JLab Site Report

Bálint Joó

USQCD All Hands Meeting
Brookhaven National Laboratory
April 19, 2013

Compute Resources @ JLab

- Installed in 2012
  - 12s Cluster: 276 nodes (4416 cores)
    - 2 GHz Sandy Bridge EP, 32 GB memory
    - QDR Infiniband
    - 2 sockets, 8 cores / socket, AVX Instructions
  - 12k Kepler GPU Cluster: 42 nodes (168 Kepler GPUs)
    - 2 GHz Sandy Bridge EP + 4 x Kepler K20m GPUs, 128 GB Memory
    - FDR Infiniband
  - 12m Xeon Phi Development Cluster: 16 nodes (64 Phi-s)
    - 2 GHz Sandy Bridge EP + 4 x Intel Xeon Phi 5110P co-processors, 64 GB Memory
    - FDR Infiniband
  - Interactive node: qcd12kmi has 1 K20m and 1 Xeon Phi

Compute Resources @ JLab

	CPU	#cores/node	#nodes	#accelerators/ node	IB	Memory/node
12s	Xeon E5-2650 (SNB) **2.0 GHz**	2 x 8	275	0	QDR	32 GB
12k	Xeon E5-2650 (SNB) **2.0 GHz**	2 x 8	42	4 NVIDIA K20m	FDR	128 GB
12m	Xeon E5-2650 (SNB) **2.0 GHz**	2 x 8	16	4 Intel Xeon Phi	FDR	64 GB
11g	Xeon E5630, (Westmere) 2.53 GHz	2 x 4	8	4 NVIDIA 2050	QDR	48 GB
10g	Xeon E5630, (Westmere) 2.53 GHz	2 x 4	53	4 Mixture	DDR/QDR	48 GB
9g	Xeon E5630, (Westmere) 2.53 GHz	2 x 4	62	4 Mixture	DDR/QDR	48 GB
10q	Xeon E5630, (Westmere) 2.53 GHz	2 x 4	224	0/1 NVIDIA 2050 in some nodes	QDR	24 GB
9q	Xeon E5530 (Nehalem) 2.4 GHz	2 x 4	328	0	QDR	24 GB

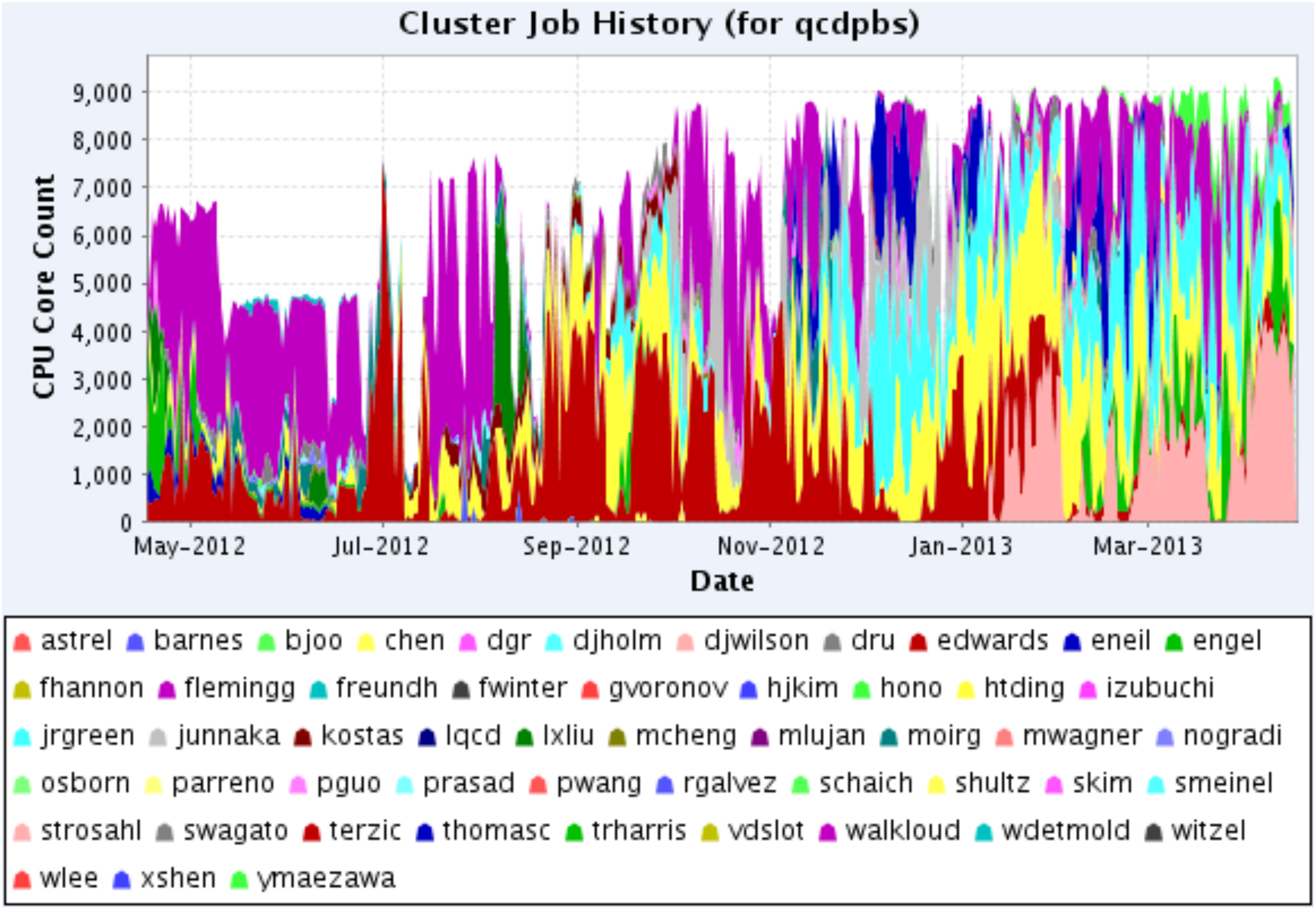
New Documentation page: https://scicomp.jlab.org/docs/?q=node/4

