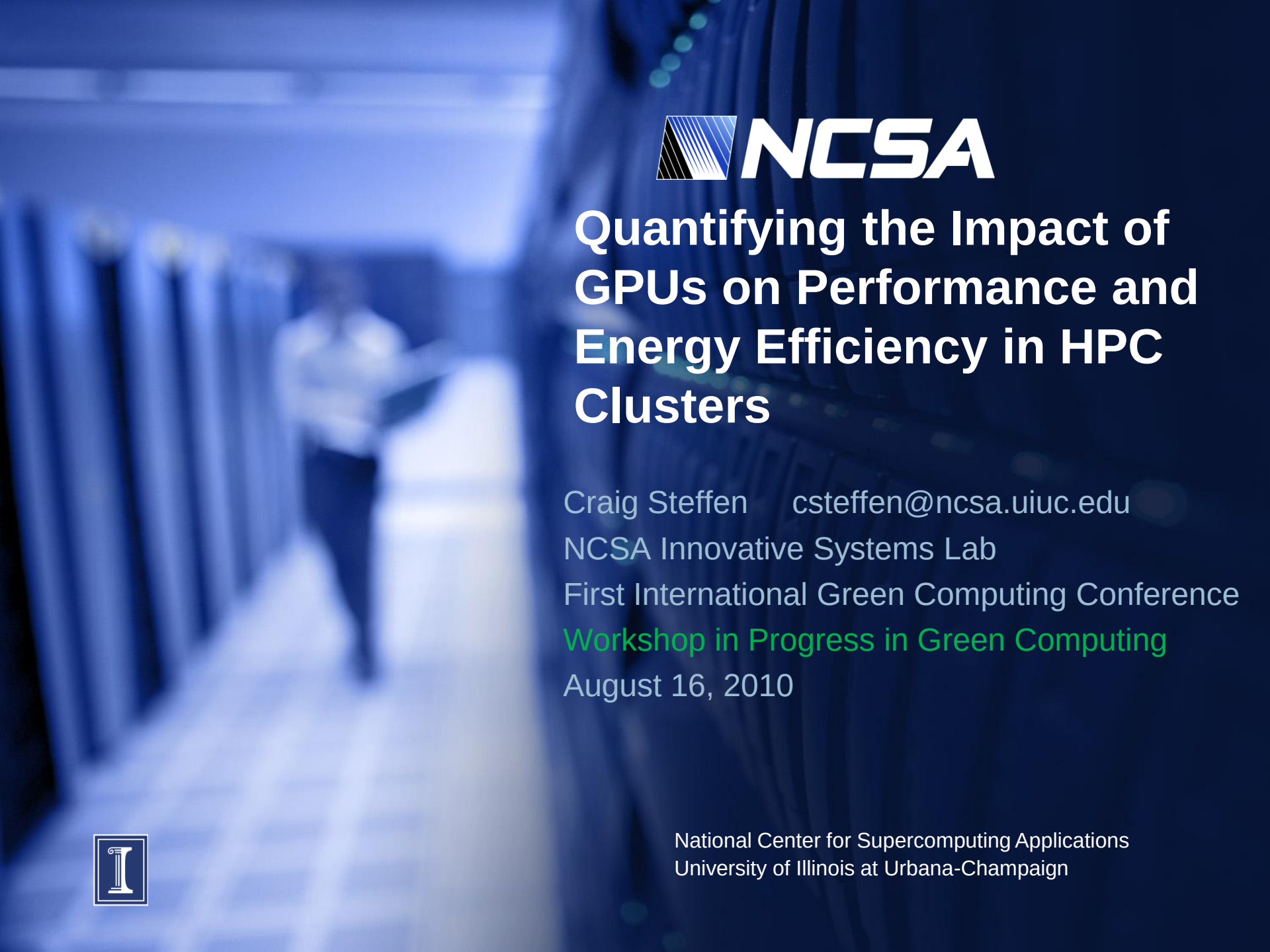


Quantifying the Impact of GPUs on Performance and Energy Efficiency in HPC Clusters

Craig Steffen csteffen@ncsa.uiuc.edu
NCSA Innovative Systems Lab
First International Green Computing Conference
Workshop in Progress in Green Computing
August 16, 2010

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

Cast of Characters

- 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

Overview

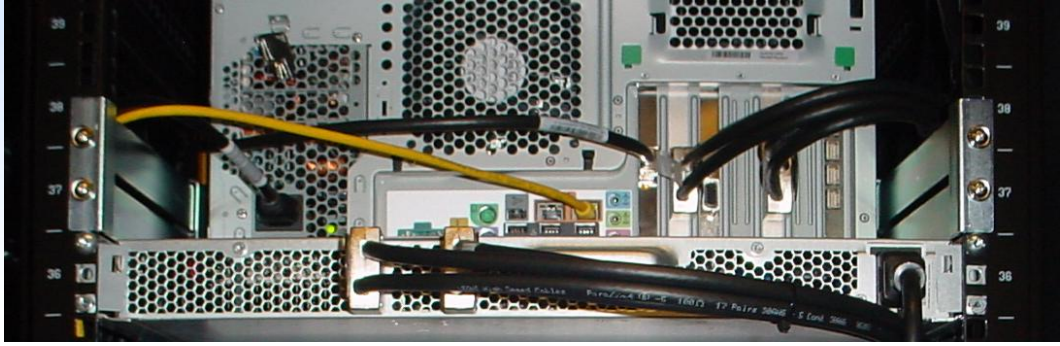
- AC GPU computing cluster
- Power monitoring
  - Search for power monitors
  - Roll our own--version 1: Tweet-A-Watt
  - Roll our own--version 2: Arduino-based power monitor
- Current power monitoring on real applications

AC cluster (Accelerator Cluster)

