Techniques and tools for measuring energy efficiency of scientific software applications

David Abdurachmanov[1], Peter Elmer[2], Giulio Eulisse[3], Robert Knight[4], Tapio Petteri Niemi[5], Jukka Nurminen[6], Filip Nyback[6], Goncalo Marques Pestana[6], Zhonghong Ou[6]

[1] Digital Science and Computing Center, Faculty of Mathematics and Informatics, Vilnius University, Vilnius, Lithuania
[2] Department of Physics, Princeton University, Princeton, NJ 08540, USA
[3] Fermilab, Batavia, IL 60510, USA
[4] Research Computing, Office of Information Technology, Princeton University, Princeton, New Jersey 08540, USA
[5] Helsinki Institute of Physics, PO Box 64, FI-00014, Helsinki, Finland
[6] Aalto University, PO Box 11100, 00076 Aalto, Finland

E-mail: Peter.Elmer@cern.ch

Abstract. As both High Performance Computing (HPC) and High Throughput Computing (HTC) are sensitive to the rise of energy costs, energy-efficiency has become a primary concern in scientific fields such as High Energy Physics (HEP). There has been a growing interest in utilizing low power architectures, such as ARM processors, to replace traditional Intel x86 architectures. Nevertheless, even though such solutions have been successfully used in mobile applications with low I/O and memory demands, it is still unclear if they are suitable and more energy-efficient in the scientific computing environment. Furthermore, there is still lack of tools to derive and compare power consumption for these types of workloads, and eventually to support software optimizations for energy efficiency.

To that end, we have performed several physical and software-based measurements of workloads from CERN running on ARM and Intel architectures, to compare their power consumption and performance. We leverage several profiling tools to extract different aspects of the experiments, including hardware usage and software characteristics. We report the results of these measurements and the experience gained in developing a set of measurement techniques and profiling tools to accurately assess the power consumption for scientific workloads. [Version of 7 August 2014]

1. Introduction

Test references: The Large Hadron Collider (LHC) [1] at the European Laboratory for Particle Physics (CERN) goes round (and round). XXX

2. Tools and techniques

2.1. External probing devices

What is it ? Refer to figure above in order to mention which machine components do the external probing devices measure

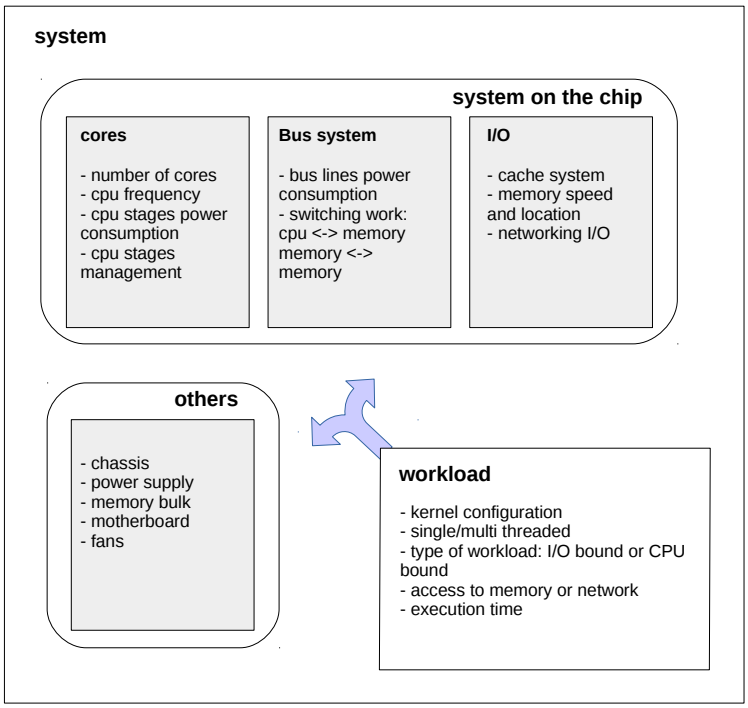

Figure 1. System's components which contribute for power consumption in HPC

2.1.1. Noninvasive clamp meters

2.1.2. Electrical plugin power and energy monitors

2.2. Internal probing chips
What is it ? Refer to figure above in order to mention which machine components do the internal probing chips monitor

2.2.1. Running Average Power Limit Running Average Power Limit (RAPL) provides a platform for monitoring and limiting power of systems on chip (SoC). It is a Intel's technology which was introduced initially on the Sandy Bridge processors. RAPL platform exposes on-chip measurements via the MSR registers. According to [2], this technology offers power measurements of the system at a granularity impossible to reach before with other tools.

As documented by Intel in [3], there are 3 different domains to sample energy consumed by different SoC components on a server. The domains are: package (accounting for the entire socket), power plan 0 or pp0 (accounting for energy consumed by the core) and dram (sum of energy consumed by memory in a given socket, excluding the core caches). The measurements are dumped in the MSR registers at a frequency of 1 kHz and are exposed to the user via /dev/cpu/¡cpu_nr¿/msr. It is also possible to read and write data from the MSR register using Intels open source tool msr-tools [4].

In addition to power monitoring of the sockets, it can limit the power consumed by the different domains. This feature, usually referred as power capping, allows the user to define the average power consumption limit of a domain in a defined time window. For more information

about RAPLs features and configurations, refer to section 14.9.1 of Intels Developers manual [3].

The advantages of using RAPL for measuring power consumption are a straightforward and already installed tool to perform fine grained measurements of energy consumption on SoC and its components.

On the other hand, the drawbacks are lack of documentation available about the monitoring chip. To the knowledge of the authors, specifications such as error degrees, accuracy and implementation diagrams can are not publicly available. In addition, the RAPL technology is vendor locked. Considering those two points, it is difficult to accurately compare and reason power measurements between SoCs from Intel and other vendors.

2.2.2. TI INA231 alike

2.3. Software based measuring tools
2.3.1. powertop and alike

2.4. Profiling tools
Filip's work

3. CMSSW and power performance in a NUMA environment. A study case
Here I would describe and present the result of the measurements done on the full capability CMSSW with different NUMA configurations. This experiments will be running in the Xeon E5-4650. It can be seen as a use case of some of the 'toos and techniques' described above.

4. Comparison of power efficiency of ARM and Intel
4.1. Workflow and experiments setup
Description of the experiments done in ARM (*Cortex-A15 Quad and Cortex-A7 Quad 1.4GH*) and Intel (*Xeon E5-4650*)

4.2. ARM results
Result of measurements done in ARM

4.3. Intel results
Result of measurements done in Intel

4.4. Analisis
Analysis and comparison of above described experiments

5. Conclusions

Acknowledgements
This work was partially supported by the National Science Foundation, under Cooperative Agreement PHY-1120138, and by the U.S. Department of Energy.

References

[1] Lyndon Evans and Philip Bryant. LHC Machine. *JINST*, 3:S08001, 2008.

[2] Balaji Subramaniam and Wu chun Feng. Towards energy-proportional computing for enterprise-class server workloads.

[3] *Intel 64 and IA-32 Software Developer Manuals - Volume 3.* www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html.

[4] Msr tools. `https://01.org/msr-tools`. Last time accessed: 28-07-2014.