UMaine VMW Group Tech Report UMAINE-VMW-TR-DRAM-RAPL-2015-08, 26 August 2015

Initial Validation of DRAM and GPU RAPL Power Measurements

Spencer Desrochers
University of Maine
spencer.desrochers@maine.edu

Chad Paradis
University of Maine
chad.paradis@maine.edu

Vincent M. Weaver
University of Maine
vincent.weaver@maine.edu

Abstract

Recent Intel processors support Running Average Power Level (RAPL) measurements that provide estimated energy metrics for the CPU, integrated GPU and DRAM. Having easy-to-access energy measurement in a system is extremely valuable when undertaking co-design tasks, especially when trying to optimize code for energy efficiency. The various RAPL metrics are not well documented and are of unknown quality, so we compare the results gathered against detailed physical hardware measurements. We find that the RAPL results match overall measured energy and power trends, but are offset from each other.

1 Introduction

Most modern Intel processors (starting with the Sandy-Bridge microarchitecture released in 2011) have had the capability of providing estimated power measurements via the Running Average Power Level (RAPL) interface [5]. This functionality is part of the power capping infrastructure, allowing the user (or operating system) to specify maximum power limits. This allows a processor to run at the highest possible speed, but automatically throttling back to stay within power or thermal bounds. In order to respect the power limits, a processor must be aware of its current power usage. This is typically not measured, but estimated based on performance counters, temperature, and other inputs, combined via a software model. In addition to its use in power capping, the results of the power model are available to the user via a model-specific register (MSR) and can be used when characterizing workloads. In addition to CPU measurements, RAPL can provide estimated energy readings for other system components including the DRAM and GPU.

Measuring power and energy in modern systems is a difficult task. Most systems do not include such measurement circuitry, and adding it often involves intrusively inserting wires and using expensive digital acquisition boards. Some parts of a system, such as the CPU, can be particularly difficult to instrument due to having scores of power lines soldered directly to the motherboard. This makes the comparative ease of using RAPL for power measurements a very attractive alternative.

Before using RAPL values for research tasks such as co-design, it is important to have an idea of the underlying accuracy. There has been some limited work in validating the counters [7, 10, 23] but this has focused on CPU power measurements. In this work we look at CPU measurements as well as DRAM and GPU. We find that RAPL measurement trends closely match those found by real hardware, but the absolute results are offset from each other. We are still investigating the cause of this divergence.

2 RAPL Background and Related Work

RAPL is documented in Chapter 14.3 of the Intel Volume3b Documentation [18] although many of the details of the interface are not as complete as they could be.

RAPL provides estimated per-package (not per-core, but aggregate for all cores in a physical CPU package) energy estimates for some combination of the CPUs (PP0), PP1 (an implementation defined part of the uncore, often the GPU), the DRAM, and the total package power. Which measurements are available may vary by chip model; for example DRAM measurements were originally only available for server systems but starting with Haswell are available on all processors.

The estimated energy is updated roughly at 1 millisecond (1kHz) intervals, but there is no timestamp so it can hard to get useful results at small timescales [10] although it is possible to mitigate this by carefully monitoring when updates happen and starting measurements at the transition [12].

The minimal energy increment can vary; on regular Haswell this can be read from a register (it is

roughly 61μJ) but on Haswell-EP it is documented elsewhere [17] as being fixed at 15μJ.

To read the RAPL registers one must have ring-0 access, meaning typically only an operating system will have permissions to read them. To access the values one either needs raw MSR access (available in Linux via the `/dev/msr` interface) or else the values may be exported by the perf_event subsystem. Due to security reasons and the full-system nature of the counts, the results are by default only available with root permissions (in theory an attacker could use the power metrics to spy on what other users of the CPU are doing).

2.1 CPU RAPL Validation

Various groups have investigated the accuracy of the RAPL counters against real hardware. Hähnel et al. [12] look at comparing CPU RAPL results on a SandyBridge processor and find results similar to ours where the patterns look the same but there is an offset in the power. They provide only a single graph of a synthetic benchmark in their validation.

Rotem et al. [23] show one validation graph of an unspecified benchmark showing a close match for RAPL CPU and package measurements to actual measurements.

Dongarra et al. [7] compare RAPL measurements on a SandyBridge machine using PAPI to the results found using PowerPack [9] on a different microarchitecture. They use LU factorization as a workload.

Demmel and Gearhart [6] validate two SandyBridge machines against RAPL Package with the STREAM [21] benchmark and a full-system wall power meter.

Hackenberg et al. [10] validate RAPL (and the similar AMD APM interface) on a variety of SandyBridge hardware. They measure both at the wall, as well as the CPU and motherboard level by intercepting the ATX power connectors. They find that RAPL accuracy can vary by workload, and that it can be confused when HyperThreading is enabled.

Mazous, Pradelle and Jalby [20] apply statistical validation to RAPL results compared to full system wall outlet measurements on IvyBridge and SandyBridge. They found some anomalies with the RAPL results when only exercising a single core or when operating at maximum frequency.