GPU Selection

	GTX285	GTX480	GTX580	C2050	M2050	K20m	Other
9g	108	45	95				
10g	28	66	10	108			
10q				10			6
11g				24	8		
12k						164	4
Total	136	111	105	142	8	164	10
Online	132	111	105	138	4	160	5

Online: as on 3/17/13
This table can be found at: http://lqcd.jlab.org/gpuinfo/

Utilization



CPU Project Utilization

(Core hours for each cluster are converted to 12-13-core hours based upon measured relative performance)

Project Name	Proj. Allocation	Proj. Used Hours	Annual Pace	Monthly Pace	Hour Remaining	Overused
NPLQCD	30,590,000	19,245,805	79%	63%	11,344,195	0
Spectrum	27,520,000	29,975,380	137%	147%	0	2,455,380
thermo	19,200,000	17,637,579	116%	97%	1,562,421	0
Structure	16,220,000	12,769,421	99%	153%	3,450,579	0
isoClover	2,000,001	1,690,476	106%	0%	309,525	0
SMhisq	1,910,000	1,033,716	68%	433%	876,284	0
emc	1,800,001	1,681,246	118%	0%	118,755	0
TMD	1,750,000	1,418,038	102%	541%	331,962	0
strangeness	1,200,001	851,550	89%	174%	348,451	0
spinpol	1,190,000	1,191,661	126%	468%	0	1,661
BFN	50,001	32	0%	0%	49,969	0
Total	103,430,004	87,494,902	106%	124%	18,392,142	2,457,041

NB: This plot can be found ‘live’ on the web:

http://lqcd.jlab.org/lqcd/maui/allocation.jsf

GPU Project Utilization

(GPU Hours are weighted by node's performance and the cost)

Project Name	Proj. Allocation	Proj. GPU Hours	Annual Pace	Monthly Pace	Hour Remaining	Overused
Spectrumg	1,521,000	1,391,049	115%	183%	129,951	0
thermog	1,170,000	1,106,819	119%	210%	63,181	0
Structureg	672,000	421,902	79%	99%	250,098	0
NPLQCDg	405,000	120,887	38%	10%	284,113	0
discogpu	110,000	112,504	129%	0%	0	2,504
gwuQCD	82,400	0	0%	0%	82,400	0
DwfHVPg	80,002	4,041	6%	1%	75,961	0
HotQCDg	50,001	0	0%	0%	50,001	0
Total	4,090,403	3,157,202	97%	145%	935,705	2,504

Note: Annual Pace appears low and Monthly pace high due to a large capacity and allocation increase in January.

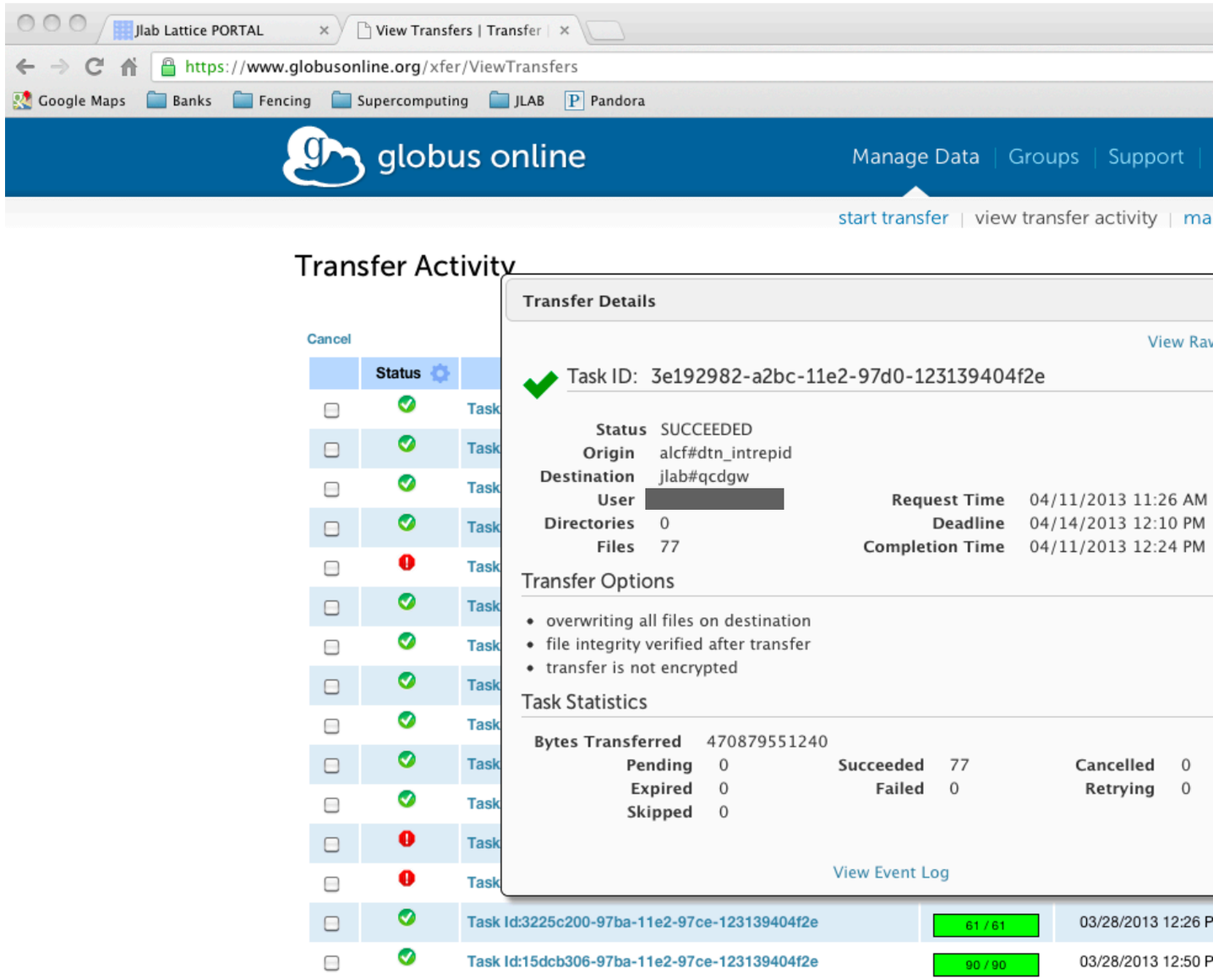
NB: This plot can be found ‘live’ on the web:

http://lqcd.jlab.org/lqcd/maui/allocation.jsf

Globus Online

- Globus Online has been deployed in production
- Endpoint is jlab#qcdgw
- Can also use Globus Connect to transfer data to/ from laptops off-site
- Whitelisting no longer needed
- No certificates needed (JLab username and password)
- Sign up at :

http://www.globusonline.org



Choice of Hardware Balance

- "How is the balance of hardware (e.g. CPU/GPU) chosen to ensure that science goals and community are well served?"
  - Before GPUs relatively few cluster design decisions needed much user input (mainly memory/node)
  - Project level purchases are coordinated with Executive Committee, budget level decisions are vetted by DOE HEP & NP program managers.
  - Balance of resources based on input from PIs of relevant largest class A allocations, and considerations for allocations for the year. SPC provides oversubscription rate.
  - Informal consultations with experts and 'site local' projects
  - With current diversity of available resources (GPU/MIC/BGQ, "regular" cluster nodes etc) perhaps more input will be needed from users, EC and SPC.

Accelerators/Coprocessors

Bálint Joó

USQCD All Hands Meeting
Brookhaven National Laboratory
April 19, 2013

Why Accelerators

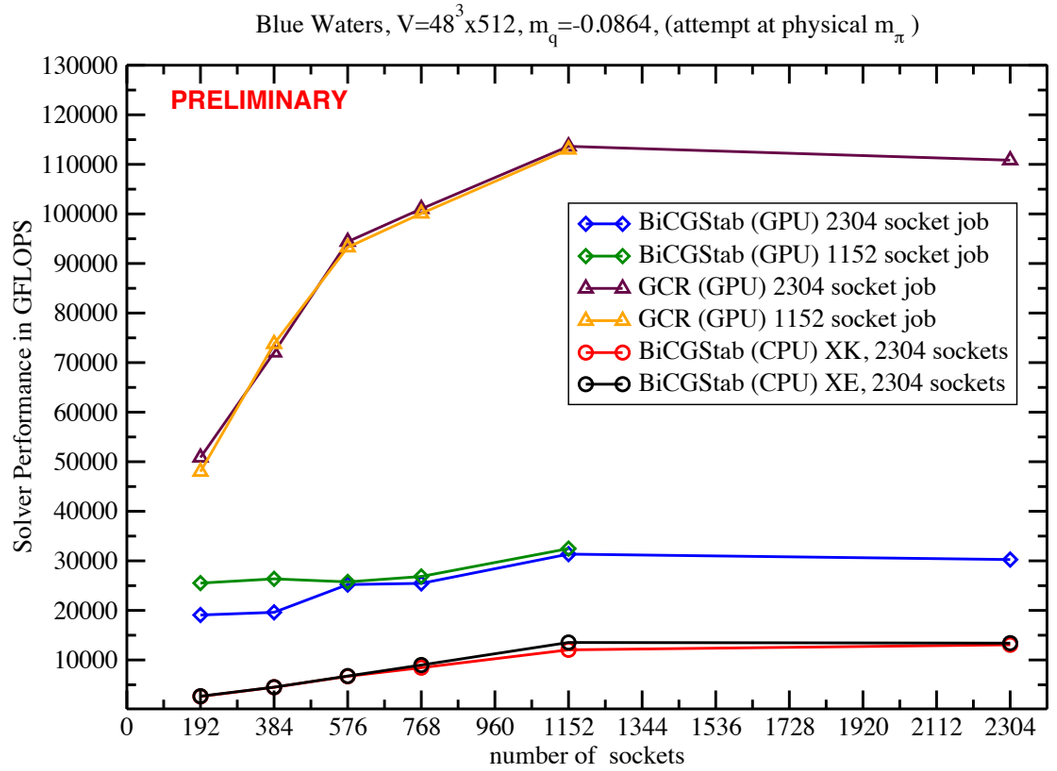
- We need to provide enough FLOPS to complement INCITE FLOPS on leadership facilities,
  - At capacity level & within $$$ constraints
  - Power Wall: clock speeds no longer increase
  - Moore's law: transistor density can keep growing
  - Result: Deliver FLOPS by (on chip) parallelism
  - Examples: Many core processors e.g. GPU, Xeon Phi
  - Current packaging: is accelerator/coprocessor form
    - Hybrid Chips are coming/here: e.g. CPU + GPU combinations

