

# *FY16 Hardware Acquisition*

*Chip Watson*

*LQCD-ext II Annual Review*

*June 28, 2016*

# Outline

## Acquisition Strategy

- Maximizing Science on a Portfolio of Machines
- Summer 2015 Alternatives Analysis

## Procurement Process

- Early RFI and feedback from vendors
- Benchmark selection

## New Hardware

- Overview of proposals
- Performance comparisons
- Configuration optimization
- Procured hardware

## Summary

# Portfolio Optimization

Optimize **the portfolio of machines** to get the most science on the **portfolio of applications**.

A single machine procurement is not driven solely by acquiring the greatest total benchmark suite result (i.e. we don't optimize just one machine).

Instead, for each procurement, the project optimizes its resources to yield the **best aggregate performance** for its **portfolio of applications** on its **portfolio of hardware**.

From 2009 to 2015, buying a combination of conventional and GPU clusters has produced the best hardware portfolio. Software availability and maturity has been a constraint on the expansion of the use of GPUs.

# Benchmarking LQCD

For more than a decade, machine performance for LQCD was measured by two key kernels:

- DWF (domain wall fermion) inverter (sparse matrix solver)
- Staggered inverter

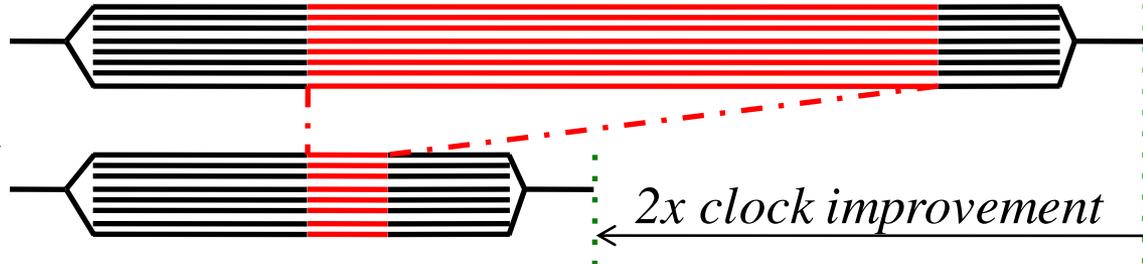
These kernels still represented a healthy (albeit shrinking) fraction of the flop/s used in LQCD. With conventional clusters and supercomputers, these kernels were very good predictors of application performance and clock time.

The first iteration of the LQCD Computing project used the average of these two as its benchmark, as a sort of Linpack for LQCD. We continue to track these two to see long range trends.

GPUs excelled at these kernels, but Amdahl's Law effects and the difficulty of porting code to CUDA precluded using GPUs exclusively in our portfolio.

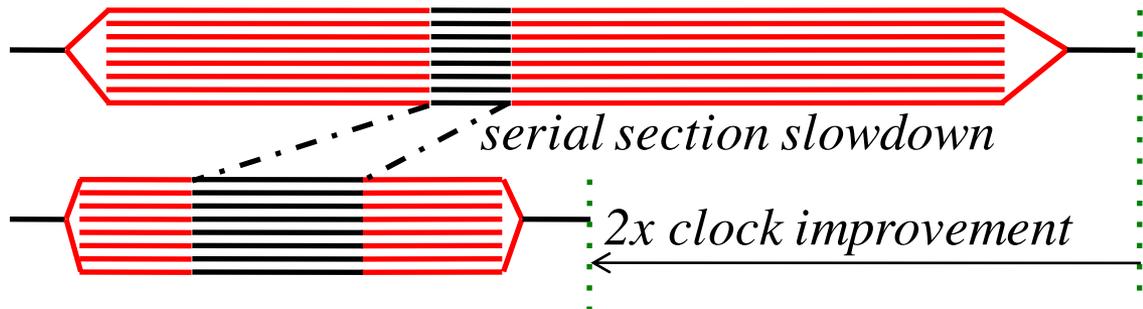
# Amdahl's Law Constraints

A major challenge in exploiting GPUs is Amdahl's Law: if 60% of the code is GPU accelerated by 6x, the net gain is only 2x.



