Statistical Validation Methodology of CPU Power Probes

Abdelhafid Mazouz, Benoît Pradelle, and William Jalby

University of Versailles St-Quentin en Yvelines, France
{first.last}@uvsq.fr

Abstract. Achieving or proving energy efficiency necessarily relies on the ability to perform power measurements at some point. In order to simplify power measurements at the CPU level, recent processors support model-based energy accounting interfaces such as Intel RAPL or AMD APM. Though such interfaces are an attractive option for energy characterization, their accuracy and reliability has to be verified before using them.

We propose a new statistical validation methodology for CPU power estimators that does not require any complex hardware system instrumentation. The methodology only relies on a single full-system AC power meter and is able to make statistically relevant decisions about the probes reliability. We also present an experimental evaluation using two Intel machines equipped with a RAPL interface and investigate the impact of multiple parameters such as the CPU frequency or the number of active cores on the probe accuracy.

Keywords: Statistical performance evaluation, Power measurement, RAPL.

1 Introduction

Reducing energy consumption is now a major concern for computing systems. Indeed, application power accounting, power modeling, power capping, and Dynamic Voltage and Frequency Scaling (DVFS) are now common tasks performed in data centers. All of them share a common requirement: they are all based on physical power measurements, including device-specific measurements. However, device-specific power measurements often require physical access to the device and expensive measurement probes.

In order to ease power measurements on processors, new CPU devices integrate model-based interfaces for energy consumption estimation like Intel *Running Average Power Limit* (RAPL) [16] or AMD *Application Power Management* (APM) [1]. In both systems, power or energy can be read directly by userspace software using *Model Specific Registers* (MSR). The spreading of such model-based interfaces provokes a considerable attraction for power measurement at the CPU package or at CPU core level. However, the accuracy and reliability of such interfaces has to be considered and validated before using them.

L. Lopes et al. (Eds.): Euro-Par 2014 Workshops, Part I, LNCS 8805, pp. 487–498, 2014.

Obviously, the most precise approach to validate power estimation interfaces is to add a power probe directly on the CPU. However, setting up an additional probe on the CPU is complex and requires a physical access to an experimental machine. To simplify the validation, we propose a statistical validation methodology that does not require precise instrumentation of the processor. Instead, we only use a full-system digital power measurement (DPM) device. A statistical quantification approach is then employed to overcome the limits of system-level instrumentation. Thus, the presented method simplifies and reduces the cost of CPU power probe validation while maintaining a high accuracy thanks to statistics. It is then possible to easily expose some of the limits of the power probes without having to void the hardware warranty because of the instrumentation process. Moreover, it allows anyone with a calibrated full system power meter to check the power probes before using them in a production mode.

The paper is organized as follows. Section 2 defines our protocol for statistical validation of the RAPL power estimation. Section 3 describes our experimental setup (software and hardware) and our measurement methodology. Section 4 shows experimental results on two Intel machines. Finally, we present some related work in Section 5, and conclude in the last section.

2 Validation Methodology

To mitigate the low precision of system-level instrumentation, the presented methodology is based on the execution of a large number of micro-benchmarks stressing only the CPU. Power consumption is measured for the whole execution of each micro-benchmark with various experimental configurations, no sampling is performed. The RAPL power estimation is compared against whole system power measurements obtained by a digital power meter (DPM). The intuition behind the experimental methodology is the following: the DPM measures power consumption for the whole system, including processors, then, if the RAPL interface indicates an increased power consumption, the DPM must report at least an equivalent increase.

2.1 Experimental Configuration and Validity Test

We define an experimental configuration as a set of experimental parameters, each one being either a hardware setting such as the CPU frequency, a software parameter such as the number of cores on which the benchmark is replicated on, or an environmental factor such as the system temperature. An experimental configuration could consist for instance in having a benchmark replicated on all the cores, the highest frequency set, the CPU temperature left to the ambient one. Such configuration is not related to any benchmark and can be used for many of them.

