



Power Monitoring At The NCSA Innovative Systems Lab

Energy-Efficient HPC Working
Group Webinar

February 8, 2011

Craig Steffen

csteffen@ncsa.uiuc.edu

217-979-2392

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign



NCSA ISL and Others

- Jim Philips and John Stone: Theoretical and Computational Biophysics Group, Beckman Institute, UIUC
- Kenneth Esler: NCSA and UIUC Physics
- Joshi Fullop: NCSA Systems Monitoring
- Jeremy Enos, Volodymyr Kindratenko, Craig Steffen, Guochun Shi, Mike Showerman: NCSA Innovative Systems Laboratory
- Wen-mei Hwu and William Gropp: UIUC ECE Department

Overview

- AC GPU computing cluster
- Power monitoring
 - Search for power monitors
 - Roll our own--version 1: Tweet-A-Watt
 - Roll our own--version 2: Arduino-based power monitor
- Power monitoring on real applications
- EcoG Cluster
- EcoG Top500 and Green500 submissions

AC cluster (Accelerator Cluster)

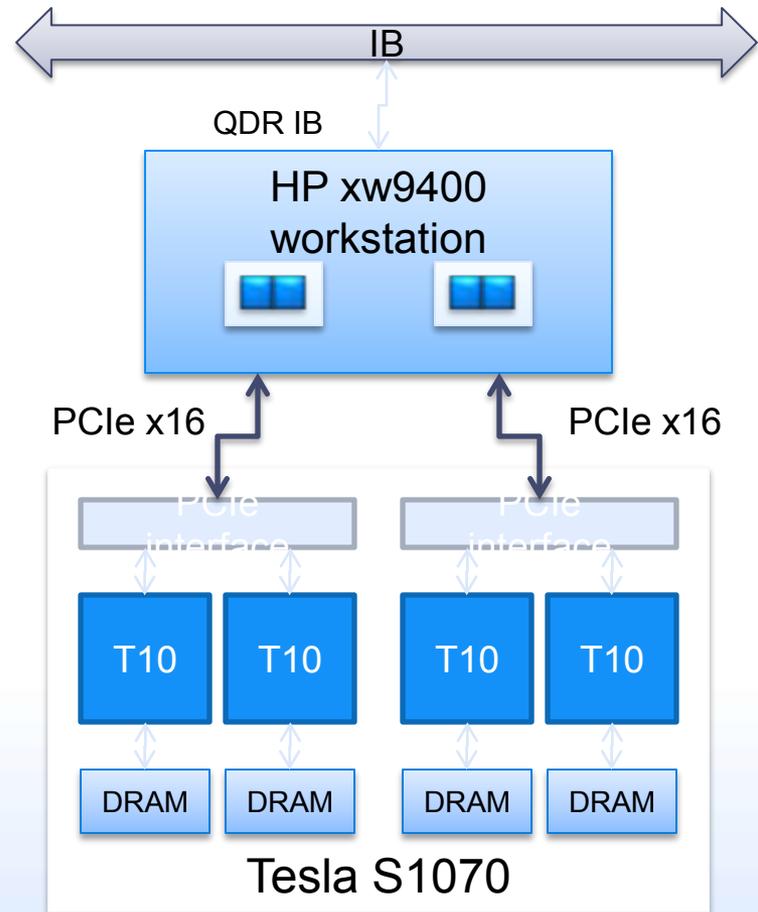
- Originally “QP” cluster for “Quadro Plex”
- 32 HP XW9400 nodes. Each node:
 - 2 dual-core 2.4 GHz Opteron 2216
 - 8 GB RAM per node
 - **NVIDIA Tesla S1070 each:**
 - **4 Tesla C1060 GPUs (128 total in cluster)**
- Interconnect network is QDR Infiniband
- CUDA 3.1 compiler/build stack
- Job control/scheduler Moab
 - **Specific resource management for jobs via Torque**
- QP first commissioned November 2007
- AC on-line since December 2008

AC Cluster



AC01-32 nodes

- HP xw9400 workstation
 - 2216 AMD Opteron 2.4 GHz dual socket dual core
 - 8GB DDR2 in ac04-ac32
 - 16GB DDR2 in ac01-03, “bigmem” on qsub line
 - PCI-E 1.0
 - Infiniband QDR
- Tesla S1070 1U GPU Computing Server
 - 1.3 GHz Tesla T10 processors
 - 4x4 GB GDDR3 SDRAM
 - 1 per host



AC cluster used for

- Virtual school for Science and Engineering (attached to the Great Lakes Consortium for Petascale Computing)
NVIDIA/CUDA August 2008,2009,2010
- Other classes in 2010:
 - “Intro to CUDA” Volodymyr Kindratenko, Singapore June 13-19
 - Barcelona Spain, Wen-Mei Hwu July 5-9
 - Thomas Scavo July 13-23
 - “Proven Algorithmic Techniques for Many-core Processors”
Thomas Scavo August 2-6
 - John Stone August 7-8

AC GPU Cluster Power Measurements

State	Host Peak (Watt)	Tesla Peak (Watt)	Host power factor (pf)	Tesla power factor (pf)
power off	4	10	.19	.31
start-up	310	182		
pre-GPU use idle	173	178	.98	.96
after NVIDIA driver module unload/reload ⁽¹⁾	173	178	.98	.96
after deviceQuery ⁽²⁾ (idle)	173	365	.99	.99
GPU memtest #10 (stress)	269	745	.99	.99
after memtest kill (idle)	172	367	.99	.99
after NVIDIA module unload/reload ⁽³⁾ (idle)	172	367	.99	.99
VMD Madd	268	598	.99	.99
NAMD GPU STMV	321	521	.97-1.0	.85-1.0 ⁽⁴⁾
NAMD CPU only ApoA1	322	365	.99	.99
NAMD CPU only STMV	324	365	.99	.99

1. Kernel module unload/reload does not increase Tesla power
2. Any access to Tesla (e.g., deviceQuery) results in doubling power consumption after the application exits
3. Note that second kernel module unload/reload cycle does not return Tesla power to normal, only a complete reboot can
4. Power factor stays near one except while load transitions. Range varies with consumption swings

Search for Power Monitors:

What questions do we want to answer?

- How much power do jobs use?
- How much do they use for pure CPU jobs vs. GPU-accelerated jobs?
- Do GPUs deliver a hoped-for improvement in power efficiency?

Hardware: Criteria for data-sampling device

- Cheap
- Easy to buy/produce
- Allows access to real data (database or USB, no CD-installed GUIs)
- Monitors 208V 16A power feed
- Scalable solution across machine room (one node can collect one-node's data)

Search for Good (and Cheap) Hardware Power Monitoring

- Laboratory units too expensive
- Commercial Units:
 - 1A granularity?
 - No direct data logging
 - No real-time data logging

Very capable

- **PS3000 PowerSight Power Analyzer**
\$ 2495.00

Capable; Closer but still too expensive

- [ElitePro™ Recording Poly-Phase Power Meter](#) Standard Version consists of:
 - US/No. America 110V 60 Hz Transformer
 - 128Kb Capacity
 - Serial Port Communications
 - Indoor Use with Crocodile Clips
 - Communications Package (Software) and Current Transformers sold separately.
- [More Information](#)
Price: \$965.00 Part Number: EP

Instrumented PDUs: poor power granularity

- 1A granularity
- 120V circuits



Watts-up integrated power monitor: CLOSE

- Smart Circuit 20 31298 \$194.95
- Outputs data to web page (how to efficiently harvest this data?)

Data Center Power—208 V, 20 or 30A



Power Monitoring Version 1: Tweet-a-Watt Receiver and Transmitter



<http://www.ladyada.net/make/tweetawatt/>
Kits available from www.adafruit.com

Tweet-a-Watt

- Kill-a-watt power meter
- Xbee wireless transmitter
- power, voltage, shunt sensing tapped from op amp
- Lower transmit rate to smooth power through large capacitor
- Readout software modified from available Python scripts to upload sample answers to local database
- We built 3 transmitter units and one Xbee receiver
- **Currently integrated into AC cluster as power monitor**



Evaluation of Tweet-a-Watt

- Limited to Kill-a-Watt capability (**120V, 15A circuit**)
- Low sampling rate (report every 2 seconds, readout every 30 seconds)
- Either TWO XBEE units required or scaling issue
- Fixed but configurable program; one set, difficult to program (low sampling rate means unit is off most of the time)

- Correlated voltage and current (read power factor and true power usage)
- 50-foot plus range (through two interior walls)
- Currently tied to software infrastructure: **Application power studies done with Tweet-a-Watt**

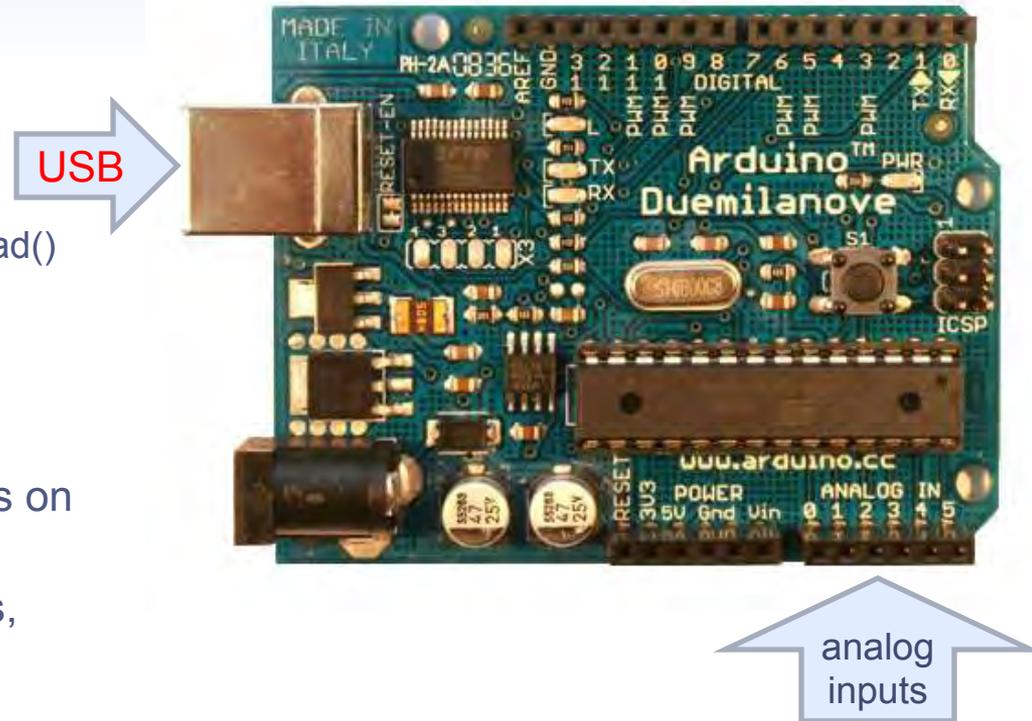
Power Monitor version 2:

