KNL Experience at BNL

Omnipath, KNL, and Slurm from an admin perspective

70 YEARS OF DISCOVERY

A CENTURY OF SERVICE

W. Strecker-Kellogg
A. Zaytsev
C. Caramarcu
T. Wong

HEPiX Spring 2017
Budapest

U.S. DEPARTMENT OF ENERGY

BROOKHAVEN NATIONAL LABORATORY

Timeline / Overview

1. Fall 2016-BNL acquires KNL machine
2. Winter 16/17-debugging hardware issues
3. February - March 2017, debugging performance & stability
4. April 2017-"beta" status, converging on final configuration

- 142-node KNL cluster deployed in RACF/SDCC data center since 2016Q4:
    - Single Intel Xeon Phi CPU 7230: 64 physical / 256 logical cores @ 1.3 GHz
        - (1.5 GHz maximum in turbo mode) per node
    - Dual Intel Omni-Path PCIe x16 HFI cards (non-blocking 200 Gbps)
    - Two level fat-tree single fabric Intel OPA interconnect system built out of 8x spine (core) + 14x leaf (edge) 48-port Intel Omni-Path switches
        - Bisection (unidirectional) bandwidth for compute nodes alone is about 14 Tbps = 1.7 TB/s
        - All the OPA uplinks are done with passive copper cables (630 uplinks total in 5 racks)

Knight's Landing Architecture

- Manycore architecture
- On-die vs. Coprocessor
- 4xHT/core, AVX512 (F, CDI, ERI, PFI subsets)
  - Advanced prefetch instruction extensions
- Requires significant code optimizations
  - Vectorization
  - NUMA--depending on memory mode (next slide)
- 2d-Mesh Interconnect on Die
  - MCDRAM off die
- 64 physical cores @ 1.3 GHz(base) / 1.5 GHz (max turbo), 256 threads
- Software: Intel Compiler and memkind library, others...

Memory Modes

16 GB MCDRAM Modes

- 3-D Stacked DRAM
  - Allowing much higher-bandwidth (multi-channel) access from cores
- Operating Modes
  - Flat
    - Single address space
    - Visible to kernel, 2 NUMA Nodes
  - Cache
    - MCDRAM acts as extended cache
    - Invisible to OS
  - Hybrid
    - Carve out x% as cache, (1-x) as usable

