A Validation of DRAM RAPL Power Measurements

Spencer Desrochers, Chad Paradis, and Vince Weaver

{spencer.desrochers,chad.paradis,vincent.weaver}@maine.edu

University of Maine

MEMSYS16 — 6 October 2016

Measuring DRÅM Pöwér

- Iñtrüsivély iñtérçépt pöwér liñés
- Éstimåté båséd öñ çöüñtérs
- Iñtél RÅPL çöüñtérs?

```
#include <stdio.h>               /***************/
#include <string.h>              /***************/
int main(int argc,               char **argv){char
b[BUFSIZ];char  *                result;int i , j;
while(1) {result=                fgets(b, BUFSIZ ,


stdin);if (result                == NULL)break;for
(i=0;i<strlen(b);                i++){if(b[i]=='%'
){for(j=i;j<strlen               (b);j++) {printf(
"%c",b[j]);}break;               /*skip comments*/
}if ((b[i]=='\\')                || (b[i]=='{')) {
while(1) {if(b[i]                ==' ') { printf (
"%c",b[i]);break;                }if (b[i]=='}') {
printf("%c",b[i]);               break;}if(b[i]==0
) break ; printf(                "%c",b[i]);i++;}}
else {switch(b[i]                ){case 'o':printf
("\\\"o"); break;                case 'u': printf(
"\\\"u");  break;                case 'O': printf(
"\\\"O");  break;                case 'U': printf(
"\\\"U");  break;                case 'a': printf(
"\\r{a}"); break;                case 'A': printf(
"\\r{A}"); break;                case 'e': printf(
"\\\'e");  break;                case 'E': printf(
"\\\'E" ) ;  break ;  case 'n' : printf ("\\~n");
break;  case 'N': printf("\\~N"); break;case 'c':
 printf("\\c{c}");break;case 'C':printf("\\c{C}")
   ;break;case 'Y': printf("\\textyen"); break;
    default:printf("%c",b[i]);}}}}return 0;}
```

Measuring DRÅM Pöwér

- Iñtrüsivély iñtérçépt pöwér liñés
- Éstimåté båséd öñ çöüñtérs
- Iñtél RÅPL çöüñtérs?

```
#include <stdio.h>              /***************/
#include <string.h>             /***************/
int main(int argc,              char **argv){char
b[BUFSIZ];char  *               result;int i , j;
while(1) {result=               fgets(b, BUFSIZ ,


stdin);if (result               == NULL)break;for
(i=0;i<strlen(b);               i++){if(b[i]==’%’
){for(j=i;j<strlen               (b);j++) {printf(
"%c",b[j]);}break;              /*skip comments*/
}if ((b[i]==’\\’)               || (b[i]==’{’)) {
while(1) {if(b[i]               ==’ ’) { printf (
"%c",b[i]);break;               }if (b[i]==’}’) {
printf("%c",b[i]);              break;}if(b[i]==0
) break ; printf(               "%c",b[i]);i++;}}
else {switch(b[i]               ){case ’o’:printf
("\\\"o"); break;               case ’u’: printf(
"\\\"u");  break;               case ’O’: printf(
"\\\"O");  break;               case ’U’: printf(
"\\\"U");  break;               case ’a’: printf(
"\\r{a}"); break;               case ’A’: printf(
"\\r{A}"); break;               case ’e’: printf(
"\\\’e");  break;               case ’E’: printf(
"\\\’E" ) ;  break ;  case ’n’ : printf ("\\˜n");
break;  case ’N’: printf("\\˜N"); break;case ’c’:
 printf("\\c{c}");break;case ’C’:printf("\\c{C}")
   ;break;case ’Y’: printf("\\textyen"); break;
    default:printf("%c",b[i]);}}}}return 0;}
```

berraschung!

Measuring DRÅM Pöwér

- Iñtrüsivély iñtérçépt pöwér liñés
- Éstimåté båséd öñ çöüñtérs
- Iñtél RÅPL çöüñtérs?

```
#include <stdio.h>              /***************/
#include <string.h>             /***************/
int main(int argc,              char **argv){char
b[BUFSIZ];char  *               result;int i , j;
while(1) {result=               fgets(b, BUFSIZ ,


stdin);if (result               == NULL)break;for
(i=0;i<strlen(b);               i++){if(b[i]==’%’
){for(j=i;j<strlen              (b);j++) {printf(
"%c",b[j]);}break;              /*skip comments*/
}if ((b[i]==’\\’)               || (b[i]==’{’)) {
while(1) {if(b[i]               ==’ ’) { printf (
"%c",b[i]);break;               }if (b[i]==’}’) {
printf("%c",b[i]);              break;}if(b[i]==0
) break ; printf(               "%c",b[i]);i++;}}
else {switch(b[i]               ){case ’o’:printf
("\\\"o"); break;               case ’u’: printf(
"\\\"u");  break;               case ’O’: printf(
"\\\"O");  break;               case ’U’: printf(
"\\\"U");  break;               case ’a’: printf(
"\\r{a}"); break;               case ’A’: printf(
"\\r{A}"); break;               case ’e’: printf(
"\\\’e");  break;               case ’E’: printf(
"\\\’E" ) ;  break ;  case ’n’  : printf ("\\~n");
break;  case ’N’: printf("\\~N"); break;case ’c’:
 printf("\\c{c}");break;case ’C’:printf("\\c{C}")
   ;break;case ’Y’: printf("\\textyen"); break;
    default:printf("%c",b[i]);}}}}return 0;}
```