One-off function Prototype Power Monitor

- Used chassis from existing (120 V) PDU for interior space
- Connectors, breaker, and wiring to carry 208V 16A power distribution
- Current sense transformers and Arduino microcontroller for current monitoring
- Prototyped (but not deployed) Python script to insert output into power monitor database

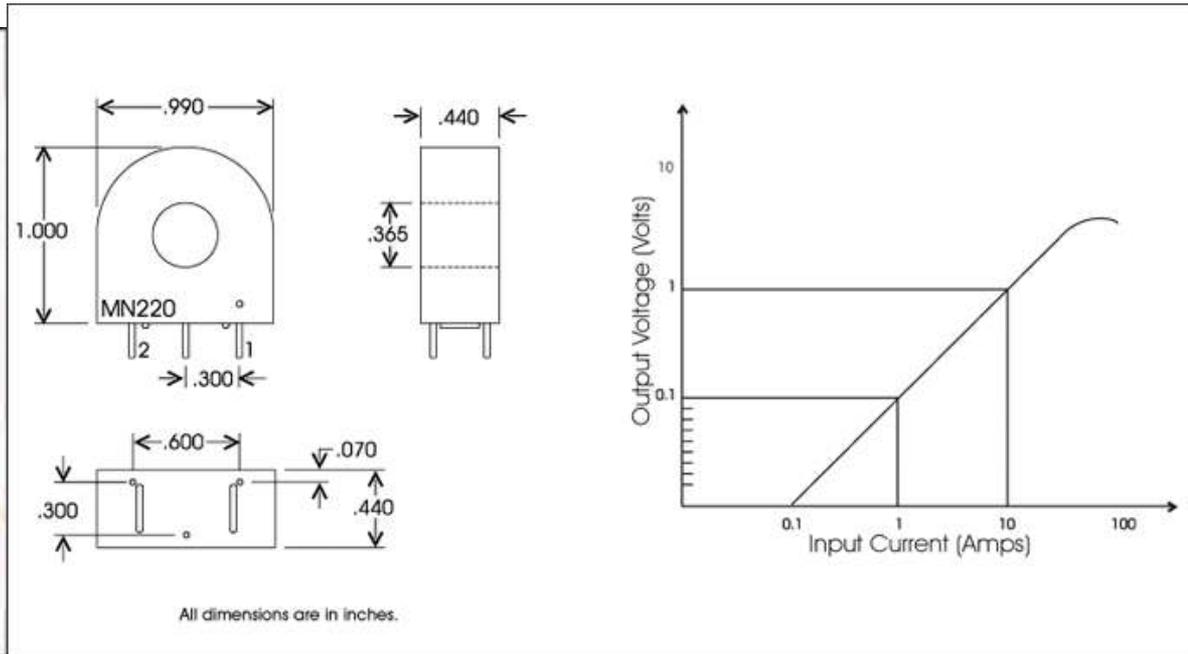
Arduino-based Power Monitor

- Based on Arduino Duemilanove
 - Runs at 16 MHz
 - has 6 **analog voltage-to-digital converters** (sampled explicitly by read() function)
 - Runs microcode when powered on (from non-volatile memory)
- Accumulates sample arrays for N samples per channel per report (N is on subsequent slides)
- Accumulates current measurements, computes RMS values, and outputs results in ASCII on USB connection
- Arduino is powered from the USB connection



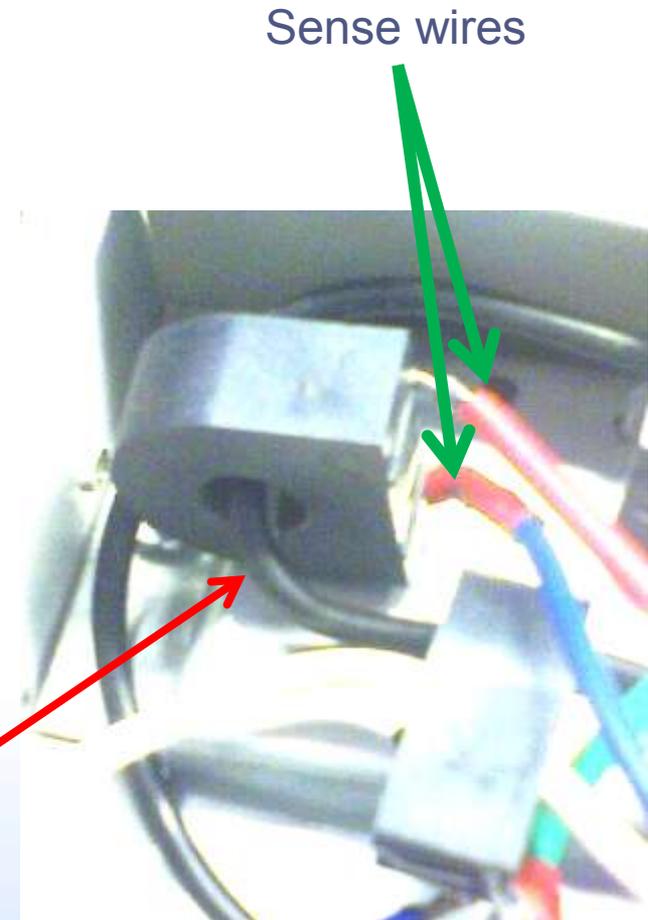
MN 220 picking transformer from Manutech

- Manutech.us
- 1000 to 1 voltage transformer; 1 to 1000 current transformer
- Suggested burden resistor: 100 Ohms.
- AC output voltage proportional to AC current input.
- Output at 100 Ohms: 100 mV/Amp.
- Various ranges of output are achievable by using different burden resistors.



Current Sense Transformer

- MN-220 current “transformer” designed for 1 to 20 amp primary
 - 1000-1 step-up current transformer
- Burden resistor sets the sensitivity; sets “volts per count” calibration constant
- Allows current monitoring without Arduino contact with high-voltage wires



AC Current carrying wire

Industrial Design

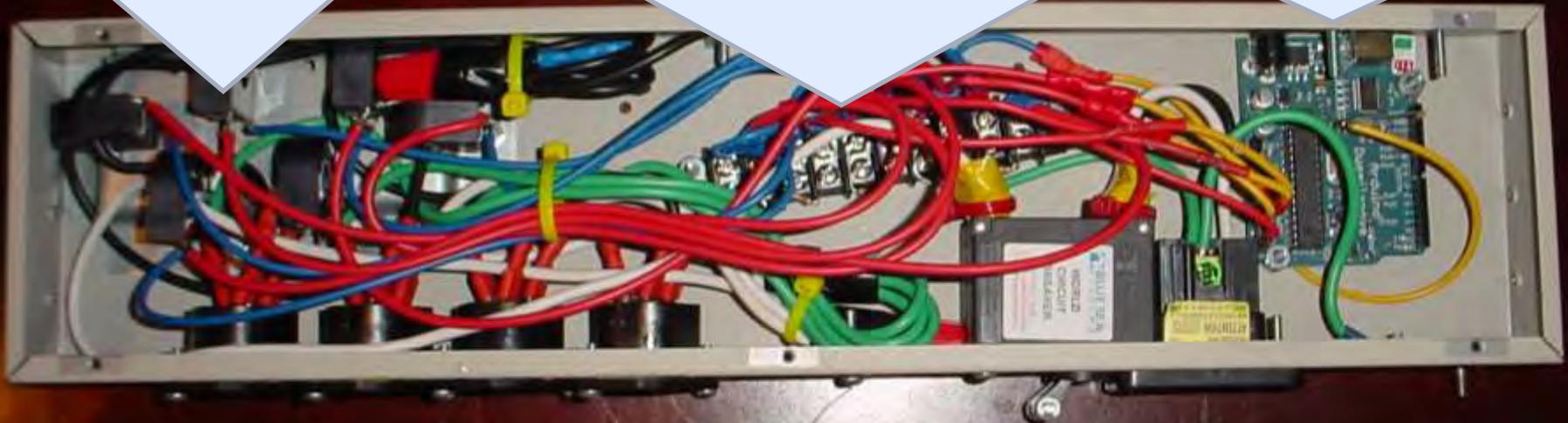
- 5 separate sense transformers for 4 power legs and opposite leg of input
- Current sense ONLY; Arduino is completely isolated from power conductors. No phase or power factor information, RMS current *only*



Current sense
transformers

Interchangeable
burden resistors

Arduino



Arduino development environment

- C-like language environment
 - #defines for calibration constants
 - Initial setup() function runs once
 - loop() function repeats forever