Let us consider that we have two experimental configurations $C1$ and $C2$ that differ only by a single parameter. $C2$ is such that it implies a higher CPU power consumption than $C1$. For instance, $C2$ could be a configuration similar

to $C1$ except that more CPU cores are used. We then expect that the power consumption measured when running a benchmark b under $C2$ is higher than that of $C1$, both at the CPU and system level. Let $Pcpu(C, b)$ be the power measured at the CPU level using the RAPL interface when running b with an experimental configuration C, and $Psys(C, b)$ be the power consumption reported by the DPM. $Pcpu(C, b)$ and $Psys(C, b)$ are the median of multiple measurements. Depending on the power increase observed at each level, we may encounter two different situations.

First, $Psys(C2, b) - Psys(C1, b) \geq Pcpu(C2, b) - Pcpu(C1, b)$. The power consumption increased more on the system than on the CPU. Consequently, we cannot conclude that the model-based CPU power estimation is inaccurate. Note that the CPU probes are however not proven accurate.

Second, $Psys(C2, b) - Psys(C1, b) < Pcpu(C2, b) - Pcpu(C1, b)$. Either the rest of the system power consumption decreased, or the DPM is wrong, or the RAPL interface is wrong. However, $C2$ is chosen to generate a higher CPU power consumption than $C1$, the benchmarks are built so that they only stress the CPU, and the rest of the system is kept idle by stopping all the non-essential processes. Thus, some components may increase their consumption (think about fans for instance) but cannot possibly consume less power. Moreover, the DPM is assumed to be working and correctly calibrated. Thus, there is a strong evidence that the CPU power estimation is inaccurate. Moreover, $C1$ and $C2$ differ only by a single parameter, which provides hints on the cause of the inaccuracy.

The test does not allow us to find all the potential flaws of the probe. Indeed, it only detects situations where the RAPL interface reports excessive power increase. When the power increase is under estimated, it cannot be detected because the extra power consumption measured at the system level can also be due to other components activity. Thus, the test may report false positives. However, it exposes all the situations where the power increase is over-approximated by the RAPL interface without having to perform complex hardware instrumentation.

The described test is the core of our methodology. The complete probe validation methodology is described hereafter along with the statistical tools required to decide whether the probe can be considered as accurate or not.

2.2 Statistical Significance

Let us define $\mathcal{B}$ be the set of micro-benchmarks. For each pair of experimental configurations $C1$ and $C2$, where $C2$ implies a higher CPU power consumption than $C1$, computing the set of power differences is done as follows:

1. For each micro-benchmark $b \in \mathcal{B}$, compute $\Delta Psys(b) = Psys(C2, b) - Psys(C1, b)$ and $\Delta Pcpu(b) = Pcpu(C2, b) - Pcpu(C1, b)$
2. Compute the set $\Delta P = \{\Delta Psys(b) - \Delta Pcpu(b) | \forall b \in \mathcal{B}\}$ of power increase differences between DPM measurements and model-based CPU power estimations.

The set ΔP contains the power increase differences between those reported by the DPM and the RAPL interface. Thus, checking for negative values in ΔP

is the simplest way to determine if there are cases where the RAPL interface miss-estimate power consumption. However, because the system-level measurements are often not precise enough, such simple test frequently reports irrelevant RAPL errors. Consequently, we use a more robust statistical method to check for positive values in ΔP.

Statistical hypothesis testing is a widely used technique in the process of decision making based on empirical or observed data. The idea behind hypothesis testing is to make a choice (accept or reject) between two hypotheses: the null hypothesis called H_0, and the alternative hypothesis called H_a. H_0 represents our general belief about a particular data set. For example, a medicine A is not more efficient than an another medicine B. On the other hand, H_a represents an another belief about the data set. Then, having a fixed risk level α, a hypothesis test evaluates whether H_0 can be proven false. If H_0 is rejected with a risk level α, then the alternative hypothesis H_a is usually considered to be true with a confidence level $1-\alpha$, although it only approximates the exact confidence level $1-\beta$ [20]. Such statistical hypothesis tests are useful to determine if a particular belief on a data-set can be considered true or not, which is exactly what needs to be done in our case.

