

# A Validation of DRAM RAPL Power Measurements

Spencer Desrochers, Chad Paradis and Vincent M. Weaver  
 Electrical and Computer Engineering, University of Maine  
 {spencer.desrochers,chad.paradis,vincent.weaver}@maine.edu

**Abstract**—Recent Intel processors support the Running Average Power Level (RAPL) interface, which among other things provides estimated energy measurements for the CPUs, integrated GPU, and DRAM. These measurements are easily accessible by the user, and can be gathered by a wide variety of tools, including the Linux perf\_event interface. This allows unprecedented easy access to energy information when designing and optimizing codes that must be energy-aware.

While greatly useful, on most systems these RAPL measurements are just estimated values, generated on the fly by an on-chip energy model. The values are not documented well, and the results (especially the DRAM results) have had only limited validation done.

We take an Intel Haswell desktop machine and instrument the hardware to provide actual, physical measurements of various components, including the DRAM. We explore the many challenges encountered when instrumenting systems for detailed power measurement. We then compare the results gathered against those provided by the RAPL interface. We find that the RAPL results match overall energy and power trends, but often have lower absolute measurements, especially when the CPU or DRAM is idle. Care should be taken when using the RAPL counters results, as they are the results of a model and may not directly correspond with actual DRAM behavior.

## I. INTRODUCTION

In 2011 (with the SandyBridge microarchitecture) Intel introduced the Running Average Power Level (RAPL) interface [1]. This is an advanced powercapping infrastructure which allows the user (or operating system) to specify maximum power limits. The processor can run at the highest possible speed while automatically throttling back to stay within power or thermal bounds. In order to respect these power limits, a processor must be aware of its current power usage. It is usually impractical to measure power directly, so instead the processor typically estimates these values using a power model based on performance counters, temperature, and other inputs. As a bonus feature, the results of the power model are made available to the user via a model-specific register (MSR) and can be used when characterizing workloads. The interface provides RAPL estimated power measurements for the total processor package, the aggregate total of all cores, and optionally for the DRAM and integrated GPU.

Power and energy are becoming important metrics when optimizing code. At the low end, energy use in cell phones and other mobile devices is important to provide long battery life. In large supercomputing environments, when there are millions of cores active, saving just 1 Watt per core can amount to megawatts (and millions of dollars) in savings. To properly optimize for power though you need some way of measuring it.

Traditionally CPU cores are the primary concern when worrying about power consumption of large systems. However DRAM can have a significant impact on overall power; on a 16-core Haswell-EP server with 80GB of DDR4 RAM running the Linpack benchmark, RAPL reports the cores average 130W and the RAM 13W. In this case the RAM consumes more power than an additional core would.

Gathering actual power and energy results is difficult in modern systems. Most devices are not instrumented for power measurement, and adding suitable interfaces usually involves intrusive modifications to the system’s power distribution network. Some parts of a system, such as the CPU, can be particularly difficult to instrument due to being soldered to the motherboard with the numerous power traces being buried inside of a multi-layer circuit board.

The difficulties with conducting actual power measurements make the RAPL interface an attractive alternative for gathering detailed power measurements. Before this can be done, however, the interface needs to be properly validated. There has been initial work in validating RAPL results [2], [3], [4] but this has focused on CPU power measurements. In this work we not only look at CPU values but also those gathered for DRAM and GPU.

We extensively instrument a Haswell desktop machine for power measurement, and detail the many difficulties found along the way. We then run various benchmarks and compare the RAPL results to the actual measurements. We find that RAPL measurement trends closely match those found by real hardware, but the absolute results are offset from each other, especially when the resource in question is idle.

## II. RAPL BACKGROUND AND RELATED WORK

RAPL is documented in Chapter 14.3 of the Intel Volume3b Documentation [5] although many of the details of

the interface are not as complete as they could be.

RAPL provides estimated per-package energy estimates, meaning the totals are at the socket (not per-core) level. Energy measurements available include: total package, Power Plane 0 (PP0) which is the aggregate total of all cores, Power Plane 1 (PP1) which is an implementation-defined part of the uncore (usually the GPU), and the DRAM. Measurement availability varies by chip model; DRAM measurements were originally only available for server systems but starting with Haswell are available on all processors. Similarly, GPU measurements are not available on server versions of the 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 [3] although it is possible to mitigate this by carefully monitoring when updates happen and starting measurements at the transition [6].

The minimal energy increment can vary; on regular Haswell this can be read from a register (it is roughly  $61\mu\text{J}$ ) but on Haswell-EP it is documented elsewhere [7] as being fixed at  $15\mu\text{J}$ .

To read the RAPL registers one must have ring-0 access, which usually requires some sort of device driver in the operating system. Linux provides at least three ways to access the values: raw MSR access (via the `/dev/msr` interface), the `perf_event` subsystem, or the powercapping interface visible under `/sys`. For various security reasons (including the system-wide nature of the measurements) reading the values requires root permissions (in theory an attacker could use the power metrics to spy on what other users of the CPU are doing).