SPECIAL WARNING: Arduino INTs are 16 bits! Summing the squares of measured voltages (in the 200 to 400 range) will OVERFLOW the accumulator INT. (Convert to float before squaring)

```
File Edit Sketch Tools Help
work_voltmeter_doublefloat
#define AMPSPERCT0 (14.796)
#define AMPSPERCT1 (9.574)
#define AMPSPERCT2 (9.574)
#define AMPSPERCT3 (14.796)
#define AMPSPERCT4 (48.828)

// correction factors
#define CORREC0 (1.0)
#define CORREC1 (1.249)
#define CORREC2 (1.193)
#define CORREC3 (1.043)
#define CORREC4 (0.967)

void setup(){
  analogReference(DEFAULT);
  pinMode(0, INPUT);
  Serial.begin(9600);
}

void loop() {
  float accum0 = 0.0, accum1 = 0.0, accum2 = 0.0, accum3 = 0.0, accum4 = 0.0, accum5 = 0.0;
  float total0=0.0, total4=0.0;
  float rmsCts = 0.0;
  int N=0;
  int relval;
}
```

Output Format (our implementation)

- Every sampling period outputs block of ASCII text to virtual console (accessed under Linux typically at `/dev/ttyUSB0`)
- No protocol or readers necessary; software can be checked with commands *tail* or *more*
- If ANY sample on a channel is within 10% of the hard limit, then the channel is flagged as “overflow” in the output stream
(note the `\r \n` double-line breaks)

```
File Edit View Terminal Help
```

```
(4)[ ]= 1335.24
```

```
analogzero=524.68 514.80
```

```
(0)[ ]= 1366.71
```

```
(1)[ ]= 7.87
```

```
(2)[ ]= 8.34
```

```
(3)[ ]= 13.22
```

```
(4)[ ]= 1329.58
```

```
analogzero=501.67 507.42
```

```
(0)[ ]= 1318.34
```

```
(1)[ ]= 8.02
```

```
(2)[ ]= 8.97
```

```
(3)[ ]= 9.76
```

```
(4)[ ]= 1315.29
```

```
analogzero=496.03 506.72
```

```
(0)[ ]= 1346.84
```

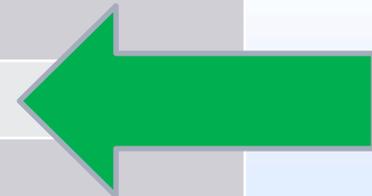
```
(1)[ ]= 8.46
```

```
(2)[ ]= 6.59
```

Calibration, Uncertainty and Readout Speed

- Arduino only does RMS summing; not synchronized with AC clock. Possible sampling errors from undersampling AC waveform (hopefully eliminated by enough samples)
- Samples-per-report is set high enough to minimize undersampling errors
- Uncertainty measured with idle node (upper uncertainty limit only)

Measurements per report	Time between reports (s)	Uncertainty (mA)
250	.28	±7
125	.2	±8
60	.15	±35



Industrial design continued

- Interchangeable burden resistors to match pickup transformer output voltage to Arduino voltage sense
- Initially configured with two 600W channels, two 1000W channels, and main leg monitor is about 3300W for 16A at 208V
- Conclusion: **no advantage** to careful matching of burden resistors.
Uncertainty of 3300W channel vs. 600W:
 - 250 samples: 6 vs 7mA
 - 125 samples: 8 vs 8
 - 60 samples: 37 vs 35
- Advantage: eliminates a LOT of wiring from the prototype

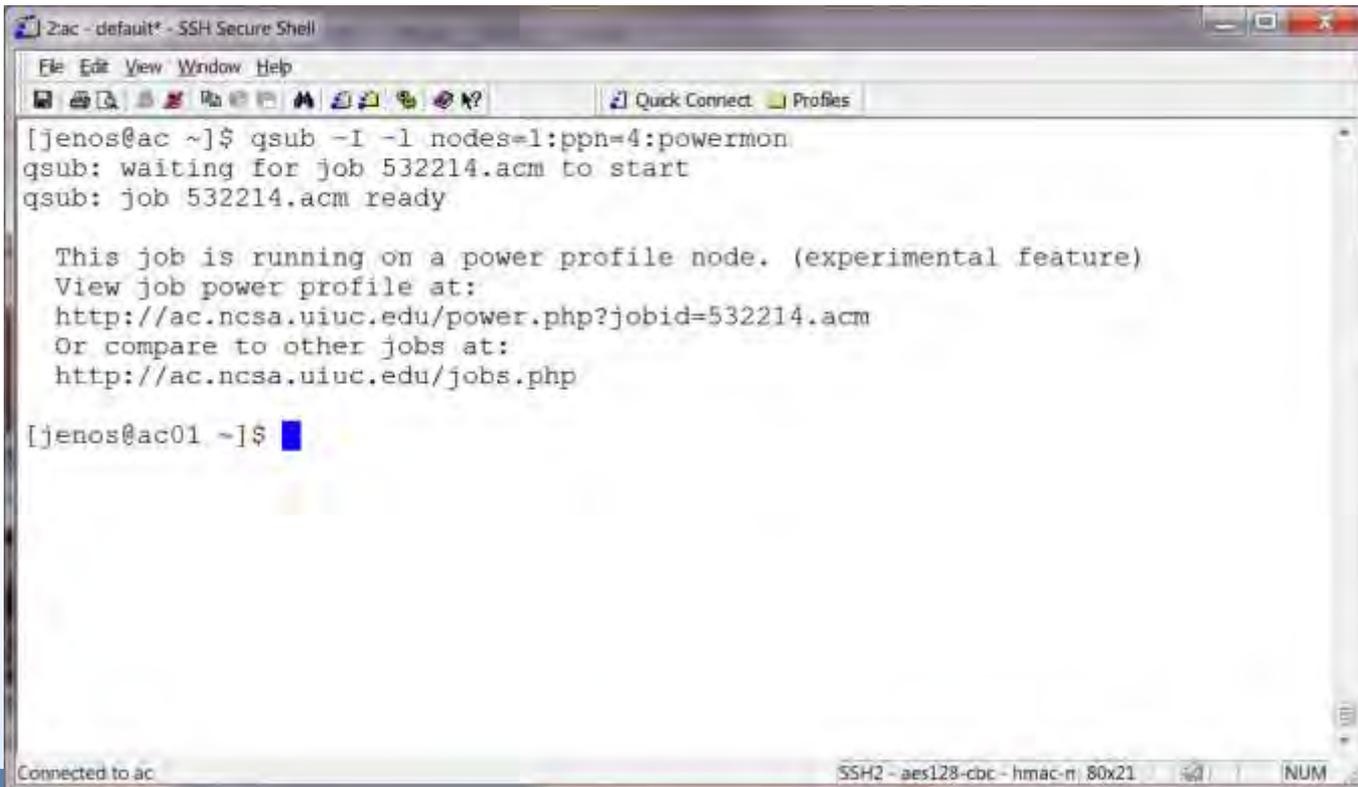


Data storage and calibration database

- Prolog scripts identify the (one) power monitored node (via Torque)
- Job history entry tags job to be attached to time window of power monitor data
- The job scripts create an automagic link to graphed output data per-sample and total usage summary

Power monitor data presentation

- <http://ac.ncsa.uiuc.edu/docs/power.readme>
- submit job with prescribed Torque resource (powermon)
- Run application as usual, follow link(s)



```
Zac - default* - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
[jenos@ac ~]$ qsub -I -l nodes=1:ppn=4:powermon
qsub: waiting for job 532214.acm to start
qsub: job 532214.acm ready

This job is running on a power profile node. (experimental feature)
View job power profile at:
http://ac.ncsa.uiuc.edu/power.php?jobid=532214.acm
Or compare to other jobs at:
http://ac.ncsa.uiuc.edu/jobs.php

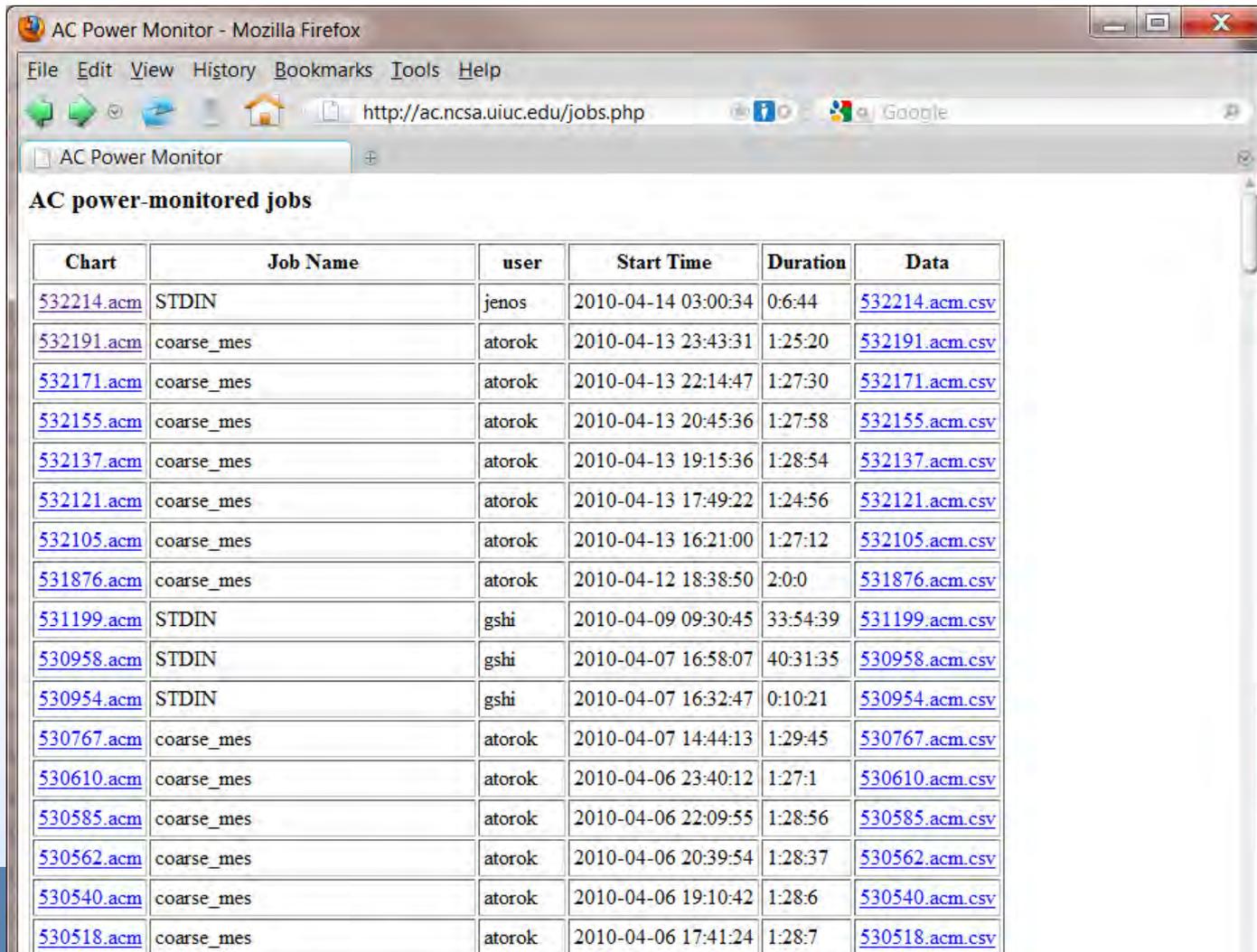
[jenos@ac01 ~]$
```