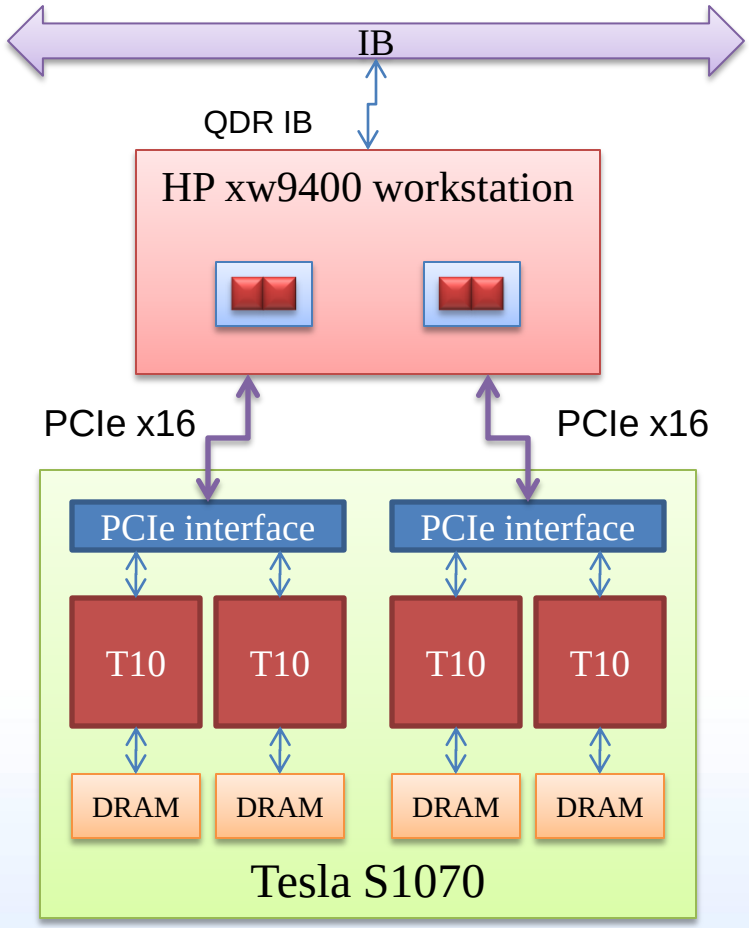
- Originally “QP” cluster for “Quadro Plex”
- 32 HP XW9400 nodes. Each node:
  - 2 dual-core 2.4 GHz Opteron 2216
  - 8 GB RAM per node
  - **NVIDIA Tesla S1070 each:**
    - **4 Tesla C1060 GPUs (128 total in cluster)**
- Interconnect network is QDR Infiniband
- CUDA 3.1 compiler/build stack
- Job control/scheduler 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)

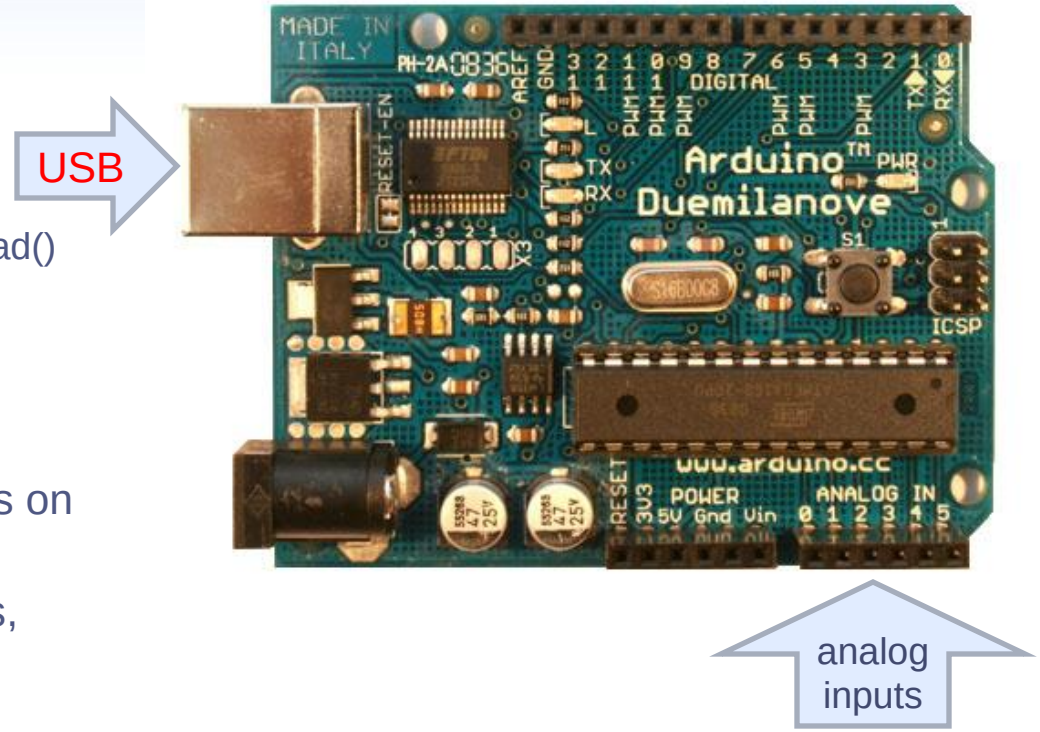
- Correlated voltage and current (read power factor and true power usage)
- 50-foot plus range (through two interior walls)
- Currently tied to software infrastructure: Application power studies done with Tweet-a-Watt

