LQCD Facilities at Jefferson Lab

Chip Watson
May 6, 2011

Outline

- Overview of hardware at Jefferson Lab
- GPU evolution since last year
- GPU R&D activities
- Operations
- Summary

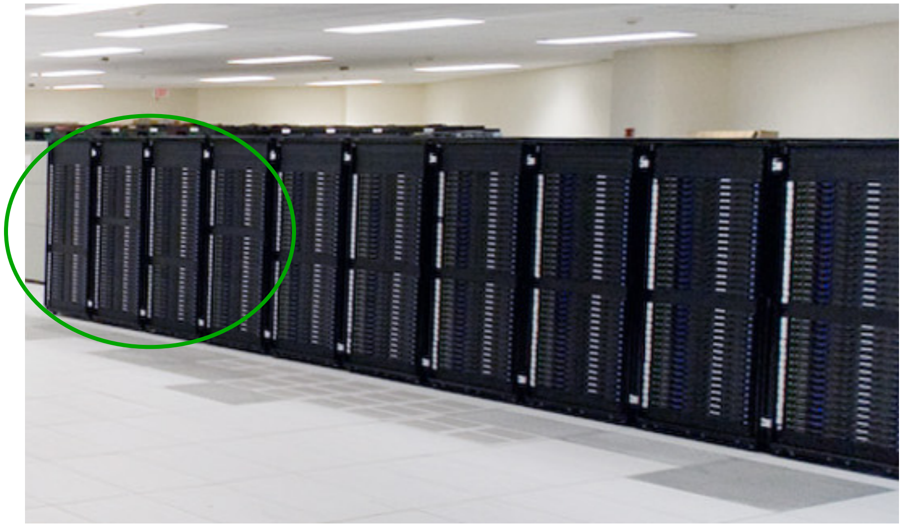
CPU + Infiniband Clusters

The LQCD ARRA clusters were expanded by 5 racks to 17 racks of 32 nodes 2.4 GHz Nehalem (10 racks) / 2.53 GHz Westmere (7 racks), each about 20 Gflops/node

The network is QDR Infiniband; nodes are mostly connected as sets of 32 (single rack), with one set of 4 racks interconnected with 2:1 oversubscription. Deploying nodes as sets of 32 reduced the cost of the Infiniband fabric while maintaining the highest efficiency for jobs up to 640 Gflops.

All racks have 2 uplinks to a core switch for file services

Note: a full homogeneous fabric with a 2:1 oversubscription would have required 13 additional switches and 200 cables, and would have had somewhat worse multi-node scaling, yielding 5%-10% lower performance per dollar. A few extra nodes on the fabric solved the problems of 1 or 2 failed nodes preventing large jobs from running.



File Servers

Lustre distributed file system, supporting 2 of our 3 areas:

/cache/<project> write through cache to tape with auto-delete
/volatile/<project> large scratch space with auto-delete

Total Lustre Capacity: 416 TB, Bandwidth > 2 GBytes/s

Phase 1: 224 TB across 14 servers

- dual Nehalem 2.26 GHz, 12 GB memory, DDR Infiniband
- 24*1TB disks, RAID-6 8+2, OS mirrored, journal separate
- bandwidth 1.4 GB/s using 6 nodes (single DDR uplink)

Phase 2: 192 TB across 4 servers

- similar to above, but with 3 RAID-6 (8+2) strips per server instead of 2
- 2 TB disks, QDR Infiniband, higher performance RAID controller (?)
- somewhat lower bandwidth / TB, but still more than necessary
- still being commissioned: performance below expectations

GPU Cluster Configurations

Fermi Tesla, quad GPU

32 nodes, 128 GPUs (C2050)
QDR Infiniband in 5th slot, half bandwidth
C2050s have ECC enabled, thus 2.67 GB memory

Fermi GTX, quad GPU

45 nodes, 180 GPUs (GTX-480)
Re-cycled SDR Infiniband in 5th slot, half bandwidth
GTX-480s have 1.5 GB memory

GT200b GTX, quad GPU

38 nodes, 156 GPUs (GTX-285)
Re-cycled SDR Infiniband in 5th slot, half bandwidth

GT200b Tesla, quad GPU

2 nodes, 8 GPUs (4 C1050, 4 in S1070)
Re-cycled SDR Infiniband in 5th slot, half bandwidth

GT200b GTX, single GPU

34 nodes, 34 GPUs (GTX-285)
QDR Infiniband, full bandwidth

- ✧ Most GPUs are in a quad GPU configuration in a 4-slot chassis, so the Infiniband card is in a lower bandwidth slot (4x vs. 8x or 16x)
- ✧ Scaling above 8-16 GPUs is poor, but 99% of the workload is single node.
- ✧ Queues:
  quads: gpu
  singles: ibg, which shares the nodes with the ib (Infiniband) queue, but with higher priority.