We rely on the *Wilcoxon Signed-Rank Test* [14], a one-sample statistical test. It imposes that H_0 is formulated as the median of a sample is equal to an arbitrarily chosen value. Then, the test computes a p-value, which is a probability that quantifies the strength of evidence to not reject H_0. If the p-value is very small, then H_0 is unlikely to be true, and the alternative hypothesis is likely to be true. In practice, if p-value $\leq \alpha$, where α is a specified risk level, then H_0 is rejected and H_a is usually accepted with a confidence level $1-\alpha$. Unlike other statistical hypothesis tests, the Wilcoxon test does not assume any specific distribution of the data set. Indeed, most of the statistical tests in the literature impose that the data distribution follows the normal distribution. However none of our experimental results follow the normal distribution. The Wilcoxon Signed-Rank test is then perfectly suited to our case. Moreover, we fulfill the two conditions that are required to correctly use the test: 1) all the values in ΔP are mutually independent, and 2) each value in ΔP comes from a continuous population (not necessarily the same).

Let us now show how we can use the test in our context. The test imposes that H_0 is formulated as $med(\Delta P) = 0$, and H_a is formulated as $med(\Delta P) > 0$ or $med(\Delta P) < 0$ or $med(\Delta P) \neq 0$. We also know that to prove that RAPL correctly estimates power, we must prove $med(\Delta P) \geq 0$. However, there is nothing in the forms of H_0 and H_a that allow us to express $med(\Delta P) \geq 0$. Consequently, instead of proving that RAPL is accurate, we try to prove that RAPL is inaccurate. Thus, we express H_a as $med(\Delta P) < 0$. If we succeed to reject H_0, then we can prove with an approximated confidence level $1-\alpha$ that RAPL is inaccurate. Otherwise, we assume that RAPL is accurate.

The Wilcoxon test allows us to verify a binary knowledge on a data-set. However, rather than providing a yes/no answer about the accuracy of the measurements with a fixed confidence level, it would be more useful to determine the

probability for the RAPL interface to incorrectly estimate power. In fact, the p-value resulting from the test can be used for that purpose. Indeed, accepting H_a, requires p-value $\leq \alpha$, then $1 - p$-value $\geq 1 - \alpha$, where $1 - \alpha$ represents the confidence level one desires in order to accept H_a. Thus, $1 - p$-value is the maximal confidence level one can have when considering H_a as true. Thus, rather than comparing the p-value with a risk level α as it is classically done in the Wilcoxon test, we consider $1 - p$-value as the confidence one can have when considering the RAPL interface as inaccurate.

2.3 Comparison to Simple Metrics

Let us show the benefits of a rigorous statistical protocol for probe validation over simple metrics like proportions or sample median. A sample proportion ρ represents the fraction N^- out of N benchmarks, where the power increase difference between that reported by the DPM and the RAPL interface is negative. One may consider that the higher this proportion is, the better is our confidence on the inaccuracy of CPU power probe. The sample median and ρ are both simple tests that could typically be used to distinguish between a failure of the RAPL interface or a success.

Figure 1 reports an observed distribution of power increase differences when the benchmarks are run on 1 or 2 cores on the `SandyBridge` machine while the idle cores remain in the level C2 C-state. In addition to the histogram, the figure reports the 1-p-value, the sample proportion ρ, and the sample median. In the presented case, $\rho = 51\%$, i.e. there are slightly more negative values than positive ones in ΔP. Moreover the median is close to 0. Thus, the conclusion about the RAPL interface precision in this case is uncertain, especially with regard to the dispersion in the set. On the other hand, the Wilcoxon test indicates that in order to declare that the RAPL interface overestimates power consumption, we should accept at most a confidence level $1 - p$-value of 47.46%. Such low confidence level clearly states that the RAPL interface can hardly be said incorrect, whereas the other metrics are unclear. In fact, the p-value resulting from the Wilcoxon test takes into account the data distribution and is then more robust than the other considered metrics. Thus, not only the Wilcoxon test provides a clearer answer in the presented case, but it is also more reliable than classical metrics one could think of.