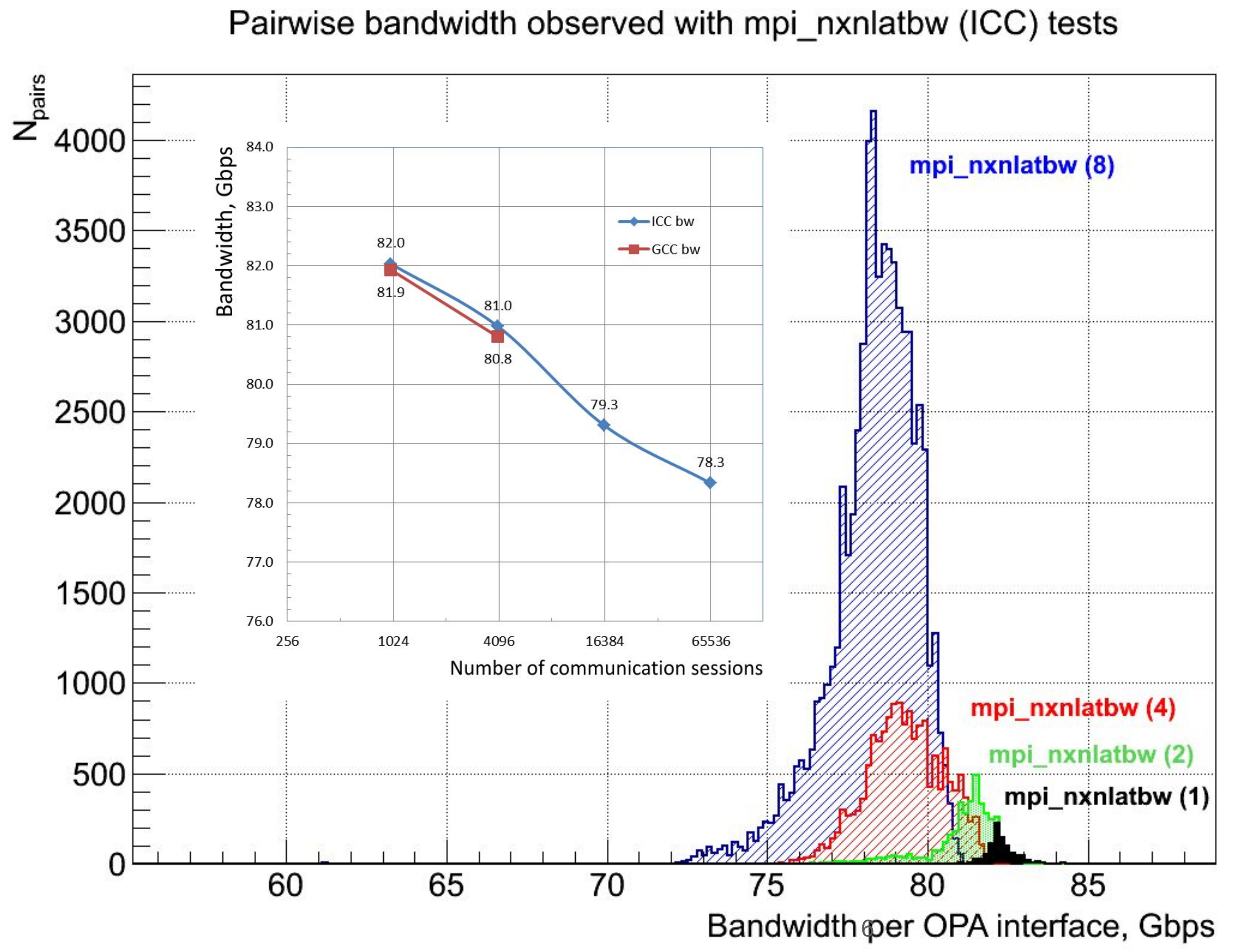
Memory Access Modes

- All-to-All
  - Addresses hashed uniformly across processor grid
- Quadrant
  - Split die into 4, hash into same quadrant
    - Sacrifices homogeneity for improved latency & bandwidth
- Sub-NUMA Clustering (SNC)
  - Quadrants exposed as numa nodes with cost matrix
  - Fastest if used correctly
  - NUMA optimization needed in code

OmniPath (OPA) Interconnect

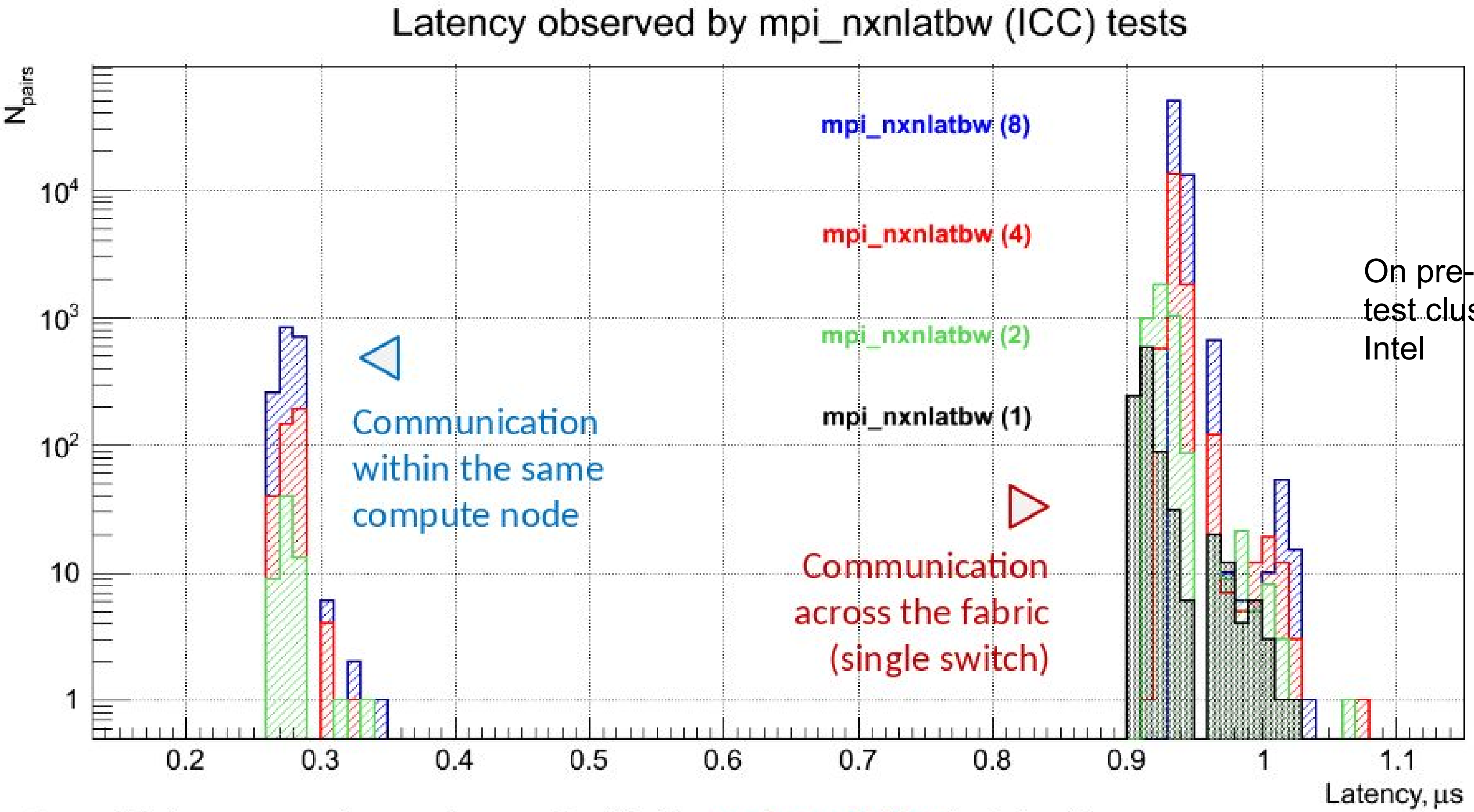
- OPA can be considered as a direct rival of the Mellanox 4X EDR IB technology (ConnectX-4 product line) with the following features giving the OPA a competitive edge:
  - Better price/performance due to the significant cost reduction on the client side (HFI cards)
  - Slightly better endpoint-to-endpoint latency
  - Higher achievable MPI message rate (for small messages and applications that can efficiently utilize such a high message rate)
  - Higher maximum number of ports on a single non-modular switch (48 vs 36 of EDR IB)
  - Additional transport / congestion control features both in hardware and software stack (Fabric Manager) that should be particularly useful for the HPC storage systems such as GPFS and Lustre
  - More fabric survivability features (partial deactivation/slowdown of the problematic uplinks is supported)
  - Software stack is fully integrated with Intel Performance Scaled Messaging 2 (PSM2) library (Intel provided MPI builds and OPA specific benchmarks in particular)