Hackenberg et al. [11] investigate RAPL on Haswell-EP processors. They find that the DRAM + Package RAPL results correlate well with total system power readings, but do not measure the individual actual power results for CPU or DRAM.

2.2 DRAM RAPL Validation

The RAPL DRAM interface was first described by David et al. [5]. While concentrating on power-capping, they do describe in detail the underlying power model which presumably is similar to that found in modern Intel chips. A parametric model is built using genetic algorithms based on various inputs and the weights are calibrated by the BIOS as boot. They validated against real hardware using a DIMM riser card and a data acquisition board sampling at 100Hz. They found accuracy of 1% when using a Nehalem server system and a DDR3 1333 4GB memory module.

Khanna et al. [19] describe the weights used in RAPL DRAM measurements. They measure actual DRAM results using a riser with a 5mΩ sense resistor sampled at 100Hz. They find RAPL results within 2.3% of actual measurements.

3 Experimental Setup

We run experiments on a Lenovo Thinkcentre desktop system with a 4-core 2.9GHz i5-4570S Haswell CPU. The "S" series of processors denotes a low-power 65W thermal design envelope. It has an integrated Intel HD Graphics 4600 GPU and main memory consists of one 4GB DDR3 DIMM. The machine is running the Jessie Debian Linux distribution, the 4.1.5 kernel for the DRAM measurements and a specially patched 4.0.5 kernel for the GPGPU measurements.

3.1 Hardware Measurement Setup

System-wide power is measured using a WattsUp-Pro? [8] device which measures power with 1Hz resolution at the wall outlet.

The CPU is instrumented by intercepting the power at the 12V "P4" 4-pin auxiliary ATX connector. This pin primarily powers the CPU [15] but may also power an unknown amount of other parts of the motherboard. Due to potentially high currents involved (in the tens of Amps) an ACS715 Hall Effect sensor [2] is used for measurement.

The DRAM is instrumented by using a JET-5464 DDR3 DIMM Extender card which has a 3.3mΩ sense resistor built in. The voltage drop across this resistor can be used to calculate the current draw and thus the power usage. This voltage drop is very small, so an INA122 instrumentation amplifier [4] is used to amplify the signal.

The DRAM and CPU voltages are logged using a Measurement Computing USB-1208FS-Plus data acquisition board, which is connected to a separate computer that conducts the logging. The results are gathered at 2kHz.

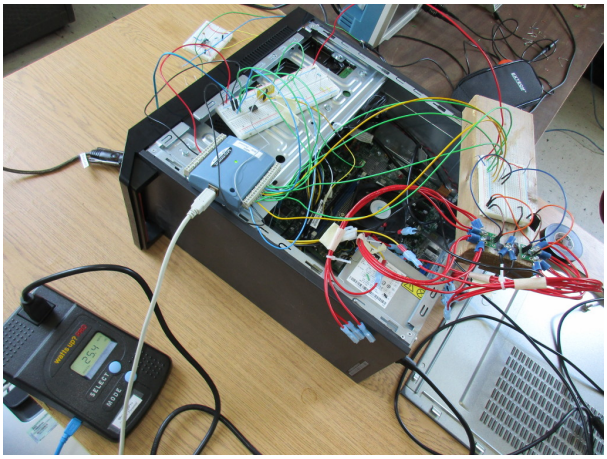

Figure 1: Our instrumented test machine.

A picture of our instrumented test machine can be seen in Figure 1. More details on our hardware measurement setup can be found on our group website [22, 27].

3.2 RAPL Measurement Setup

The RAPL values are gathered using the perf tool that comes with the Linux kernel and uses the perf_event [26] interface. We also gather other hardware performance counter values at the same time, including cycles and cache misses. An example command line used:

```
perf stat -a -e cycles,instructions,
        cache-misses,cache-references,
        uncore_imc/data_reads/,
        uncore_imc/data_writes/,
        power/energy-cores/,
        power/energy-pkg/,
        power/energy-cpu/,
        power/energy-ram/
        -I 100 -x , ./run_test.sh
```

The perf tool is modified to toggle a serial port DTR signal when starting measurement; this line is hooked to our data logger and is used to synchronize the software measurements to the hardware measurements.

To allow gathering system-wide measurements as a normal user the `/proc/sys/kernel/perf_event_paranoid` setting is set to "0".

We only gather the perf results at 10Hz (100ms) resolution. This is a relatively low frequency, as the RAPL counters update at 1kHz. The perf tool operating in interval (`-I`) mode will not let you measure at faster than 100ms. We found that by removing the limit and trying to gather data at 100Hz caused a noticeable 0.5W jump in power consumption due to the measurement overhead.

We investigated writing a custom tool that would use the perf_event interface's sampling/mmap() ring-buffer recording to provide lower-overhead access, but when we tried to record at 1KHz the kernel's interrupt throttling kicked in due to the performance interrupts taking up over 25% of CPU time. For now we are using the lower sampling frequency. Possible ways to avoid this would be to use a different performance interface such as LIKWID [25] or to read the MSRs directly.

3.3 CPU/DRAM Benchmarks

We investigate a variety of benchmarks commonly used in high-performance computing.

