Per-Task Energy Metering and Accounting in the Multicore Era

Qixiao Liu, Miquel Moreto, Jaume Abell, Francisco J. Cazorla and Mateo Valero
Barcelona Supercomputing Center, Universitat Politecnica de Catalunya
{Qixiao.liu,Miquel.moreto,Jaume.abella,Francisco.cazorla,Mateo.valero}@bsc.es

Abstract*-Energy has become arguably the most expensive resource in a computing system. As multi-core processors are the preferred processing platform across different computing domains, measuring the energy usage draws vast attention.*

In this thesis, for the first time, we formalize the need for per-task energy measurement in multicore by establishing a two-fold concept: per-task energy metering and sensible energy accounting. The former, for a task running in a multi-core system, provides estimates on the actual energy consumption corresponding to its resource usage. The latter provides estimates on the energy the task would have consumed running in isolation with a given fraction of the shared resources. We have shown how these two concepts can be applied to the main components of a computing system: the processor and the memory system.

I. INTRODUCTION

Chip multi-core processors (CMPs) are the preferred processing platform across different domains such as data centers, real-time systems and mobile devices. In all those domains, energy is arguably the most expensive resource in a computing system, in particular with fastest growth. Therefore, measuring the energy usage draws vast attention. Current studies mostly focus on obtaining finer-granularity energy measurement, such as measuring power in smaller time intervals, distributing energy to hardware components or software components. Such studies focus on scenarios where system energy is measured, and

level mechanism proposed to distribute the system energy to multiple running programs in a resource sharing multi-core system in an exact way.

To elaborate on the need of accurate per-task energy measurement, Figure 1 shows the average power dissipation when executing all the SPEC CPU2006 benchmarks on a POWER7-based system. As shown, different tasks incur different average power dissipation, with the maximum variation being 16%, between *453.povray* and *410.bwaves*. Hence, if a povray-like and a bwaves-like program execute undisturbed in a computing system for a period of time, they will incur significantly different energy consumptions. However, the same amount of energy would be attributed to each, which sum up to the total energy consumption of the system.

In this work, we formalize the need for per-task energy measurement in multicore by establishing a two-fold concept: ***Per-Task Energy Metering (PTEM)*** and ***Sensible Energy Accounting (SEA)***. Given a workload composed by *n* tasks T_1, T_2, ..., T_n running in a processor with n cores, we define per-task

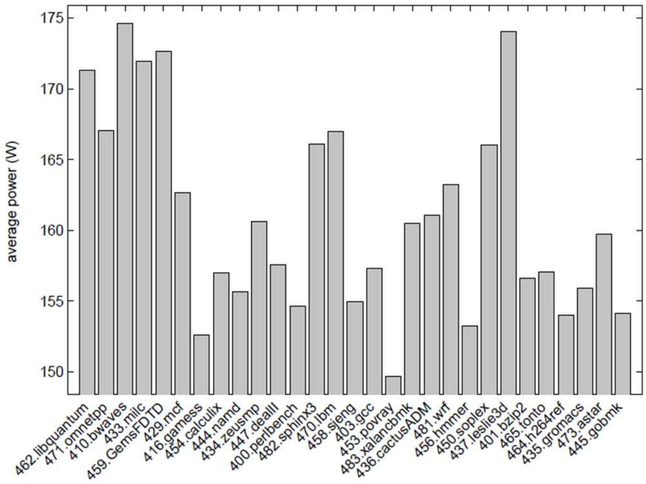

Figure 1: Power consumption of SPECCPU 2006 benchmarks on a PS701 system with an IBM POWER7 processor

energy metering and accounting as follows. ***PTEM*** consists in tracking the energy that a given task, T_i, consumes during a given period of time. ***SEA*** consists in deriving for a given task T_i, the energy that T_i would have consumed if it had run in isolation with a *fraction* of the hardware resources. Both, per-task energy metering and accounting, are complex to derive with the increasing number of hardware shared resources that can serve requests from different tasks concurrently using different resources and/or with different latencies[i].

It is our position that accurately measuring the energy consumed by each task in a computer, instead of considering only the whole energy consumed by the computer, will have plenty of important applications.

Billing. When a customer requests the same computing power to run the same task using the same input, the same energy cost should be accounted[ii].

Energy/Performance optimization. Metering and accounting the energy consumed per task would allow finding the processor setup (e.g. number of cores) and software setup (e.g. mapping) that leads to the lowest system energy consumption.

Selection of appropriate co-runners. Task interaction in hardware shared resources may negatively affect tasks hurting performance and increasing energy requirements. Metering per-task energy can help the OS/runtime scheduler to decide which task to run and when, reducing systems' energy profile.