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

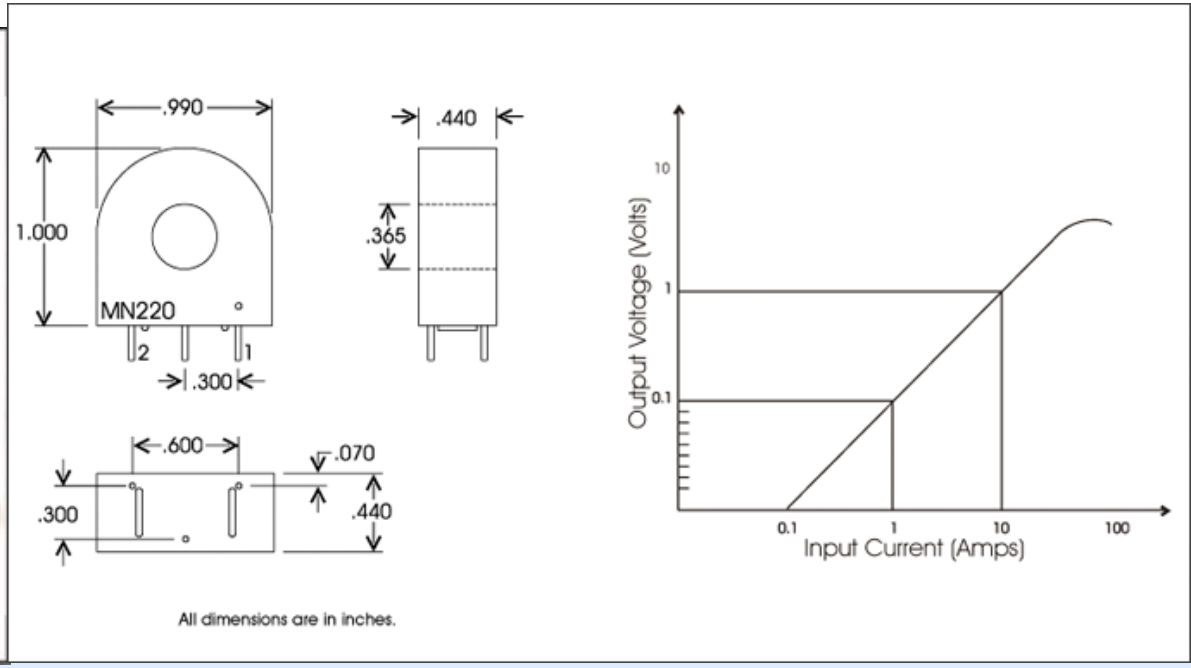
Arduino-based Power Monitor

- 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



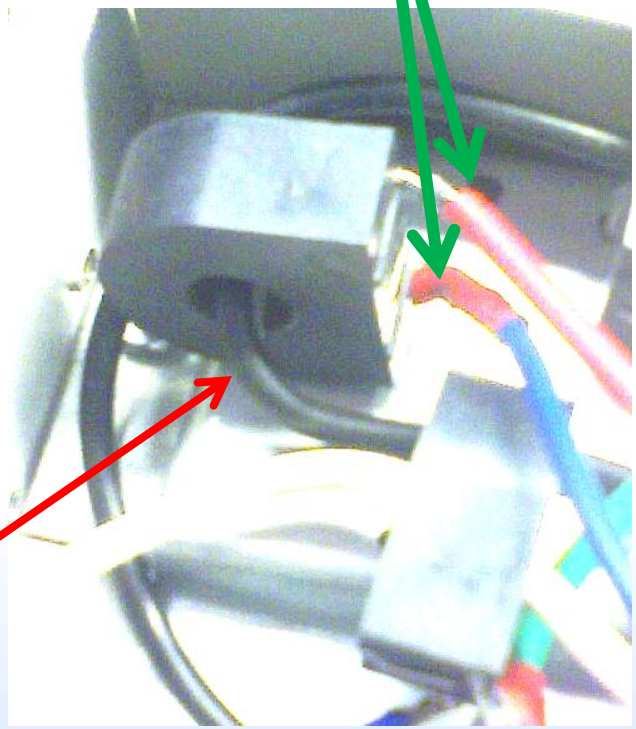
MN 220 picking transformer from Manutech

- 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.

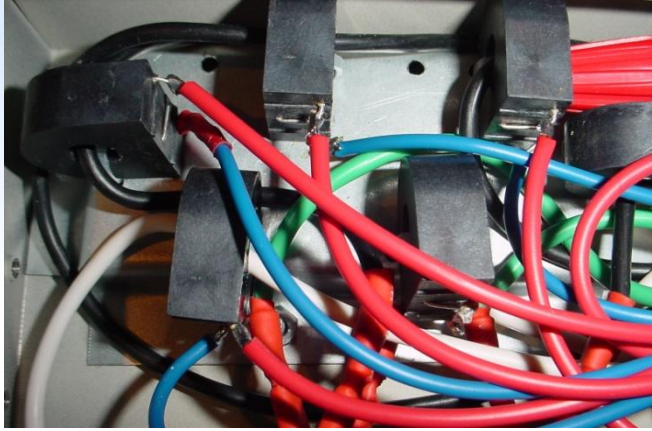


Current Sense Transformer

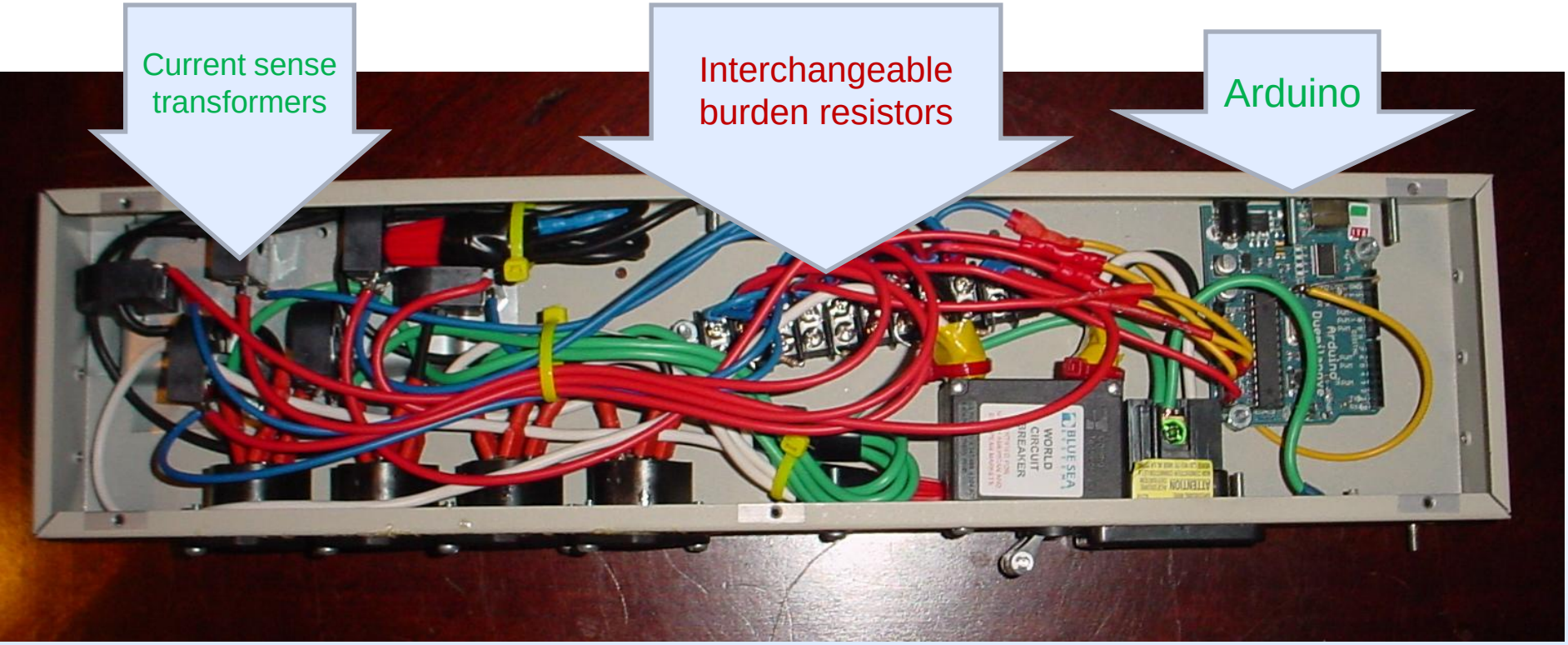
- MN-220 current "transformer" designed for 1 to 20 amp primary
  - 1000-1 step-up current transformer
- Burden resistor sets the sensitivity; sets "volts per count" calibration constant
- Allows current monitoring without Arduino contact with high-voltage wires