For a basis, we look at the idle system, which is just recording system behavior when a "sleep" command is issued. For other benchmarks when possible we include a 1 second sleep command at the beginning and end of the benchmark runs so that the perf measurements will include an amount of rest system state for comparison.

In order to exercise the DRAM we look at the STREAM [21] benchmark which tests a machine's memory performance. STREAM performs operations such as copying bytes in memory, adding values together, and scaling values by another number. We use the OpenMP version of the benchmark to try to use all of the cores in the system.

To exercise the CPU we use the high-performance Linpack HPL benchmark. We use it with three different BLAS libraries:

- The version of Automatically Tuned Linear Algebra Software (ATLAS) [28] that ships with Debian Linux,
- OpenBLAS [1] optimized for Haswell processors (including using the new FMA fused-multiply-add) instruction,
- and a statically linked version that comes with Intel's MKL libraries [14].

HPL is configured with a problem size of N=15000 and to use a 2x2 grid of processors, which gives high performance for all of the BLAS implementations and nearly uses all 4GB of available memory.

3.4 GPU Benchmarks

GPU measurement is difficult to quantify with the integrated GPU, as it is not possible to intercept GPU power alone. There are hardware performance counters available for the GPU [16] but as of yet the Linux support for reading these is not complete.

The first benchmark we look at is SmallptGPU2 [3], an OpenCL ray-tracer. We use Beignet [13] which is an

OpenCL implementation for the Intel HD series of integrated GPUs. We use the default ray-trace setup, ending after 25s of tracing.

For an OpenGL intensive video game benchmark we use the game Kerbal Space Program [24]. We record a 25s long snapshot of behavior while launching a rocket in-game.

4 Results

We use the perf tool to measure the RAPL package and DRAM results on a number of benchmarks in addition to the cycles per instruction (CPI) and last level cache (LLC) misses. In general the DRAM power closely matches the LLC rate, and the package power matches the CPI metric. For the GPU benchmarks we additionally measure the RAPL cores and GPU counts. Finally we take actual hardware measurements of total system power, the P4 ATX connector (which should be mostly the same as package power), as well as the actual DIMM power.

4.1 CPU Benchmark Results

In Figure 2 we show the results of an idle system. This has surprisingly high CPI and cache variability; the system is not completely idle and the operating system and background jobs are running. The RAPL and actual behaviors match each other fairly well, but the RAPL readings tend to be low. For CPU this might be due to other devices on the power connector as well as power conversion losses by the voltage regulators, but it is unclear why the DRAM results are low.

Figure 3 shows the results when DRAM is being stressed by the STREAM benchmark. It can be seen that the RAPL DRAM results are much closer to actual measurements when the system is not idle.

Figure 4, 5 and 6 show Linpack running with various BLAS libraries. These have much more dynamic phase behavior. The DRAM values for all three benchmarks are lower than expected.

4.2 GPU Benchmark Results

Figure 7 shows the results when the GPU is being used for OpenCL work. It is interesting to see the CPU is almost totally idle and the GPU is using the majority of the package power. The DRAM behavior is complex and the RAPL readings do not seem to capture this, possibly due to the low sampling frequency.

Figure 8 shows the results when the GPU is being used to play a 3D video game. The power profile is very similar to that of the OpenCL demo with slightly more CPU being used (though the game is only using 1 core). There is more DRAM activity that does not seem to be captured by the RAPL results.

Table 1: CPU results. In the split rows, the top row is actual measurement and bottom row is RAPL.

Benchmark	Time (s)	$GFlops$	Energy (J)	Average Power (W)	$\frac{GFlops}{W}$
Sleep	9.7	—	65.2	6.7	—
			43.2	4.4	—
STREAM	12.7	—	292.4	23.0	—
			249.8	19.6	—
HPL-ATLAS	61.2	40.9	2670.8	43.6	0.94
			2340.4	38.3	1.07
HPL-OpenBLAS	36.1	113.9	1600.6	44.3	2.57
			1382.8	38.3	2.97
HPL-mkl	25.5	106.8	1404.5	55.1	1.93
			1211.5	47.5	2.25
OpenCL-raytrace	26.1	—	577.5	22.09	—
			523.9	20.0	—
OpenGL-kerbal	26.7	—	710.0	26.6	—
			617.6	23.1	—

Table 2: DRAM results. In the split rows, the top row is actual measurement and bottom row is RAPL.

Benchmark	Time (s)	$GFlops$	Energy (J)	Average Power (W)	$\frac{GFlops}{W}$
Sleep	9.7	—	7.7	0.79	—
			4.2	0.43	—
STREAM	12.7	—	27.5	2.16	—
			26.6	2.09	—
HPL-ATLAS	61.2	40.9	131.3	2.15	19.0
			96.2	1.57	26.1
HPL-OpenBLAS	36.1	113.9	69.0	1.91	59.6
			53.2	1.47	77.5
HPL-mkl	25.5	106.8	62.0	2.43	44.0
			53.9	2.11	50.6
OpenCL-raytrace	26.1	—	24.8	0.95	—
			22.3	0.85	—
OpenGL-kerbal	26.7	—	36.9	1.38	—
			31.2	1.17	—

4.3 Overall Totals

Table 1 shows overall summaries for the CPU results and Table 1 shows overall summaries for the DRAM results. It can be seen that for both CPU and DRAM the RAPL results are consistently below actual measurements, both for total energy as well as average power. It is unclear if this is an actual difference or an artifact in our measurement methodology. Despite the offset, metrics such as GFlops/Watt still gave the same rankings whether sorted by actual or RAPL results.