#### A. CPU RAPL Validation

Various groups have investigated the accuracy of the RAPL counters against real hardware. Hähnel et al. [6] 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. [4] provide one validation graph of an unspecified benchmark showing a close match of RAPL CPU and package measurements to actual measurements.

Dongarra et al. [2] compare RAPL measurements on a SandyBridge machine using PAPI to actual measurements found using PowerPack [8] on a completely different (non-Sandybridge) microarchitecture. They use LU factorization as a workload.

Demmel and Gearhart [9] validate two SandyBridge machines against RAPL Package with the STREAM [10] benchmark and a full-system wall power meter. They do not look at the DRAM measurements.

Hackenberg et al. [3] validate RAPL (and the similar AMD APM interface) on a variety of SandyBridge hardware. They measure at the wall outlet, 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 [11] 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. [12] 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.

#### B. DRAM RAPL Validation

The RAPL DRAM interface was first described by David et al. [1]. 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 validate 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. [13] describe the weights used in RAPL DRAM measurements. They measure actual DRAM results using a riser with a  $5\text{m}\Omega$  sense resistor sampled at 100Hz. They find RAPL results within 2.3% of actual measurements.

### III. 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.

#### A. Hardware Measurement Setup

System-wide power is measured using a WattsUpPro? [14] 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. This is typically how previous work [16], [17], [18] has measured CPU power, although on another Haswell system we own the connector is specifically marked as “CPU/NIC/USB” so it is quite possible that these other hardware components are interfering with pure CPU measurements. Due to potentially high currents

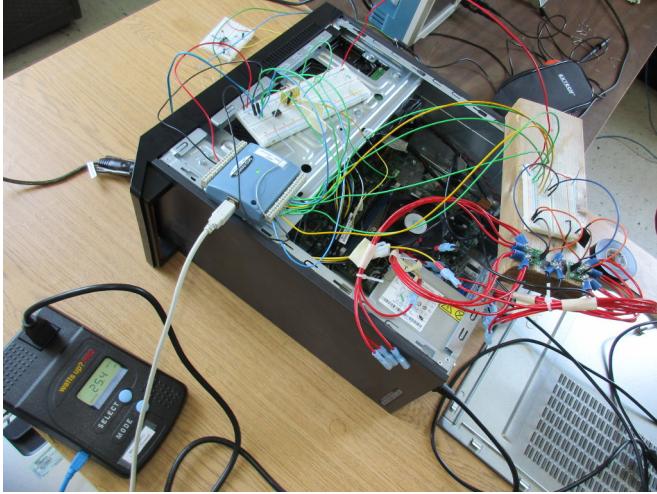


Fig. 1. Our instrumented test machine.

involved (in the tens of Amps) an ACS715 Hall Effect sensor [19] is used for measurement rather than a sense resistor. The hall effect sensor provides a voltage output that is proportional to the current flowing through the device.

The DRAM is instrumented by using a JET-5464 DDR3 DIMM Extender card which has a  $3.3m\Omega$  sense resistor built in. The voltage drop across this resistor can be used to calculate the current draw via Ohm's Law  $I = \frac{V}{R}$  where V is the voltage drop and R is  $3.3m\Omega$ . This current can be passed into the equation  $P = IV$  to calculate the power, with this V being the DDR3 RAM supply voltage of 1.5V. The original voltage drop being measured is very small due to the small resistor value, so an INA122 instrumentation amplifier [20] is used to amplify the signal before measurement.

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.

A picture of our instrumented test machine can be seen in Figure 1.