Intel Cluster Bandwidth Benchmarks



On pre-production Xeon-based test cluster from Intel

Intel Cluster Latency Benchmarks



On pre-production test cluster from Intel

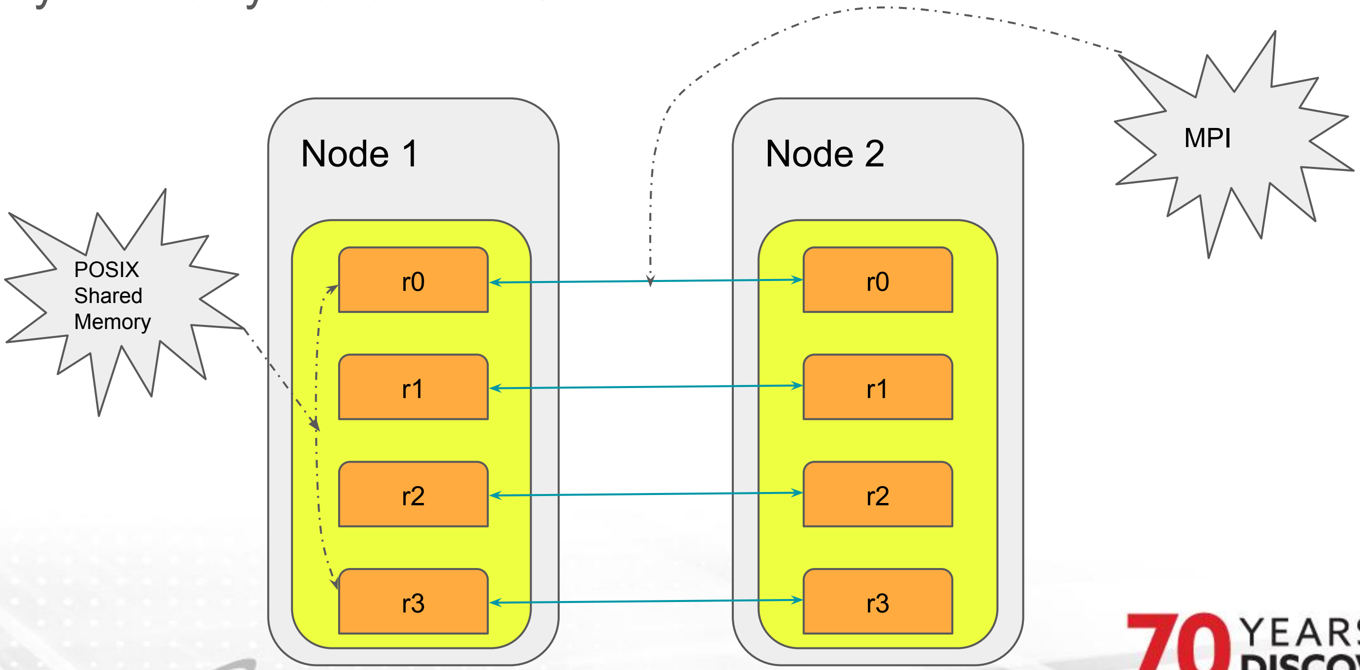
Overall latency spectrum observed with the mpi_nxnlatbw test for the cases of 1, 2, 4 and 8 test processes running on every compute node in the cluster