Industrial Design



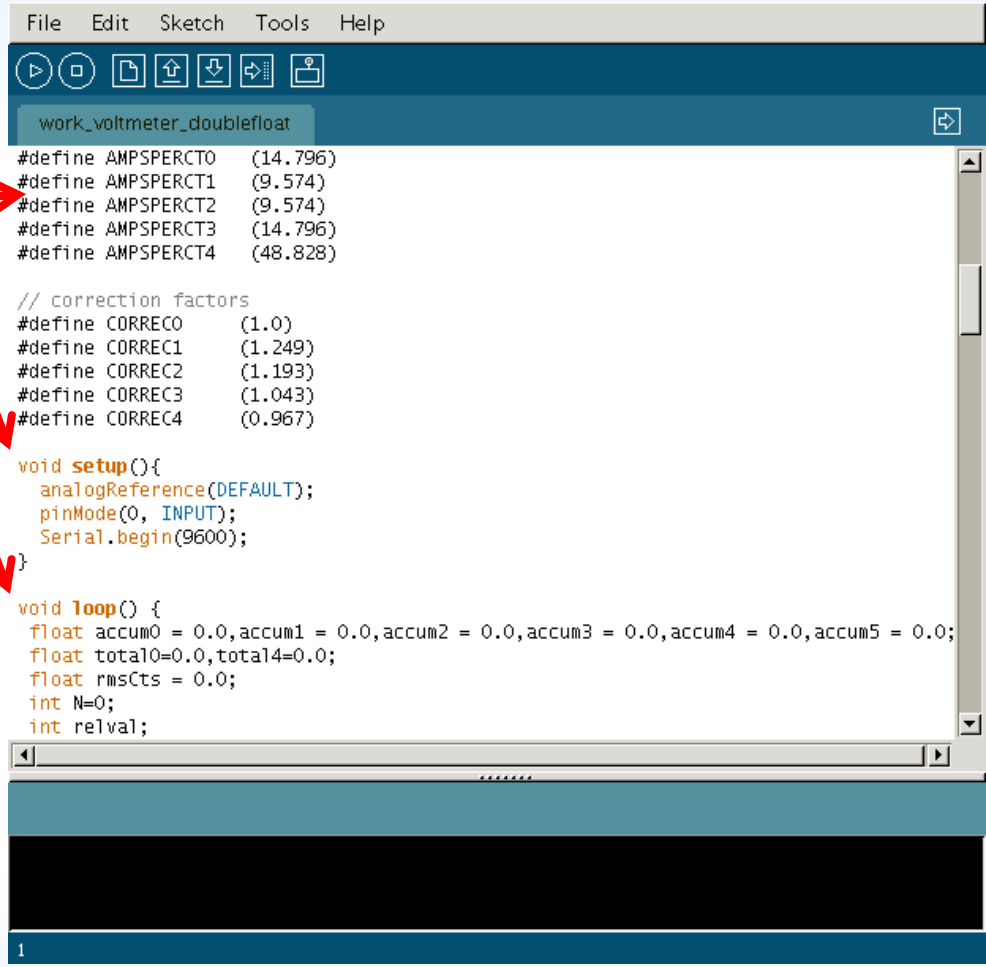
- 5 separate sense transformers for 4 power legs and opposite leg of input
- Current sense ONLY; Arduino is competely isolated from power conductors. No phase or power factor information, RMS current *only*



Arduino development environment

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

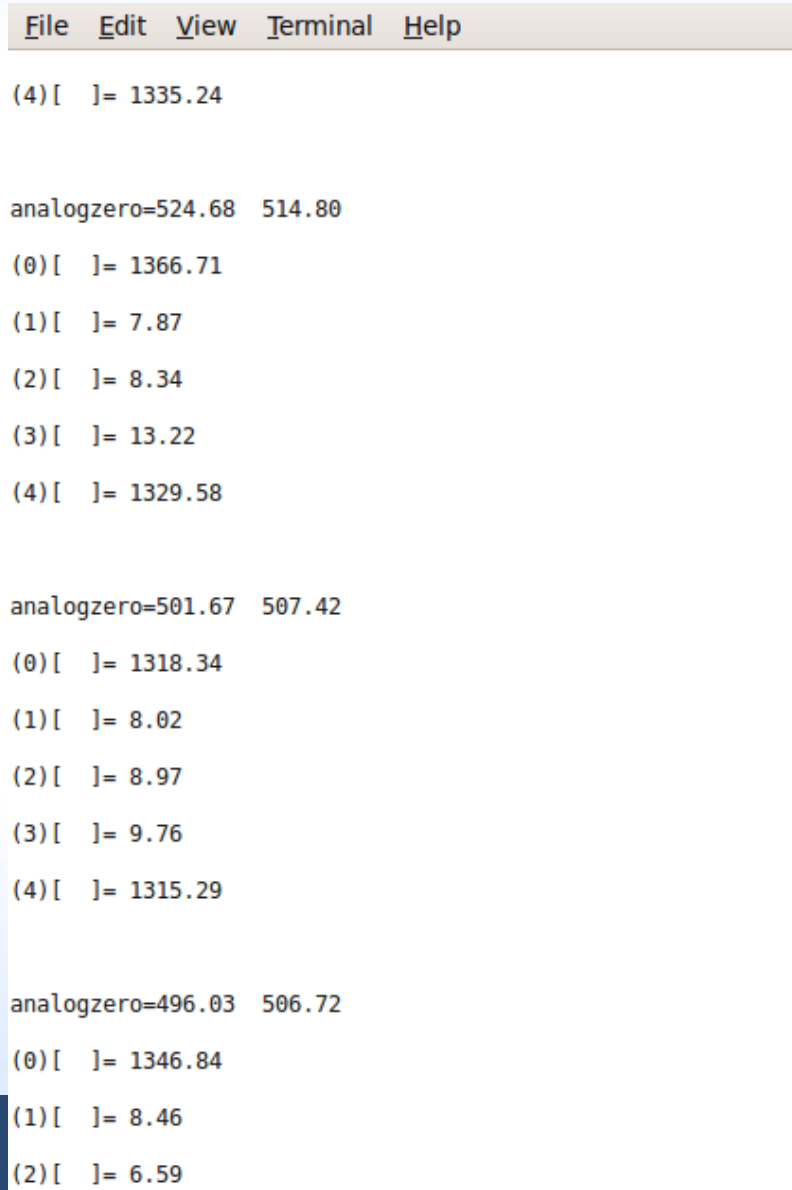
SPECIAL WARNING: Arduino INTs are 16 bits! Summing the squares of measured voltages (in the 200 to 400 range) will OVERFLOW the accumulator INT. (Convert to float b efore squaring)