5 Conclusion and Future Work

Our work shows that RAPL power measurements on our Haswell machine closely track actual power measurements, although the RAPL results have lower absolute results.

Figure 2: DRAM metrics for idle (sleep).

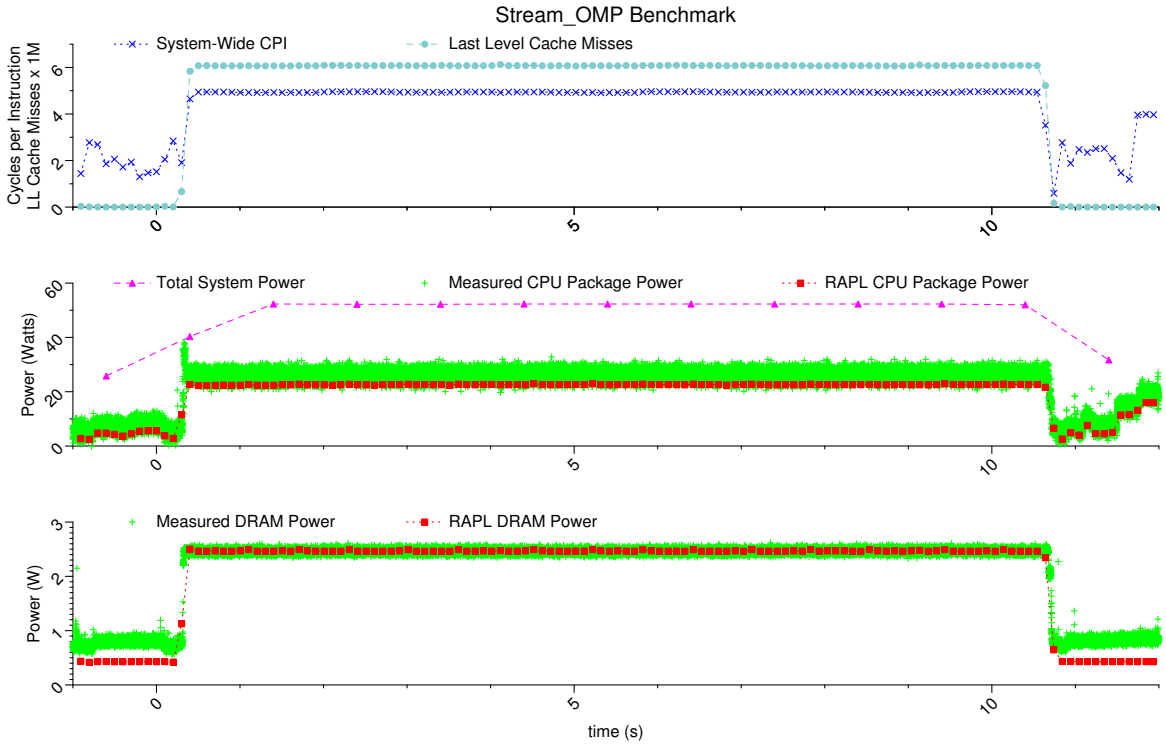

Figure 3: DRAM metrics for STREAM benchmark.