Quick Update on GPUs

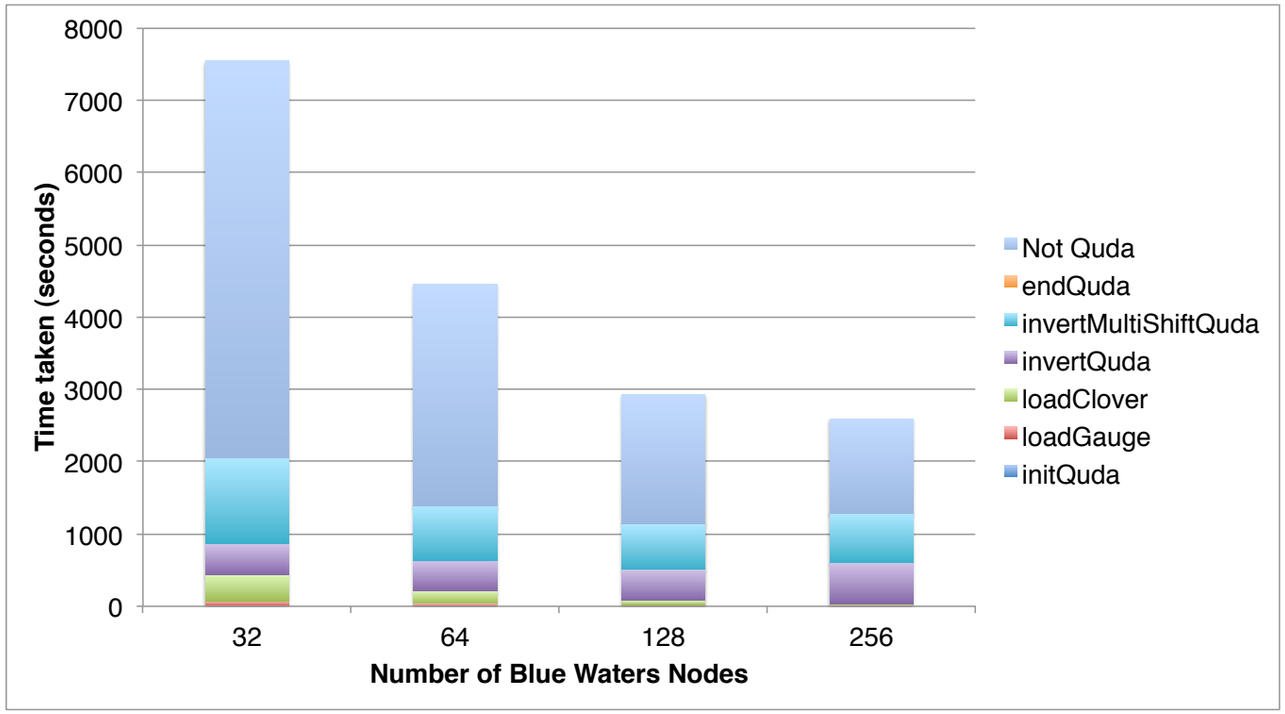
- GPUs discussed extensively last year
- Recently installed Kepler K20m GPUs in JLab 12k cluster
- 12k nodes have large memory: Host: 128 GB, Device: 6 GB
- Software:
  - QUDA: http://lattice.github.com/quda/ (Mike Clark, Ron Babich & other QUDA developers)
  - QDP-JIT & Chroma developments (by Frank Winter)
    - QDP-JIT to NVIDIA/C is production ready (interfaced with QUDA)
    - JIT to PTX is full featured, but needs some work to interface to QUDA
    - Makes Analysis and Gauge Generation, via Chroma, available on GPUs.
  - GPU enabled version of MILC code (Steve Gottlieb, Justin Foley)
  - Twisted Mass fermions in QUDA (A. Strelchenko)
  - QUDA interfaced with CPS (Hyung-Jin Kim)
  - Thermal QCD code (Mathias Wagner)
  - Overlap Fermions (A. Alexandru, et. al.)

GPU Highlights

- Chroma + QUDA propagator benchmark
  - up to 2304 GPU nodes of BlueWaters
  - 48^3x512 lattice (large), light pion
  - Speedup factors (192-1152 nodes)
    - FLOPS: 19x - 7.66x
    - Solver time: 11.5x-4.62x
    - Whole app time: 7.33x - 3.35x