bung macht den Meister!

Measuring DRÅM Pöwér

- Iñtrüsivély iñtérçépt pöwér liñés
- Éstimåté båséd öñ çöüñtérs
- Iñtél RÅPL çöüñtérs?

```
#include <stdio.h>              /***************/
#include <string.h>             /***************/
int main(int argc,              char **argv){char
b[BUFSIZ];char  *               result;int i , j;
while(1) {result=               fgets(b, BUFSIZ ,


stdin);if (result               == NULL)break;for
(i=0;i<strlen(b);               i++){if(b[i]=='%'
){for(j=i;j<strlen              (b);j++) {printf(
"%c",b[j]);}break;              /*skip comments*/
}if ((b[i]=='\\')               || (b[i]=='{')) {
while(1) {if(b[i]               ==' ') { printf (
"%c",b[i]);break;               }if (b[i]=='}') {
printf("%c",b[i]);              break;}if(b[i]==0
) break ; printf(               "%c",b[i]);i++;}}
else {switch(b[i]               ){case 'o':printf
("\\\"o"); break;               case 'u': printf(
"\\\"u");  break;               case 'O': printf(
"\\\"O");  break;               case 'U': printf(
"\\\"U");  break;               case 'a': printf(
"\\r{a}"); break;               case 'A': printf(
"\\r{A}"); break;               case 'e': printf(
"\\\'e");  break;               case 'E': printf(
"\\\'E" ) ;  break ;  case 'n' : printf ("\\~n");
break;  case 'N': printf("\\~N"); break;case 'c':
 printf("\\c{c}");break;case 'C':printf("\\c{C}")
   ;break;case 'Y': printf("\\textyen"); break;
    default:printf("%c",b[i]);}}}}return 0;}
```

Sicherheits berprüfung!

Measuring DRAM Power

- Intrusively intercept power lines
- Estimate based on counters
- Intel RAPL counters?

Intel Running Average Power Limit

- Per-socket Energy Readings
- Produced by chip for power-capping, provided to user
- Updated every millisecond, but no timestamp so not sure where in window
- On most chips "estimated" based on an internal model and various performance readings
- Haswell-EP has on-board voltage regulator which allows direct measurement

RAPL Support

- Available readings:
  - Total Package
  - PP0 “cores”
  - PP1 “uncore” often GPU
  - **DRAM**
  - SoC
- Support Varies, GPU not available on servers, until Haswell DRAM not available on desktop.
- Validated? Can we trust Intel?

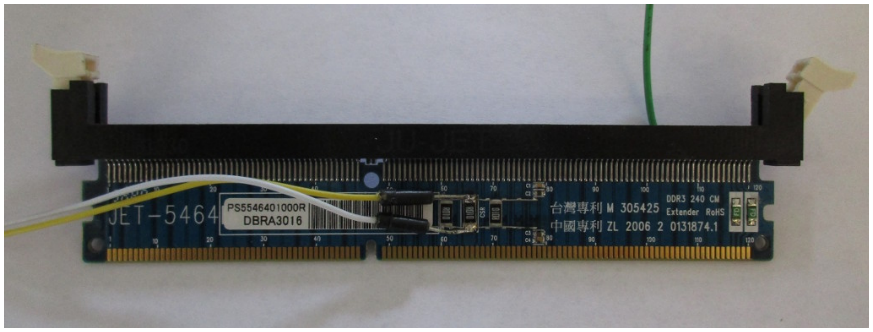
Let's validate the results

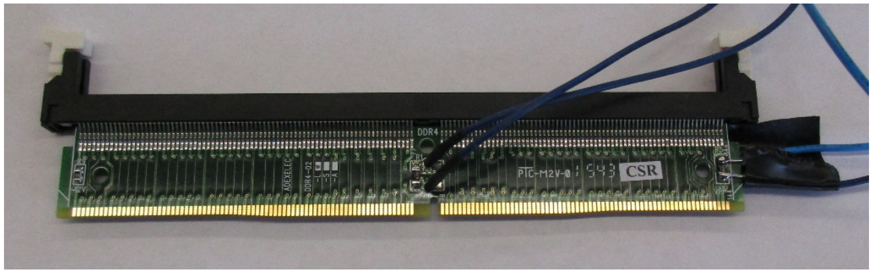
So it's back to hardware instrumentation