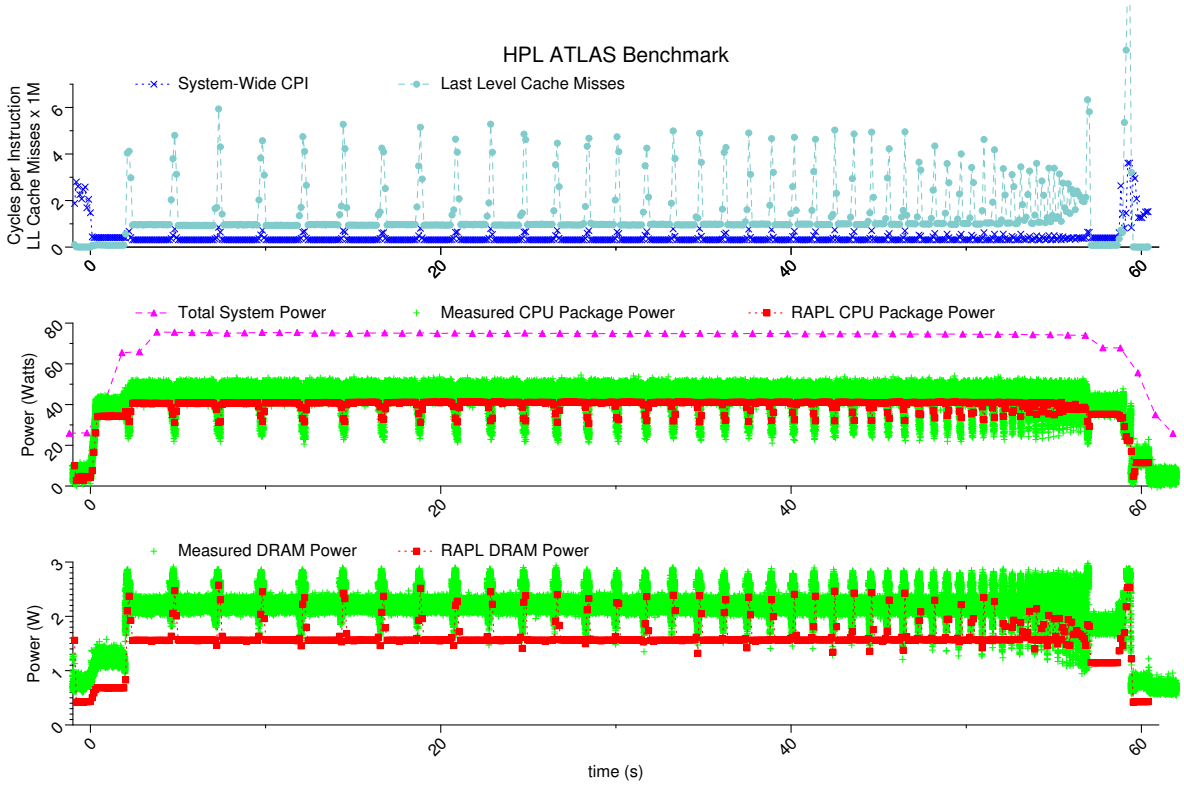

Figure 4: HPL Atlas Benchmark

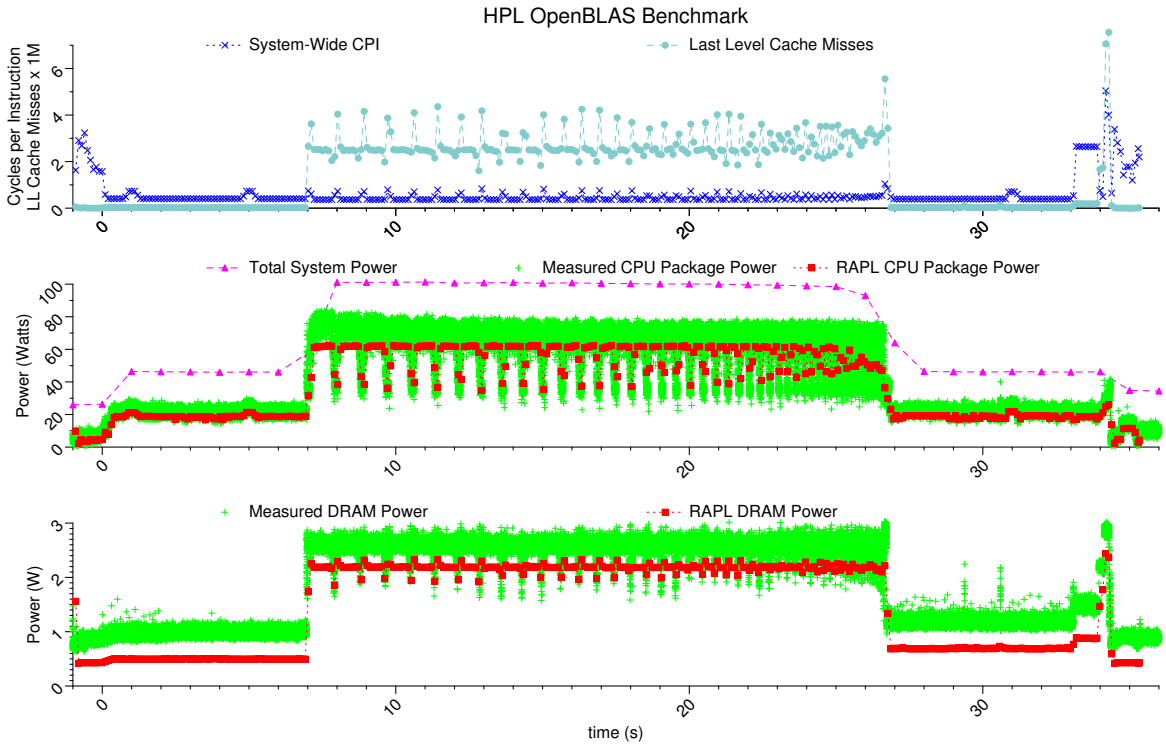

Figure 5: HPL OpenBLAS Benchmark. Yes, anonymous reviewer #2, the CPU line crosses over the total line; this can happen when your total power measurement is only sampling at 1Hz.