OPA Performance Limitations

- Kernel Driver (hfi1) implementation "feature"
  - The current version of the driver is implemented in such a way that limits the maximum combined bandwidth achievable through a single client side OPA port (or a set of ports for the multi-rail installations) for the MPI jobs utilizing a low number of ranks per compute node.
  - Bottleneck is maximum compute power of a single physical core on the client
    - Not an issue on Xeon-based systems where 1 core is sufficient to drive full bandwidth
    - One MPI rank on KNL core with max turbo is unable to drive a single OPA interface (2 are sufficient)
      - Means 4 ranks / node to drive both OPA interfaces
  - Intel is aware of this problem; significantly impairs the bandwidth-limited scientific applications requiring maximum unidirectional bandwidth between compute nodes, such many LatticeQCD codes
  - Pure software problem; solution from intel is expected Q3-4 2017
  - A workaround for certain kinds of LQCD codes is already developed and contributed to the Grid library https://github.com/paboyle/Grid by P. Boyle (School of Physics and Astronomy, University of Edinburgh).

LQCD Solution

1. 4 MPI Ranks per Node
   a. First tried monkey-patching MPI to intercept local calls and replacing with
   b. Using POSIX Shared Memory calls for communication, eliminates a second memcpy() operation between same-node ranks
2. Communities other than LQCD are eagerly awaiting a fix from Intel
3. Affects only KNL-only nodes with OPA