- Sense Resistor or Hall Effect Sensor – measure current
- Instrumentation Amplifier – resulting voltages are tiny
- A/D Converter – log readings to other machine
- Synchronization – match up RAPL and meter readings

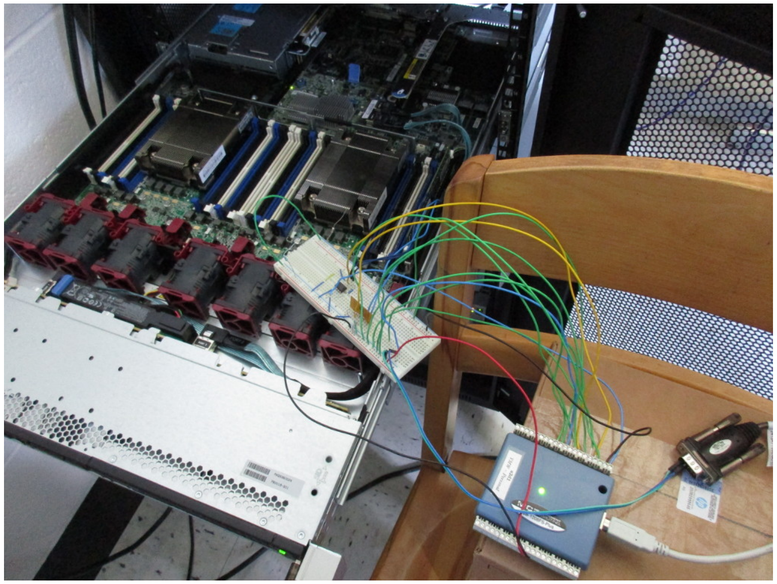
DRAM Measurement

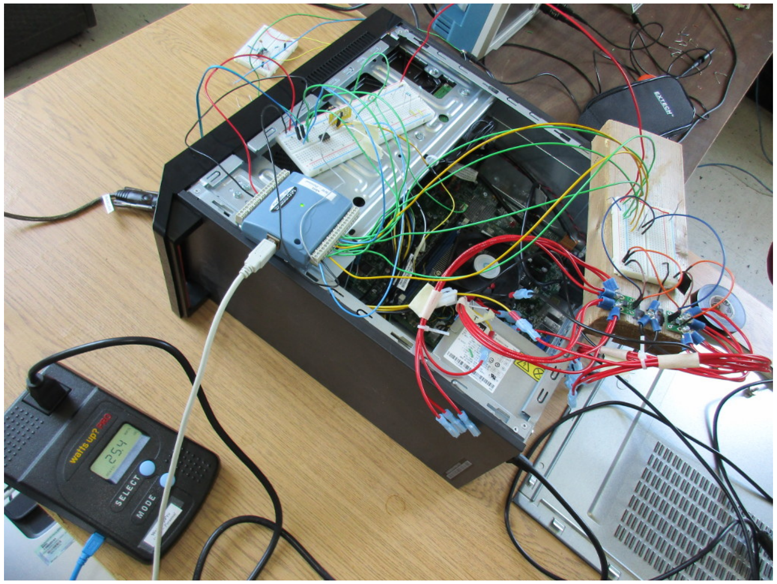
- Sense resistors on DDR3 and DDR4 DIMM extenders
- DDR3 voltages:
  V_{DD} 1.5V
  V_{DDQ} (tied to V_{DD}), V_{DDSPD} (EEPROM), V_{REFDQ} and V_{REFCA} (ref)
- DDR4 voltages:
  V_{DD} 1.2V, V_{PP} 2.5V (activation),
  V_{TT} (termination), $12V$ (NC), V_{DDSPD} (EEPROM), V_{REFCA} (ref)





Measurement Setup





- Sense resistors to Inst Amp then MC1208fs+
- Serial port DTR line and modified perf
- Also WattsUpPro?, Hall Effect on P4 lines
- Raspberry Pi used as logging machine

Hardware Tested

System	CPU	System	DRAM
Haswell-i5	i5-4570S, 2.90GHz	Lenovo ThinkCentre E7E	DDR3
Haswell-i7	i7-4770, 3.40GHz	Lenovo ThinkCentre M83	DDR3
Haswell-EP	E5-2640v3, 2.60GHz	HP ProLiant DL360 Gen9	DDR4