GPU Comparison

Card	GPU	#cores	clock speed (GHz)	memory size (GB)	raw memory bandwidth (GB/s)	clover inverter (Gflops)[1]	cost
GTX-285	GT200b	240	1.47	2	159	135	$500
C1060	GT200b	240	1.30	4	102	100	$1500
GTX-480	Fermi	480	1.40	1.25	177	270	$500
C2050[2]	Fermi	448	1.15	2.67	144	185	$2100

[1] Newest development code gets up to 310 Gflops on GTX-480; data is this talk uses older 270 Gflops; all numbers are for mixed precision (half + single)

[2] C2050 evaluated with ECC enabled

The Fermi Tesla line of cards (C2050) has a significant advantage in having ECC memory so that more than just inverters can be safely executed. This comes at a steep price: 4x on GPU price, and 1.5x on lower performance. Integrated into a host this yields a price performance difference between them of 3x.

Conclusion: judicious use of gaming cards is a very good idea as long as we have inverter heavy loads (which we do).

GTX-480 Problems

The 192 GTX-285 cards we bought in phase 1 were very stable, and exhibited no errors in running a 2 hour memory test program.

The 210 GTX-480 cards did much worse:

86 encountered no errors in a 2 hour test
80 encountered 1-10 errors in 2 hours
26 encountered memory errors at 1-10 per minute
2 encountered memory errors at about 1 per second
4 undetected by the NVIDIA driver
2 bad fan & hung running CUDA code
10 hung running CUDA code

We were an early buyer of GTX-480 cards for computing, and apparently caught some early quality control issues.

The first set of 86 was put into production fairly early, and the second set a while later (after experience showed they delivered the same results). We set the threshold (arbitrarily?) at 10 errors per two hours for production running, where it remains today, although 95% are now at 0 or 1 per two hours, so we could further reduce the threshold at modest cost. (Feedback appreciated.)

GTX-480 Problem Resolution

1. The manufacturer PNY wasn't helpful, even with cards that would not run at all. Only 2 were replaced under manufacturer warranty (bad fans).
2. JLab tried several other memory test programs, tried under-clocking the poor cards, all to no avail. LQCD software Chroma+QUDA, however, ran successfully on all functioning cards, despite the memory errors (evidence that low error rates are not a problem).
3. We developed a more rigorous testing procedure, running a 2 hour test on every GPU every week to catch any further degradation (1.5% load), and removing from the production queue any GPU with more than 10 errors in 2 hours. Users were warned that the cards were only suitable for inverters, and that applications should test inversion residuals.
4. The cluster vendor Koi eventually agreed to replace 35 cards with new cards from ASUS. All but 1 of these passed our tests with low error rates.
5. Today 4*45 = 180 GTX-480s are in production (and consequently we have only 1 rack of single GPU instead of the planned 2). Memory testing is ongoing, and has caught one GTX-285 and one C2050 failure.

Gaming GPUs: An Early Taste of Exascale

Reliability *– System architecture will be complicated by the increasingly probabilistic nature of transistor behavior due to reduced operating voltages, gate oxides, and channel widths/lengths resulting in very small noise margins. Given that state-of-the-art chips contain billions of transistors and the multiplicative nature of reliability laws, building resilient computing systems out of such unreliable components will become an increasing challenge. This cannot be cost- effectively addressed with pairing or TMR* [Triple Modular Redundancy]*; rather, it must be addressed by X-stack software and perhaps even scientific applications.*

-- from The International Exascale Software Project Roadmap
http://www.exascale.org/

GPU Job Effective Performance

Comparing GPUs to regular clusters can't be done on the basis of inverter performance (Amdahl's Law problem), so instead we compare job clock times, and from that derive an "effective" performance, which is the cluster inverter performance multiplied by the job clock time reduction.

