Sustainable Computing: Informatics and Systems xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Understanding hardware and software metrics with respect to power consumption

Julian Kunkel[a], Manuel F. Dolz[b,*]

[a] German Climate Computing Center, DKRZ GmbH, 20.146 Hamburg, Germany
[b] Dept. of Computer Science, University Carlos III of Madrid, 28.911 Leganés, Spain

ARTICLE INFO

Article history:
Received 14 June 2016
Received in revised form 9 July 2017
Accepted 31 October 2017
Available online xxx

Keywords:
HPC
Data analysis
Power modeling
Statistical methods
Performance counters
Energy consumption

ABSTRACT

Analyzing and understanding energy consumption of applications is an important task which allows researchers to develop novel strategies for optimizing and conserving energy. A typical methodology is to reduce the complexity of real systems and applications by developing a simplified performance model from observed behavior. In the literature, many of these models are known; however, inherent to any simplification is that some measured data cannot be explained well. While analyzing a models accuracy, it is highly important to identify the properties of such prediction errors. Such knowledge can then be used to improve the model or to optimize the benchmarks used for training the model parameters. For such a benchmark suite, it is important that the benchmarks cover all the aspects of system behavior to avoid overfitting of the model for certain scenarios. It is not trivial to identify the overlap between the benchmarks and answer the question if a benchmark causes different hardware behavior. Inspection of all the available hardware and software counters by humans is a tedious task given the large amount of real-time data they produce.

In this paper, we utilize statistical techniques to foster understand and investigate hardware counters as potential indicators of energy behavior. We capture hardware and software counters including power with a fixed frequency and analyze the resulting timelines of these measurements. The concepts introduced can be applied to any set of measurements in order to compare them to another set of measurements. We demonstrate how these techniques can aid identifying interesting behavior and significantly reducing the number of features that must be inspected. Next, we propose counters that can potentially be used for building linear models for predicting with a relative accuracy of 3%. Finally, we validate the completeness of a benchmark suite, from the point of view of using the available architectural components, for generating accurate models.

1. Introduction

Power and energy consumption have been identified as the single largest challenges in the design of future Exascale high-performance computing (HPC) systems [1]. Basically, the vast increase of levels of parallelism to the point of millions of processors working concurrently is a challenge that will need radical changes in hardware and software design (e.g., programming models, compilers, I/O libraries, etc.) [2]. Thus, understanding how computers use power is key to develop a new hardware and software stack in order to face the Exascale challenge.

Nevertheless, the implementation and deployment of Exascale systems calls for a holistic approach that, with the use of power and performance tracing tools and wattmeters, allows the inspection of power bottlenecks and energy hotspots of current scientific parallel software. However, acquisition costs and deployment of power measurement devices can be, due to the nature of the platforms and number of nodes, infeasible. Recent research has significantly demonstrated that a promising alternative in order to mitigate this issue is the design of power models [3–5]. Taking into account that most of the current processors feature a large set of hardware counters, temperature sensors, and resource usage statistics provided by the operating system, one could cleverly use this information to predict power drawn by individual components and system power consumption. For instance, a per-component power model could be easily exploited to make energy-aware scheduling with the aim of reducing the power consumption while preserving performance [6].

As highlighted in [7], a good power model should always be accurate, simple, inexpensive and portable. Indeed, recent works

[*] Corresponding author.
E-mail addresses: kunkel@dkrz.de (J. Kunkel), mdolz@inf.uc3m.es (M.F. Dolz).

https://doi.org/10.1016/j.suscom.2017.10.016
2210-5379/© 2017 Elsevier Inc. All rights reserved.

have shown that many power models can fulfill all these properties while providing fairly good estimations [3–5]. In this sense, a classical methodology to validate the accuracy of power models has only been carried out by calculating the preciseness and responsiveness of their predictions [8]. However, this approach may lead to optimistic results because often (i) the static power is included when computing relative accuracy, (ii) the model is applied to complete application runs where statistical effects negate individual errors, (iii) the selection of similar training and validation data may lead to optimistic errors, (iv) a few strong outliers of the model may still lead to a good relative accuracy. Thus, while analyzing model accuracy is important, we believe it is even more relevant to identify the properties of such predictors, i.e., relations between metrics in order to understand the source of these errors. Therefore, in this paper we aim at understanding and identifying behavior of hardware counters in order to explain and model power consumption. We also validate the completeness of a benchmark set, from the point of view of utilizing available architectural features, for generating accurate models. Note that we leave the building of power models part as future work.