DDR3

Manufacturer	Model	Stats
SK Hynix	HMT451U6AFR8C-PB	4GB 1Rx8 PC3 12800U-11-12-A1
Samsung	M378B5173DBO-LKO	4GB 1Rx8 PC3 12800U-11-12-A1
Micron	MT16JTF1G64AZ-1G6E1	8GB 2Rx8 PC3 12800U-11-13-B1

DDR4

Manufacturer	Model	Stats
SK Hynix	HMA41GR7MFR4N-TF	8GB 1Rx4 PC4-2133P-RC0-10
Kingston	KTH-PL421/16G	16GB 2Rx4 PC4-2133P-RA0-11

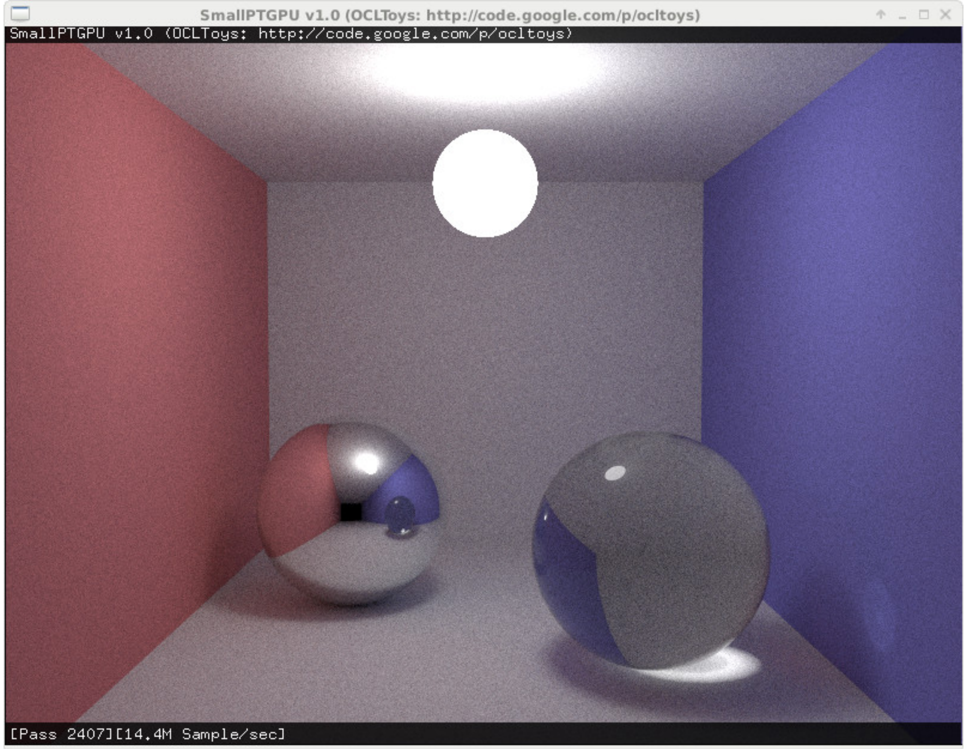
Benchmarks

Things I run a lot

- Idle
- gcc (compiling PAPI)
- HPL – ATLAS, OpenBLAS, IntelMKL

GPU Benchmarks

OCLToys Raytracer. Integrated Intel Beignet OpenCL



```
SmallPTGPU v1.0 (OCLToys: http://code.google.com/p/ocltoys)
```

```
[Pass 2407][14.4M Sample/sec]
```