Connected to ac

SSH2 - aes128-cbc - hmac-sha1 - 80x21

NUM

Each monitored job shows up as a link at <http://ac.ncsa.uiuc.edu/jobs.php>



AC Power Monitor - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://ac.ncsa.uiuc.edu/jobs.php

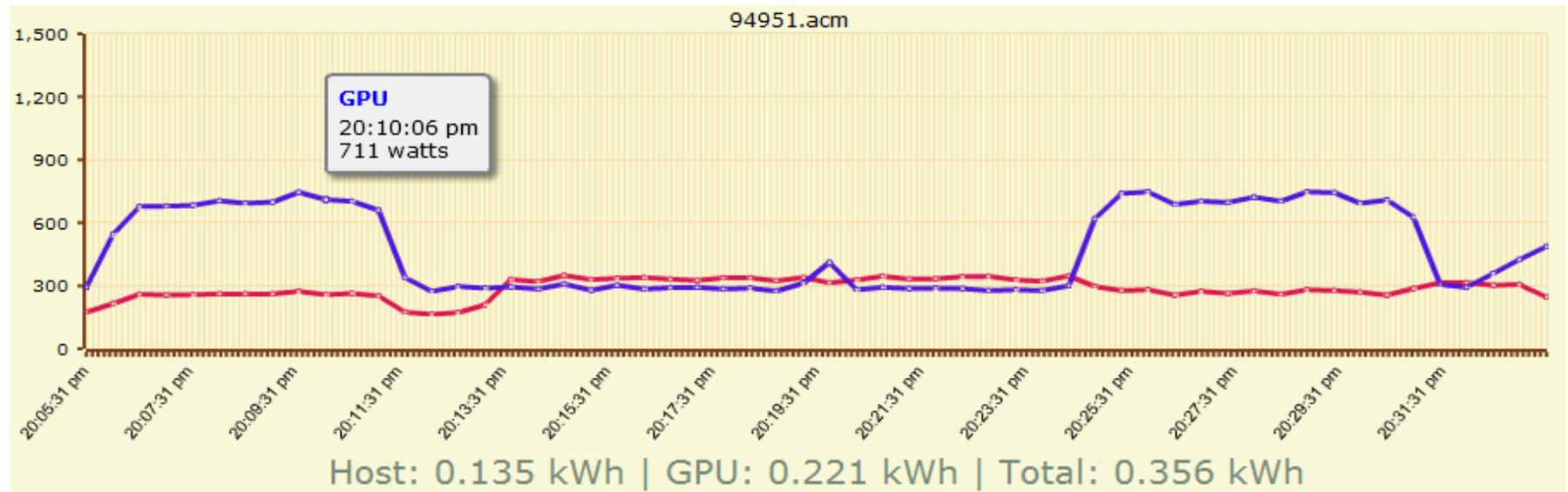
AC Power Monitor

AC power-monitored jobs

Chart	Job Name	user	Start Time	Duration	Data
532214.acm	STDIN	jenos	2010-04-14 03:00:34	0:6:44	532214.acm.csv
532191.acm	coarse_mes	atorok	2010-04-13 23:43:31	1:25:20	532191.acm.csv
532171.acm	coarse_mes	atorok	2010-04-13 22:14:47	1:27:30	532171.acm.csv
532155.acm	coarse_mes	atorok	2010-04-13 20:45:36	1:27:58	532155.acm.csv
532137.acm	coarse_mes	atorok	2010-04-13 19:15:36	1:28:54	532137.acm.csv
532121.acm	coarse_mes	atorok	2010-04-13 17:49:22	1:24:56	532121.acm.csv
532105.acm	coarse_mes	atorok	2010-04-13 16:21:00	1:27:12	532105.acm.csv
531876.acm	coarse_mes	atorok	2010-04-12 18:38:50	2:0:0	531876.acm.csv
531199.acm	STDIN	gshi	2010-04-09 09:30:45	33:54:39	531199.acm.csv
530958.acm	STDIN	gshi	2010-04-07 16:58:07	40:31:35	530958.acm.csv
530954.acm	STDIN	gshi	2010-04-07 16:32:47	0:10:21	530954.acm.csv
530767.acm	coarse_mes	atorok	2010-04-07 14:44:13	1:29:45	530767.acm.csv
530610.acm	coarse_mes	atorok	2010-04-06 23:40:12	1:27:1	530610.acm.csv
530585.acm	coarse_mes	atorok	2010-04-06 22:09:55	1:28:56	530585.acm.csv
530562.acm	coarse_mes	atorok	2010-04-06 20:39:54	1:28:37	530562.acm.csv
530540.acm	coarse_mes	atorok	2010-04-06 19:10:42	1:28:6	530540.acm.csv
530518.acm	coarse_mes	atorok	2010-04-06 17:41:24	1:28:7	530518.acm.csv

Power Profiling – Walk through

AC Power Utilization



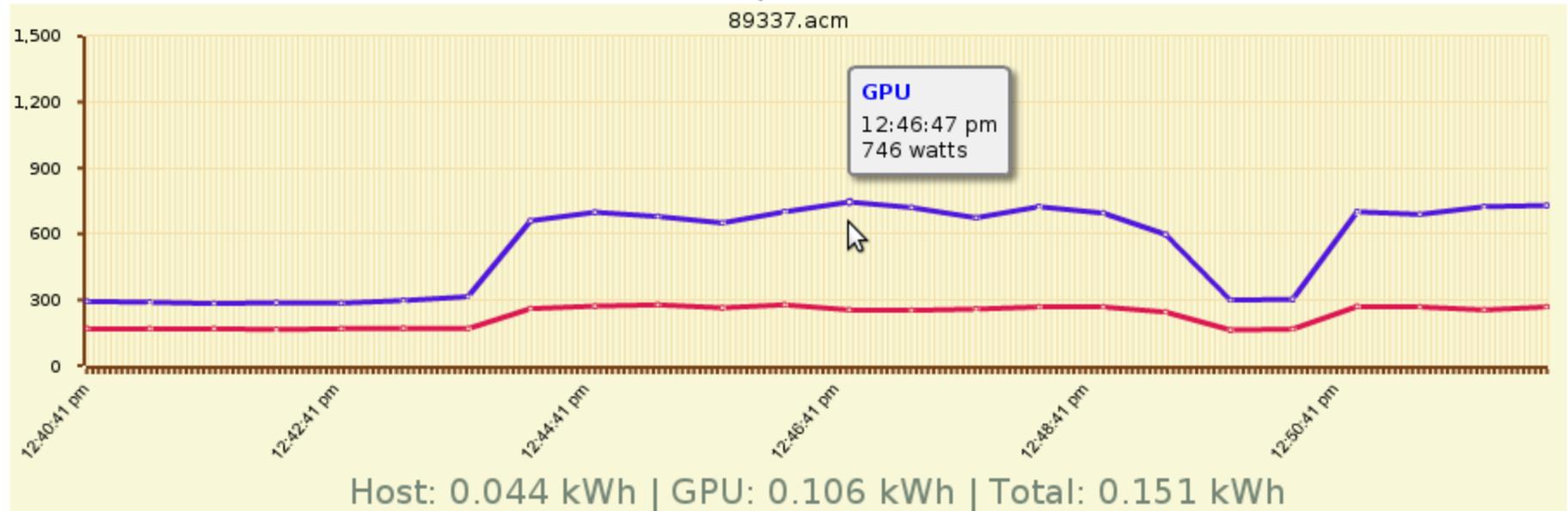
[JSON Data](#)

- Mouse-over value displays
- Under curve totals displayed
- If there is user interest, we may support calls to add custom tags from application

Output Graphs



AC Power Utilization



Unique Features of this Hardware+Software Setup

- Hardware solution
 - Cheap
 - Scalable
- Presentation integrated with job software
- Simple to use with jobs.php link
- Not required; can be ignored by other users