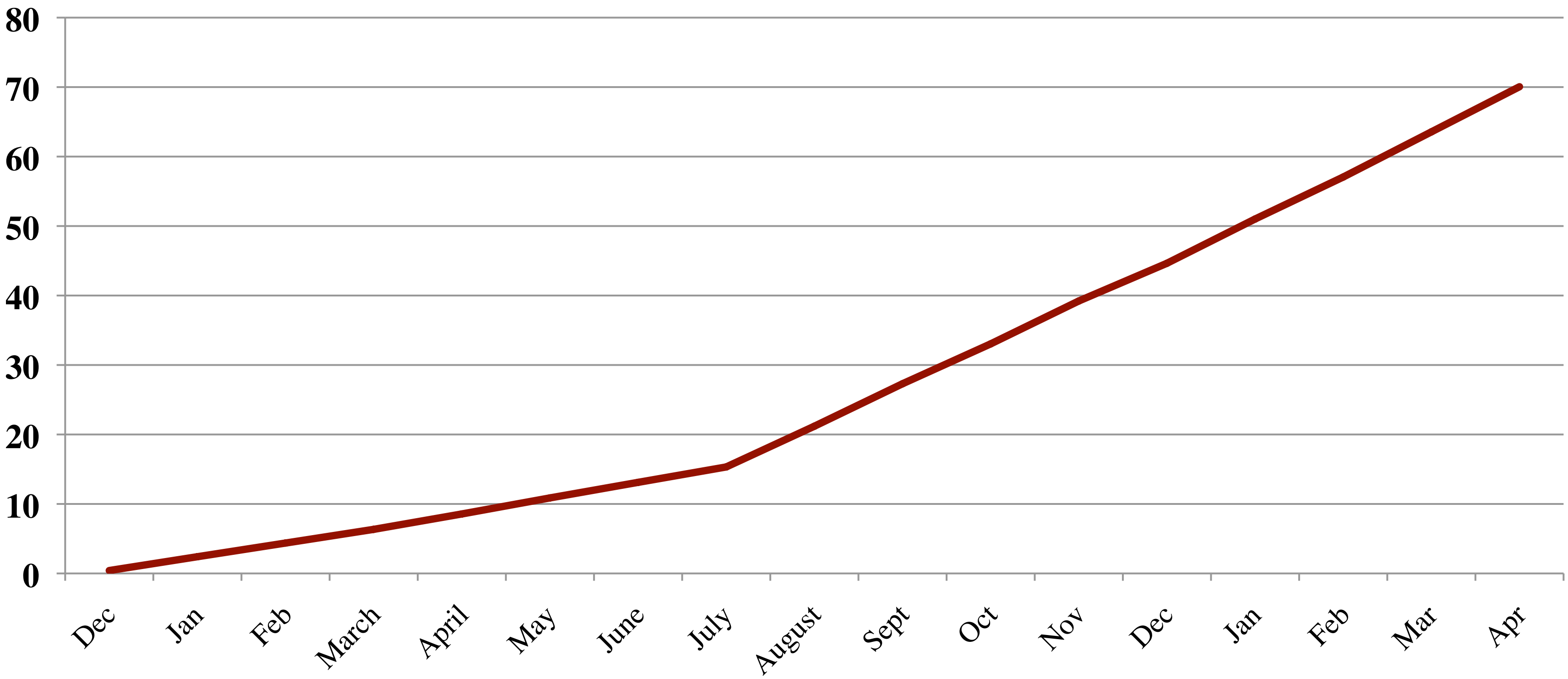
The following table shows the number of core-hours in a job needed to match one GPU-hour in a job. Last project used 32 single GPU nodes and was I/O bound.

The allocation-weighted performance of the cluster is **63 TFlops**.

Project	2010-2011 Hours	#GPUs, nodes	Jpsi core hours / GPU hour (job time)	Effective Performance Gflops/node	GPU used
Spectrum	1,359,000	4, 1	180	800	(average)
thermo	503,000	4, 1	90	400	(average)
disco	459,000	4, 1	92	410	C2050
Tcolor	404,000	4, 1	40	175	GTX285
emc	311,000	4, 1	80	350	(average)
gwu	136,000	32, 32	47	50	GTX285

*Not all projects shown.