### B. RAPL Measurement Setup

The RAPL values are gathered using the perf tool that comes with the Linux kernel and uses the perf\_event [21] 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
```

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

### C. Data Synchronization

The measurements we gather end up on two different machines. The RAPL measurements are collected locally on the machine running the benchmarks (unfortunately possibly skewing the results due to measurement overhead). The actual power measurements are collected at the same time on a separate machine using the Measurement Computing USB Data Acquisition board. When collating the results it is necessary to line up the start and stop times of the measurements as closely as possible.

There are various ways to do this and all have their limitations. For these experiments we modify the perf tool to toggle the DTR line of a serial port. This serial port line is connected to one of the inputs on our data acquisition device, allowing our recorded traces to have a clear signal of when perf measurements were started on the device under test.

### D. RAPL Measurement Overhead

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 has a convenient "print every interval" (-I) mode but it is hard-coded to not allow measurement 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 measurement/interrupt overhead. We investigated writing a custom tool that would use the perf\_event interface's sampling/mmap() ring-buffer recording mode 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 (10Hz) sampling frequency. Possible ways to avoid this would be to use a different performance interface such as LIKWID [22] or to read the MSRs directly.

### E. CPU/DRAM Benchmarks

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

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.

For a baseline we look at an idle system, which is just recording system behavior when a "sleep" command is issued. Note that we do have a full Debian Jessie environment running so the system is not truly idle (i.e. we are not running

in single-user mode with all unnecessary processes killed). This is because we are interested in the power behavior of a real-life system that is sitting unused.

In order to exercise the DRAM we look at the STREAM [10] 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) [23] that ships with Debian Linux,
- OpenBLAS [24] 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 [25].

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.

#### F. GPU Benchmarks

It is difficult to obtain power measurements for the integrated GPU, as it is on-die and there is no way to intercept the input voltages. There are additional (non-power related) hardware performance counters available for the integrated GPU [26] but as of yet the Linux support for reading these is not complete.