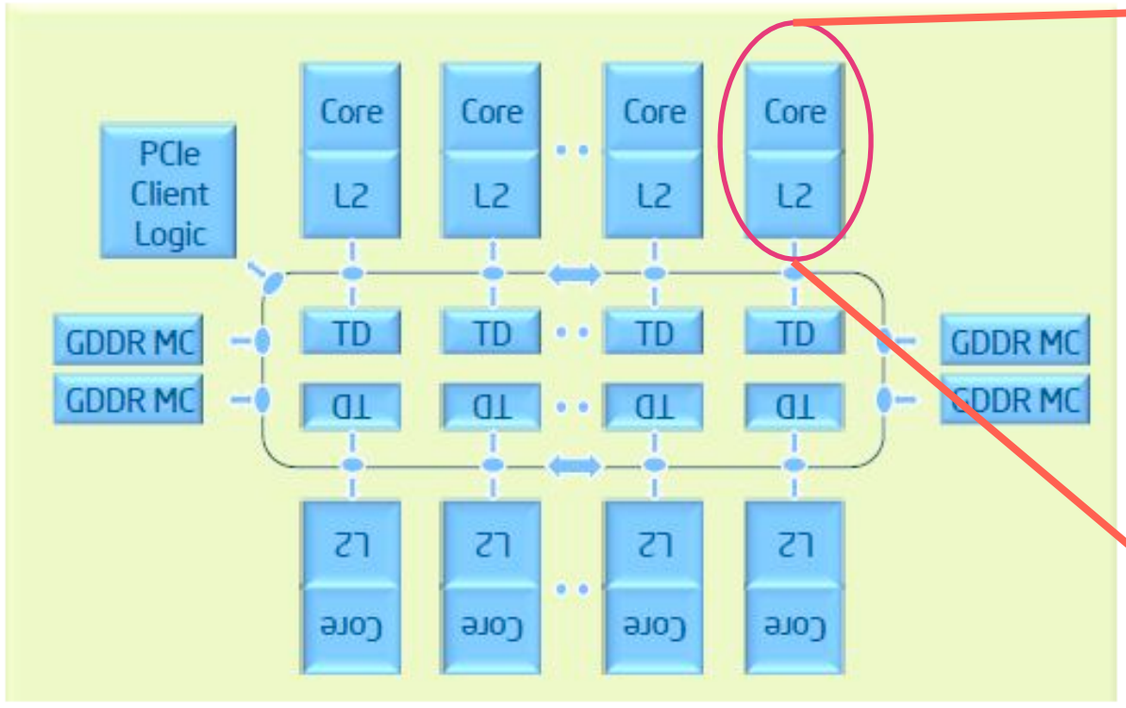


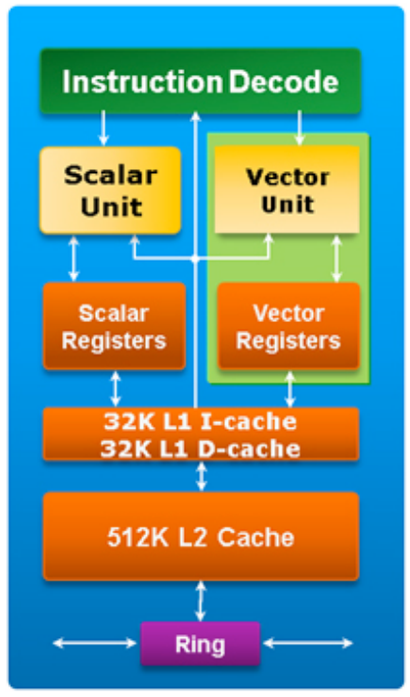
- Stout smeared, clover gauge generation with QDP-JIT/C+Chroma+QUDA
  - on GPU nodes of BlueWaters
  - 32^3x96 lattice (small), BiCGstab solver
  - BiCGStab solver reached scaling limit
  - expect better solver scaling from DD+GCR (coming soon)



Xeon Phi Architecture

- Xeon Phi 5110P (Knights Corner) - 60 cores, 4 SMT threads/core
- Cores connected by ring, which also carries memory traffic
- 512 bit vector units: 16 floats/8 doubles
- 1 FMA per clock, 1.053 GHz => 2021 GF peak SP (1010 GF DP)
- L2 cache is coherent, 512K per core, “shared” via tag directory
- PCIe Gen2 card form factor





Images from material at:

http://software.intel.com/mic-developer

Xeon Phi Features

- Full Linux O/S + TCP/IP networking over PCIe bus
  - SSH, NFS, etc
- Variety of usage models
  - Native mode (cross compile)
  - Offload mode (accelerator-like)