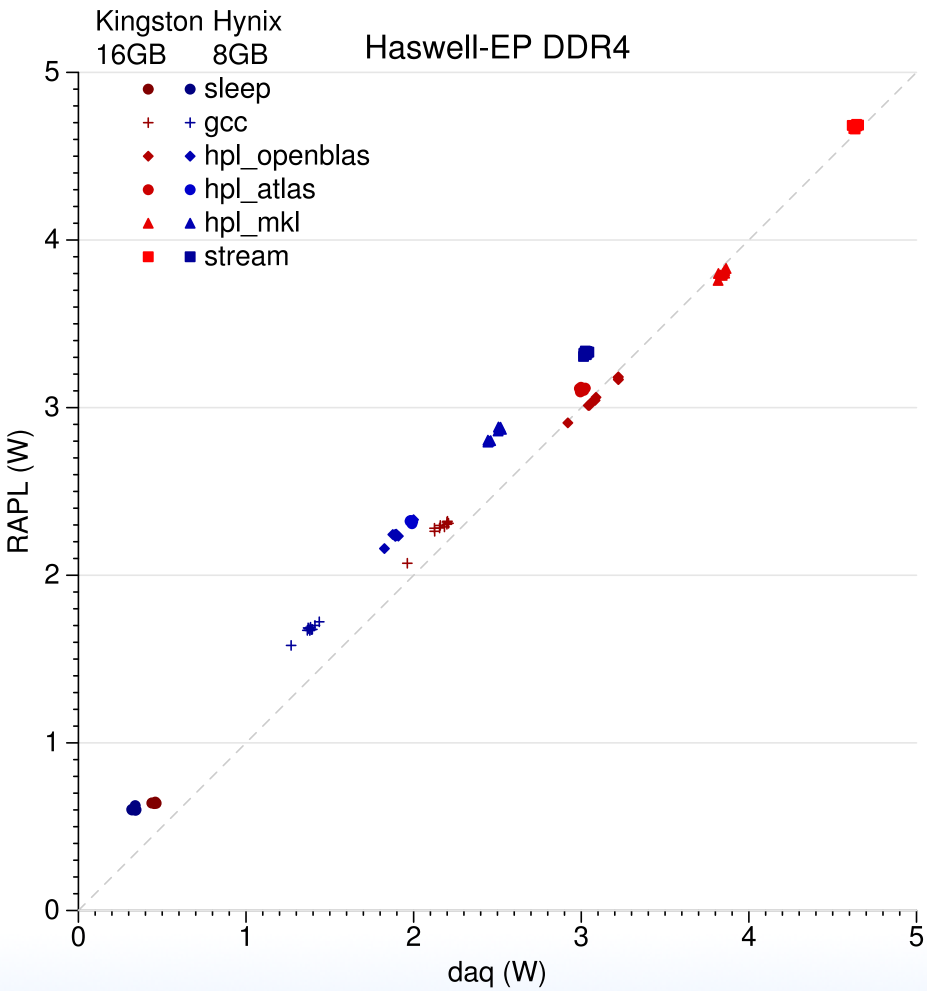
GPU Benchmarks

Kerbal Space Program, OpenGL game

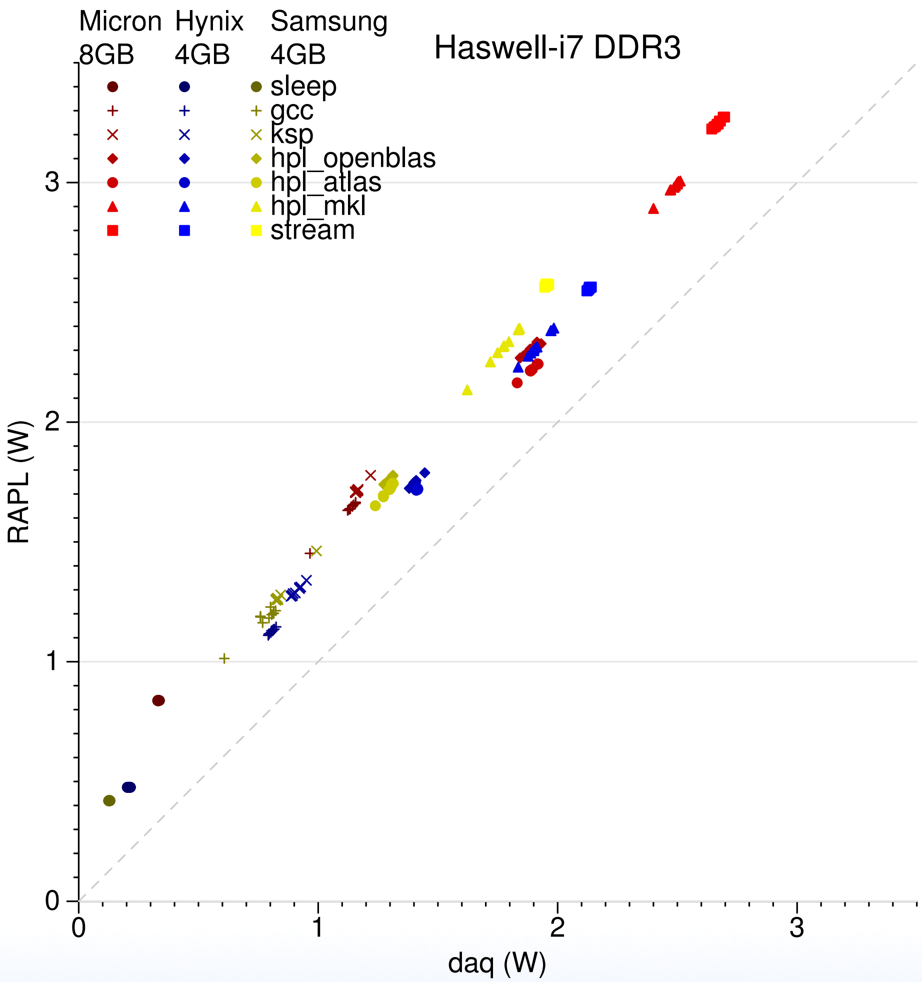
■ ■ ■

T+ 0y, 83d, 00:49:41
MET
048123m
ATMOSPHERE
Overheat!
Overheat!
STAGE
001
Surface
SAS
HDG
Detrix Kerman
Sigbert Kerman