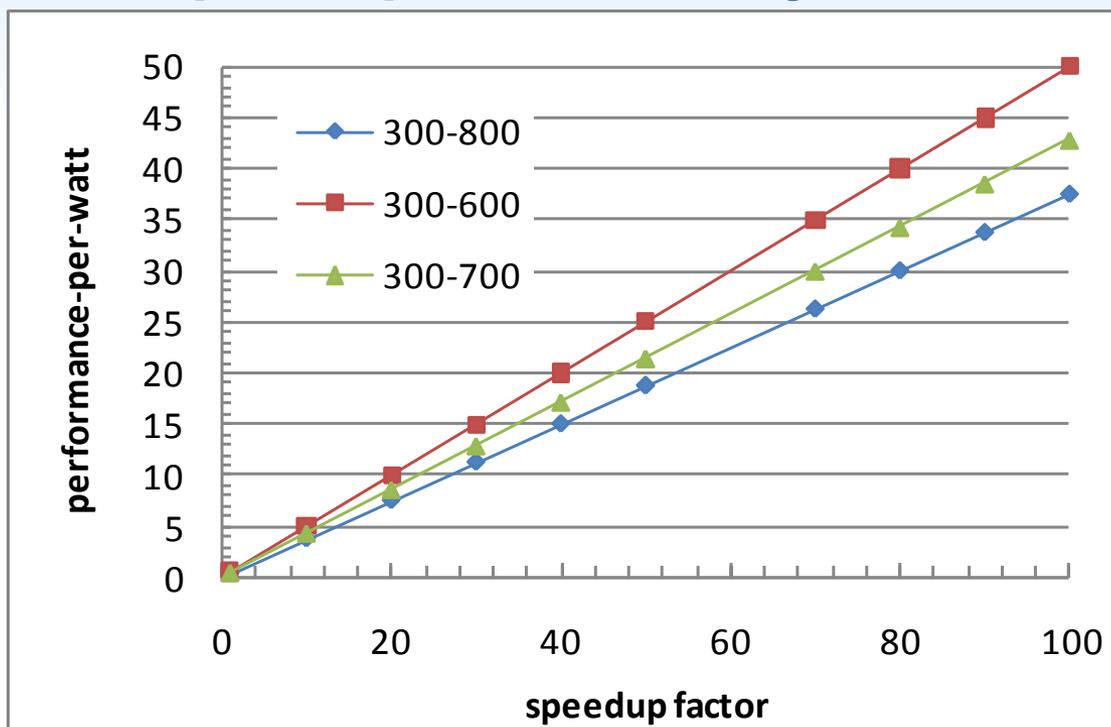
Real Application Speed and Efficiency

- Speedup measured in terms of **wall clock time** for whole application to run
- Power consumption measurements made over at least 20 sample runs
- Removed power measurements from startup and shutdown phases of applications

NOTE: The NVIDIA cards have internal power measuring. We didn't use them because

- That leaves out the power supply of the Tesla
- We got inconsistent node-to-node results
- We wanted to understand the systematics of the data

Current State: Speedup to Efficiency Correlation



- The GPU consumes roughly double the CPU power, so a 3x GPU is required to break even
- Performance-per-watt is asymptotically roughly half speedup factor or less

Real Applications Speedup Summary

- NAMD: raw speedup: 6 speedup-per-watt: 2.8
- VMD: raw: 26x XperW: 10.5
- QMCPack: raw: 62 XperW: 23
- MILC: raw: 20 XperW: 8

SAAHPC 2011

- **Symposium on Application Accelerators in High Performance Computing 2011**
- Covers all accelerators including GPUs, FPGAs, Cell
- Co-hosted by **NCSA, University of Illinois** and **University of Tennessee**, Knoxville
- 2011 dates and location not announced (June or July)
- Submissions due in April/May 2011

Current news can be found at: saahpc.org

EcoG: Tesla 2050-based Cluster

- 128 Tesla 2050 GPU cards donated by NVIDIA
- Significant parts of infiniband fabric donated by QLogic
- Ethernet cables, power cables, PDUs, recycled from retired NCSA “Mercury” and “Tungsten” systems
- EcoG cluster sits on food service shelves and occupies 18 square feet

System Assembled and Installed by Students

~13 students from UIUC ECE/CS departments in cluster-building independent study

2 graduate students from the chemistry department

Mike Showerman, Jeremy Enos, Luke Scharf, and Craig Steffen from ISL

Sean Treichler from NVIDIA

EcoG Design Goals

- Experiment with low-power, high performance GPU-based architecture
- Maps to GPU math capabilities
- Frequent but not constant node-to-node updates
- Likely target apps:
 - Molecular dynamics
 - Fluid dynamics
 - HPL works passably well
- High-performance GPUs, lower power CPUs
- RAM (which also consumes power) just bigger than GPU
- NFS root file system (no hard drive on nodes)

EcoG Final Configuration

- Tesla 2050 GPUs primary computing element; single modest CPU per node
- Single-socket motherboard
- Each node:
 - Intel® Core i3 2.93 GHz CPU
 - 4 GB RAM main memory
 - 1 two-port QDR infiniband card

HPL Function Division

- Intel CPU:
 - main application loop
 - panel factorization
 - DTRSM update
 - final triangular solve
 - residual check
- Tesla GPU:
 - Update DGEMM
 - Rowswap scatter/gather

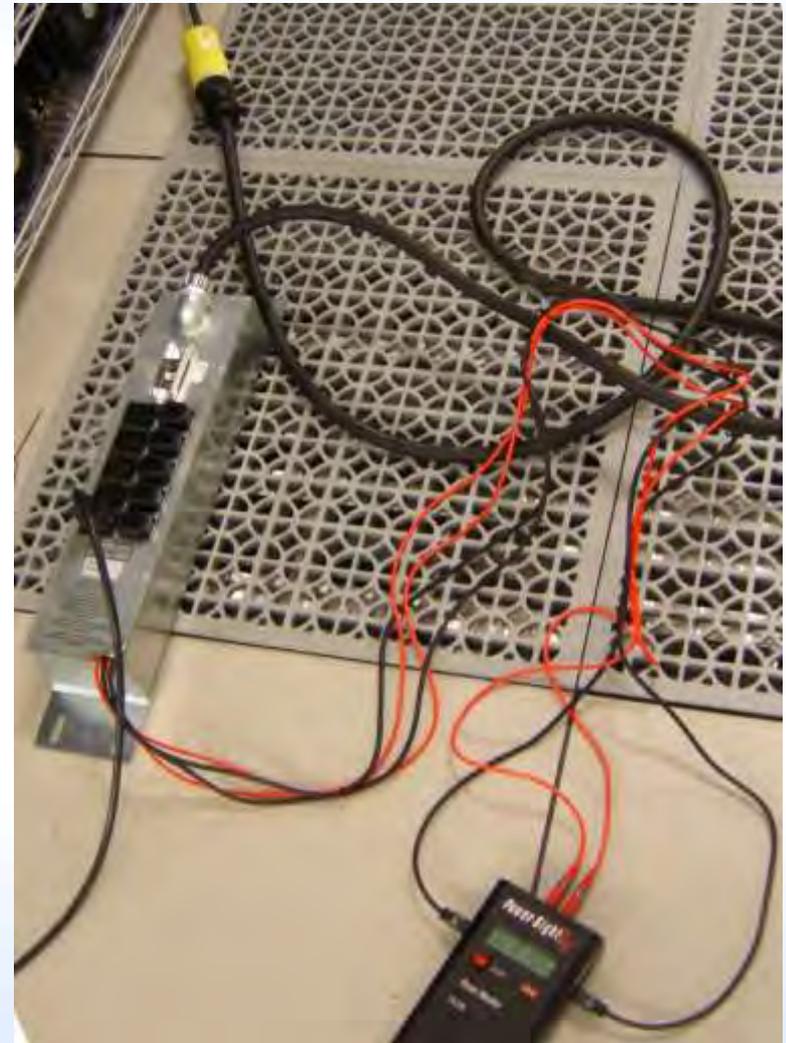
Power Monitoring Setup: Voltage and Current Probes

- Re-used rack-mounted PDU
- 2 voltage probes for 208V power legs
- 2 clamp-on current probes for current measurement
- Probes secured INSIDE enclosure



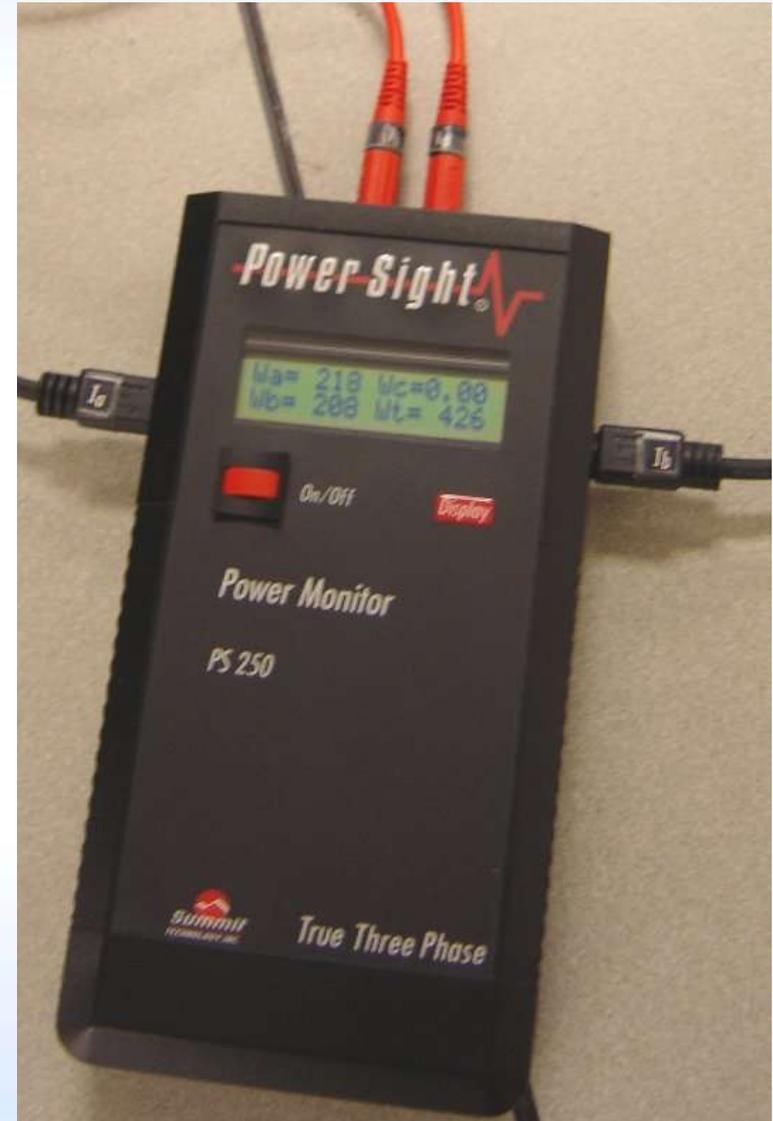
Final Power Monitoring Setup: Enclosed for Convenience and Safety