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

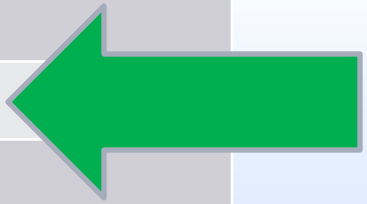
(note the \r \n double-line breaks)



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

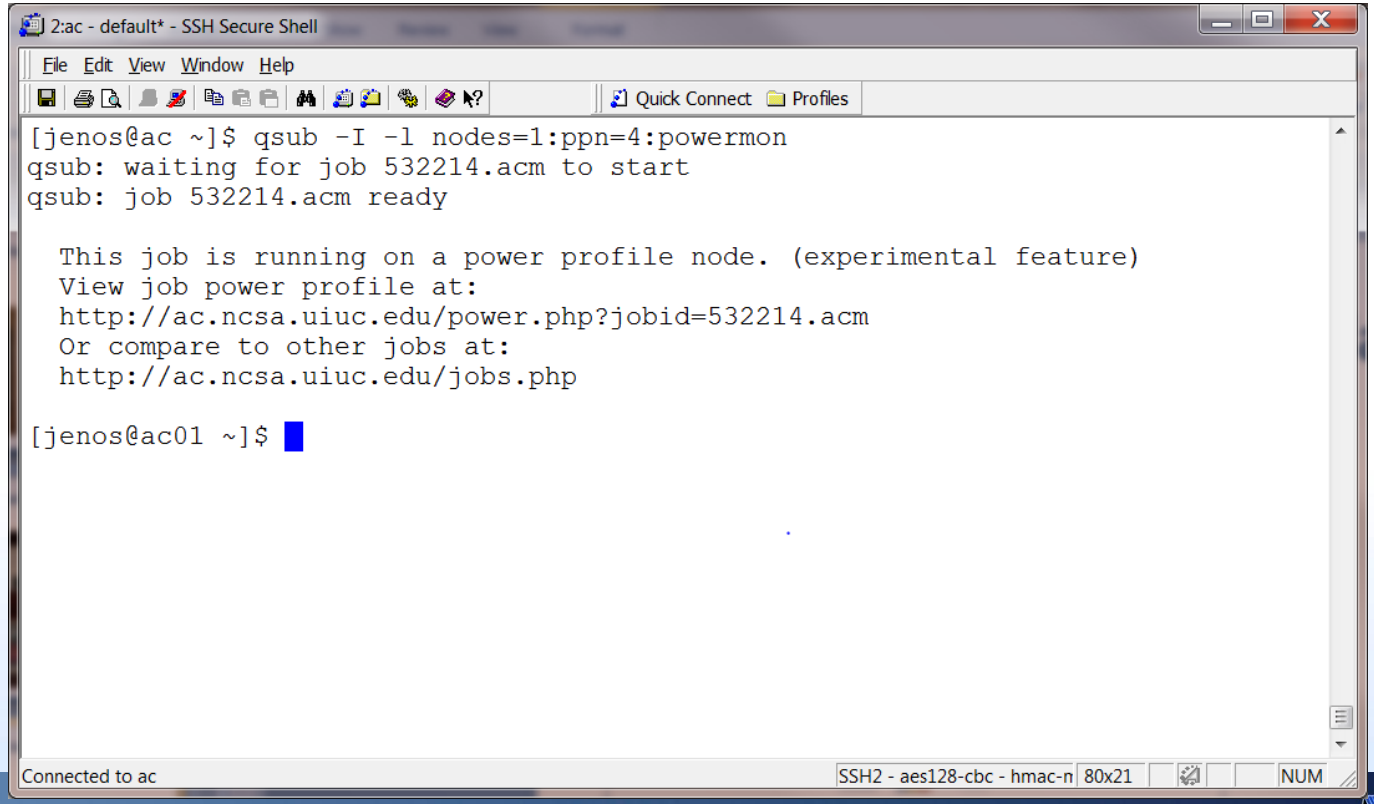
- Interchangable burden resistors to match pickup transformer output voltage to Arduino voltage sense
- Initially configured with two 600W channels, two 1000W channels, and main leg monitor is about 3300W for 16A at 208V
- Conclusion: no advantage to careful matching of burden resistors. Uncertainty 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