Also disappointing in this scenario: the GPU is idle 80% of the time!

A similar impact happens when the clock speed is reduced as the parallelism of a homogeneous chip is increased.



Both of these effects (limited acceleration, and the slowdown of serial code on slower clocked but more parallel chips) make the hardware performance strongly dependent upon the application (i.e. harder to define), and forces us to do more careful benchmarking than was done in the previous decade.

# Effective Performance

The approach we have followed since 2009 to adjust for Amdahl's Law is to use “effective performance” for these advanced architectures, defined as:

*the performance of a conventional set of node(s) needed to achieve the same clock time on the same application*

Thus, if a quad GPU node gives the same performance (application clock time) as a cluster of 8 un-accelerated nodes, then we rate each GPU node for that application as 8x the performance of the un-accelerated node.

*This keeps the same units of “inverter flops”, while making the benchmarking process more application oriented.*

Further, multiple applications representative of the actual workload on the nodes are used so as to come up with an overall, average, effective performance number.

# Anticipated Hardware (Summer 2015)

## Conventional x86 Cluster

- ✓ Runs all software at least OK
- ✓ Easily integrated & used
- ✓ Most user friendly for development

## NVIDIA Pascal GPU Cluster

- ✓ High flop/s, high memory bandwidth
- ✓ Should run all existing GPU software

## Intel Xeon Phi / Knights Landing Cluster

- ✓ High flop/s, high memory bandwidth
- ✓ Might run most software at least OK

# Typical Configurations

## Conventional x86

- Dual socket, 16 core Xeon (64 threads), 64 GB memory
- Infiniband, 1:1 QDR or 2:1 FDR

## NVIDIA GPU

- Quad GPU + dual socket CPU (typical: host = 4x-6x GPU memory)
- on package high bandwidth memory
- fatter node therefore higher speed Infiniband, FDR or faster

## Intel Xeon Phi (KNL)

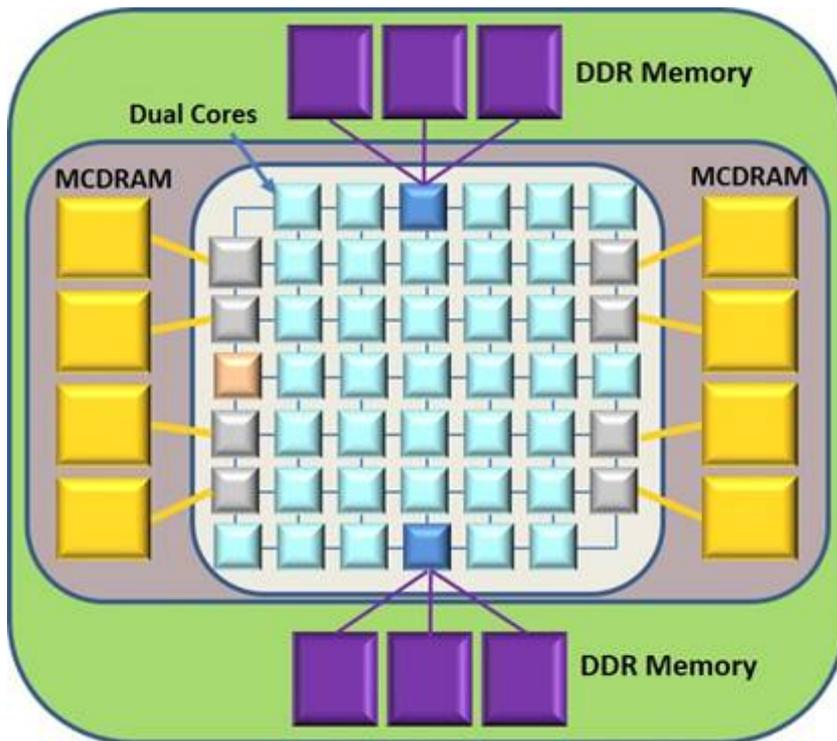
- single socket, 64+ core, 256+ threads, 512 bit SIMD
- 96 or 192 GB main memory (6 slots, “up to 384 GB”)
- on-package high bandwidth memory “up to 16 GB”