The first benchmark we look at is SmallptGPU2 [27], an OpenCL ray-tracer. We use Beignet [28] 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 [29]. We record a 25s long snapshot of behavior while launching a rocket in-game.

## IV. 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 RAPL DRAM power follows the LLC rate, and the RAPL package power follows the CPI metric. For the GPU benchmarks we additionally measure the RAPL core and RAPL GPU values. Finally we take actual hardware measurements of total system power, the P4 ATX connector (which should be closely related to package power), as well as the actual DIMM power.

### A. CPU Benchmark Results

In Figure 2 we show the results of an idle system. No attempts were made to limit the number of background jobs running, or in any way artificially limiting the background noise. We wanted to measure the power behavior of a typical system sitting unused. It turns out that this setup has surprisingly high CPI and cache variability.

The CPU RAPL and actual power measurements match each other fairly well, although RAPL seems to underestimate the power slightly (but this could be due to the P4 connector powering devices other than the CPU, as well as losses in the power converters that convert the 12V input to the much lower voltages used by the CPU).

The DRAM RAPL values are much lower than actual values, possibly RAPL has trouble estimating power if the DIMM has entered a low-power mode.

Figure 3 shows the results when DRAM is being stressed by a multi-core aware OpenMP version of the STREAM benchmark.

The total CPU package measurements match closely the CPI results from the performance counters, and the DRAM results match closely the last level cache misses. Again, the CPU RAPL estimates read a bit lower than the actual measurements. While under high utilization the DRAM RAPL results closely match measured results, but when memory utilization drops toward idle the RAPL values read low.

Figure 4, 5 and 6 show Linpack running with various BLAS libraries. Despite being the same benchmark, the underlying BLAS libraries lead to markedly different phase behaviors. The phase behavior is also much more complex than the other benchmarks we investigate. In some of the figures it appears as though the total CPU package power is higher than the wall-outlet power measurement; this is just an artifact due to the much lower sampling frequency of the WattsUpPro? device.