The paper is structured as follows: First, we present related work in Section 2. In Section 3, we describe the statistical methods that are used to draw conclusions over a set of hardware counters and benchmarks. Afterwards, in Section 4 we demonstrate their use on several experiments to (i) identify properties for the highest and lowest energy consumption, (ii) localize outliers of a linear model based on the Intel RAPL (Running Average Power Limit) interface, (iii) identify different phases of the `linpack` benchmark, and (iv) investigate the impact of benchmarks that encompass a training suite for models. Finally, we conclude the paper in Section 5.

2. Related work

We classify the work related to this paper in three different categories: (i) building of power and energy models using statistical analysis using hardware counters; (ii) techniques leveraging power models to dynamically limit energy consumption; and (iii) integrated interfaces in current architectures to provide on-line power measurements.

In the literature, we find a large collection of works using statistical analysis for building power and energy models based on hardware counters. For instance, the approach by Xiao et al. [9] presents a methodology for building system-level power models based on regression analysis without the need of power measurements at component level. In this sense, the regression models describe the aggregate power consumption of the processors, the wireless network interface and the display using hardware performance counters. Similarly, Shang et al. [10] present an efficient adaptive regression-based high-level power model to estimate FPGA power consumption. In order to improve on-line power estimation accuracy, they use adaptive regression methods to lessen the problem of biased training sequences and to finally achieve a good trade-off between efficiency and accuracy. To deal with the accuracy issue, the work by McCullough et al. [11] investigated on the accuracy of hardware counters-based models and concluded that the inherent complexity of the system architectures and the current microprocessors are, in most of the cases, the root cause of the model errors. Nevertheless, they stated that hardware counters are, as of today, the only source to obtain fine-grain information about the platform.

Research leveraging power models to dynamically limit energy consumption can also be found in the literature. For instance, we encounter works using models for controlling dynamic voltage and frequency scaling (DVFS) [12] and for guiding compilers to generate energy-efficient codes [13]. Similar works have also used models to reduce power in system components other than processors, such as RAM [14] and disks [15]. Researchers have also analyzed the impact of techniques using such models to control a single knob —either DVFS or dynamic concurrency throttling (DCT) – for dynamic power management on shared-memory [16,17], and on distributed-memory parallel systems [18,19]. The work from Curtis-Maury et al. [20] differs from earlier research since it uses multiple knobs in several key aspects, such as for DVFS and DCT. To do so, this work proposes methods to generalize multi-dimensional prediction models that leverage statistical analysis for estimating how DVFS and DCT influence the performance of applications. On the other hand, we find that many hardware manufacturers have integrated hardware counters in order to provide on-line measurements and reduce the energy consumption. For example, the Intel Running Average Power Limit (RAPL) counters [21], the AMD Application Power Management (APM) interface [22] and the IBM Power7 interface [23] provide power measurements based on models and sensors of the processors. Besides, recent NVIDIA GPUs report power usage via the NVIDIA Management Library (NVML) [24]. It is important to highlight also the Intel Intelligent Platform Management Interface (IPMI) [25] which measures total server power using on-board sensors and supports the reading of additional sensors.

All in all, we conclude that the application of statistical methods for data analysis [26] in the domain of energy-efficiency has been used as a fundamental technique to derive power models. However, none of those studies has used advanced statistical methods prior ensuring that independent variables of the models are reliable and robust enough for building them. From the techniques we apply in this paper, and to the best of our knowledge, only the Pearson product-moment correlation coefficient has been previously used [27]. In this sense, the novelty of this paper lies in the use of statistical methods, normally applied to other sciences, to the specific field of power modeling. To some extent, as well, such knowledge can be used to improve the models or to develop benchmarks that aid in the design of models for future architectures. In sum, this paper uses advanced statistical methods (i) to analyze measured data in much more detail, (ii) to reveal interesting properties and (iii) to validate the completeness of a benchmark set for generating accurate models.

3. Methodology

In this section, we explain in detail the methodology used to apply the statistical methods for analysis performed in this paper. First, we give details about the HPC platform and the benchmark suite used to build the target data set. Next, we formalize this data set for further analysis and, finally, we describe three advanced statistical methods used along the paper.

3.1. Target platform and benchmark suite

Our measurements were gathered on an Intel Xeon CPU “Sandy Bridge” E31275 processor with 4 cores running at 3.40 GHz (with the `performance` governor and active Turbo Boost),[1] and 16 GB of DDR3 RAM (1333 MHz).[2] We collected the following information:

1. *Power consumption* is captured with a frequency of 20 Hz from an external ZES-Zimmer LMG450 [28], a highly advanced precision wattmeter, using the PMLib framework [29].

[1] We cover Turbo Boost on purpose, as several HPC centers enable it for specific workloads.

[2] Due to space limits, we are only able to carry out the evaluation using a single platform.