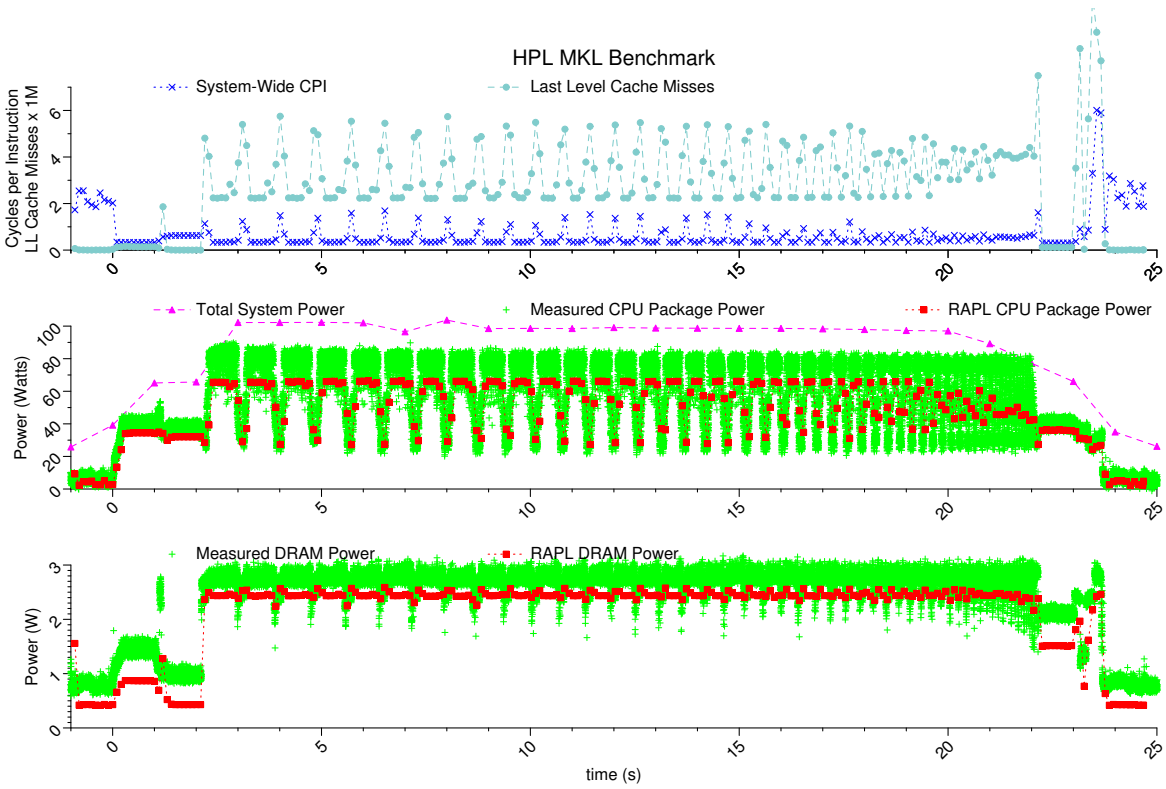

Figure 6: HPL MKL Benchmark

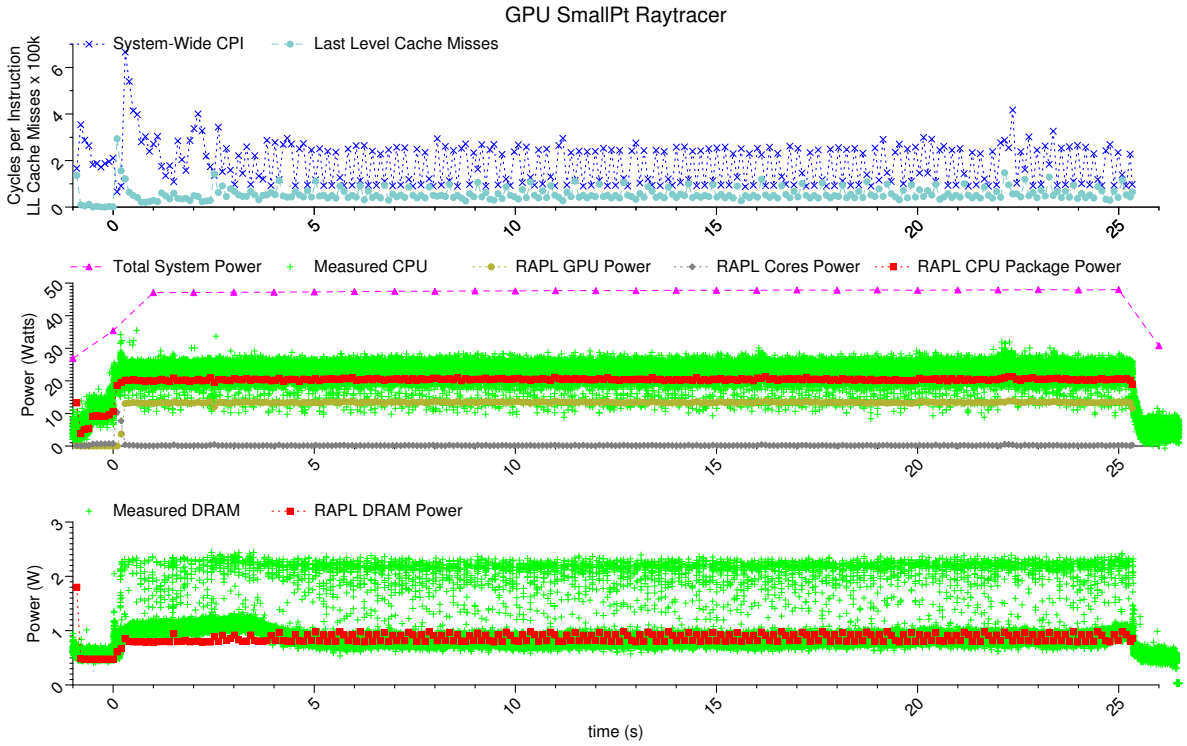

Figure 7: smallpt OpenCL Raytracer

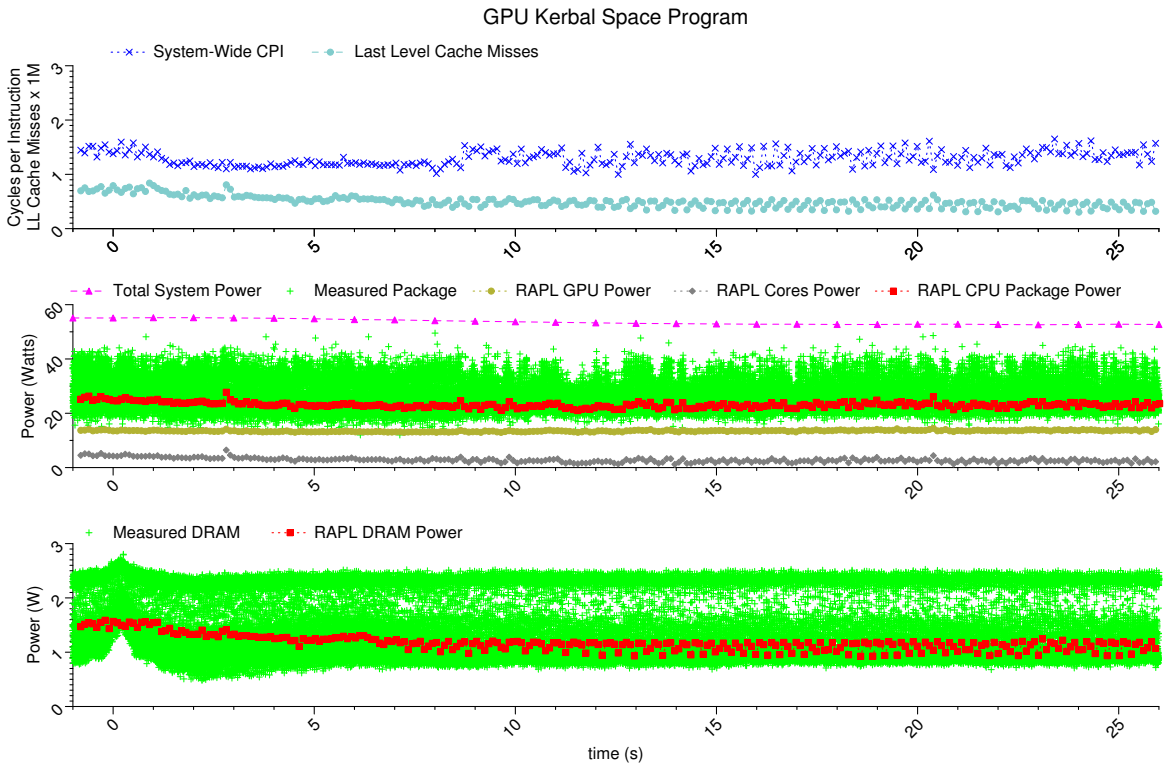

Figure 8: Kerbal Space Program Game

We plan to investigate the cause of the lower readings further to see if they are related to our test setup or if they are intrinsic in the counters. We will investigate whether filtering the actual results (as done by Hackenberg et al. [10]) might make the results easier to analyze.

We also plan to compare various machines in the same Haswell microarchitecture as well as across different microarchitectures.

In addition we would like to expand our results by measuring with multiple DIMM slots installed, enabling monitoring of NUMA workloads.

We also would like to explore a server system, although our Sandybridge-EP machine is missing DRAM support, and our Haswell-EP server has DDR4 DIMMs which will require obtaining different DIMM measurement risers. Hackenberg et al. [11] report that Haswell-EP machines have integrated voltage regulators and more advanced RAPL hardware that includes RAPL "DRAM Mode 1" readings which include actual measurement (rather than the "DRAM Mode 0" pure estimation found on earlier processors) so it will be interesting to see what effect this has on the accuracy of the counts.

Acknowledgment

Spencer's contribution was sponsored by the University of Maine Center for Undergraduate Research (CUGR). The authors would also like to thank Nicholas Nethercote and Thomas Ilsche for their comments on an earlier version of this document.

References

[1] OpenBLAS an optimized BLAS library website. `http://www.openblas.net/`.

[2] ALLEGRO MICROSYSTEMS LLC. *ACS715: Automotive Grade, Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor Lightweight Profiling Specification*, 2013.

