On the reliability of performance counters for power models

[Position paper]

Arliones Hoeller, Giovani Gracioli, and Antônio Augusto Fröhlich*
{arliones,giovani,guto}@lisha.ufsc.br

Modern processors have been equipped with *Hardware Performance Counters* (HPCs) and *Performance Monitoring Units* (PMUs) to keep track of hardware events, building adequate tools to debug and analyze systems. Recent research projects addressed the problem of building processor power models by correlating variations on power to hardware events. Here, we intent to tease the workshop participants claiming that HPC-based power models are helpful only in very strict and controlled conditions and do not deliver reliable estimates to support power management algorithms.

The main reason identified so far for the problems shown here is HPC register multiplexing. Intel processors feature over 200 monitorable events, and multicore ARM processors are reaching the count of 100. One can think that the more, the best, but due to the increase of circuit power and complexity, event signals need to be routed to the PMUs, making it impossible to monitor all events simultaneously. The number of HPCs in Intel and ARM processors are model-dependent but are, at most, 8 and 6, respectively. Besides the reduced quantity, the use of HPCs is further restricted by hardware design when sets of events are mappable only to one HPC or some HPC is fixed to a unique event.

To support our claim we consider the impact of HPC multiplexing in two recently published HPC-based power models for Intel processors. Both are empirical models that correlate power measurements to HPCs' registry. Bircher and John [2] propose abstract submodels for CPU and peripherals while Bertran et al. [1] provides detailed submodels for CPU components such as the pre-fetcher (FE) and the branch-prediction unit (BPU). Nevertheless, what is significant about these models to sustain our position is the fact that the models use, respectively, 7 and 11 distinct events.

The cited papers show that the models approximate the power of their systems with acceptable errors; therefore, we do not question their correctness. However, the experimentation scenario used in their evaluation methodology is, in fact, very controlled and, perhaps, unrealistic. In real systems, several events would be monitored to implement not only the power model, but also schedulers [4], memory allocators [3], and so on, increasing the multiplexing degree.

We produced data showing that, in the presence of HPC multiplexing, the result of event accounting gets unreliable: the greater the multiplexing degree, the higher the errors in HPCs. Figure 1 shows the growth in HPC error with increased multiplexing degree of two events used in the power models. Similar errors are shown by virtually all events under the same conditions. The figures show 11 of the 13 applications from the PARSEC-3.0 benchmark ran on an Intel i7-2600 processor. The HPCs were configured and accessed using the Linux Perf-Tool (the *perf*). The error shown in the figures is the difference between the events accounted without multiplexing, and the result when events are measured grouped, randomly, 4-by-4, 8-by-8, or 16-by-16.

Event accounting is a powerful tool to allow systems to introspect on their behavior, but current PMU implementations limit the full potential of the technology. How do we solve

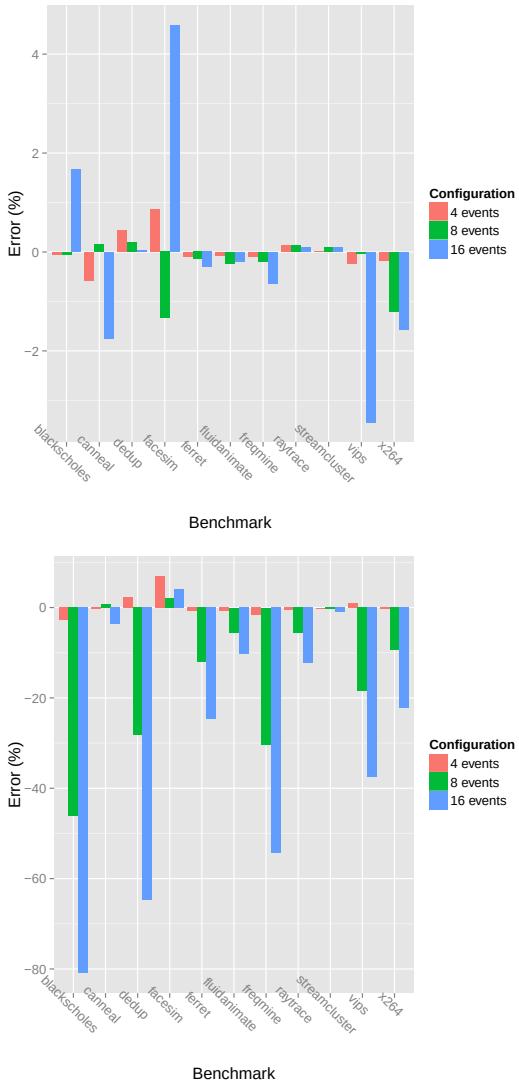

Figure 1: Impact of HPC multiplexing to measure Instructions (top) Last-level cache miss (bottom).

that? Will we find efficient ways to route signals from all corners of a silicon die to a PMU? Can we implement better software mechanisms to, for instance, forecast event counts or explore the correlation among events? The best solution is still unclear to us, but we believe that PMUs can be used to generate more reliable power estimates of complex systems, and our future work will explore ways to achieve such power models.

REFERENCES

[1] R. Bertran, M. Gonzelez, X. Martorell, N. Navarro, and E. Ayguade. A systematic methodology to generate decomposable and responsive power models for cmps. *Computers, IEEE Trans. on*, 62(7):1289–1302, 2013.

[2] W. Bircher and L. John. Complete system power estimation using processor performance events. *Computers, IEEE Trans. on*, 61(4):563–577, 2012.

[3] G. Gracioli, A. A. Fröhlich, R. Pellizzoni, and S. Fischmeister. Implementation and evaluation of global and partitioned scheduling in a real-time os. *Real-Time Systems*, 49(6):669–714, 2013.

[4] K. Singh, M. Bhadauria, and S. A. McKee. Real time power estimation and thread scheduling via performance counters. *SIGARCH Comp. Arch. Nws*, 37(2):46–55, 2009.

* *Mr. Hoeller, Prof. Gracioli, and Prof. Fröhlich are with the Software/Hardware Integration Lab (LISHA) of the Federal University of Santa Catarina (UFSC), at Florianópolis, Brazil.* *Authors acknowledge and thank the contribution of Prof. Sebastian Fischmeister (University of Waterloo) in helpful discussions. This work was supported in part by CAPES/DFAIT grant 1269/2011. Mr. Hoeller and Prof. Gracioli are supported in part by CAPES grant RHTVD 006/2008.*
Benchmarking of Embedded Systems Workshop 2013 Montreal, Canada.