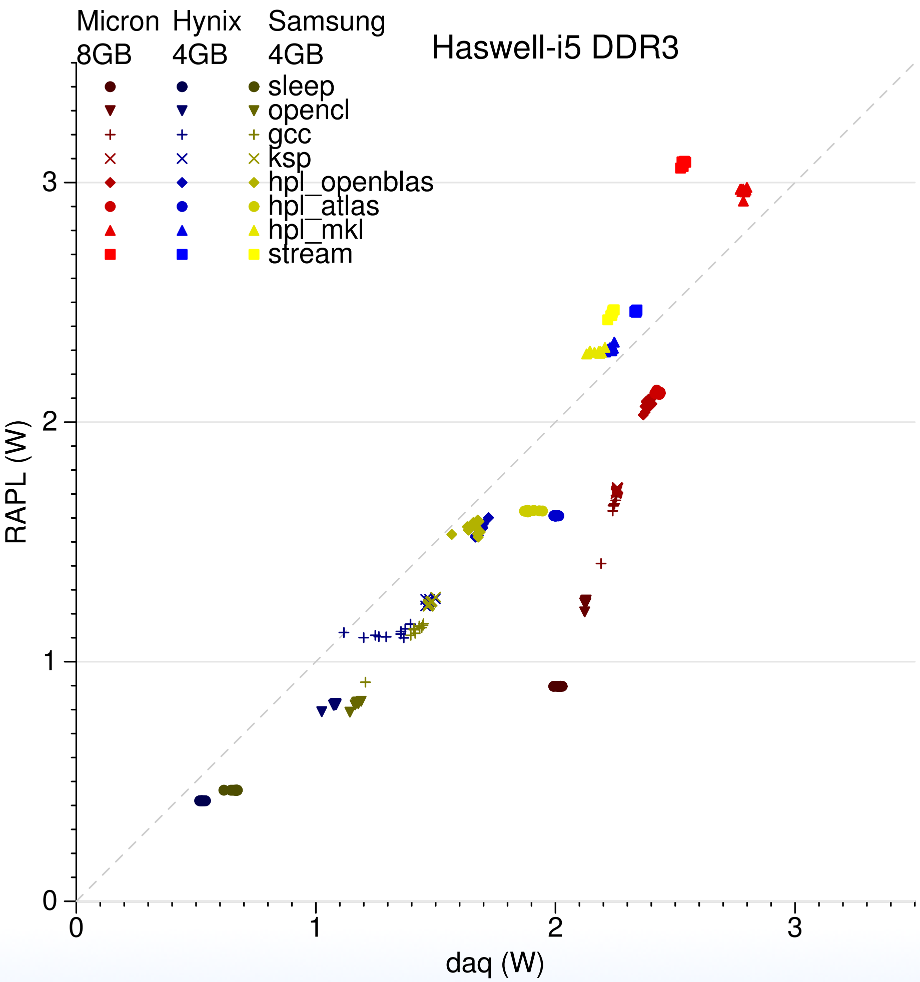
Haswell EP Results



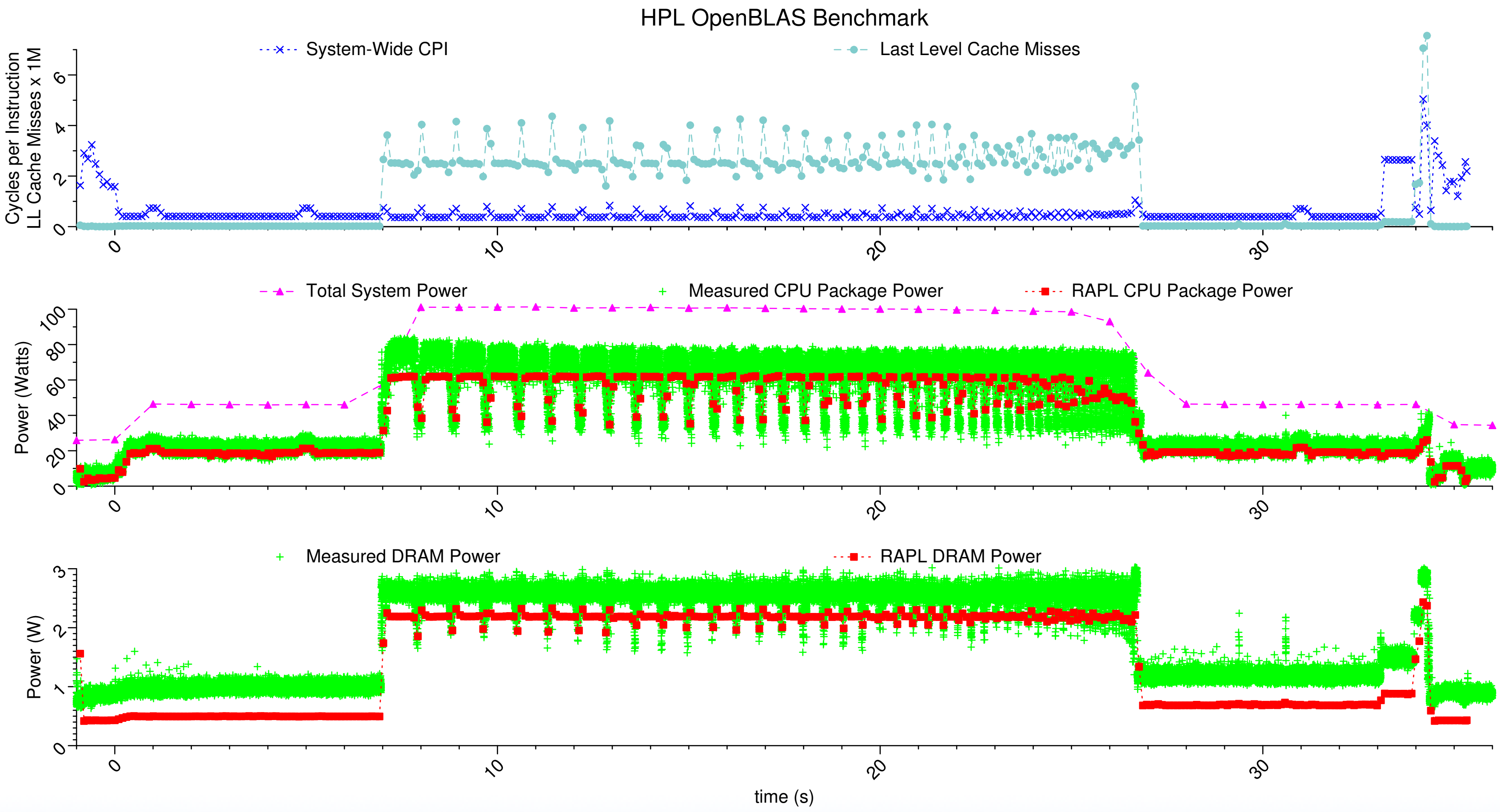
Haswell i7 Results



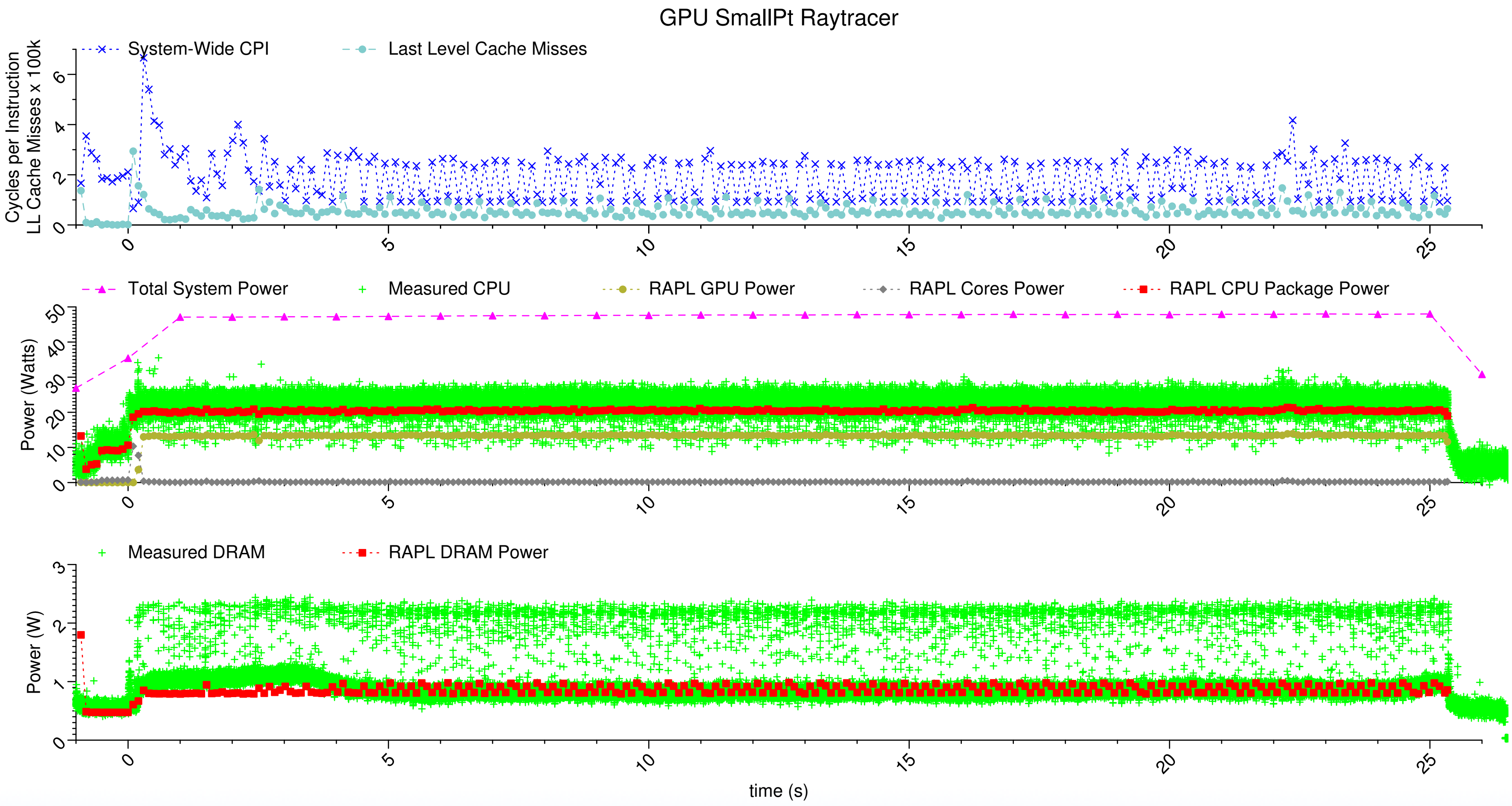
Haswell i5 Results