ARRA Delivered "Effective" Tflops-years



Notes

1. GPU nodes are rated based upon relative performance of equivalent infiniband cluster jobs for the production projects, weighted by the projects allocations, to give "effective" Tflops.
2. Clover performance (Gflops, single-half, per GPU in 4 GPU job):
   GTX-285 = 130, C2050 with ECC on = 176, GTX-480 = 273
3. Current aggregate performance ("effective"): 63 Tflops GPU, 10 Tflops conventional, 73 Tflops total

GPU – R&D

Remaining ARRA contingency funds spent on GPU R&D system:

New motherboard (Tyan)

- ✓ 8 PCIe2 x16 slots
  dual 5520, 3-port PLX switch chip on each x16 to double number of slots
- ✓ 4 GPUs on one 5520
  2 QDR Infiniband cards on the other

Explorations enabled:

1. 4 GPUs on a single 5520 allows us to better exploit the new GPU-direct capability of the Tesla cards: transfers directly from one GPU's memory to the other's memory, without going through host memory; this cuts message latency in half (effectively doubles bandwidth)
2. Dual QDR cards (a modest cost when using expensive Tesla cards), yields better bandwidth per GPU than a dual GPU / single QDR; for 2D communications, half of all messages are consumed in the box, lowering external (box) I/O requirements
3. 4 GPUs per host lowers the cost per GPU
4. 4-8 GPUs per box might help with domain decomposition approaches: GPU, box, cluster, where the in-box communications is 10x faster than in-cluster.

Planned evolution of these 6 nodes

Funding only allowed procuring 6 nodes and 8+4 GPUs

- M2050 GPUs (2 quad nodes) are somewhat higher clock speed Tesla Fermi with no on-card fan (uses the chassis airflow); buying these allows us to do performance comparison against the slightly more expensive C2050s
- The other 4 nodes will have C2050s moved into them from the 10g cluster; this 6 node cluster can be used for quad GPU dual QDR strong scaling studies at high bandwidth per GPU (emulate FDR Infiniband performance)
- We also procured 4 GTX-580 cards (512 cores, full Fermi); if these cards perform well, we will purchase 12 more and use these 4 sets of 4 to backfill the C2050s, otherwise we will backfill with GTX-480 (funding constraint).

If we can find funding for 2 more nodes, we could support some useful larger scale production jobs, such as the 32 GPU calculations done in the last 5 months. (Alternative: create a 16 or 32 dual partition again.)

Whenever we are not testing, these 6 nodes will be accessible via the gpu queue, essentially the same as the 10g C2050 nodes but with better scaling.

Operations

Fair share: (same as last year)

- Usage is controlled via Maui, "fair share" based on allocations
- Fair share is adjusted every month or two, based upon remaining time
- Separate projects are used for the GPUs, treating 1 GPU as the unit of scheduling, but still with node exclusive jobs

Consumption of tape and disk is deducted from allocations at defined rates (a total of 5% of total allocations year to date, 0%-10% of individual allocations).

New: we have allowed exchanges between core hours and GPU hours as 8:1, reflecting cost, not performance (which is ~100:1), at small scale so as to not impact allocations too much (works if flows both ways roughly balance)

Disk Space, 3 name spaces:

/work (user managed, on SUN ZFS systems)

/cache (write-through cache to tape, on Lustre)

/volatile (daemon keeps it from filling up)