2.4 Discussion

The methodology detects when RAPL estimations are incorrect either because power consumption in the low power configuration $C1$ is under-estimated or because it is over-estimated in the high power configuration $C2$. However, the methodology cannot distinguish which one of the two cases happened. Moreover, the methodology is also not sensitive to over-estimations of power in $C1$ nor under-estimations in $C2$. Although the two limitations restrict the precision of the diagnostic, the methodology is already sufficient to uncover many issues with the power probes, as shown in the experiments.

The methodology allows one to detect incorrect power estimations with only whole system instrumentation. If one is also interested in estimating the importance of the detected flaws in ΔP, other metrics such as confidence intervals or probability density functions can be used. Such tools are then useful to quantify the errors detected by the test.

Table 1. Going from the minimal to the maximal frequency on the `IvyBridge` machine

Cores used	1	2	3	4
$1-p$-value (%)	100	0	0	0
ρ (N^-/N in %)	99.8	0	0	0

Table 2. Going from the minimal to the maximal frequency on the `SandyBridge` machine

Cores used	1	2	3	4
$1-p$-value (%)	0	0	0	0
ρ (N^-/N in %)	0	0	3	4.6

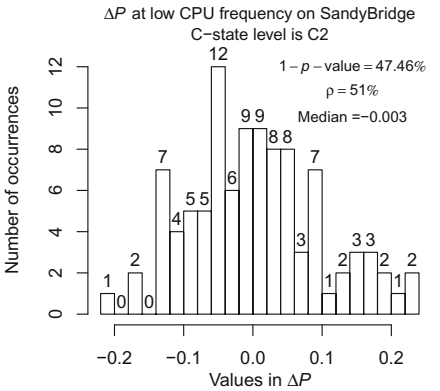

Fig. 1. The observed power increase difference when comparing a 1 core configuration to a 2 cores one on `SandyBridge`

3 Experimental Setup and Methodology

We performed measurements on two distinct Intel machines highlighting different use cases. First, a `SandyBridge` machine consisting in a Intel Xeon E3-1240 processor with 4 cores, where Hyper-Threading is disabled. On that machine, the minimal CPU frequency is 1.6 GHz and the maximal one is 3.3 GHz. Second, a `IvyBridge` machine consisting in a Intel Core i7-3770 processor of 4 cores with Hyper-Threading enabled. There are 8 hardware threads available in the machine and frequencies range from 1.6 GHz to 3.4 GHz. All the test machines run a x86_64 Linux kernel version higher than 3.2.

Each processor has a RAPL interface to estimate energy consumption. The modeled energy values are obtained by combining the status of a set of architectural performance events and energy weight across the set of cores on the chip. The RAPL interface works at the granularity of power planes that enclose the various CPU parts. Usually, three power planes are provided: one for the whole package, one for the cores, and one for the uncore part, sometimes replaced by DRAM power consumption on some server chips [5]. During our experiments, we considered the package power plane and accessed it through MSRs. Our testing processors do not provide any DRAM power plane.

In this study, RAPL power estimation is compared against a digital power meter (DPM). The RAPL and the DPM were configured to measure energy consumption instead of power. Power consumption is then computed using energy and measurement duration. We use a *Yokogawa* WT210 measurement device located between the power supply of the computer and the electrical plug, integrating the overall energy consumption of the system from power measurements performed every 0.1 s.