i5/Hynix HPL OpenBLAS Phase Plots



i5/Hynix OpenCL Phase Plots

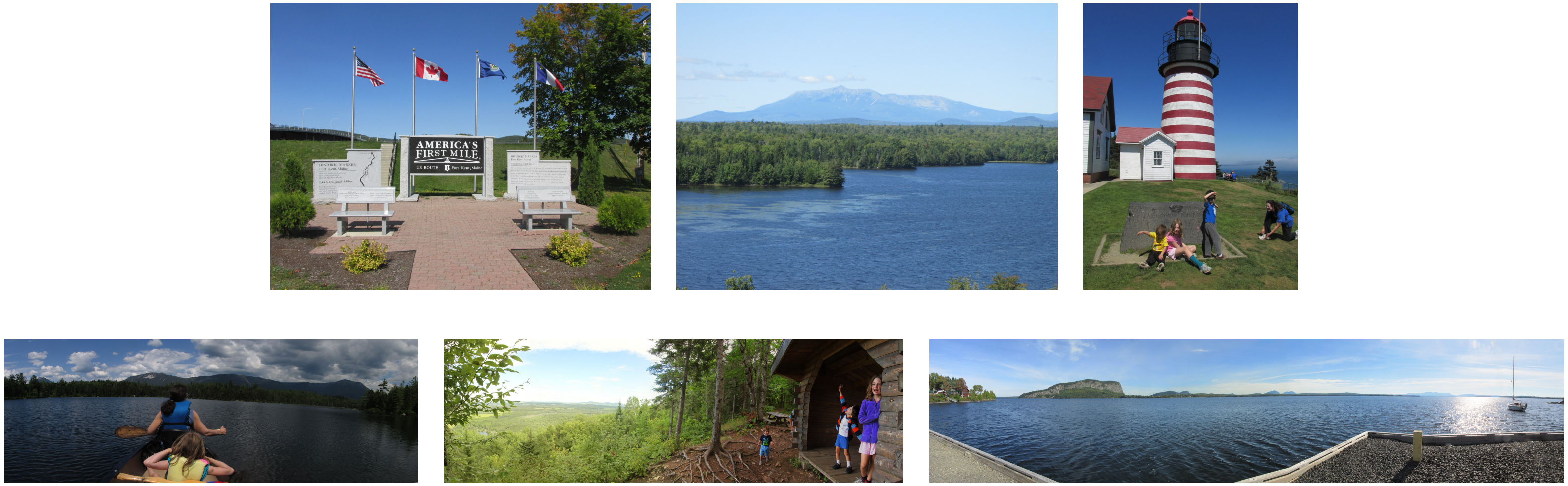


Future Work

- Non-Haswell machines (Broadwell, Skylake)
- Multiple-DIMM sockets
- Dedicated power measurement board with real PCB

Questions?

`vincent.weaver@maine.edu`



Full data can be found here:

`http://web.eece.maine.edu/~vweaver/projects/rapl/rapl_validation.html`