[3] BUCCIARELLI, D. Smallptgpu2. `http://davibu.interfree.it/opencl/smallptgpu2/smallptGPU2.html`.

[4] BURR-BROWN. *INA122: Single Supply, MicroPower Instrumentation Amplifier*, Oct. 1997.

[5] DAVID, H., GORBATOV, E., HANEBUTTE, U., KHANNA, R., AND LE, C. RAPL: Memory power estimation and capping. In *ACM/IEEE International Symposium on Low-Power Electronics and Design* (Aug. 2010), pp. 189–194.

[6] DEMMEL, J., AND GEARHART, A. Instrumenting linear algebra energy consumption via on-chip energy counters. Tech. rep., Electrical Engineering and Computer Sciences, University of California at Berkeley, June 2012.

[7] DONGARRA, J., LTAIEF, H., LUSZCZEK, P., AND WEAVER, V. Energy footprint of advanced dense numerical linear algebra using tile algorithms on multicore architecture. In *Proc. of the 2nd International Conference on Cloud and Green Computing* (Nov. 2012).

[8] ELECTRONIC EDUCATIONAL DEVICES. Watts Up PRO. `http://www.wattsupmeters.com/`, May 2009.

[9] GE, R., FENG, X., SONG, S., CHANG, H.-C., LI, D., AND CAMERON, K. PowerPack: Energy profiling and analysis of high-performance systems and applications. *IEEE Transactions on Parallel and Distributed Systems 21*, 6 (May 2010).

[10] HACKENBERG, D., ILSCHE, T., SCHOENE, R., MOLKA, D., SCHMIDT, M., AND NAGEL, W. E. Power measurement techniques on standard compute nodes: A quantitative comparison. In *Proc. IEEE International Symposium on Performance Analysis of Systems and Software* (Apr. 2013).

[11] HACKENBERG, D., SCHÖNE, R., ILSCHE, T., MOLKA, D., SCHUCHART, J., AND GEYER, R. An energy efficiency feature survey of the Intel Haswell processor. In *Proc of the 11th Workshop on High-Performance, Power-Aware Computing* (May 2015).

[12] HÄHNEL, M., DÖBEL, B., VÖLP, M., AND HÄRTIG, H. Measuring energy consumption for short code paths using RAPL. In *Proc. Greenmetrics Workshop* (June 2012).

[13] INTEL. Beignet. `http://www.freedesktop.org/wiki/Software/Beignet/`.

[14] INTEL. *Intel, Math Kernel Library (MKL)*.

[15] INTEL. Voltage regulator-down (vrd) 11.0 processor power delivery design guidelines for desktop lga775 socket. http://www.intel.com/content/dam/doc/design-guide/voltage-regulator-down-11-0-processor-power-delivery-guide.pdf, Nov. 2006.

[16] INTEL. *Open Source Intel®HD Graphics Programmer's Reference Manual (PRM) Observability Performance Counters for Intel®Core ™ Processor Family*, 2013.

[17] INTEL. *Intel®Xeon®Processor E5-1600 and E5-2600 v3 Product Families, Volume 2 of 2, Register Data Sheet*, June 2015.

[18] INTEL CORPORATION. *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3: System Programming Guide*, June 2015.

[19] KHANNA, R., ZUHAYRI, F., NACHIMUTHU, M., LE, C., AND KUMAR, M. Unified extensible firmware interface: An innovative approach to DRAM power control. In *Proc. International Conference on Energy Aware Computing* (Nov. 2011).

[20] MAZOUZ, A., PRADELLE, B., AND JALBY, W. Statistical validation methodology of CPU power probes. In *Proc. of 1st International Workshop on Reproducibility in Parallel Computing* (Aug. 2014).

[21] MCCALPIN, J. STREAM: Sustainable memory bandwidth in high performance computers. `http://www.cs.virginia.edu/stream/`, 1999.

[22] PARADIS, C. Detailed low-cost energy and power monitoring of computing systems. Master's thesis, University of Maine, Aug. 2015.

[23] ROTEM, E., NAVEH, A., RAJWAN, D., ANATHAKRISHNAN, A., AND WEISSMANN, E. Power-management architecture of the Intel microarchitecture code-named Sandy Bridge. *IEEE Micro 32*, 2 (2012), 20–27.

[24] SQUAD. *Kerbal Space Program*.

[25] TREIBIG, J., HAGER, G., AND WELLEIN, G. LIKWID: A lightweight performance-oriented tool suite for x86 multicore environments. In *Proc. of the First International Workshop on Parallel Software Tools and Tool Infrastructures* (Sept. 2010).

[26] WEAVER, V. perf_event_open manual page. In *Linux Programmer's Manual*, M. Kerrisk, Ed. Dec. 2013.

[27] WEAVER, V. VMW group power measurement techniques. Tech. Rep. UMAINE-VMW-TR-POWER-MEASURE-2015-08, University of Maine, Aug. 2015.

[28] WHALEY, R. C., AND DONGARRA, J. Automatically tuned linear algebra software. In *Proc. of Ninth SIAM Conference on Parallel Processing for Scientific Computing* (1999).