ECE 571 – Advanced Microprocessor-Based Design Lecture 7

Vince Weaver

http://web.eece.maine.edu/~vweaver
vincent.weaver@maine.edu

9 February 2017
Announcements

• HW#4 will be posted, some readings
Measuring Power and Energy
Why?

- New, massive, HPC machines use impressive amounts of power
- When you have 100k+ cores, saving a few Joules per core quickly adds up
- To improve power/energy draw, you need some way of measuring it
Energy/Power Measurement is Already Possible

Three common ways of doing this:

• Hand-instrumenting a system by tapping all power inputs to CPU, memory, disk, etc., and using a data logger
• Using a pass-through power meter that you plug your server into. Often these will log over USB
• Estimating power/energy with a software model based on system behavior
Measuring Power and Energy

- Sense resistor or Hall Effect sensor gives you the current.
- Sense resistor is small resistor. Measure voltage drop. Current $V = IR$ Ohm’s Law, so $V/R = I$.
- Voltage drops are often small (why?) so you made need to amplify with instrumentation amplifier.
- Then you need to measure with A/D converter.
- $P = IV$ and you know the voltage.
- How to get Energy from Power?
Hall Effect Current Sensors

- Output voltage varies based on magnetic field.
- Current in wire causes magnetic field
- Voltage output is linear proportional to current
- Ideally little to no resistance (unlike sense resistor)
- Can measure higher current. 5, 20, 30A
- Need that? 100W CPU at 3.3V is roughly 30A
Existing Related Work

Plasma/dposv results with Virginia Tech’s PowerPack
Powerpack

- Measure at Wall socket: WattsUp, ACPI-enabled power adapter, Data Acquisition System

- Measure all power pins to components (intercept ATX power connector?)

- CPU Power – CPU powered by four 12VDC pins.

- Disk power – measure 12 and 5VDC pins on disk power connector
- Memory Power – DIMMs powered by four 5VDC pins
- Motherboard Power – 3.3V pins. Claim NIC contribution is minimal, checked by varying workload
- System fans
PowerMon 2

• PowerMon 2 is a custom board from RENCI
• Plugs in-line with ATX power supply.
• Reports results over USB
• 8 channels, 1kHz sample rate
• We have hardware; currently not working
Shortcomings of current methods

• Each measurement platform has a different interface
• Typically data can only be recorded off-line, to a separate logging machine, and analysis is done after the fact
• Correlating energy/power with other performance metrics can be difficult
• How often can you measure (a lot happens on a CPU at 2GHz)
Watt’s Up Pro Meter
Watt’s Up Pro Features

- Can measure 18 different values with 1 second resolution (Watts, Volts, Amps, Watt-hours, etc.)
- Values read over USB
- Joules can be derived from power and time
- Can only measure system-wide
Watt’s Up Pro Graph

PLASMA Cholesky Factorization N=10,000 threads=2

Measured on Core2 Laptop
Estimating Power

- Popular thing to do. One example: *Real Time Power Estimation and Thread Scheduling via Performance Counters* by Singh, Bhadauria and McKee.
- Have some sort of hardware measurement setup.
- Then measure lots of easy-to-measure things. Performance counters. Temperature. etc.
- Create a model (machine learning?) that can estimate
- Apparently using as few as 4 counters can give pretty good results
RAPL

- **Running Average Power Limit**
- Part of an infrastructure to allow setting custom per-package hardware enforced power limits
- User Accessible Energy/Power readings are a bonus feature of the interface
How RAPL Works

• RAPL is *not* an analog power meter (usually, Haswell-EP exception)
• RAPL uses a software power model, running on a helper controller on the main chip package
• Energy is estimated using various hardware performance counters, temperature, leakage models and I/O models
• The model is used for CPU throttling and turbo-boost, but the values are also exposed to users via a model-specific register (MSR)
Available RAPL Readings

- **PACKAGE_ENERGY**: total energy used by entire package
- **PP0_ENERGY**: energy used by “power plane 0” which includes all cores and caches
- **PP1_ENERGY**: on original Sandybridge this includes the on-chip Intel GPU
- **DRAM_ENERGY**: on Sandybridge EP this measures DRAM energy usage. It is unclear whether this is just the interface or if it includes all power used by all the DIMMs too
• SoC energy (skylake and newer?)
RAPL Measurement Accuracy

- Intel Documentation indicates Energy readings are updated roughly every millisecond (1kHz)
- Rotem et al. show results match actual hardware

Rotem et al. (IEEE Micro, Mar/Apr 2012)
RAPL Accuracy, Continued

- The hardware also reports minimum measurement quanta. This can vary among processor releases. On our Sandybridge EP machine all Energy measurements are in multiples of 15.2nJ
- Power and Energy can vary between identical packages on a system, even when running identical workloads. It is unclear whether this is due to process variation during manufacturing or else a calibration issue.
RAPL Validation

- The Dresden Paper
- My MEMSYS paper (include some plots?)
RAPL Power Plot

PLASMA Cholesky Factorization N=30,000 threads=16

Measured on SandyBridge EP
RAPL Energy Plot

Cholesky Factorization \( N=30,000 \) threads=16

Measured on SandyBridge EP
**NVML**

- Recent NVIDIA GPUs support reading power via the NVIDIA Management Library (NVML)
- On Fermi C2075 GPUs it has milliwatt resolution within ±5W and is updated at roughly 60Hz
- The power reported is that for the entire board, including GPU and memory
NVML Power Graph

MAGMA LU 10,000, Nvidia Fermi C2075
Recent AMD Family 15h processors also can report “Current Power In Watts” via the Processor Power in the TDP MSR

Support for this can be provided similar to RAPL

Have had bad luck getting accurate readings. Have found various chip errata on fam15h and fam16h hardware
Other ways to measure Power

- IPMI – many server machines have built in (low frequency) measurement of power supply values.