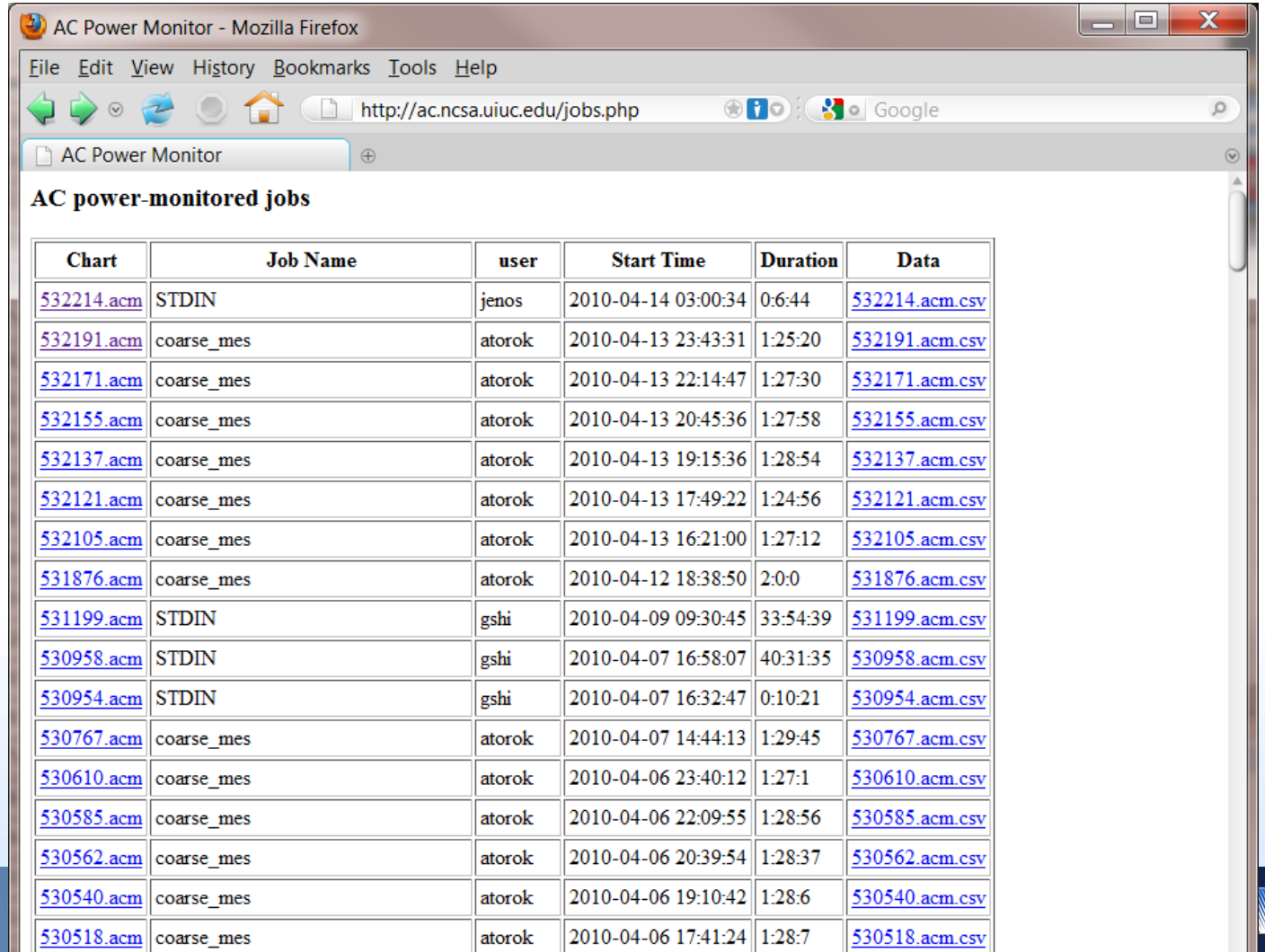
- Prolog scripts identify the (one) power monitored node (via Torque)
- Job history entry tags job to be attached to time window of power monitor data
- The job scripts create an automagic link to graphed output data per-sample and total usage summary

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)



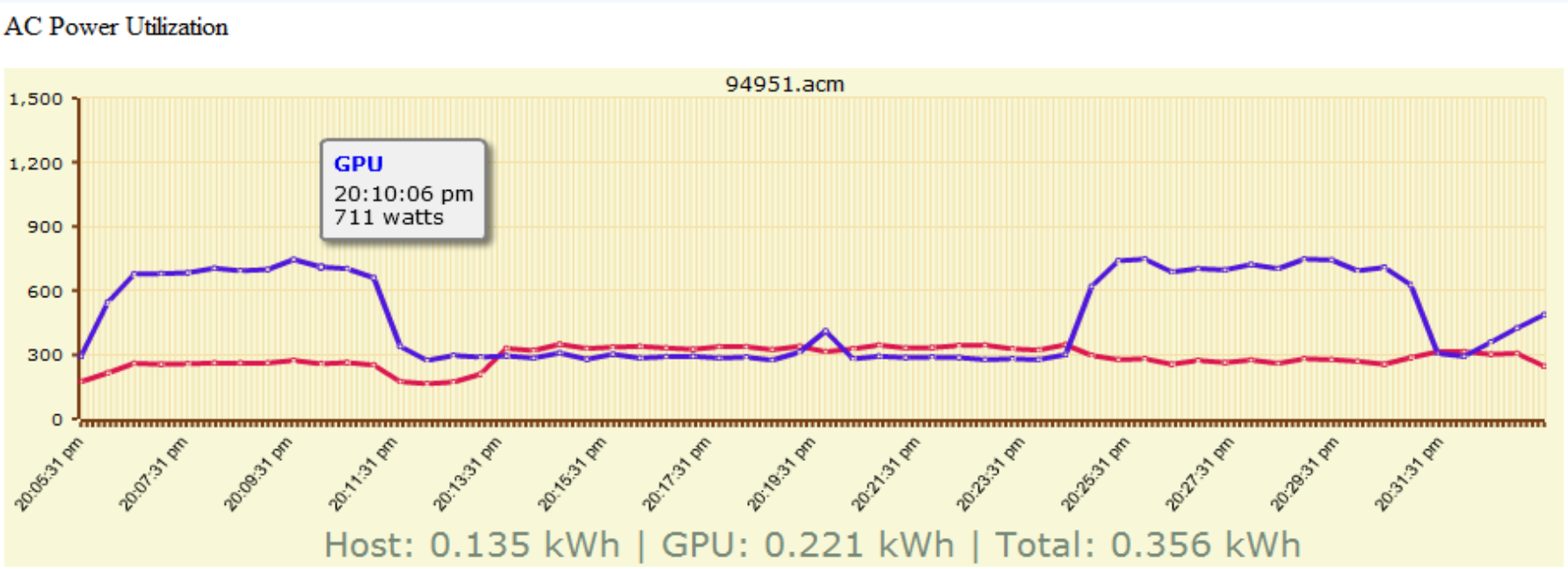
Each monitored job shows up as a link at http://ac.ncsa.uiuc.edu/jobs.php