In the ATLAS results (Figure 4) there are periodic spikes in cache misses which correspond to increased memory power usage as well as dips in cpu power usage. The DRAM RAPL measurements seem to be consistently lower than measured, even when the memory system is busy. This could be a measurement artifact due to the higher sampling rate of the hardware measurement compared to the lower rate that we sample the RAPL counters.

The OpenBLAS results (Figure 5) have different underlying behavior to the ATLAS results, but the power values show similar trends, with the DRAM results being consistently lower.

The Intel MKL results (Figure 6) again have similar trends, with the DRAM results being lower.

### B. GPU Benchmark Results

Figure 7 shows the results when the GPU is being used for OpenCL raytracing calculations. According to the RAPL

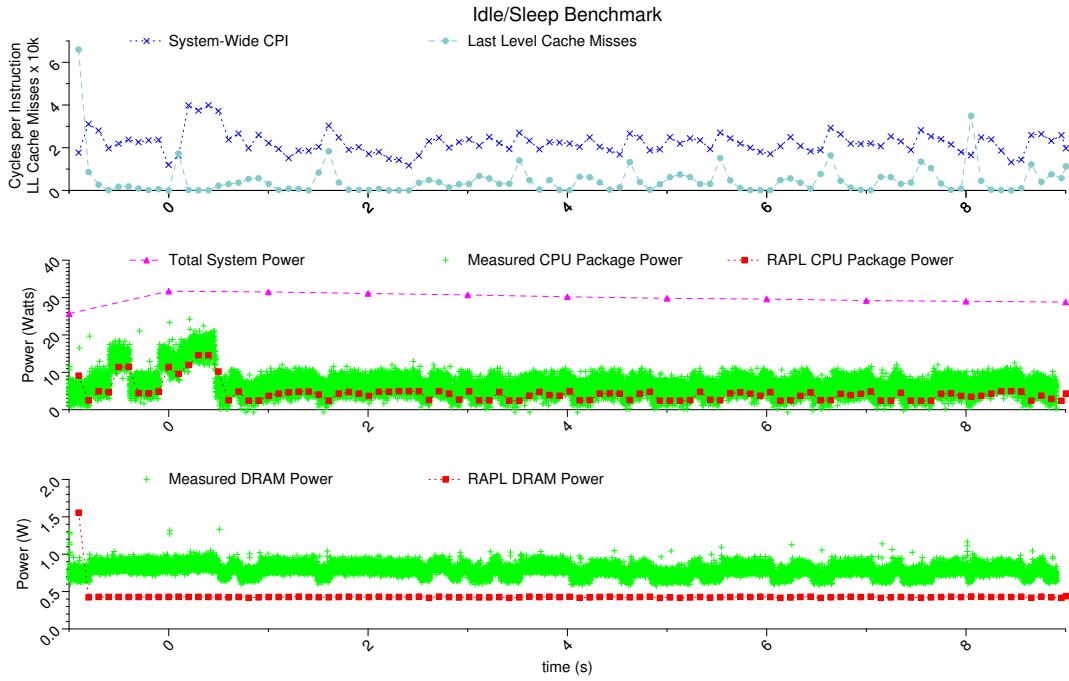


Fig. 2. Power measurements for an idle system (perf is run on a call to the `sleep` command). While CPU actual vs estimated is close, RAPL DRAM measurements underestimate the power used.

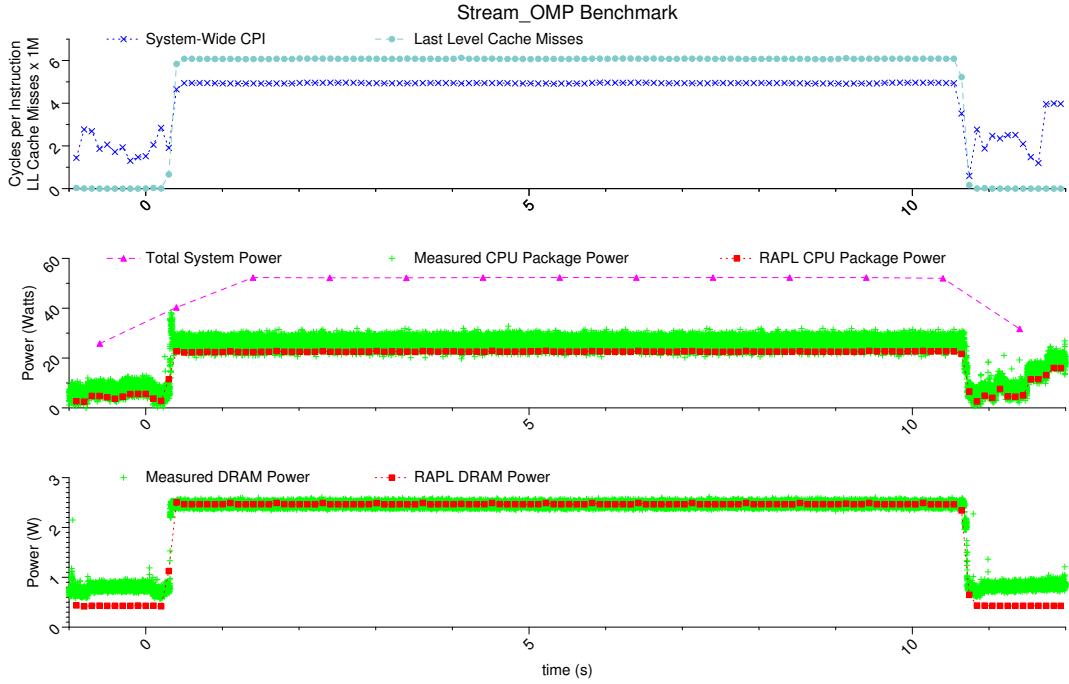


Fig. 3. Power measurements while running an OpenMP version of the memory-intensive STREAM benchmark. The DRAM measurements match estimated RAPL results when under heavy memory stress, but when memory usage drops RAPL again underestimates the power.

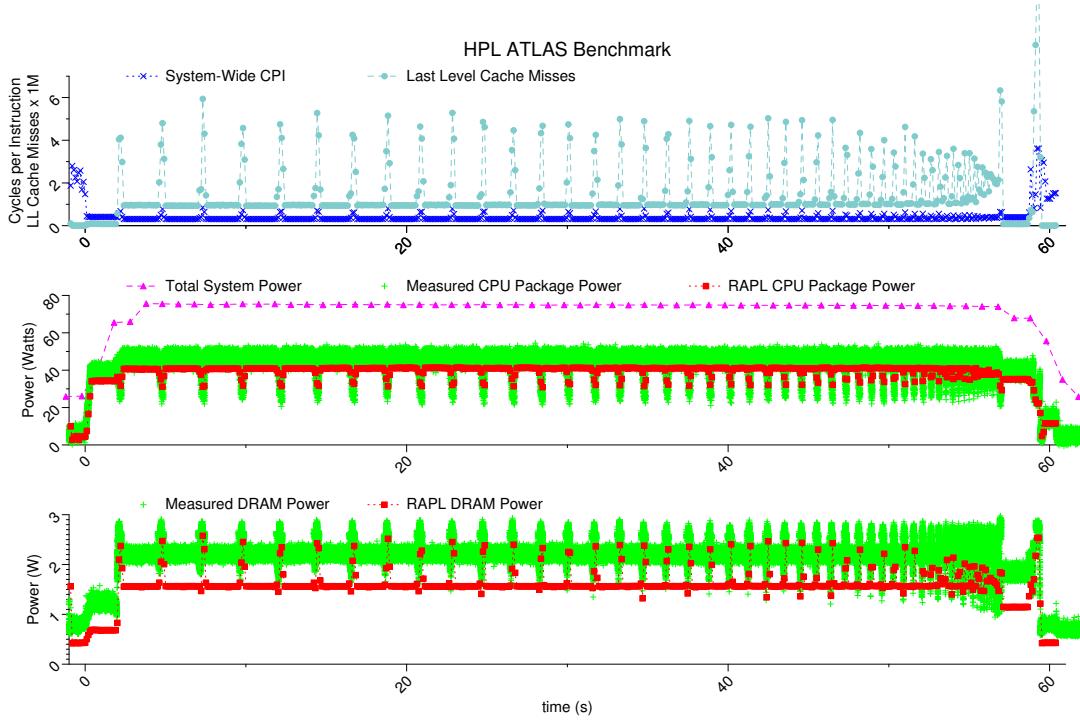


Fig. 4. Linpack (HPL) using Atlas BLAS. The periodic spikes in cache misses correspond with rises in DRAM power but dips in CPU power. It appears that package power is higher than total system power, but this is an artifact of the low sampling period of the WattsUpPro meter.

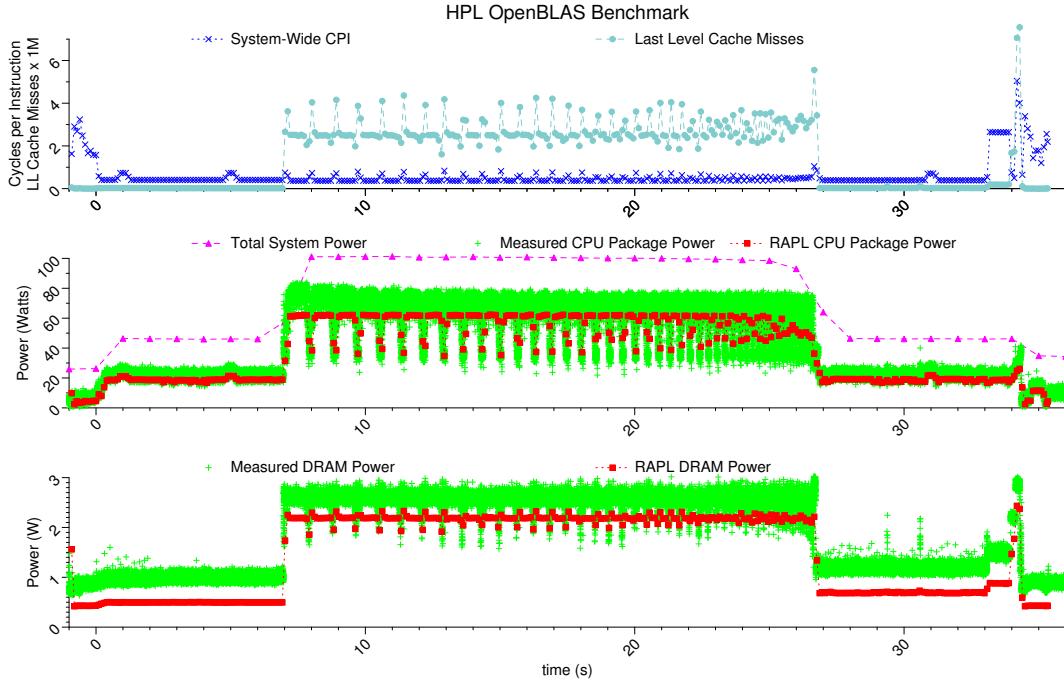


Fig. 5. Linpack (HPL) using OpenBLAS. The DRAM estimated RAPL power is consistently less than total power. It appears that package power is higher than total system power, but this is an artifact of the low sampling period of the WattsUpPro meter.

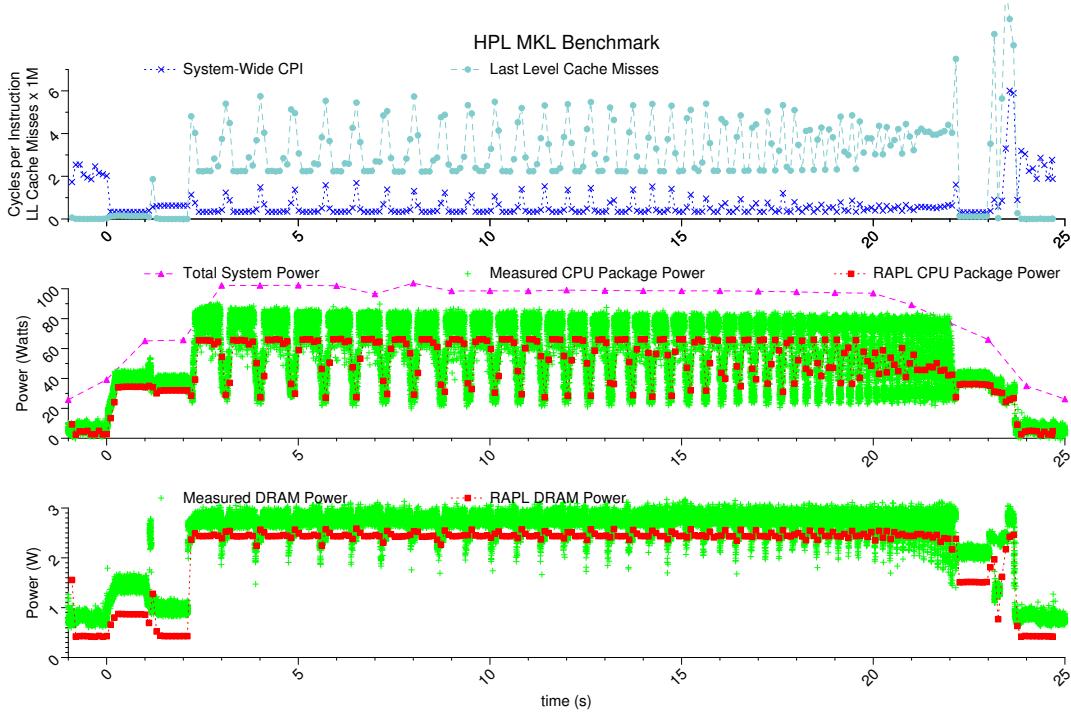


Fig. 6. Linpack (HPL) using Intel MKL BLAS. The DRAM estimated RAPL power is lower than measured when the DRAM is less active. It appears that package power is higher than total system power, but this is an artifact of the low sampling period of the WattsUpPro meter.

results the actual cores are almost completely idle and contributing very little power. The GPU is using the bulk of the power, and there is an interesting 5W reported by the package not accounted for by the GPU (some other aspect of the uncore?)

The DRAM behavior is complex and the RAPL readings do not seem to capture this, possibly due to the low sampling frequency. Another possibility is that the GPU is doing extensive DMA transfers which might not be accounted for by the RAPL model.

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). Again, the DRAM RAPL count seems to not be accounting for GPU interactions.