PROPOSALS

We break energy into its main three components. Dynamic energy corresponds to the energy spent to perform those useful activities that circuits are intended to do. Maintenance energy corresponds to the energy consumed due to useless activity not triggered by the program(s) being run. Leakage corresponds to the energy wasted due to imperfections of the technology used to implement the circuit.

We focus on the case of a shared cache as it is a good representative of the challenges to address when carrying out per-task energy metering and accounting. We assume a multicore architecture where each core has private data and instruction first level caches plus a shared on-chip Last-Level Cache (LLC).

PTEM

Dynamic energy: In order to split dynamic energy across running tasks we consider the number and type of accesses that each task performs to each resource. This requires per-task access counters and the energy consumption per access to be provided by the chip vendor.

Maintenance energy: Maintenance energy is consumed by useless activities in idle resources. It must be attributed to tasks depending on the amount of space they occupy if those resources are stateful (such as caches).

Leakage energy: Splitting leakage energy among running tasks is also proportional to both time and space.

SEA

Dynamic energy: To account the dynamic energy for a task we estimate the number of accesses that each task would perform when it runs with a fraction of LLC space alone.

Maintenance energy: Accounting the maintenance energy based on the execution time and activities estimation when a task runs with a fraction of resources alone.

Leakage energy: Accounting leakage energy relies on the execution time estimation.

Note that we devise hardware mechanisms for both PTEM and SEA in the processor and the memory system[iii,iv,v]. Through which, the runtime estimation is done for all tasks running in workloads in a multicore processor system.

Experiments

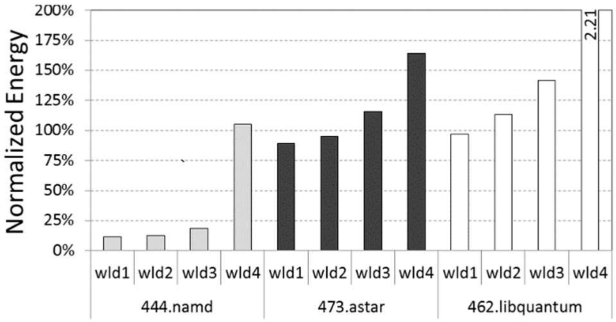

Figure 2: Energy usage of namd, astar, and libquantum in different workloads w.r.t their energy usage when executed in isolation with a fair share of resources.

Our proposed mechanisms have achieved high estimation accuracy in all components, with affordable overheads.

We show in Figure 2, the actual energy metered for three benchmarks when they run in different workloads. The energy consumed by a program will largely depend on its co-runners, due to the fact that they have been sharing resources in the multicore processor system.

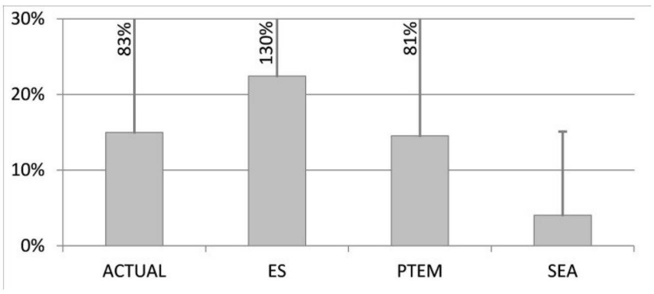

Figure 3: The deviation of mispredicted energy account to tasks running in 8-task workloads under 4-core SMT setup and 16-way LLC

In Figure 3, we show that our proposal significantly improves the estimation of the energy a task would have consumed with a given fraction of resources (SMT core and LLC), compare to other model that measures the energy

.

EXPERIMENTS

In this thesis, we have advanced the field of quantifying per-task energy cost in the multicore systems by proposing a two-fold concept: PTEM and SEA. PTEM derive an estimation of the actual energy a task consumes in a real workload based on resource utilization, and SEA gives an estimation of energy a task would have consumed when it has been given a fraction of resources. These works have already been published in international conferences and journals.

REFERENCES

[1] Q. Liu, et.al. Per-task Energy Accounting in Computing Systems. In IEEE Computer Architecture Letters, 2014.

[1] V. Jimenez, et.al. *Energy-Aware Accounting and Billing in Large-Scale Computing Facilities*. In IEEE Micro, 2011.

[1] Q. Liu, et.al. *Hardware Support for Accurate Per-Task Energy Metering in Multicore Systems.* In ACM Transactions on Architecture and Code Optimization (TACO), 2013.

[1] Q. Liu, et.al. *DReAM: Per-Task DRAM Energy Metering in Multicore Systems*. Euro-Par 2014 Parallel Processing international conference, Porto, August, 2014.

[1] Q. Liu, et.al. *Sensible Energy Accounting with Abstract Metering for Multicore Systems.* In ACM Transactions on Architecture and Code Optimization (TACO), 2015.