- Variety of (on chip) programming models
  - MPI between cores, OpenMP/Pthreads
  - Other models: TBB, Cilk++, etc
- MPI Between devices
  - Peer 2 Peer MPI Calls from native mode do work
  - Several Paths/Bandwidths in system (PCIe, IB, QPI, via Host...)
  - Comms speed can vary depending on path

Programming Challenges

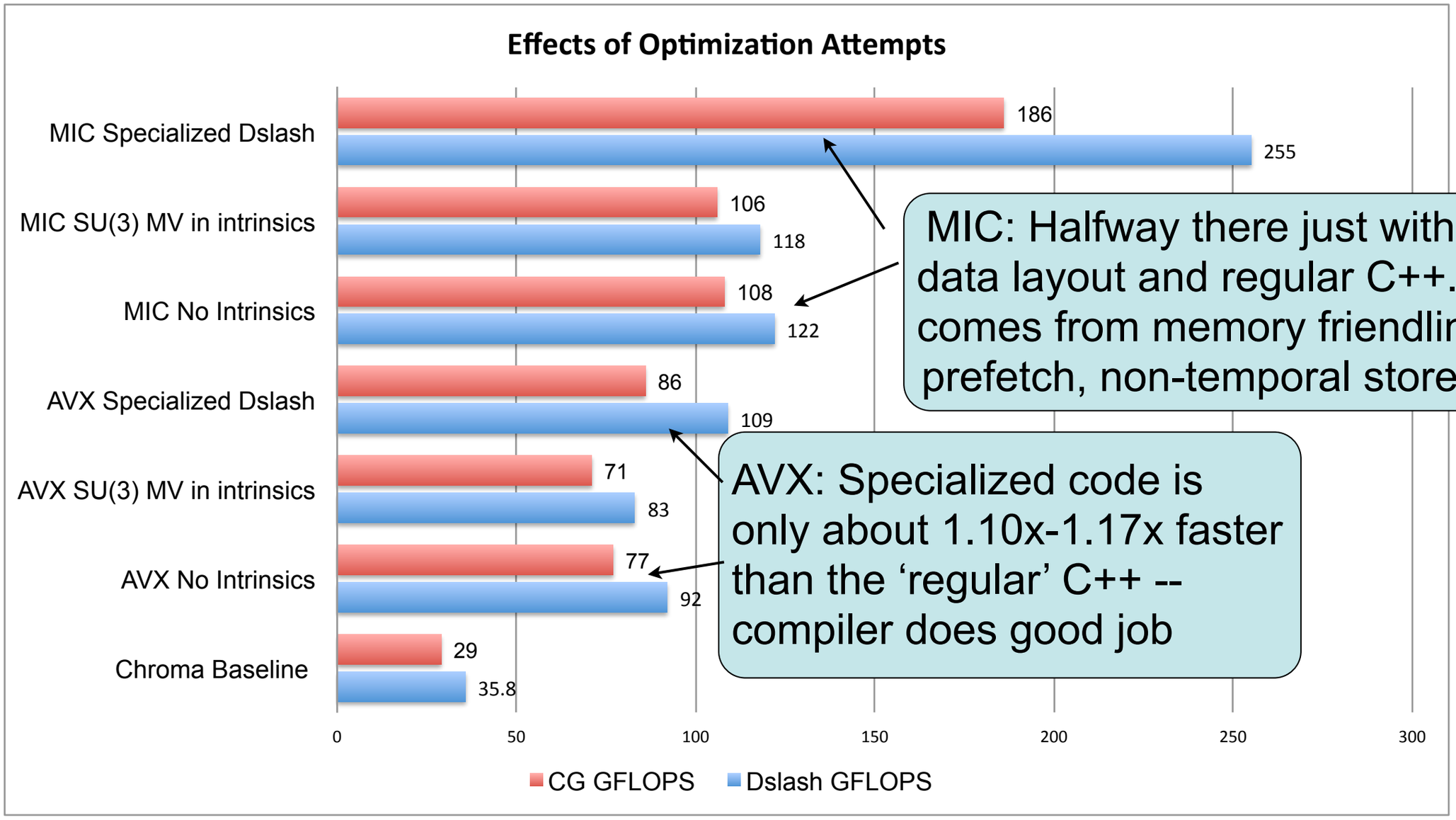
- Vectorization: Vector length of 16 maybe too long?
  - vectorize in 1 dimension: constraints on lattice volume
  - vectorize in more dimensions: comms becomes awkward
  - vector friendly data layout is important
- Maximizing number of cores used, maintaining load balance
  - 60 cores, 59 usable. 59 is a nice prime number
  - Some parts have 61 cores, 60 usable, more comfortable
- Minimize bandwidth requirements:
  - exploit reuse via caches (block for cache)
  - compression (like GPUs)
- KNC needs software prefetch (for L2 & L1)