### C. Overall Totals

In addition to instantaneous power results, it is often useful to compare the total energy used across a benchmark run. Table I shows overall summaries for the CPU results and Table II 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. The undercount by RAPL is not a fixed amount, but it varies with the workload.

Despite the undercount by RAPL, overall metrics such as GFlops/Watt, Total Energy, and Average power still give the same rankings when sorted by actual results as they do when sorting by RAPL results.

At least with our LINPACK example, if we were evaluating the power efficiency of the three BLAS libraries we would get the same overall power/performance rankings with either methodology, even if the exact measurements do not match up. This gives hope that RAPL measurements are “good enough” in the common situation where detailed, fine-grained power measurement cannot be undertaken.

## V. CONCLUSION AND FUTURE WORK

Our work confirms previous results that CPU RAPL power measurements closely follow actual CPU power. However we find, that on our Haswell machine at least, that DRAM RAPL values do not always match actual measurements. When under load the DRAM RAPL and real values match well, but when idle or when the GPU is active the RAPL readings tend to underestimate the actual results.

Our current work is done on a desktop system with a low-power envelope processor. We also would like to explore a server system, although this is made difficult by various hardware limitations. Our Sandybridge-EP machine is unable to gather RAPL DRAM measurements due to firmware issues, and our Haswell-EP server has DDR4 DIMMs which

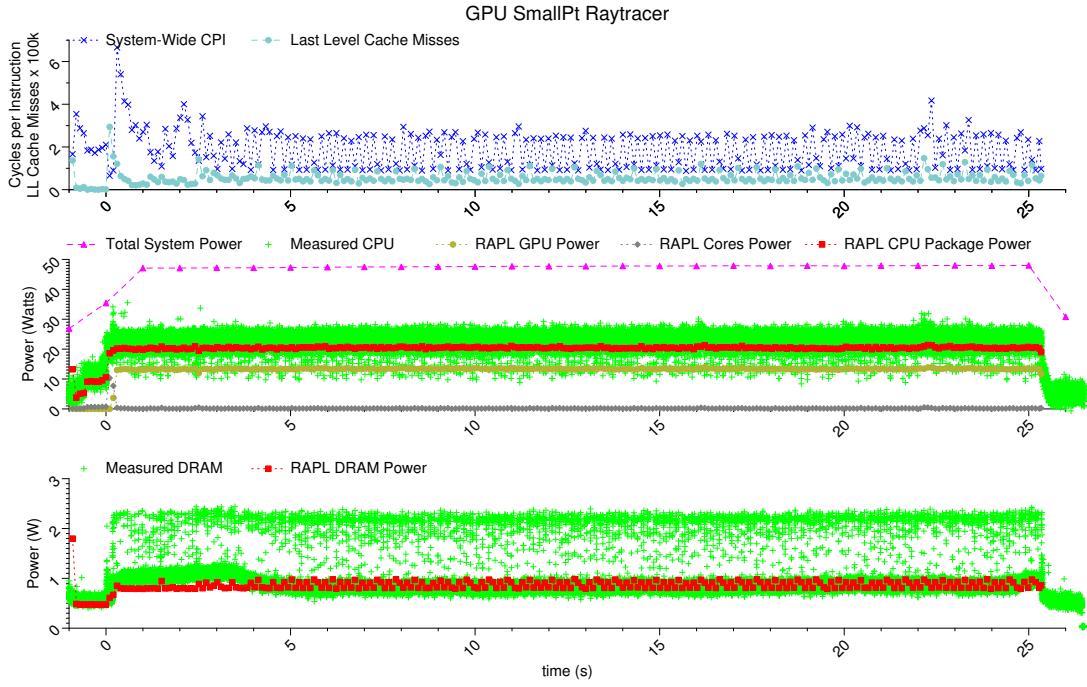


Fig. 7. Smallpt OpenCL Raytracer. The majority of package power is consumed by the GPU, with the CPU cores mostly idle. The complex DRAM power behavior is not captured well by RAPL.

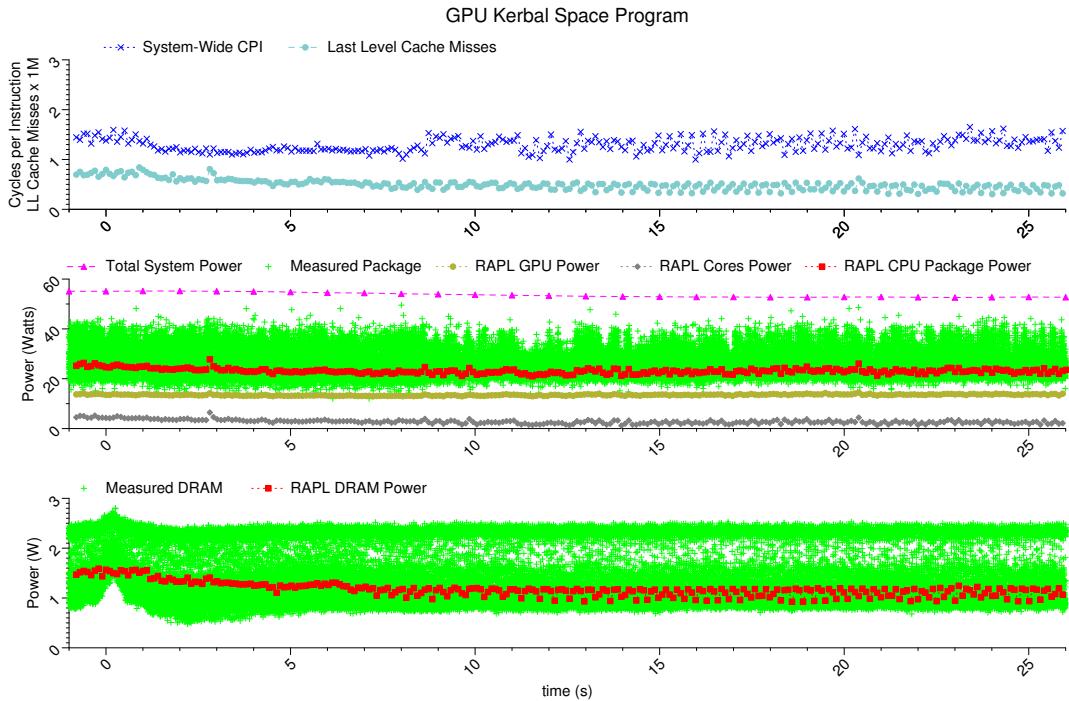


Fig. 8. Kerbal Space Program, a 3D/GPU intensive Game. The majority of package power is consumed by the GPU, with the CPU cores mostly idle. The complex DRAM power behavior is not captured well by RAPL.

TABLE I  
CPU RESULT SUMMARY. IN THE SPLIT ROWS, THE TOP ROW IS ACTUAL MEASUREMENT AND BOTTOM ROW IS RAPL.

Benchmark	Time (s)	<i>GFlops</i>	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	22.1	—
			523.9	20.0	—
OpenGL-kerbal	26.7	—	710.0	26.6	—
			617.6	23.1	—

will require obtaining different DIMM measurement risers which are proving difficult to source. DDR4 DIMMs also have multiple voltages (1.2V VDD and 2.5V wordline boost) which will make power measurement more complicated. In addition, Hackenberg et al. [12] 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.