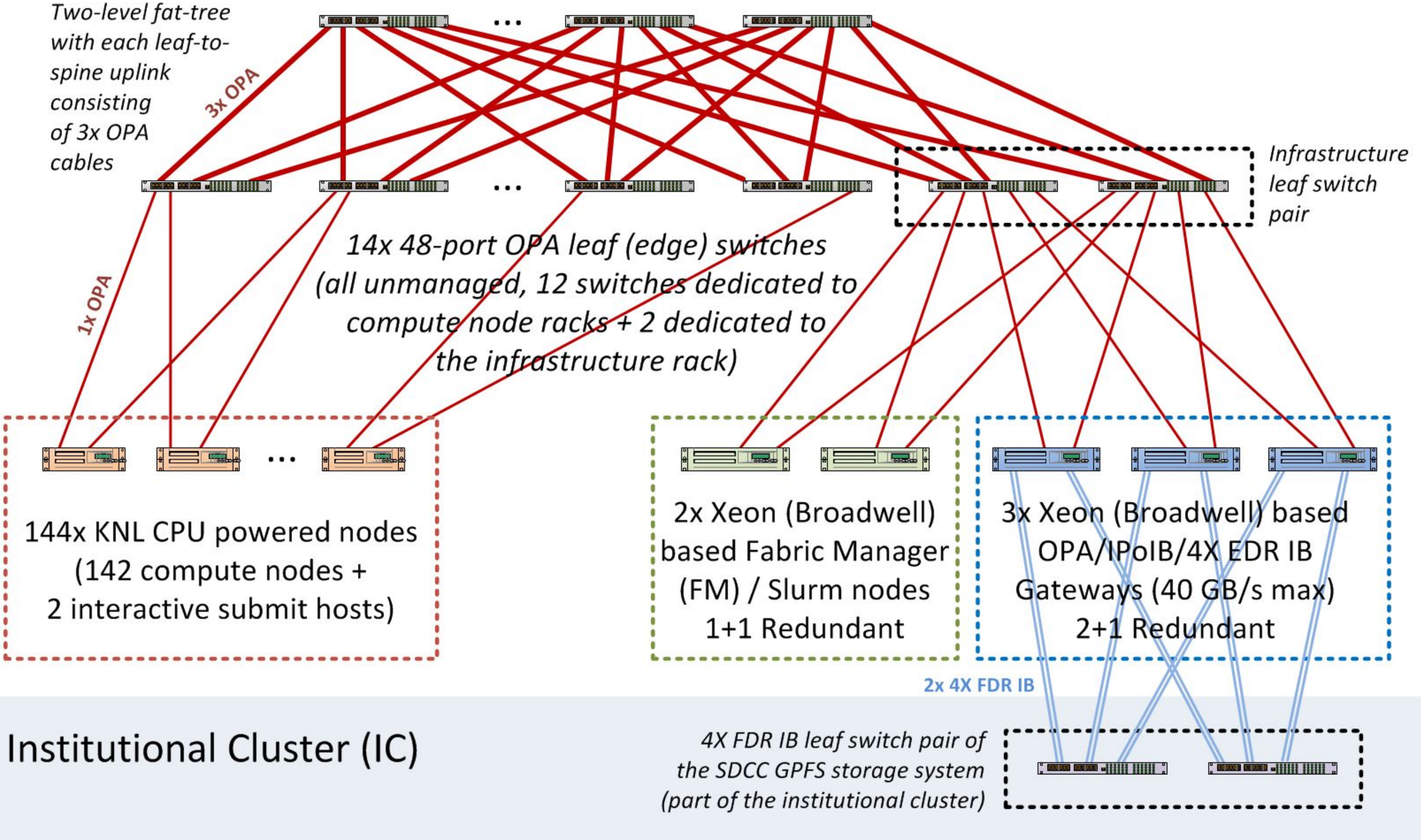
Fabric Manager

- OPA Fabric Manager on managed switches
  - The Fabric Manager (FM) on the switches are only capable of scaling up to 128 client interfaces in the fabric (which is significantly lower than the Subnet Manager scalability limit of 648 compute nodes of the Mellanox 4X EDR IB managed switches), which prompts the need for a dedicated Fabric Manager nodes
  - Additional cost and OPA ports on the fabric must be reserved for these FM nodes in all clusters with more than 128 client interfaces in the fabric
- Issue using the redundant pair of Fabric Managers in the dual-rail single-fabric fat-tree OPA fabric with non-default subnet prefix values
  - The easy workaround is to stay with the default prefix and disable the MPI false alarm warning messages about the single physical fabric being a requirement for such a configuration

GPFS Integration (OPA/IB Gateway)

- GPFS Storage exists for Institutional Cluster (IC)
    - Infiniband connected storage, RDMA
    - Access only through IB fabric
- Integration with OPA fabric
    - 3 nodes acting as gateway
        - Load balancing between them via route configuration
    - Each 2xOPA, 4xIB, forwarding between
    - Many sysctl and other tweaks for performance tuning
    - Single Node performance
        - 1.2 GB/s single thread seq. reads, 1.4 GB/s single thread seq. Writes
        - 5.5 GB/s of aggregate unidirectional I/O bandwidth in multiple threads
    - Aggregate Performance
        - About 20GB/s through gateways
        - Less than GPFS storage capability
            - A good thing for now, given primary use is IC cluster

KNL Cluster



8x 48-port OPA spine (core) switches (2 managed + 6 unmanaged)

Two-level fat-tree with each leaf-to-spine uplink consisting of 3x OPA cables

14x 48-port OPA leaf (edge) switches (all unmanaged, 12 switches dedicated to compute node racks + 2 dedicated to the infrastructure rack)

Infrastructure leaf switch pair

144x KNL CPU powered nodes (142 compute nodes + 2 interactive submit hosts)

2x Xeon (Broadwell) based Fabric Manager (FM) / Slurm nodes 1+1 Redundant

3x Xeon (Broadwell) based OPA/IPoIB/4X EDR IB Gateways (40 GB/s max) 2+1 Redundant

Institutional Cluster (IC)

4X FDR IB leaf switch pair of the SDCC GPFS storage system (part of the institutional cluster)

*see A. Zaytsev CHEP 2016 Talk

Hardware Maintenance Experience

- Initial assembly of cluster identified a few CPUs needing replacing, done in November 2016
- Delayed shipping of additional switches. Rewiring fabric after cable tray switch and still awaiting final cable runs
- Deployment of BIOS updates to current version in December 2016
  - BIOS Version
    - S72C610.86B.01.01.0231.101420161754
  - BMC Firmware
    - Op Code 0.28.10202, Boot Code 00.07
  - ME Firmware Version
    - 03.01.03.018
  - SDR Version
    - SDR Package 1.22