# NVIDIA Pascal

- Expected in 2016, hardware now starting to emerge
- Just announced at ISC 16:
  - PCle version of Pascal (P100) with no other links
- On package “3D Stacked Memory” with “up to 1 TB/s” bandwidth (*early numbers still NDA*)
- NVLink on chip (later version of chip)
  - 4 point to point bidirectional 100 Gb/s links per GPU, 5x faster than PCIe
  - Enables tight coupling of 4 GPUs within a node
  - Enables direct access to host memory for special hosts
- As for previous NVIDIA GPUs, key LQCD kernels have been optimized for Pascal by Kate Clark of NVIDIA (a former LQCD theorist) and show impressive performance

# KNL many core

Not an accelerator. Not a heterogeneous architecture. It is an x86 single socket node, with an improved core compared to KNC:



- ✧ Out-of-order execution
- ✧ Advanced branch prediction
- ✧ Scatter gather
- ✧ 8 on package MCDRAMs, up to 16 GB
- ✧ 6 DDR4 ports, up to 384 GB
- ✧ 1 MB L2 cache per 2 core tile (figure shows up to 72 cores if all are real & operational)

# JLab collaboration with Intel on Xeon Phi

Balint Joo key LQCD developer working closely with Intel:

- High Performance Dslash for Xeon Phi Knight's Corner, B. Joo, D. Kalamkar, M. Smelyanskiy, K. Vaidyanathan et. al. ISC'13 publication; Code developed into first release of QPhiX library
- Collaboration with Intel and U. of Regensburg, using QPhiX & code-generator for Domain Decomposition Preconditioner FGMRES-DR; S. Heybrock, T. Wettig et. al. published at SC'14
- Collaboration with NERSC and Intel for NESAP; Prepared QPhiX for KNL with: Thorsten Kurth (NERSC), D. Kalamkar (Intel PCL), Ashish Jha (Intel PCL), K. Vaidyanathan (Intel PCL), A. Walden (ODU), Presented at IXPUG Satellite Workshop of ISC'16.
- Continuing work with NESAP, loose collaboration with U. Regensburg on Multi-Grid

# KNL Readiness

USQCD Xeon Phi software maturity is growing:

- ◆ 2013 saw LQCD running at TACC / Stampede (Knights Corner), with an optimized Dirac inverter from Balint Joo's QPhiX code generator matching the performance of a contemporary GPU (NVIDIA k20)
- ◆ Developments are now under way on multiple codes by many in USQCD, driven by large future resources:
  - **Cori**, 2016; with **9,300+** chips,
  - Joined by ANL's **Theta** (KNL) in 2016; **2,500** chips, and
  - ANL's **Aurora** (KNH – Knights Hill) in 2018, with “**50,000 nodes**”
- ◆ Commodity hardware OEMs now offering KNL machines.

Conclusion: KNL is viable as an LQCD capacity resource in 2016.

# 2016 LQCD Procurement Benchmarks

Since applications see different gains on advanced chips, the procurement used several metrics to measure real science performance. The suite included

- Streams triad: LQCD is first and foremost memory bandwidth bound to vendors: “we don’t buy performance, we buy memory bandwidth”
- D-slash operator: a key piece in all variants of inverters, and sufficiently mature and optimized on Xeon Phi (QphiX code generator) to make it suitable for directly comparing KNL to GPU with both using guru code
- Robert Edwards’ contraction code: heavy use of batched zgemm, with a serial component and a dependence upon moving data into high bandwidth memory; a simplified benchmark was written for both GPU and KNL / x86

Extreme scalability, such as LQCD requires on capability machines for gauge configuration generation, is not a primary driver in USQCD procurements; the focus is more on analysis and high performance capacity mode running.