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

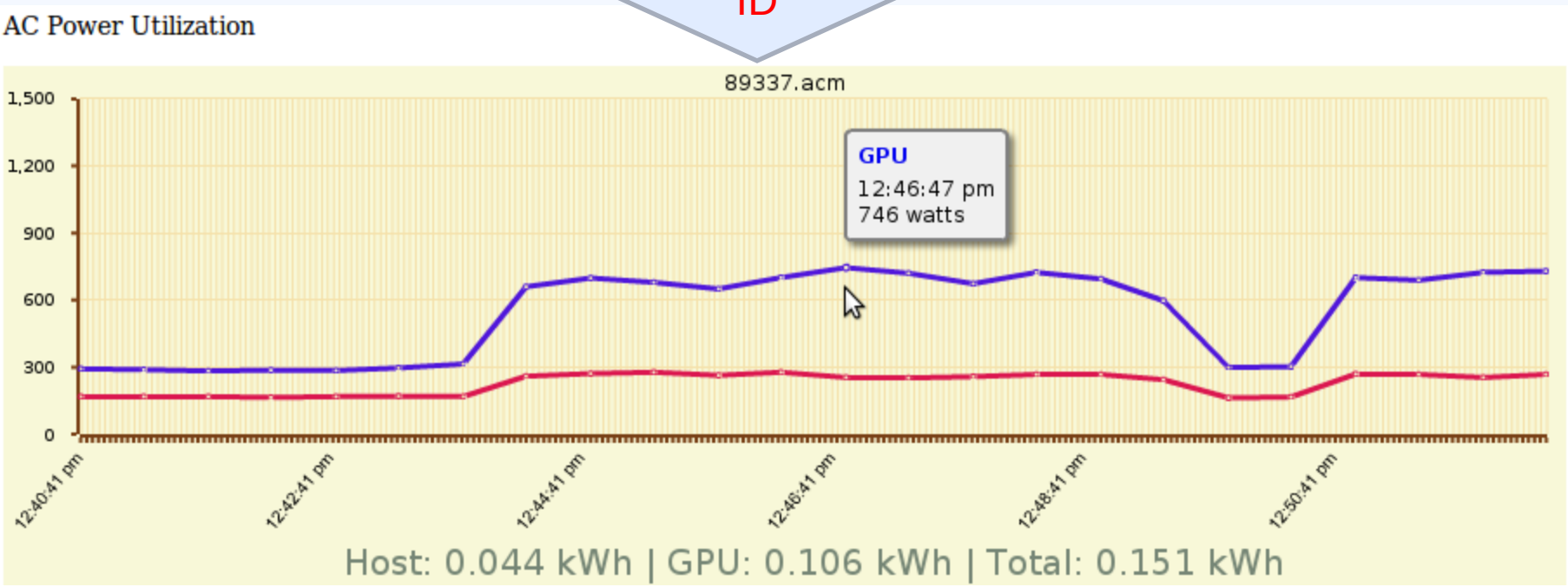


JSON Data

- 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

Job ID



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

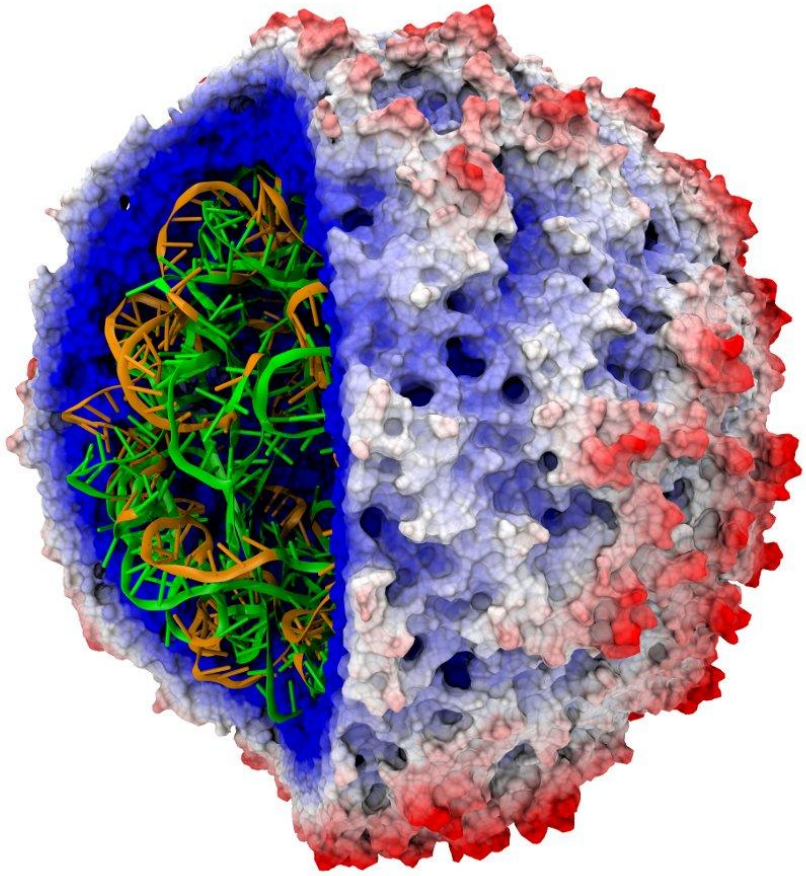
- Speedup measured in terms of **wall clock time** for whole application to run
- Power consumption measurements made over at least 20 sample runs
- Removed power measurements from startup and shutdown phases of applications