- Thermal? IR camera? Can see how much individual parts of chip use. Overheat? Use IR transparent liquid to cool it?
Using RAPL

• On Linux, at least 3 ways to get these values

• Read msr directly, either with instruction or /dev/msr. Need root as you can do bad things with msrs. “safemsr”

• perf_event

• hwmon/powercap (/sys/class/powercap/)
Listing Events

$ perf list
...
  power/energy-cores/      [Kernel PMU event]
  power/energy-gpu/        [Kernel PMU event]
  power/energy-pkg/        [Kernel PMU event]
  power/energy-ram/        [Kernel PMU event]
  ...
Measuring

$ perf stat -a -e power/energy-cores/,power/energy-ram/,instructions,cycles /opt/ece571/401.bzip2/bzip2 -k -f ./input.source

Performance counter stats for 'system wide':

63.79 Joules power/energy-cores/
2.34 Joules power/energy-ram/
21038123875 instructions # 1.06
19782762541 cycles

3.407427702 seconds time elapsed
Measuring

- The key is -a which enables system-wide mode (needs root too if not configured as such)

- Why do you need system-wide?

- What does that do to the other metrics?
Power and Energy Concerns

Table 1: OpenBLAS HPL $N=10000$ (Matrix Multiply)

<table>
<thead>
<tr>
<th>Machine</th>
<th>Processor</th>
<th>Cores</th>
<th>Freq</th>
<th>Idle Power</th>
<th>Load Power</th>
<th>Time</th>
<th>Total Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Pi 2</td>
<td>Cortex-A7</td>
<td>4</td>
<td>900MHz</td>
<td>1.8W</td>
<td>3.4W</td>
<td>454s</td>
<td>1543J</td>
</tr>
<tr>
<td>Dragonboard</td>
<td>Cortex-A53</td>
<td>4</td>
<td>1.2GHz</td>
<td>2.4W</td>
<td>4.7W</td>
<td>241s</td>
<td>1133J</td>
</tr>
<tr>
<td>Raspberry Pi 3</td>
<td>Cortex-A53</td>
<td>4</td>
<td>1.2GHz</td>
<td>1.8W</td>
<td>4.3W</td>
<td>178s</td>
<td>765J</td>
</tr>
<tr>
<td>Jetson-TX1</td>
<td>Cortex-A57</td>
<td>4</td>
<td>1.9GHz</td>
<td>2.1W</td>
<td>13.4W</td>
<td>47s</td>
<td>629J</td>
</tr>
<tr>
<td>Macbook Air</td>
<td>Broadwell</td>
<td>2</td>
<td>1.6GHz</td>
<td>10.0W</td>
<td>29.1W</td>
<td>14s</td>
<td>407J</td>
</tr>
</tbody>
</table>

1. Which machine has the lowest under-load power draw? Pi 2
2. Which machine consumes the least amount of energy? 
   Broadwell Macbook Air

3. Which machine computes the result fastest? 
   Broadwell Macbook Air

4. Consider a use case with an embedded board taking a picture once every 60 seconds and then performing a matrix-multiply similar to the one in the benchmark (perhaps for image-recognition purposes). Could all of the boards listed meet this deadline? 
   No, only the Jetson and Macbook Air can meet the
5. Assume a workload where a device takes a picture once a minute then does a large matrix multiply (as seen in Table 1). The device is idle when not multiplying, but under full load when it is.

(a) Over a minute, what is the total energy usage of the Jetson TX-1?

Each Minute = (13s Idle * 2.1W) + (47s Load * 13.4W)
= 657J

(b) Over a minute, what is the total energy usage of the Macbook Air?
Each Minute = (46s * 10W) + (14*29.1) = 867J
Pandaboard Power Stats

- Wattsuppro: 2.7W idle, seen up to 5W when busy


- **With Neon and CPU burn:**

<table>
<thead>
<tr>
<th>Idle system</th>
<th>550 mA</th>
<th>2.75W</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpuburn-neon</td>
<td>1130 mA</td>
<td>5.65W</td>
</tr>
<tr>
<td>cpuburn-1.4a (burnCortexA9.s)</td>
<td>1180 mA</td>
<td>5.90W</td>
</tr>
<tr>
<td>ssvb-cpuburn-a9.S</td>
<td>1640 mA</td>
<td>8.2W</td>
</tr>
</tbody>
</table>
Easy ways to reduce Power Usage
DVFS

• Voltage planes – on CMP might share voltage planes so have to scale multiple processors at a time
• DC to DC converter, programmable.
• Phase-Locked Loops. Orders of ms to change. Multiplier of some crystal frequency.
• Senger et al ISCAS 2006 lists some alternatives. Two phase locked loops? High frequency loop and have programmable divider?
• Often takes time, on order of milliseconds, to switch
frequency. Switching voltage can be done with less hassle.
When can we scale CPU down?

- System idle
- System memory or I/O bound
- Poor multi-threaded code (spinning in spin locks)
- Thermal emergency
- User preference (want fans to run less)