Although there was no network benchmark in the final suite, a 100 Gbps fabric was required for KNL and GPUs, and OmniPath was tested on both Broadwell and KNL clusters at Intel in advance of the fabric selection.

# FY16 Deployment Goal

The deployed performance goal for FY16, in units of *effective LQCD sustained performance* on historical solvers:

49 TFlops

This goal was projected from recent deployments of mixed GPU and conventional resources (recently 40% GPU, 60% conventional by cost, with the performance roughly 65% GPU, 35% CPU).

Our ability to deploy the most effective and high performance architectures is always constrained by the fraction of our applications' running times which exploit their higher performance per dollar.

This plus the need to support codes for which there is no CUDA code, precluded pure GPU accelerated node purchases in FY16.

# FY16 Procurement Timeline

July 2015 – Alternatives Analysis & Site Selection (done)

Aug 2015 – Review by Executive Committee (done)

Sept 2015 – FY16 budget finalization (done)

Oct 2015 – Detailed Acquisition Plan (done)

Nov 2015 – RFI (done end of Sept)

Due to unavailability of silicon, the following internal dates were intentionally slipped by 2 months without effecting the delivery milestone:

Jan 2016 – Benchmark Suite determination (done Mar)

Feb 2016 – Benchmarks frozen (done April)

Mar 2016 – RFP (done late April)

Apr 2016 – Award (done mid June)

May 2016 – Delivery & Acceptance Testing (expected Aug)

July 2016 – Early Running (expected Sept; allocations Oct 1)

# Proposals Sought

Two types of proposals were sought:

1. Homogeneous, self hosted KNL, with the largest available on-package memory, 96 GB main memory with an option for 192 GB, EDR or OmniPath fabric, over subscribed 2:1
2. 50:50 split by cost for a GPU plus conventional resource
  - a. Conventional reference design: x86 16 core, 64 GB memory, FDR or better, 2:1 over-subscribed fabric
  - b. GPU reference design: 4 GPUs, 256 GB with an option for 512 GB, EDR or OP fabric, single switch (at \$500K, less than than 34 nodes); quad K80, with 8 logical GPUs, was explicitly allowed.

Vendors were free to propose other designs & architectures.

GPUs were kept in the mix, especially since in mid 2016 the large ARRA funded GPU resources at Jefferson Lab would be de-commissioned, and there existed considerable mature software on that platform.

Three such *diverse platforms* made this the *most complicated procurement* done by the LQCD Computing project to date.

# Proposals Received (overview)

Four types of proposals were received:

1. KNL, all following the reference design
2. Quad K80 plus conventional (reference design)
3. Octal K80 plus conventional
4. Octal Pascal plus conventional

This last design was not really expected due to a 4<sup>th</sup> quarter anticipated earliest delivery. We explicitly allowed late delivery with a 2½% per month performance penalty, and with a requested end of September latest date, and 8 weeks after receipt of order as the baseline.

# Evaluation Summary

Notes: (1) Performance is an aggregate metric, not a per chip or per node metric: USQCD resources are primarily high performance high throughput, not capability.

(2) GPU and conventional resources were priced separately, and scaled to a nominal \$500K cost, and their performances added to yield a score for those proposals.

➤ *KNL proposals achieved the highest aggregate score on all performance metrics and sub-metrics.*

In addition to winning on performance metrics, KNL was shown to be capable of running non-guru MPI code at a better performance per dollar than conventional, validating the approach of a homogeneous KNL cluster vs. a 50:50 split for GPU + CPU.