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

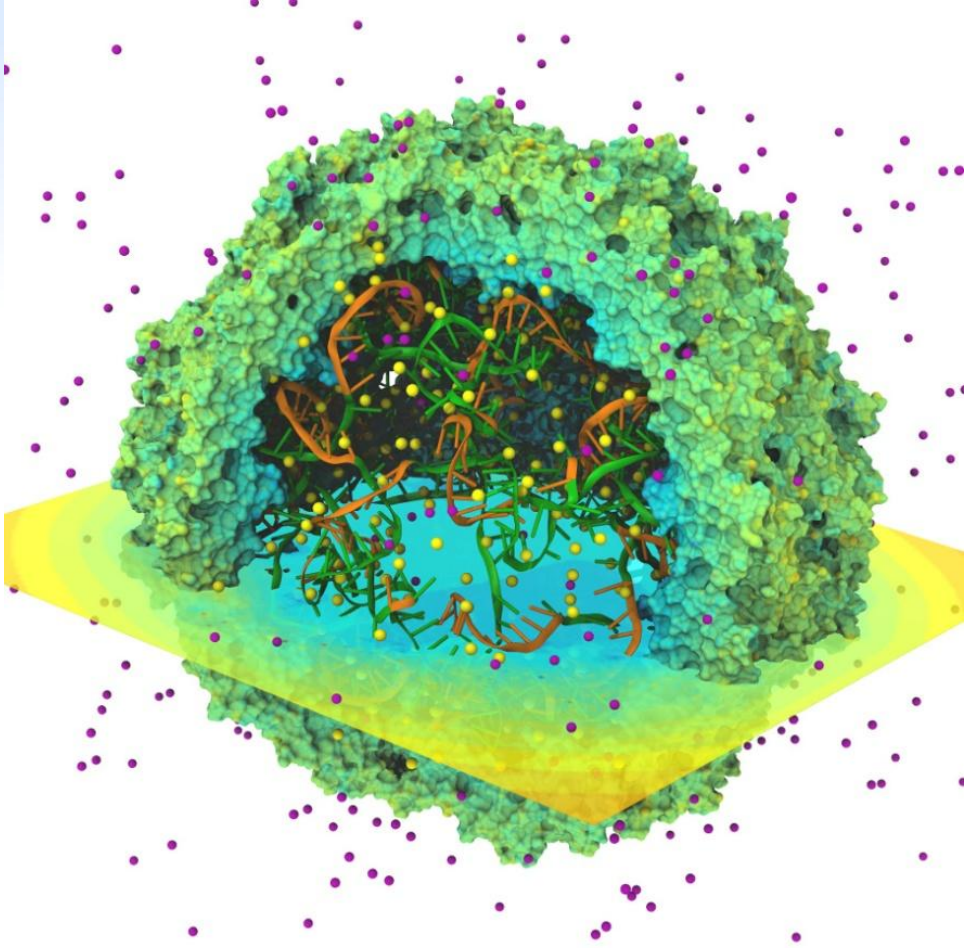
Case Study: NAMD

- Molecular Dynamics based on Charmm++ parallel programming environment; contains support for GPU codes
- Sample set was “STMV” 1 million atom virus simulation
  - Performance measure is simulatior time step per wall clock time
- CPU-only: 6.6 seconds per timeste 316 Watts
- CPU+GPU: 1.1 seconds per timestep; 681 Watts
- Speedup: 6
- Speedup-per-watt: 2.8



Case Study: VMD

- Molecular visualization and analysis tool
- Computes 3D electrostatic potential fields, forces, and electron orbitals
- Computation problem: 685,000 atom STMV trajectory using multilevel summation method
- CPU-only: t=1465 seconds; 300 watts
- CPU+GPU: t=58 seconds; 742 watts
- Speedup factor: 26
- Speedup-per-watt: 10.5



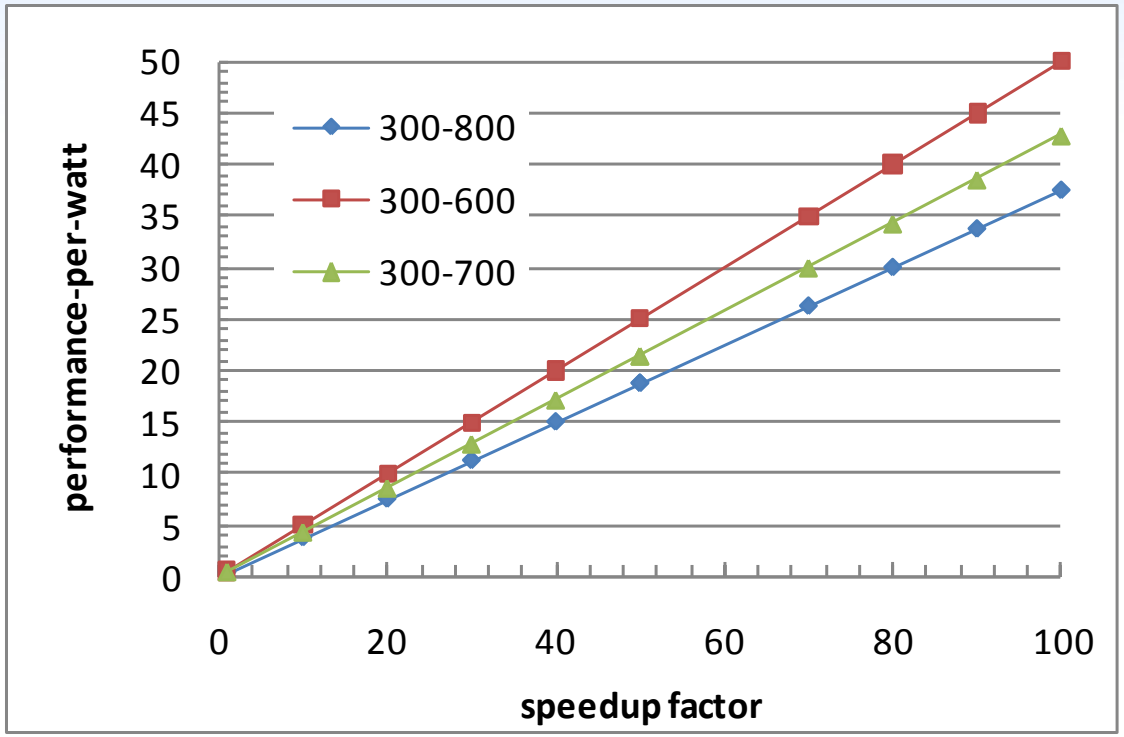
Case Study: QMCPACK

- Quantum Monte Carlo for tracking movement of interacting QM particles
- Simulating 128-atom simulation cell of bulk diamond, including 512 valence electrons
- Caviat:
  - CPU-only version uses double precision
  - CPU/GPU version uses mostly single-precision
  - Results are consistent within result uncertainty
- Results reported for Diffusion Monte Carlo; results for variational Monte Carlo are similar
- CPU-only: 1.16 walker generations per second
- CPU+GPU: 71.1 walker generations per second
- Speedup: 62
- Speedup-per-watt: 23

Application Case Study: MILC

- MIMD Lattice Computation of QCD
  - MILC calculations must now include electromagnetic effects in the calculations
- Problem set 28x28x28x96 lattice
- Single-core version only at this time
  - Computation on single-core vs. single-core+GPU
  - Power monitoring is for whole CPU node or CPU node+GPU
- CPU core: 77324 s
- CPU+GPU: 3881 s
- Speedup: 20
- Speedup-per-watt: 8

Current State: Speedup to Efficiency Correlation



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

Future Work

- Arduino-based power monitor not yet commissioned
- Data-collection issues with higher granularity:
  - Only take data for jobs (no monitoring idle nodes)
  - User selects sampling rate (high-res power monitoring only when it will be used
- Future: user tags application phases to refine data analysis (startup, shutdown, computation, communication)

HPRCTA 2010 (workshop at SC 2010 in New Orleans)

- **Fourth International Workshop on High-Performance Reconfigurable Computing Technology and Applications**
- All day Sunday, November 14
- The workshop is co-organized by the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign, the George Washington University, OpenFPGA, and Xilinx
- Submissions due September 3, 2010
- http://www.ncsa.illinois.edu/Conferences/HPRCTA10/
- **http://tinyurl.com/hprcta2010**

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**