Stability and Performance Under Various Kernels

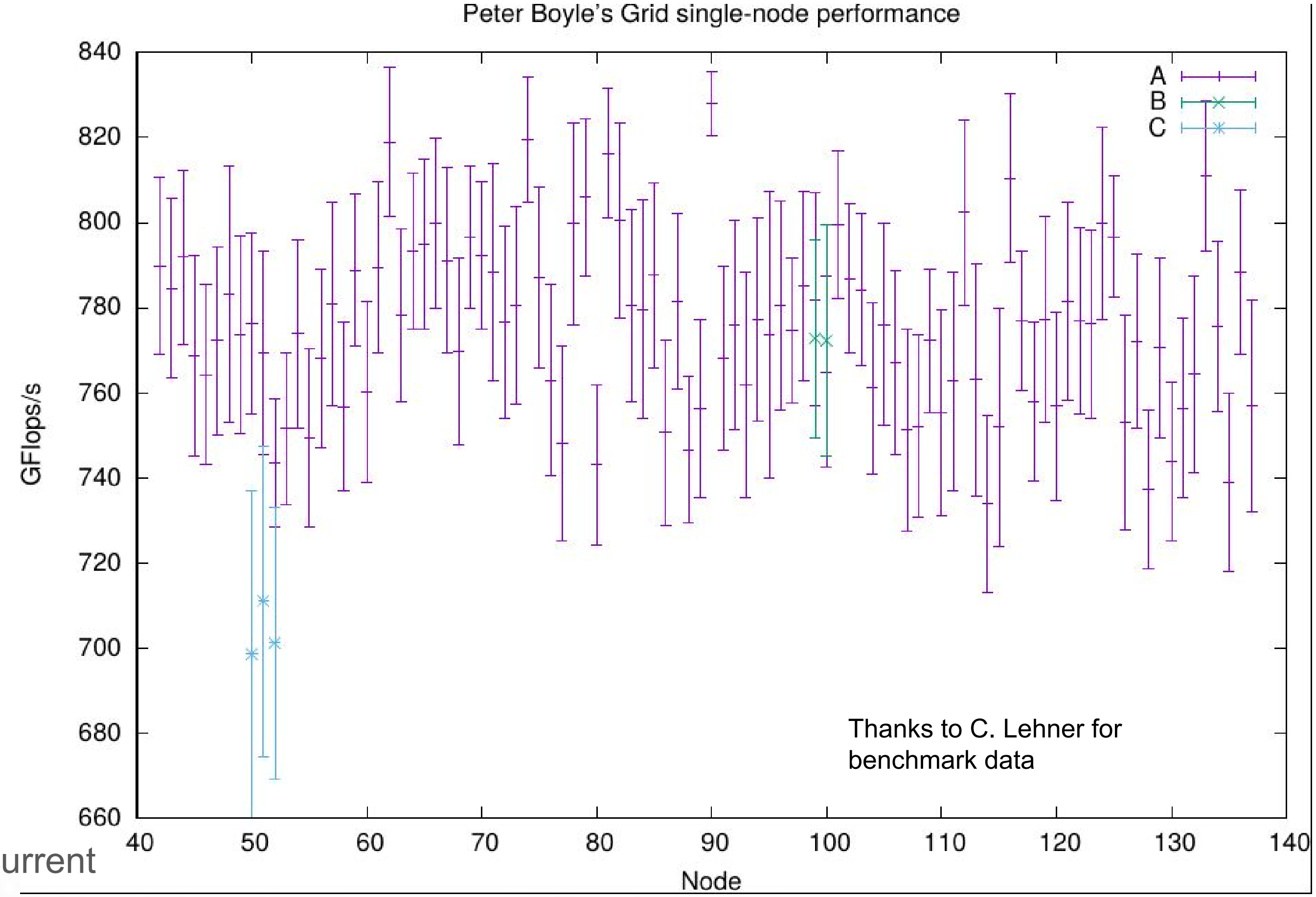
- Image shipped with system was RHEL7.2 kernel from Nov 2015
  - Kernel-3.10.0-327.el7.x86_64.rpm
  - IntelOPA-Basic.RHEL72-x86_64.10.3.0.0.81
  - xppsl-1.5.0
  - Performs as expected
- Our RHEL7.2+ image with Sept. 2016 kernel
  - Kernel-3.10.0-327.36.1.el7.x86_64.rpm
  - 10-20% single-node performance reduction!
- Vanilla 4.8.10 kernel w/ RHEL specfile
  - Same 10-20% single-node performance reduction
- RHEL7.3 image (current)
  - Performance back to where stakeholders expect
  - kernel -3.10.0-514.16.1.el7.x86_64
  - IntelOPA-Basic.RHEL73-x86_64.10.3.1.0.22
  - xppsl-1.5.1

Building and Provisioning

- Intel hfi1 driver and OPA software available in tarballs from intel.com
- Tarballs contain packages, but also a 15kloc perl ./INSTALL script
  - Intel says: "do not extract RPMs yourself"
- Ugly puppet installation (alternative was from kickstart)
  - exec[download-tarball] { creates=>dir notify=>Exec[install] }, exec[install] { refreshonly=>true }