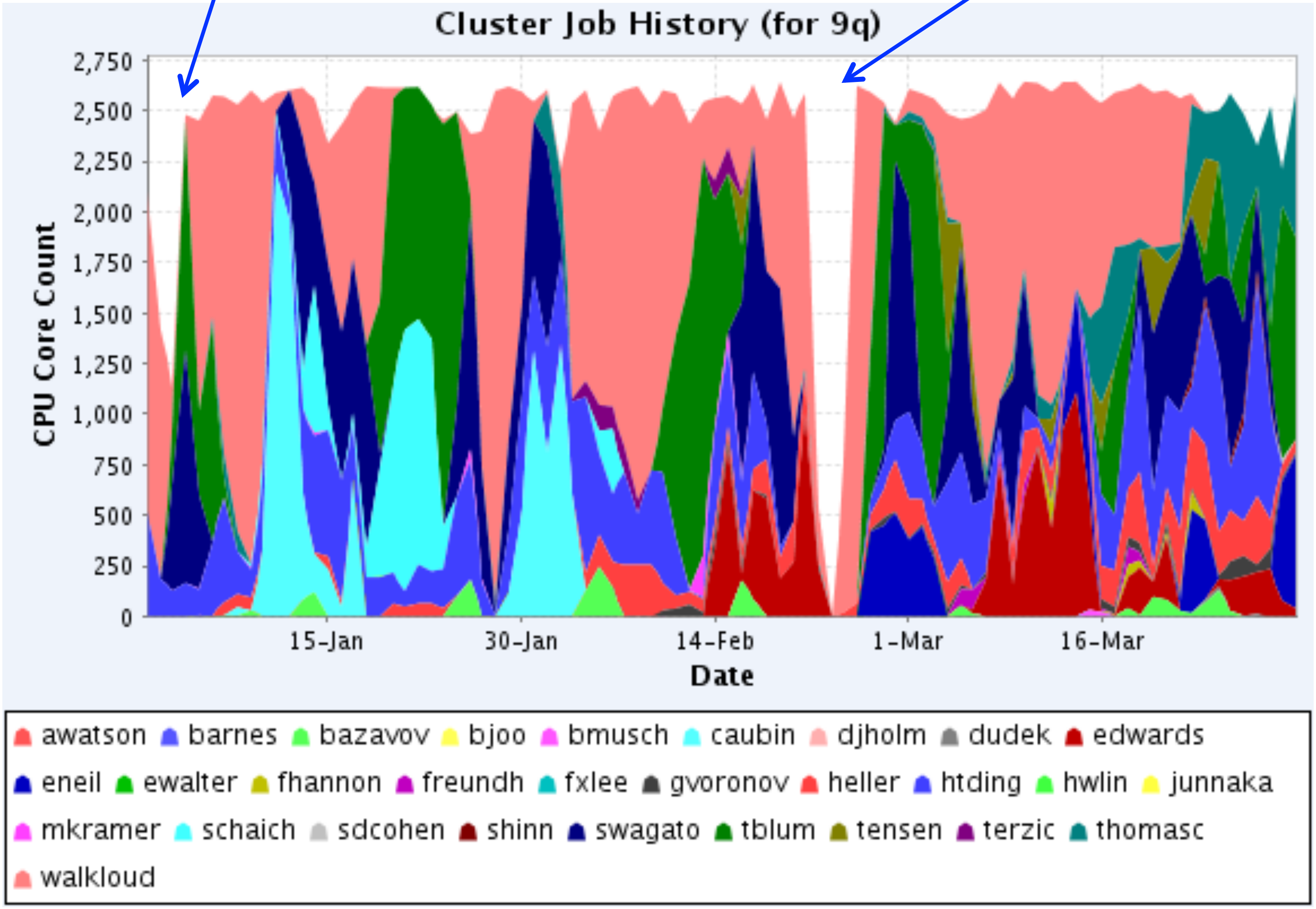
Both /cache and /volatile have notions borrowed from ZFS: project specific "reservations" and "quotas" that the daemon uses to decide what to delete.

Users may "pin" files to make them the last files eligible for deletion.