- L6-30 208V 30A input
- Voltage and current instrumented PDU
- 2 outputs each for 4 cluster nodes
- Powersight voltage/current monitor external



PowerSight power monitor

- Records sampled data to internal memory
- Time-stamped data read out later via serial



Power Data File

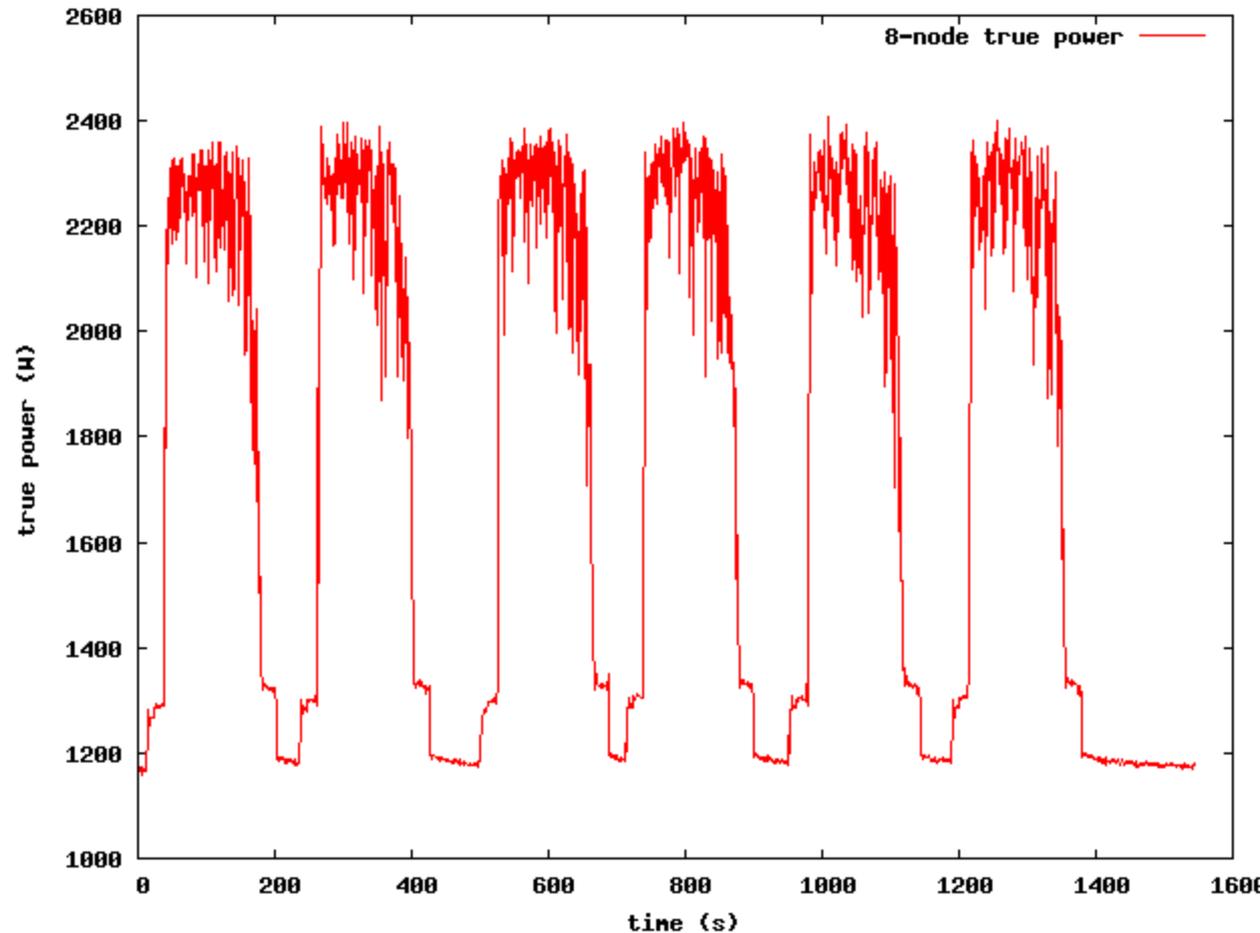
- *
- * Batch Log Began 11/02/10 at 14:16:51
- *
- * Data Type : 0x52 phase-phase
- * Data Period : 62500
- * Data Frames : 1545
- * Mon Period : 1
- * FreqMode : 2
- * Date Format : 1
- * Log Type : 1
- * Software Version : 3.3R
- * Firmware Version : 2.a5
- * Hardware Version : 6.00
- * Serial Number : 25663

Power Data File

* Start	Start	V12	V23	V31	I1	I2	I3
In	W1	W2	W3	Wt	VA1	VA2	VA3
VAt							
* Date	Time	Avg	Avg	Avg	Avg	Avg	Avg
Avg	Avg	Avg	Avg	Avg	Avg	Avg	Avg
Avg							
• 11/02/10	14:16:51	208.3	100.7	107.2	5.767		
5.804	0.000		0.000		603.8	568.2	0.0
	1172.0	620.5	584.8	0.0	1204.8		
• 11/02/10	14:16:52	208.2	100.9	107.3	5.759		
5.819	0.000		0.000		601.0	570.6	0.0
	1171.2	617.8	587.5	0.0	1204.8		
• 11/02/10	14:16:53	208.5	100.8	107.3	5.767		
5.815	0.000		0.000		604.2	569.6	0.0
	1173.6	621.0	586.4	0.0	1207.2		
• 11/02/10	14:16:54	208.1	100.9	107.3	5.704		
5.797	0.000		0.000		596.2	568.5	0.0
	1164.0	611.6	585.3	0.0	1196.8		

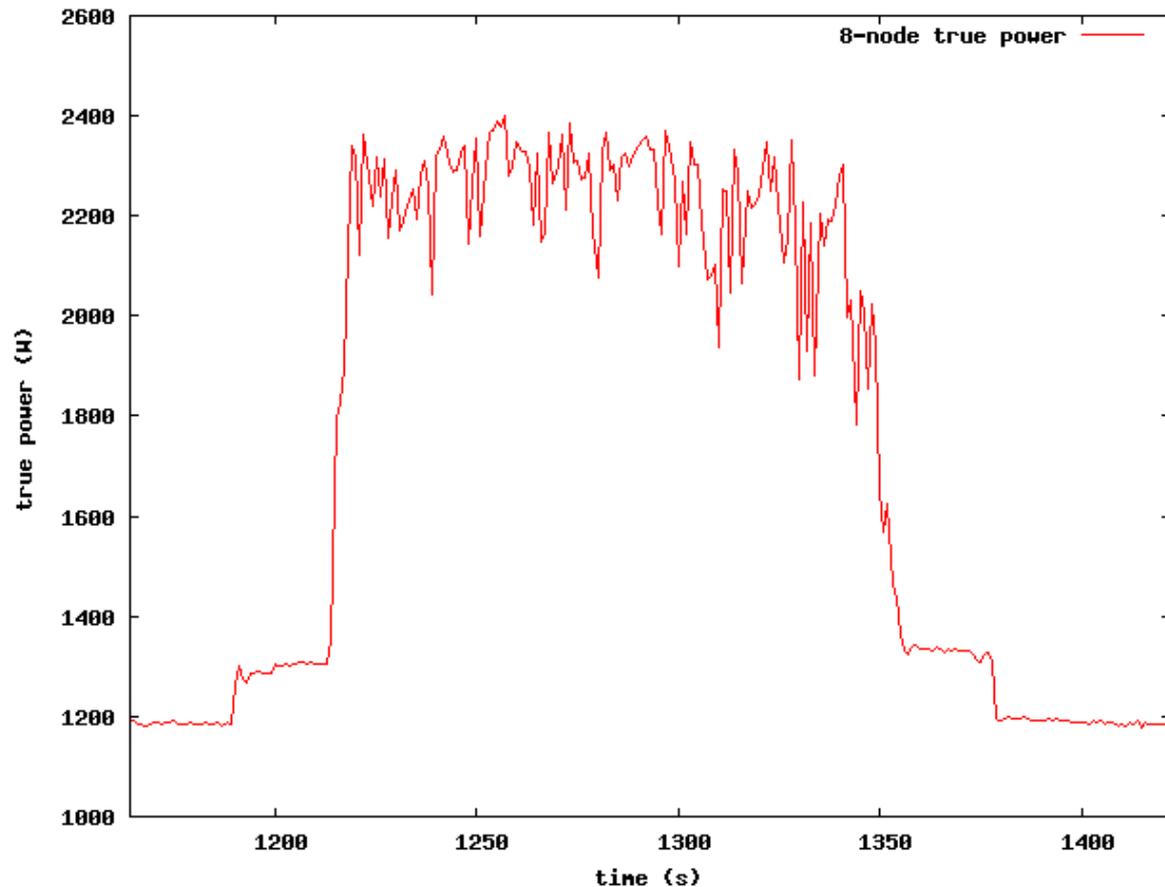
Overall Green500 Entry Test Period (6 HPL Runs)

- 6 HPL runs to get closest match to top500 run and allow for warm-up
- Last (#6) run closest to top500 submission speed