- First puppet run can take 40+ minutes (yum 1 pkg at a time!)
  - Single threaded tooling, ~1GHz CPU, like running a Pentium III
- Multi-stage run, puppet creates IPoIB interfaces and updates hfi1
  - Reboot, then GPFS can start (uses IPoIB)

Single Node Job Performance



A = 3.10.0-327
C = 3.10.0-327.36
B = 3.10.0-514.16.1 <-- current

Multi-Node Job Performance

Most recent (7.3) configuration

Good weak-scaling (scale problem size with job size) of LQCD Grid code

- 16 nodes: 300 Gflops/s/node
- 32 nodes: 300 Gflops/s/node
- 64 nodes: 280 Gflops/s/node
- 128 nodes: 250 Gflops/s/node

Thanks to C. Lehner for benchmark data

Stability and Mode Switching Issue

- Switching cache modes dynamically is supported with Slurm (see here)
  - Automatically runs BIOS syscfg util to switch and reboots into job
    - Issue ensuring Slurm starts after GPFS is *really* available
  - Manually switching to fit queue is not being considered as too manpower-intensive
- However...
  - Our tests showed about 1/2 of rebooted nodes would be unstable
  - After discussions with Intel:

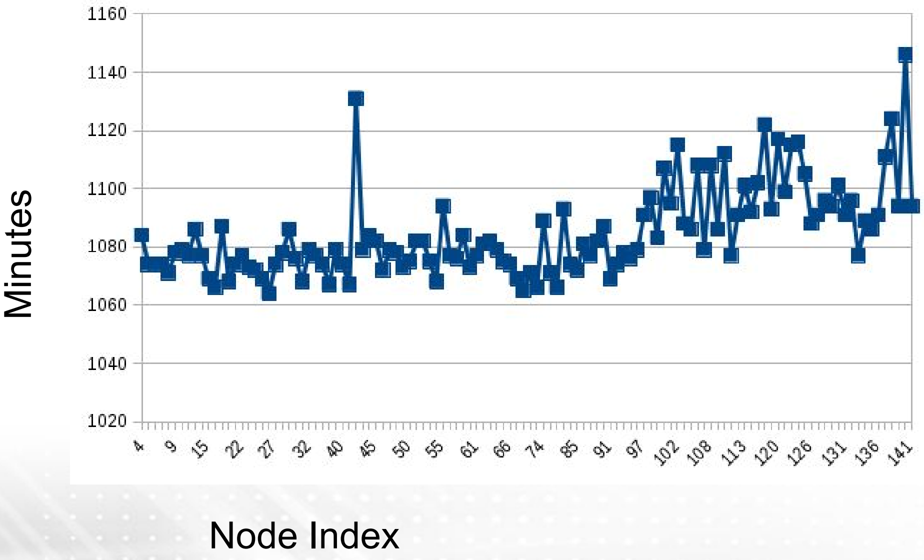
> *"If you don't powercycle between mode switches you get garbage"*

- Well...
- But... BIOS Update that fixes this issue is due out soon^{TM}
  - Hopefully in a few weeks

Benchmarking Other Applications

ATLAS Code

- Identical sample jobs, separate work directories, finished on >98% of nodes
  a. Scratch space on GPFS
- Job processes 4k events in ~18 hours
  a. Avg: 1084.6 min