Infiniband Cluster Utilization

Re-set after Christmas power problems

Upgrade of Lustre to redundant meta-data server

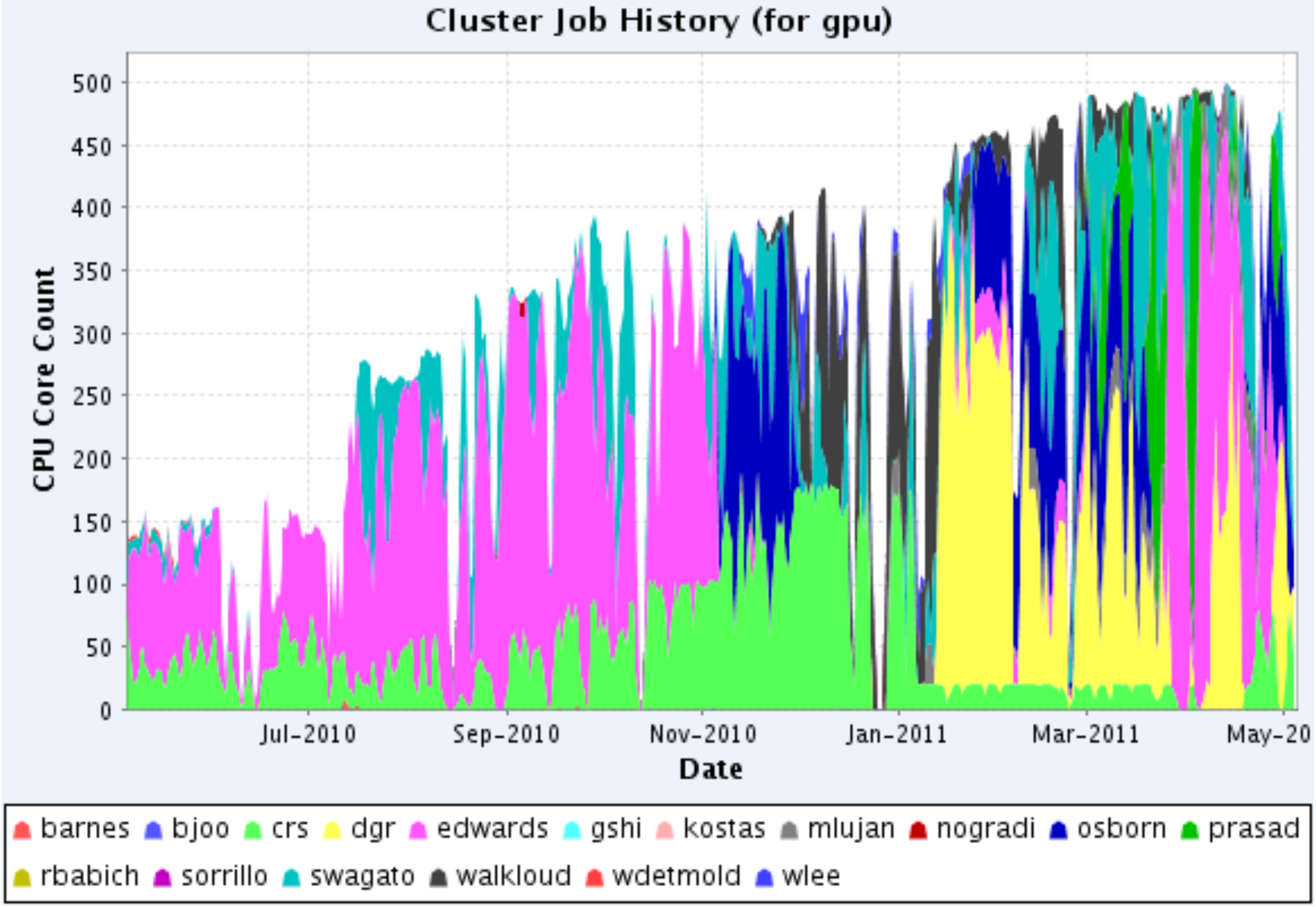


9q cluster recent utilization is very good (shown).

10q has apparent dips as nodes in 1 rack dynamically flip to GPU running (not shown here).

7n frequently has low utilization (not shown).

12 Month GPU Utilization



Our production GPU count has risen steadily over the last 12 months as the cluster was expanded (July) and as problems with the GTX480s were resolved (Jan).

GPU Utilization Advice

JLab has 5 flavors of GPU nodes, and if everyone ignores one flavor, utilization drops; use composite types to avoid this:

gpu queue:

4:Fermi	will select quad GTX-480 or C2050 or M2050
4:GT200b	will select quad GTX-285 or C1050 or S1070
16	will select 4 quads on the same IB switch

ibg queue:

256:GTX285	will select 32 nodes (8 cores), thus 32 GPUs (nodes are shared with ib queue, so they must advertise the same resource, namely CPU cores)

Additional info on available tags to select different node and GPU types is available online.

Summary

USQCD resources at JLab

- 14 Tflops in conventional cluster resources (7n, 9q, 10q)
- 63 Tflops of GPU resources

  (and as much as 100 Tflops using mixed half-single precision)

Challenges and Opportunities Ahead

- Continuing to re-factor workloads to put heavy inverter usage onto GPUs
- Finishing production asqtad, hisq and dwf inverters
- Increasingly using Fermi Tesla cards with ECC memory to accelerate more than just the inverters (software challenge)