The accuracy of RAPL power estimation is studied in various experimental configurations. The goal of the experiments is to determine if each parameter affects the accuracy of RAPL. We consider the following parameters: First, CPU frequency: minimal and maximal frequency. Second, number of cores: the micro-benchmark is replicated over 1 to the maximal number of cores available. Third, temperature: cold or warm CPU. Finally, idleness: idle or active CPU. Though the `IvyBridge` machine has Hyper-Threading, we did not study its accuracy. In fact, it is not obvious whether using all the hardware threads (8 in our case) may lead to a higher power consumption than using only 4 hardware threads (1 hardware thread per core). With an 8 threads execution, the result is an interleaved execution of the 8 threads. Consequently, due to context switching, we may observe a lower power consumption for some micro-benchmarks.

Knowing that RAPL power estimation accounts only for processor power consumption, our experimental methodology considers only workloads that stress the processor components. Indeed, workloads accessing memory create off-chip activity that is not accounted by the RAPL interface but that is measured by the DPM. Moreover, it is often unclear how memory power consumption evolves when an experimental parameter varies. For instance, although we can safely expect the CPU power consumption to increase when more cores are used, it is not certain that using more cores will increase memory consumption. Indeed, resource contention and the increased number of opportunities for batching memory accesses may in fact lead to a slightly lower memory power consumption. Consequently, to ensure the predictability of our experimental results, we consider compute-intensive workloads with negligible memory traffic.

For our evaluation, we automatically generated 500 distinct random compute-bound micro-benchmark. The average LLC miss rate is around $1.6e^{-6}$ indicating a negligible memory activity. Each micro-benchmark exhibits a distinct mix of scalar and vector instructions. The instructions are randomly taken from the most represented instructions in the binary programs available in our `/bin` directory and are expressed as inline assembly. The benchmarks are compiled using the `gcc-4.6` compiler with flag `-O3`. While the execution of our micro-benchmarks is repeated 5 times, each of them was sized to run for at least 10 s (shorter runs lead to unstable results). Considering long measurements allows us to ensure the reproducibility of the results. Measurement probes (time and energy) are inserted before and after the execution of the micro-benchmarks, limiting the introduction of noise or overhead in our measurements.

To achieve high precision in our measurements, we use thread affinity for better performance stability and the time stamp counter (TSC) for precise time

measurements. To access TSC, we follow the measurement technique proposed by Intel [15]. For our test machines, the time stamp counter increments at a fixed rate [16] and is not affected by CPU frequency change. Furthermore, it ensures accurate time measurements regardless of the used CPU frequency. We use the `userspace` Linux governor to select a particular CPU frequency. The test machines were entirely dedicated during the experiments to a single user. The experiments were done on a minimally-loaded machine (disable all inessential OS services), minimizing I/O and memory activity.

Raw data, including the micro-benchmark source code, results, and the scripts used to process them are also provided at `http://github.com/BenoitP/eprobe_validation`. As can be seen on the repository, we used the R software to process the data.

4 Experimental Results and Analysis

4.1 CPU Frequency

The goal of the first set of experiments is to study the impact of CPU frequency on the accuracy of the RAPL interface. We analyze the power differences between the DPM and RAPL interface when setting the minimal and maximal CPU frequencies. All the other experimental parameters remain fixed during the measurements in order to isolate the impact of frequencies on the RAPL interface accuracy. Then, we have to check whether the power increase between the minimal and maximal CPU frequencies estimated by RAPL is at least the same as the one reported by the DPM.

The statistical protocol leads to the results presented in Tables 1 and 2 that report the maximal accepted confidence level $1 - p$-value to declare if RAPL is inaccurate for different number of cores. They also report the sample proportion ρ of benchmarks having negative power increase difference between DPM and RAPL. As far as CPU frequency is considered, and except for the case of single thread executions on the `IvyBridge` machine, the methodology reveals no errors in the RAPL interface. Indeed, regardless of the test machine, all the computed confidence levels are equal or close to 0 %, where the p-values are close to 1. We can also observe that all the reported proportions are very small.