HS06 Basic

- 64 proc -> 208 HS
- 256 proc -> 320 HS

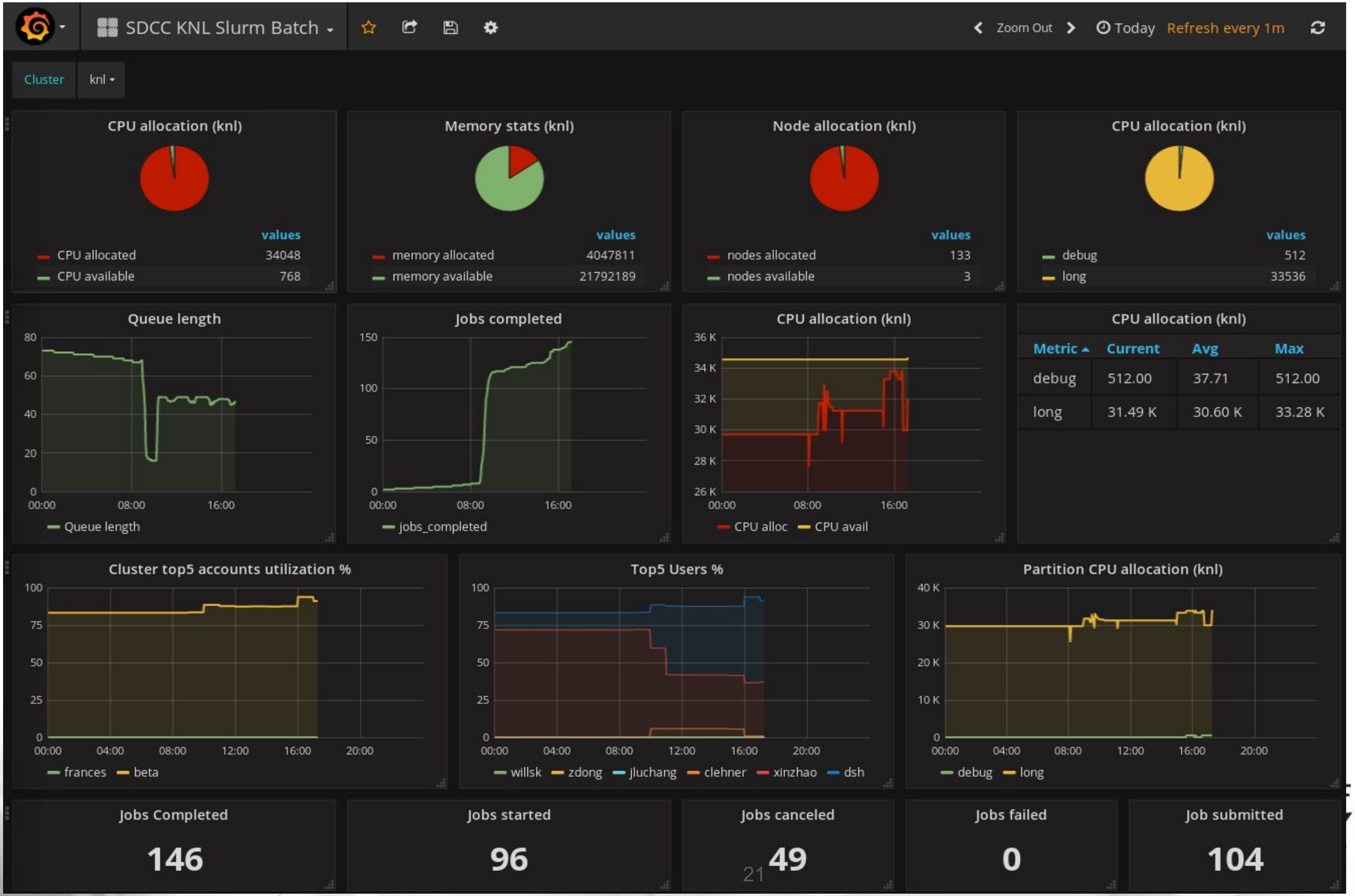
Vectorization x86_64 HS06

- GCC 5.3.0 on SL7 auto-vectorization:
  `-march=knl -mavx512f -mavx512cd -mavx512er -mavx512pf`
  ○ 64-bit on Broadwell, +10%
- 64 proc -> 315 HS (+50%)
- 256 proc -> 451 HS (+40%)

KNL Slurm Monitoring

- Custom Scripts dumping data to our graphite instance, displaying with Grafana
- Monitors Slurm cluster & queue state and per-allocation and per-user data
- Gathering done via simple script using both pyslurm bindings and screen-scraping the CLI tools
- Welcome discussion of LQCD-wide monitoring (Fifemon-like)

Grafana Status Page



CPU allocation (knl)

	values
🟥 CPU allocated	34048
🟩 CPU available	768

Memory stats (knl)

	values
🟥 memory allocated	4047811
🟩 memory available	21792189

Node allocation (knl)

	values
🟥 nodes allocated	133
🟩 nodes available	3

CPU allocation (knl)

	values
🟩 debug	512
🟨 long	33536

CPU allocation (knl)

Metric ▴	Current	Avg	Max
debug	512.00	37.71	512.00
long	31.49 K	30.60 K	33.28 K

Jobs Completed	Jobs started	Jobs canceled	Jobs failed	Job submitted
146	96	49	0	104

Conclusions

US LQCD group purchasing new cluster this Fall

Considering KNL, GPU (Pascal), or Skylake-based machines

Machine is finally reliably producing scientific results

Took more effort than expected to get there

For unoptimized code, Broadwell HS06/$ is 1.5x KNL

- Simple auto-vectorization gets to parity

Thank You!

Questions? Comments?

HEPiX Spring 2017
Budapest

U.S. DEPARTMENT OF ENERGY

BROOKHAVEN NATIONAL LABORATORY

70 YEARS OF DISCOVERY
A CENTURY OF SERVICE