Relation to other platforms

	Xeon Phi	"Regular" Xeon (Sandy Bridge)	GPU	BG/Q
"Vectorized" data layout	Yes	Yes	Yes	Yes
Explicit vectorization	Yes	Yes	No (This is good)	Yes
Blocking	Yes	Yes	Yes (shared memory)	Yes
Threading	Yes	Yes	Yes (Fundamental)	Yes
Prefetching/ Cache management	Yes	less important (Good H/W prefetcher)	less important, (small caches)	Maybe (HW prefetcher + L1P unit)
MPI + OpenMP (MPI+Pthreads) available	Yes	Yes	No	Yes

Thesis: Efficient code on Xeon Phi should be efficient on Xeon and BG/Q as well (at least at the single node level)

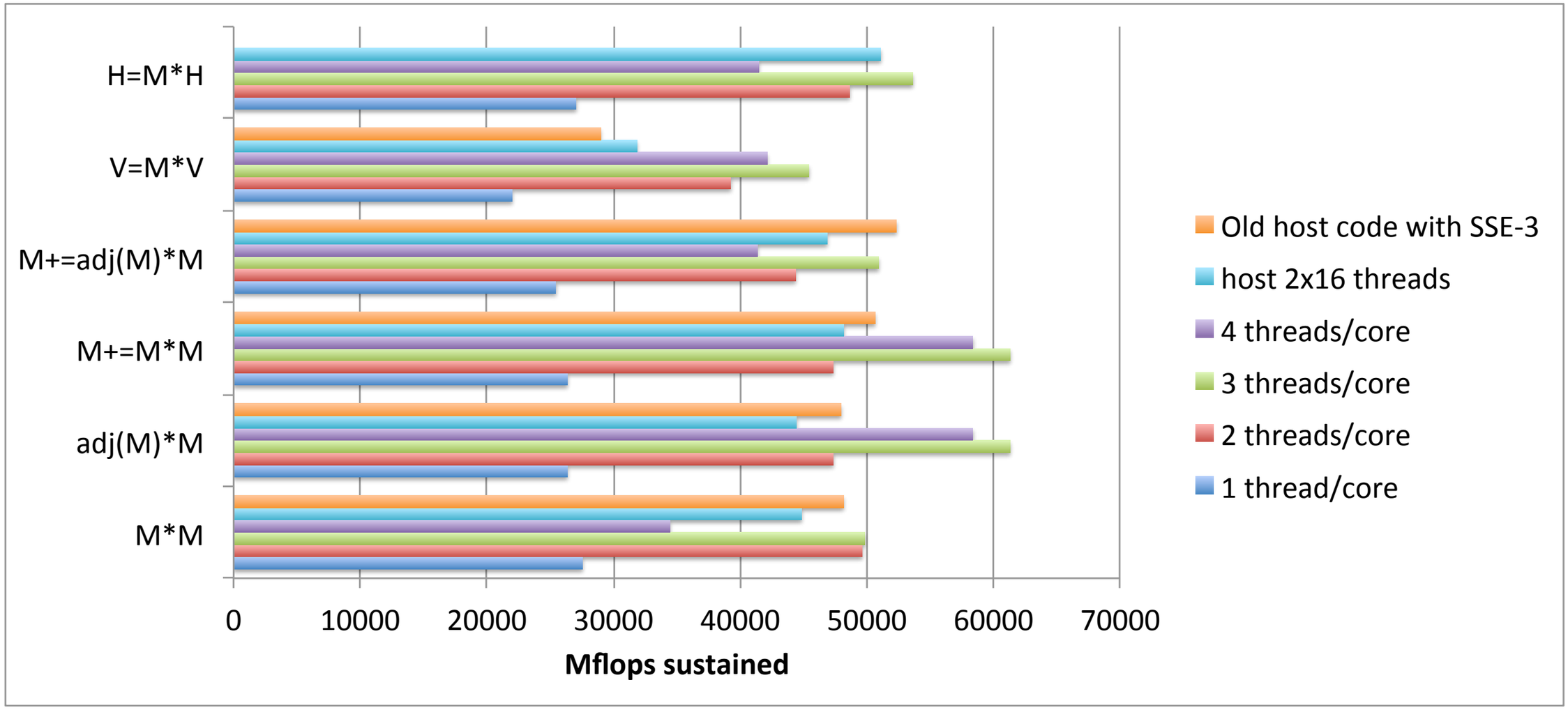
Ninja code vs. non Ninja code



Status in Nov 2012

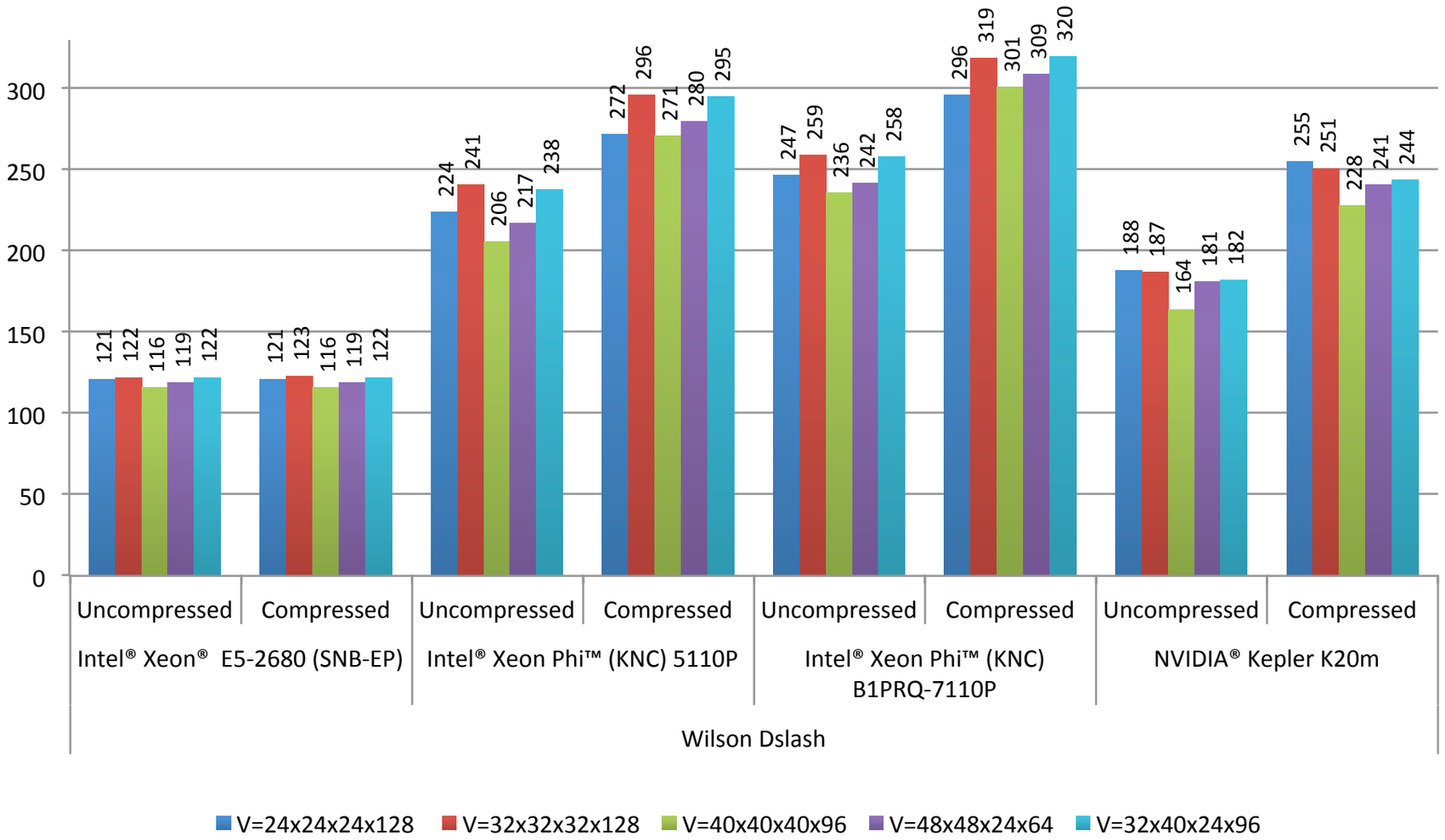
Production Xeon Phi 5110P, Si Level: B1, MPSS Gold, 60 cores at 1.053 GHz, 8GB DDR5 at 2.5GHz, with 5 GT/sec
Only 56 cores used. Lattice Size is 32x32x32x64 sites, 12 compression is enabled for Xeon Phi results, except for the 'MIC No Intrinsics case'. Xeon Phi used large pages and the "icache_snoop_off" feature.
Baseline and AVX on Xeon E5-2650 @ 2 GHz, I used the ICC compiler from Composer XE v. 13.

Optimizing QDP++



- QDP++ ‘parscalarvec’ - work by Jie Chen
  - vector friendly layout in QDP++
  - Single Xeon Phi comparable to 2 SNB sockets (no intrinsics, no prefetch)
  - parscalarvec intrinsic free host code comparable to SSE optimized host code