However, when using a single core on the `IvyBridge` machine, all the values in ΔP but one are negative. We then conclude that the RAPL interface estimation is inaccurate. The methodology however does not determine if it is due to under-estimations with the minimal frequency, over-estimation with the maximal frequency, or both. As a conclusion, it is clear that the RAPL interface is inaccurate when a single core is used on that machine and care must be taken when considering the RAPL interface power information in such situation.

4.2 Number of Active Cores

Let us now analyze the RAPL power estimation accuracy while the benchmarks are replicated over an increasing number of cores. For a fixed CPU frequency, we

analyze power differences between the DPM and RAPL interface for each pair of increasing number of cores. For example, with a quad-core CPU, our protocol will test the pairs (1,2), (1,3), (1,4), (2,3), (2,4) and (3,4). Using more processor cores should always translate into a higher power consumption.

Table 3. Impact of the number of cores used on the `IvyBridge` machine

Cores used	Minimal frequency		Maximal frequency	
	$1-p$-value (%)	ρ (%)	$1-p$-value (%)	ρ (%)
$1 \Rightarrow 2$	100	100	0	0
$1 \Rightarrow 3$	100	100	0	0
$1 \Rightarrow 4$	0	0	0	0
$2 \Rightarrow 3$	100	84.4	0	0
$2 \Rightarrow 4$	0	0	0	0
$3 \Rightarrow 4$	0	0	0	0

Table 4. Impact of the number of cores used on the `SandyBridge` machine

Cores used	Minimal frequency		Maximal frequency	
	$1-p$-value (%)	ρ (%)	$1-p$-value (%)	ρ (%)
$1 \Rightarrow 2$	100	86.8	100	86.2
$1 \Rightarrow 3$	0	0	100	89.2
$1 \Rightarrow 4$	0	0	100	79.2
$2 \Rightarrow 3$	0	0	100	89.2
$2 \Rightarrow 4$	0	0	100	68.6
$3 \Rightarrow 4$	100	66.8	0	14.2

Table 3 reports $1-p$-value resulting from the Wilcoxon test and the proportion ρ of benchmarks which have negative power increase difference between DPM and RAPL on the `IvyBridge` machine. All the metrics agree on the absence of power miss-estimation from the RAPL interface for any number of active cores when the maximal frequency is used. On the other hand, this observation does not hold when the minimal CPU frequency is used. Indeed, while half of the tested configurations exhibits significant confidence level (100%), the remaining half exhibits negligible ones (0%). Among the configurations where the RAPL power estimation accuracy is low, two of them involve the case of using 1 core. The results can be correlated to the data from Table 1: both tables indicate that the RAPL interface tends to report incorrect power consumption when the frequency is low and when a small number of cores is used on that machine.

Similarly, Table 4 reports the same power metrics for the `SandyBridge` machine. Unlike `IvyBridge`, the maximal frequency seems to be a problematic case for the RAPL interface as it nearly always reports inconsistent values when increasing the number of cores. Note that, in the presented case, the statistical methodology is not only more robust but it also provides a clearer decision on the RAPL interface accuracy compared to simple proportions. On the other hand, setting the minimal CPU frequency shows that only 2 out 6 configurations exhibit important confidence on the inaccuracy of the RAPL interface. As shown in Table 2, while changing the CPU frequency on the `SandyBridge` platform does not lead to incorrect power estimation, changing the number of cores, may lead to inaccurate power estimation with RAPL interface on that machine.

The methodology reveals flaws in both platforms. The measurements performed under the problematic conditions should then be considered with care and, ideally, validated with another measurement tool. However, the exact solution to handle such inaccuracy depends on the ultimate goal of the measurements and is then out of the scope of the methodology.

