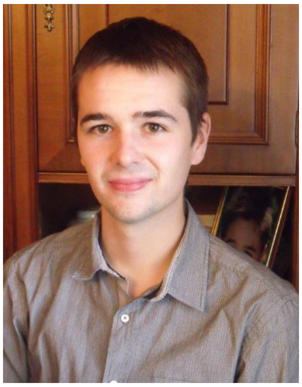
The proposed environment opens a number of future research lines involving not only the Intel Xeon Phi but other types of accelerated, non-accelerated and hybrid systems. Our future research will leverage the capabilities of the environment to characterize and compare different workloads, data transfer and power saving strategies on the Xeon Phi and/or GPUs from the power perspective; we will also extend it to support hybrid accelerated architectures—analyzing 12V power rails on the motherboard using an analogous infrastructure.

Acknowledgments This work is supported by Projects CICYT-TIN 2008/508 and 2012-32180. We thank the technical staff at DACYA for their support in setting up the measurement environment.

References

1. (1999) Standard commands for programmable instruments, vol 1. SCPI Consortium
2. (2013) How a single chip causes massive power bills. GPUSimPow: a GPGPU power simulator. In: ISPASS 2013, IEEE, pp 97–106
3. (2013) Intel Xeon Phi Coprocessor Datasheet. Intel Corporation
4. Barrachina S, Barreda M, Catalán S, Dolz MF, Fabregat G, Mayo R, Quintana-Ortí ES (2013) 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, pp 114–119
5. Chan E, Quintana-Ortí ES, Quintana-Ortí G, van de Geijn R (2007) SuperMatrix out-of-order scheduling of matrix operations for SMP and multi-core architectures. SPAA '07. San Diego, CA, USA, pp 116–125
6. Dongarra JJ, Luszczek P, Petitet A (2003) The linpack benchmark: Past, present, and future. CC: P&E 15:2003
7. Jackson J (2013) Comparison and analysis of GPU energy efficiency for CUDA and OpenCL
8. Shao Y, Brooks D (2013) Energy characterization and instruction-level energy model of Intel's Xeon Phi processor. In: Low power electronics and design, pp 1–6
9. Ukidave Y, Ziabari AK, Mistry P, Schirner G, Kaeli D (2013) Quantifying the energy efficiency of FFT on heterogeneous platforms. In: ISPASS 2013, IEEE, pp 235–244
10. Zee FGV (2009) libflame: the complete reference. http://www.lulu.com

Francisco D. Igual received his BS degree from the University Jaume I in Castellon in 2006, and his MS degree from the same University in 2007. He obtained his Ph.D. in Computer Science in 2011 from the Department of Computer Science and Engineering of the University Jaume I. His research interests include the optimization and adaptation of numerical libraries to platforms with one GPU or multiGPU systems, runtime support for automatic parallelization of numerical codes, and development of accelerated routines for distributed memory architectures. Additionally, he is involved in the implementation and optimization of biomedical and image processing applications on hardware accelerators.



Luis M. Jara is a student of Electronic Engineering at the Complutense University of Madrid, Spain. He also has a bachelor degree in Physics by the same University, obtaining the speciality in Device Physics and Control. He has been doing the final degree project in the Computer Architecture department.

José I. Gómez-Pérez is an assistant professor at Complutense University of Madrid, where he earned his Ph.D. in 2007. His current research interests include energy efficient computing in massively parallel architectures with special focus in the management of the memory system.

Luis Piñuel received the Ph.D. degree in computer science from the Universidad Complutense de Madrid (UCM) in 2003. He is an associate professor in the Departamento de Arquitectura de Computadores y Automática, UCM. His research interests include computer architecture, high-performance computing, low-power microarchitectures, embedded systems, and resource management for emerging computing systems.

Manuel Prieto-Matías received the Ph.D. degree in computer science from the Complutense University of Madrid (UCM) in 2000. He is now associate professor in the Department of Computer Architecture at UCM and serves as vice-dean for External Relations and Research at the School of Computing Science. His research interests include areas of parallel computing and computer architecture. Most of his activities have focused on leveraging parallel computing platforms and on complexity-effective microarchitecture design. His current research addresses emerging issues related to herogeneous systems and energy-aware computing, with a special emphasis on the interaction between the system software and the underlying architecture. He has co-written numerous articles in journals and for international conferences in the field of parallel computing and computer architecture.