# Performance

Metric	x86 (Broadwell) 32 cores	½ K80 1 GPU	KNL 64 cores
Streams triad	130 GB/s	180 GB/s	450 GB/s
d-slash	117 Gflop/s	235 Gflop/s	490 Gflop/s
<i>bicgstab*</i>	<i>110 Gflop/s</i>	<i>230 Gflop/s</i>	<i>380 Gflop/s</i>
batched zgemm batch: 64 - 1536	650-1050 GF	540 GF	780-1560 GF
C/C++*	Yes	---	Yes, 64 & 128 MPI processes

\* *Not procurement benchmarks*

# Normalized Performance per Dollar

Metric	x86 node 32 cores	quad K80 node	50:50 mix (required)	KNL
Streams triad	0.3	0.9	<b>0.6</b>	<b>1.0</b>
d-slash	0.25	1.0	<b>0.65</b>	<b>1.0</b>
bicgstab	0.3	1.1	<b>0.7</b>	<b>1.0</b>
batched zgemm batch: 64 - 1536	0.8 – 1.3	1.2	<b>1.0 – 1.3</b>	<b>1.0 – 2.0</b>
C/C++	1.0	-----	<b>0.5</b>	<b>1.0</b>

Conclusion: the GPU is still a viable and useful platform, but is limited in its reach without significant additional software development costs.\*\*\*

# Configuration Optimization

The following priorities drove the final optimized configuration:

1. Highest performance for  $2^n$  or  $3 \cdot 2^n$  nodes, so as to support lattices with factors of  $2^n$  in their dimensions, with an occasional 3 or 5 in the mix e.g.  $32^3 \times 128$ ,  $48^3 \times 96$  ...
2. Largest possible job memory footprint, especially for large Eigenvector, Deflation and All Mode Averaging
3. Fastest possible network (including 1 or 2 network links)

Optimizing on these 3 axes involves trading off one application's potential performance against another's. Doubling memory or network bandwidth is typically a 10% - 15% cost, thus lowers the performance available to applications that don't need these factors, while improving performance (or even capability) for those that do.

# Selected Configuration

- KNL cluster of 192 + 4 nodes
  - 3 \* 2<sup>6</sup> nodes, plus 4 to accommodate node outages
- FY 17 option to expand to 256 + 8 nodes
- 192 GB main memory / node
  - 36.8 TB on 192 node partition (matches cluster at FNAL)
  - 49 TB once upgrade to 256 nodes (modest growth and / or concurrent running of large and small jobs)
- OmniPath fabric (single)
  - 48 port leaf switches convenient for 2:1 fabric since 32 hosts will always consume ½ of messages within the switch
  - Three 48-port switches for the core, to support growth to 256 + 8 + multiple LNET routers for file system access

# Additional Components

1. Two Lustre LNET routers with OP and dual QDR Infiniband connections, to connect this cluster to Jefferson Lab's shared file systems at high bandwidth
2. Two new file servers (Lustre OSS), in an active-active configuration, each with two 12 Gbps SAS JBOD controllers (4 cables per OSS), connecting to two 40 disk arrays with dual attach backplanes, 8 TB disks, using open ZFS RAID z2 8+2, thus a total of 8 OSTs, 512 TB after raid overhead

This brings Jefferson Lab's Lustre system to 2.2 PB.  
(~1.2 LQCD, ~1.0 Experimental Physics, usable 80%)

# FY17 Expansion Option

1. KNL expansion to 256 + 8 nodes ( $2^8$  + spares)
2. Additional one or two OSS each with 256 TB after raid

Sept 2016 – Evaluation and Recommendation

Oct 2016 – Award

Dec 2016 – Delivery & Acceptance Testing (abbreviated)

Jan 2017 – Operations

These are working dates and may slip in the *unlikely* event of a continuing resolution.

# Summary

- Following a rigorous selection and optimization process, an Intel Xeon Phi “Knights Landing” cluster has been ordered which exceeds the KPI for FY16 delivered performance:

Goal: 49 TFlops

Procured: ~70 TFlops

- The resource will support both highly optimized codes as well as our large base of reasonably data parallel C and C++ codes, using either or both of openMP and MPI, **opening this advanced architecture’s performance gains to all of USQCD**
- The larger memory option as exercised will support the largest memory footprint now used on USQCD resources, and with the FY17 expansion will allow for even larger problems next year.