4.3 Other Parameters

Along the frequency and the number of active cores, we also evaluated two other parameters that may have an impact of the RAPL interface accuracy. First, we varied the CPU temperature by running a long and intense workload before performing the measurements. Second, we also evaluated the RAPL interface accuracy when the CPU is idle compared to having one or several active cores. Both parameters were evaluated but our methodology did not expose any issue with such configurations.

5 Related Work

Many energy-related work in the past exploited power measurement for various purposes. Some research efforts focused on power efficiency of large scale HPC systems [18,9]. In the context of power monitoring tools, the Power Pack framework [8] aims at isolating power consumption of devices like disks, memory, inter-connect networks and processors in HPC clusters. Georgiou et al. [11] propose a framework integrated to SLURM [21] allowing energy accounting for distinct jobs at the cluster node level. Power measurements are also widely used for performance-profile based estimations. In [2,19,17] total energy consumption measurements are combined to hardware performance counters to estimate energy usage of either hardware or software components. Obviously, precise power measurements can only be performed if the probes themselves are accurate. Thus, it is of primary importance for the work based on measurements to be able to assess the probes accuracy. The presented methodology can then help improving the correctness of any results based on power measurements.

Despite its large usage, only a few research efforts focused on the accuracy of on-chip model-based power estimation for x86 architectures. In [6,7] the accuracy of RAPL interface is studied using linear algebra kernels and algorithms. For the tested algorithms, they concluded that RAPL power estimation represents a viable alternative to physical power meters. Hackenberg et al. [12] performed a quantitative comparison of various power measurement techniques on compute nodes. They showed that the RAPL interface is accurate in most of the cases. However, the RAPL interface was showed to be inadequate to measure energy for short codes [13,3]. All previous studies validate the accuracy of the RAPL interface either by simply comparing RAPL estimation to full system power meter measurements or to dedicated CPU power devices. On the other hand, we propose a more rigorous statistical validation approach of CPU power probes.

Statistical analysis has recently gained more focus in the computer science community. However, the majority of the proposed analysis techniques address only temporal performance. Georges et al. [10] proposed statistical measurement methodologies based on the analysis of variance to compare the performance of Java programs. Touati et al. [20] proposed a performance analysis protocol that computes statistically significant speedups. The proposed protocol relies on well-known parametric and non-parametric hypothesis tests. Similarly, to compare the performance of computers, Chen et al. [4] proposed a statistical protocol that

relies on non-parametric tests. We extend previous works to check the accuracy of power measurement probes by means of statistical techniques.

6 Conclusion

We propose a rigorous statistical approach to validate the accuracy of model-based CPU power estimation. The main advantages are twofold: portability and low cost. First, the statistical protocol can be extended to support the validation of any kind of power measurement probe. Second, the approach does not require complex hardware instrumentation as only full-system instrumentation such as a DPM or an IPMI-based probe is needed. Statistical validation is also more robust than simple metrics such as the sample median or proportions. With this regard, the proposed method outperforms the techniques commonly used.

The proposed methodology is able to pinpoint the couple of experimental parameters that influence the most the accuracy of power probes, although it does not exactly indicate which parameter is the source of the observed flaws. As an illustration, we applied our methodology on two Intel based machines and report the incorrect estimations detected and the associated parameters that seem to cause them. We also observed that in overall, RAPL power estimation is more accurate on `IvyBridge` than on `SandyBridge`, reflecting an increased accuracy in newer processors.