In addition to server systems, it would be interesting to gather results on both older (IvyBridge) and newer (Broadwell and Skylake) to see if the RAPL interfaces have improved with time. In addition we would like to expand our results by measuring with multiple DIMMs installed, enabling monitoring of NUMA workloads. These research efforts may be difficult, as they involve finding instrumented DDR4 memory risers, and also gathering and correlating multiple simultaneous DRAM measurements.

Despite the under-estimation of power on our test machine, the RAPL counters do track overall program behavior and can be a useful measurement methodology especially when compared to the alternatives of either complex hand-instrumentation of every machine of interest or else having no memory energy information at all.

## REFERENCES

- [1] H. David, E. Gorbatov, U. Hanebutte, R. Khanna, and C. Le, “RAPL: Memory power estimation and capping,” in *ACM/IEEE International Symposium on Low-Power Electronics and Design*, pp. 189–194, Aug. 2010.
- [2] J. Dongarra, H. Ltaief, P. Luszczek, and V. Weaver, “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.
- [3] D. Hackenberg, T. Ilsche, R. Schoene, D. Molka, M. Schmidt, and W. E. Nagel, “Power measurement techniques on standard compute nodes: A quantitative comparison,” in *Proc. IEEE International Symposium on Performance Analysis of Systems and Software*, Apr. 2013.
- [4] E. Rotem, A. Naveh, D. Rajwan, A. Anathakrishnan, and E. Weissmann, “Power-management architecture of the Intel microarchitecture code-named Sandy Bridge,” *IEEE Micro*, vol. 32, no. 2, pp. 20–27, 2012.
- [5] Intel Corporation, *Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3: System Programming Guide*, June 2015.
- [6] M. Hänel, B. Döbel, M. Völp, and H. Härtig, “Measuring energy consumption for short code paths using RAPL,” in *Proc. Greenmetrics Workshop*, June 2012.
- [7] Intel, *Intel®Xeon®Processor E5-1600 and E5-2600 v3 Product Families, Volume 2 of 2, Register Data Sheet*, June 2015.
- [8] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. Cameron, “PowerPack: Energy profiling and analysis of high-performance systems and applications,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, May 2010.
- [9] J. Demmel and A. Gearhart, “Instrumenting linear algebra energy consumption via on-chip energy counters,” tech. rep., Electrical Engineering and Computer Sciences, University of California at Berkeley, June 2012.
- [10] J. McCalpin, “STREAM: Sustainable memory bandwidth in high performance computers.” <http://www.cs.virginia.edu/stream/>, 1999.