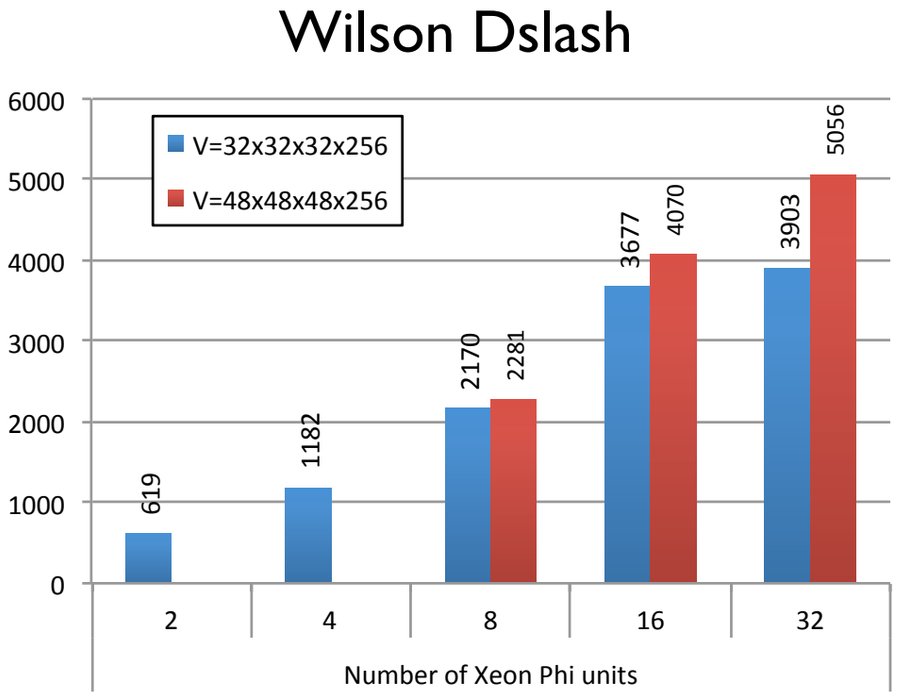
Ninja Code: Wilson Dslash

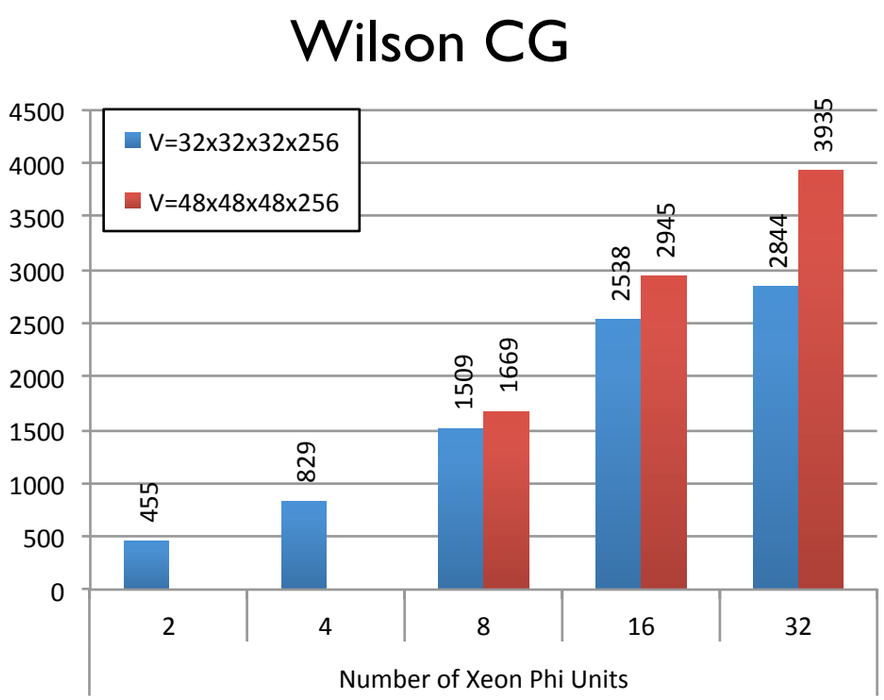


From: B. Joo, D. D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnani, V. W. Lee, P. Dubey, W. Watson III "Lattice QCD on Intel(R) Xeon Phi(tm) Coprocessors", Proceedings of ISC'13 (Leipzig) Lecture Notes in Computer Science Vol 7905 (to appear),

- Blocking scheme maximises #of cores used
- SOA layout with tuned 'inner array length'
- CPU Performance is excellent also (used 2.6 GHz SNB)
- Here Xeon Phi is comparable to 4 sockets

Multi-Node Performance





- 2D Comms only (Z&T)
  - vectorization mixes X & Y
- Intel Endeavor Cluster
- 1 Xeon Phi device per node
- MPI Proxy:
  - pick fastest bandwidth path between devices (via host in this case)
  - similar to GPU strong scaling at this level (expected)

From: B. Joo, D. D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnani, V. W. Lee, P. Dubey, W. Watson III, "Lattice QCD on Intel(R) Xeon Phi(tm) Coprocessors", Proceedings of ISC'13 (Leipzig) Lecture Notes in Computer Science Vol 7905 (to appear),

Clover Progress

- Two forms of clover operator: $(A^{-1}_{ee} D_{eo})$ and $(A_{oo}-D_{oe})$
- Use to construct EO operator:
  - $A_{oo}-D_{oe}A^{-1}_{ee}D_{eo}$
- Single node ops coded and pass correctness test
- Still need to perform prefetching optimizations
- as of 3/15/2013,
  - $(A^{-1}_{ee}D_{eo})$ operator ~100 GF (SP) (with 2 row compress)
  - $(A_{oo}-D_{oe})$ operator ~ 143 GF (SP) (with 2 row compress)
  - EO Preconditioned CG ~ 125-133 GF (SP)
- Prefetching, AVX version, Multi-node is work in progress

Summary

- Increasing parallelism is industry trend - driven by power constraints
- Xeon Phi: a many core CPU
  - Ninja code on Xeon Phi is competitive with Ninja code on GPU
  - Xeon Phi will compile and run your non-Ninja code today
  - **But no free lunch:** need to invest effort for performance
- Unlocking all levels of parallelism takes some effort
  - multiple cores, multiple threads per core, short vectors
  - Currently we have “Ninja Gap” (also on GPUs and BG/Q)
  - Threading + vectorized layout already brings benefits
- Payoff: Performance portability (at least on single node)
  - Excellent performance on Xeon
  - Expect good (single node) performance from BG/Q too
- JLab 12m cluster is ideal development resource