References

1. AMD: Amd opteron 6200 series processors, linux tuning guide (2012), `http://developer.amd.com/wordpress/media/2012/10/51803A_OpteronLinuxTuningGuide_SCREEN.pdf`
2. Bellosa, F.: The benefits of event: Driven energy accounting in power-sensitive systems. In: Proceedings of the 9th Workshop on ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System, EW 9, pp. 37–42. ACM, New York (2000)
3. Cao, T., Blackburn, S.M., Gao, T., McKinley, K.S.: The yin and yang of power and performance for asymmetric hardware and managed software. In: 39th International Symposium on Computer Architecture (ISCA), pp. 225–236. IEEE (2012)
4. Chen, T., Chen, Y., Guo, Q., Temam, O., Wu, Y., Hu, W.: Statistical performance comparisons of computers. In: Proceedings of the 18th IEEE International Symposium on High-Performance Computer Architecture, HPCA 2012, pp. 1–12. IEEE Computer Society, Washington, DC (2012)
5. David, H., Gorbatov, E., Hanebutte, U.R., Khanna, R., Le, C.: Rapl: Memory power estimation and capping. In: ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), pp. 189–194 (2010)
6. Demmel, J., Gearhart, A.: Instrumenting linear algebra energy consumption via on-chip energy counters. Tech. Rep. UCB/EECS-2012-168, EECS Department, University of California, Berkeley (June 2012), `http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-168.html`

7. Dongarra, J., Ltaief, H., Luszczek, P., Weaver, V.M.: Energy footprint of advanced dense numerical linear algebra using tile algorithms on multicore architectures. In: Second International Conference on Cloud and Green Computing (CGC), pp. 274–281. IEEE (2012)
8. 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)
9. Ge, R., Feng, X., Subramanya, S., Sun, X.-H.: Characterizing energy efficiency of i/o intensive parallel applications on power-aware clusters. In: IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), pp. 1–8. IEEE (2010)
10. Georges, A., Buytaert, D., Eeckhout, L.: Statistically rigorous java performance evaluation. In: Proceedings of the 22nd Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems and Applications (OOPSLA 2007), pp. 57–76. ACM, New York (2007)
11. Georgiou, Y., Cadeau, T., Glesser, D., Auble, D., Jette, M., Hautreux, M.: Energy accounting and control with SLURM resource and job management system. In: Chatterjee, M., Cao, J.-n., Kothapalli, K., Rajsbaum, S. (eds.) ICDCN 2014. LNCS, vol. 8314, pp. 96–118. Springer, Heidelberg (2014)
12. Hackenberg, D., Ilsche, T., Schone, R., Molka, D., Schmidt, M., Nagel, W.E.: Power measurement techniques on standard compute nodes: A quantitative comparison. In: 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 194–204 (2013)
13. Hähnel, M., Döbel, B., Völp, M., Härtig, H.: Measuring energy consumption for short code paths using rapl. SIGMETRICS Perform. Eval. Rev. 40(3), 13–17 (2012)
14. Hollander, M., Wolfe, D.A.: Nonparametric Statistical Methods, 2nd edn. Wiley Interscience (January 1999)
15. Intel Corporation: How to benchmark code execution times on Intel IA-32 and IA-64 instruction set architectures (2000), `http://download.intel.com/embedded/software/IA/324264.pdf`
16. Intel Corporation: Intel 64 and IA-32 architectures software developer's manual: System programming guide (2013), `http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html`
17. Isci, C., Martonosi, M.: Runtime power monitoring in high-end processors: Methodology and empirical data. In: Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 36, p. 93. IEEE Computer Society, Washington, DC (2003)
18. Kamil, S., Shalf, J., Strohmaier, E.: Power efficiency in high performance computing. In: IEEE International Symposium on Parallel and Distributed Processing (IPDPS), pp. 1–8. IEEE (2008)
19. Kansal, A., Zhao, F.: Fine-grained energy profiling for power-aware application design. SIGMETRICS Perform. Eval. Rev. 36(2), 26–31 (2008)
20. Touati, S.-A.-A., Worms, J., Briais, S.: The speedup-test: a statistical methodology for programme speedup analysis and computation. Concurrency and Computation: Practice and Experience, 22 (2012)
21. Yoo, A., Jette, M., Grondona, M.: Slurm: Simple linux utility for resource management. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2003. LNCS, vol. 2862, pp. 44–60. Springer, Heidelberg (2003)