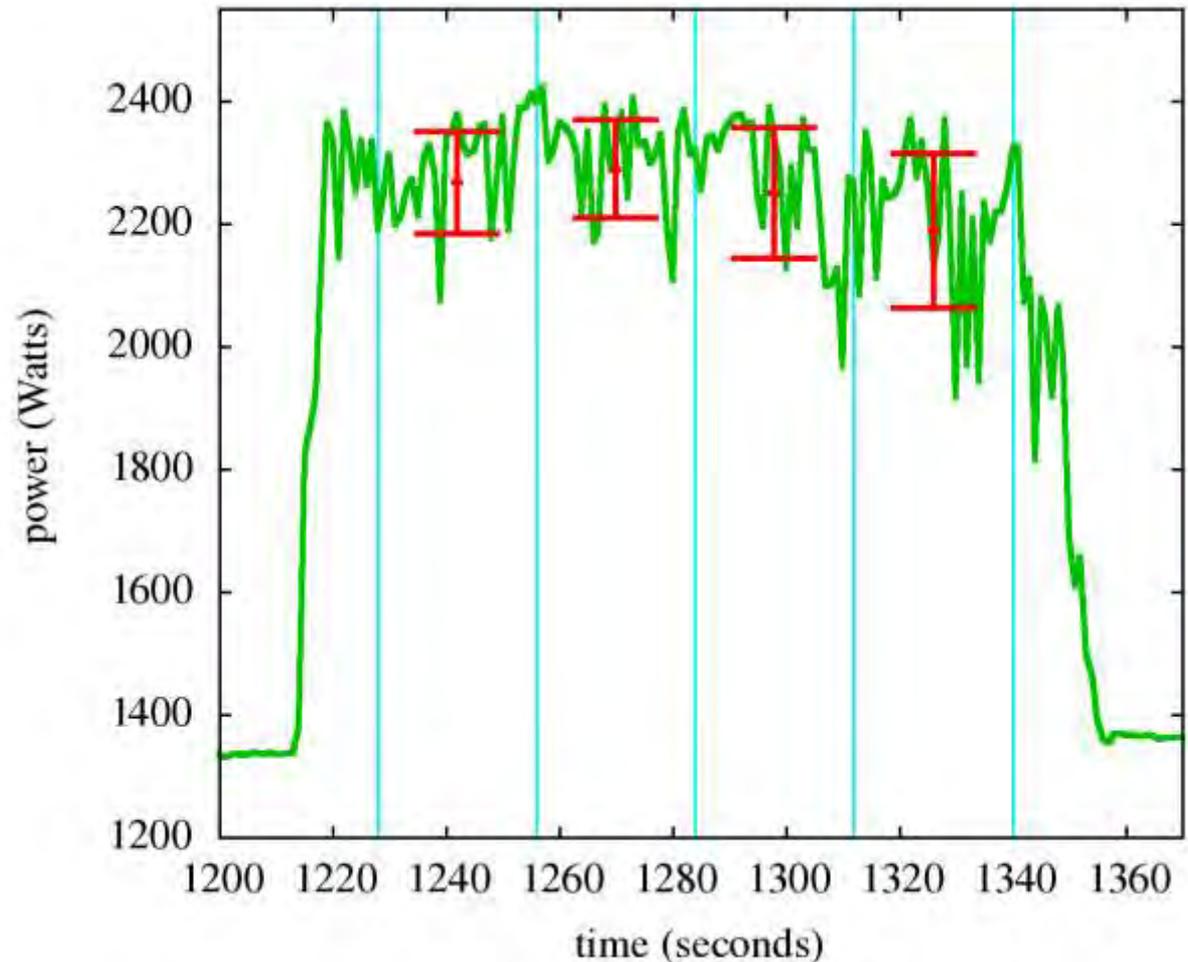
Power Graph for Measured Single HPL Run

- 2 shoulders: front porch for setup, back porch for answer validation
- Features:
 - Negative spikes
 - Power drops slightly over run



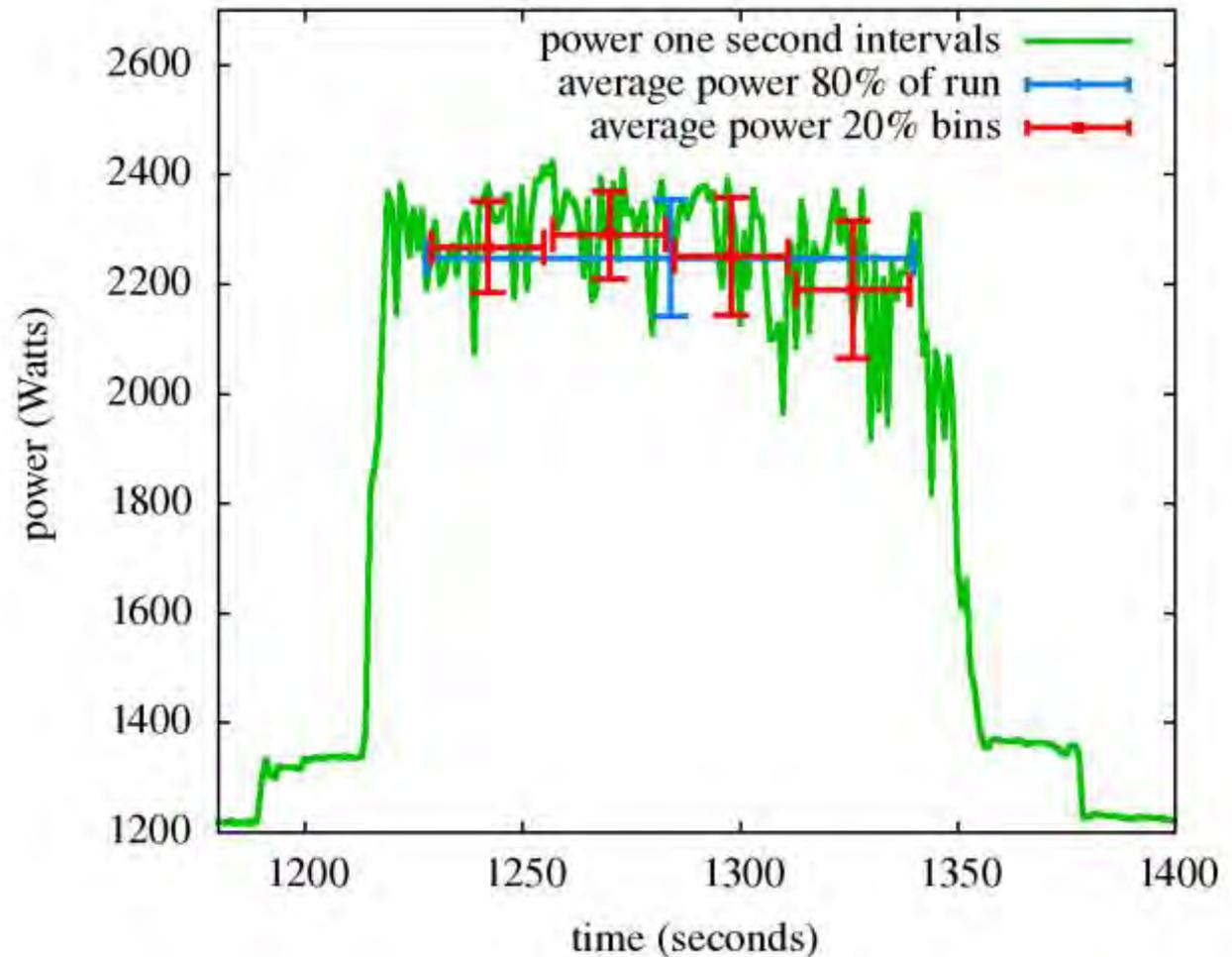
Average 8-node Power Draw In 20% Bins

- Spec for green500 is average power over 20% of run or more
- 4 20% bins in run middle: average 8-node power varies from 2289 W to 2189 W
- Power lowering is real physical effect; GPUS start to run out of computations to do