- [11] A. Mazouz, B. Pradelle, and W. Jalby, “Statistical validation methodology of CPU power probes,” in *Proc. of 1st International Workshop on Reproducibility in Parallel Computing*, Aug. 2014.
- [12] D. Hackenberg, R. Schöne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer, “An energy efficiency feature survey of the Intel Haswell processor,” in *Proc. of the 11th Workshop on High-Performance, Power-Aware Computing*, May 2015.
- [13] R. Khanna, F. Zuhayri, M. Nachimuthu, C. Le, and M. Kumar, “Unified extensible firmware interface: An innovative approach to DRAM power control,” in *Proc. International Conference on Energy Aware Computing*, Nov. 2011.
- [14] Electronic Educational Devices, “Watts Up PRO.” <http://www.wattsupmeters.com/>, May 2009.
- [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] H. Chen, S. Wang, and W. Shi, “Where does the power go in a computer system: Experimental analysis and implications,” in *International Green Computing Conference*, pp. 1–6, July 2011.
- [17] S. Khoshbakht and N. Dimopoulos, “Relating application memory activity to processor power,” in *Proc. International Conference on Parallel Processing*, pp. 849–857, Oct. 2013.
- [18] M. C. no, S. Catalán, R. Mayo, and E. Quintana-Ortí, “Reducing the cost of power monitoring with DC Wattmeters,” *Computer Science – Research and Development*, vol. 30, pp. 107–114, May 2015.
- [19] Allegro MicroSystems LLC, *ACS715: Automotive Grade, Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1*

TABLE II  
DRAM RESULT SUMMARY. IN THE SPLIT ROWS, THE TOP ROW IS ACTUAL MEASUREMENT AND BOTTOM ROW IS RAPL.

Benchmark	Time (s)	<i>GFlops</i>	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	—

- kVRMS Voltage Isolation and a Low-Resistance Current Conductor Lightweight Profiling Specification*, 2013.
- [20] Burr-Brown, *INA122: Single Supply, MicroPower Instrumentation Amplifier*, Oct. 1997.
  - [21] V. Weaver, “perf\_event\_open manual page,” in *Linux Programmer’s Manual* (M. Kerrisk, ed.), Dec. 2013.
  - [22] J. Treibig, G. Hager, and G. Wellein, “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.
  - [23] R. C. Whaley and J. Dongarra, “Automatically tuned linear algebra software,” in *Proc. of Ninth SIAM Conference on Parallel Processing for Scientific Computing*, 1999.
  - [24] “OpenBLAS an optimized BLAS library website.” <http://www.openblas.net/>.
  - [25] Intel, *Intel Math Kernel Library (MKL)*.
  - [26] Intel, *Open Source Intel®HD Graphics Programmer’s Reference Manual (PRM) Observability Performance Counters for Intel®Core™ Processor Family*, 2013.
  - [27] D. Bucciarelli, “Smallptgpu2.” <http://davibu.interfree.it/opencl/smallptgpu2/smallptGPU2.html>.
  - [28] Intel, “Beignet.” <http://www.freedesktop.org/wiki/Software/Beignet/>.
  - [29] Squad, *Kerbal Space Program*.