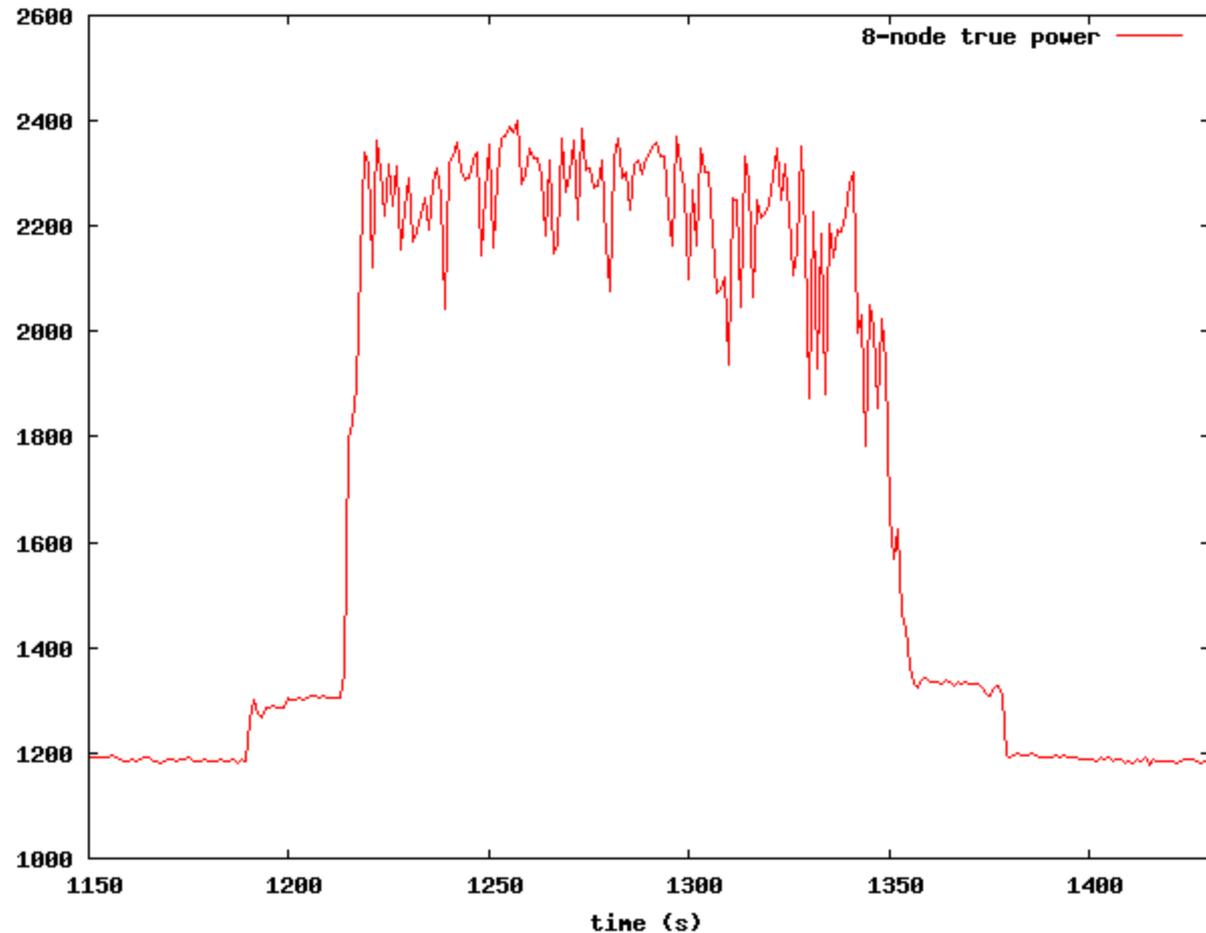
Final Average Power Calculation

- Average power calculated over 10%-90% range
- Calculated to be 2248 W (8 nodes) = **35.97 kW** for cluster



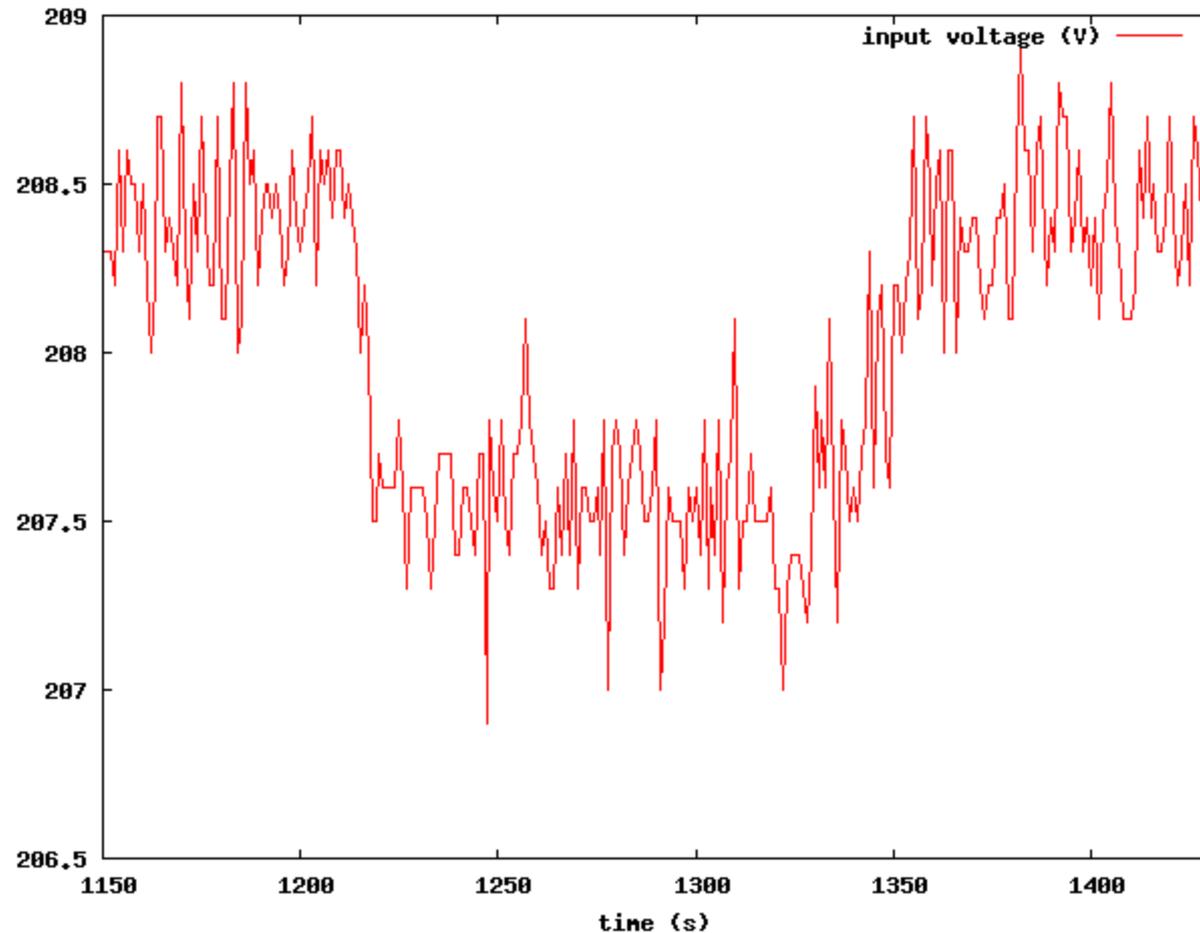
Power Draw for Voltage and Power Factor

- Expanded time range



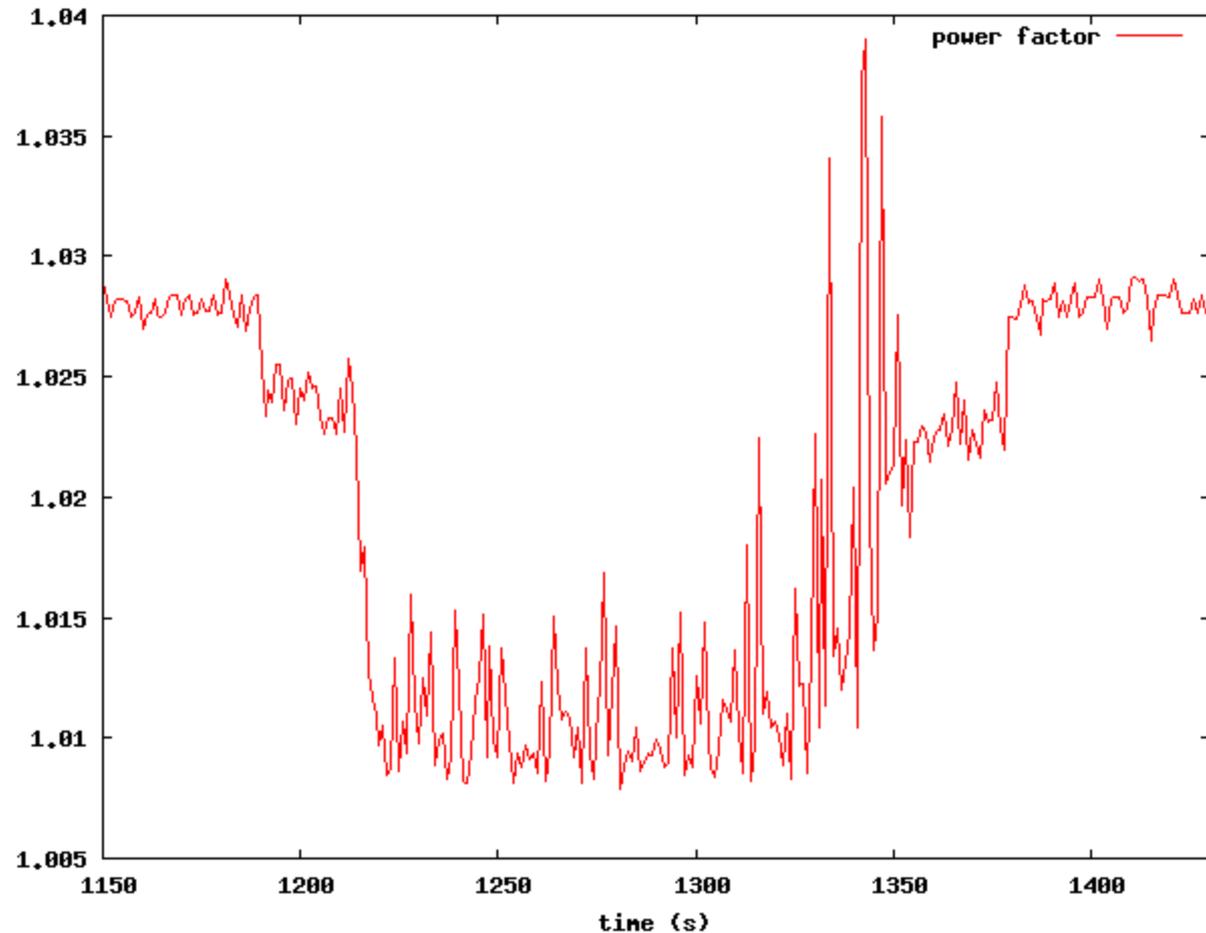
Input Voltage During HPL Runs

- Voltage drops but remains within spec
- Shown here for validation and as a sanity check
- Remains about 207.5 during HPL run



Power Factor

- Power factor remains below 1.035 for whole run including idle time
- Efficient power supplies, not overspecified



Current Questions and Next Steps

- What are the downward power spikes?
 - 1 second resolution *too coarse* to resolve cleanly
 - Need to use .2 second resolution current meter
- What are similar results with 1, 2, 4 nodes?
- How do the high-resolution timing results vary with application phase and input parameters? (Memory saturation tests have smooth graphs.)

- For more info see:
<http://www.ncsa.illinois.edu/News/Stories/GreenGPU/>

Next Steps to Work On:

- High-resolution Application Testing
- Arduino-based power monitor integrated into cluster control
- Instantaneous power available to running application; application control of power monitoring granularity