Improving the Performance of the MILC Code on Intel Knights Landing, An Overview

IXPUG 2017 Fall Meeting

September 26th – 28th, 2017

Texas Advanced Computing Center (TACC)

Austin, TX

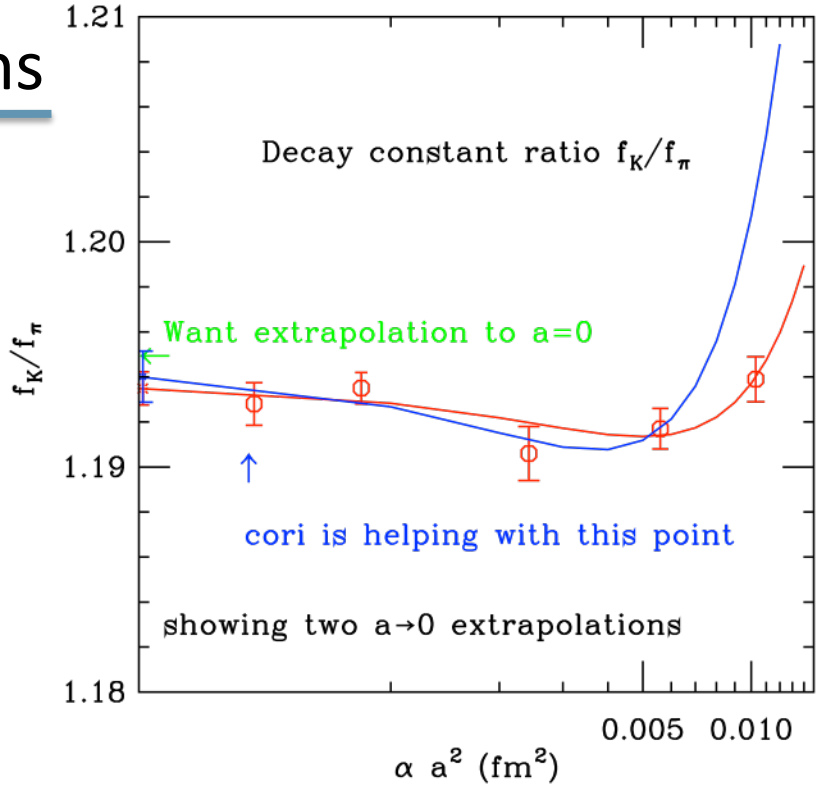
MILC on KNL Working Group

1. Indiana University: S. Gottlieb, R. Li
2. University of Utah: C. DeTar
3. University of Arizona: D. Toussaint
4. Intel: K. Raman, A. Jha, D. Kalamkar, M. Tolubaeva, T. Phung, R. Malladi
5. Tata Consultancy Services (TCS): G. Bhaskar, P. Gaurav, J. Bhat
6. Jefferson Lab: B. Joó
7. LBL/NERSC: D. Doerfler

- This effort is supported by Intel Parallel Computing Center at Indiana University
- And is a NERSC/NESAP Tier 1 code in the Cori Phase 2 (KNL) Project

Impact of MILC QCD Simulations

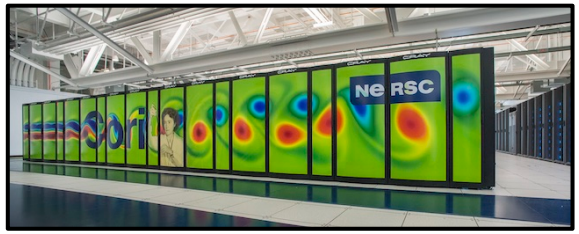
- Measuring the fundamental parameters of the Standard Model of particle physics
- And looking for deviations which suggest physics NOT accounted for,
  i.e. New Physics!
- Method is to use Monte Carlo evaluation of the quantum mechanical path integral
- Plot shows results achieved over the last several years, using resources from multiple facilities
- As the physical grid (lattice) spacing decreases, the computational complexity increases
- Cori is helping with the the calculations associated with a = 0.043 femtometers



Ratio of decay constants of the K meson to the Pion

Benchmarking Platforms

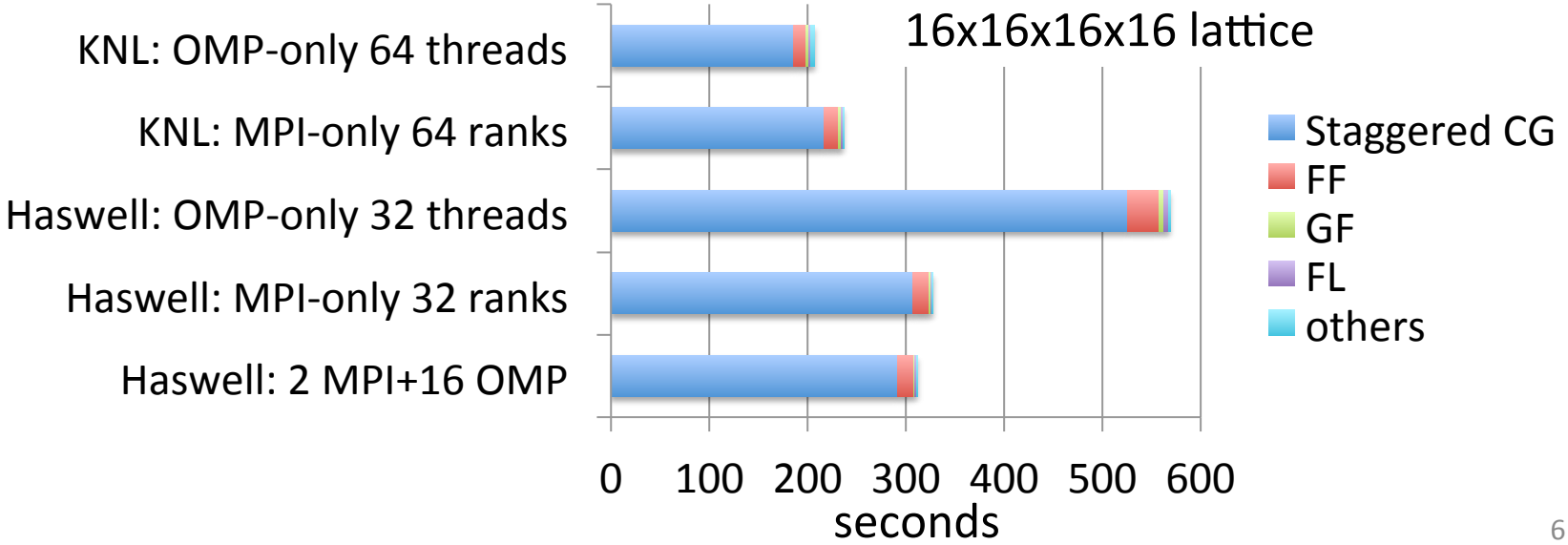
- Endeavor - Intel
  - Intel's internal development cluster
  - Knights Landing (KNL) and Sky Lake (SKX) nodes
  - Intel OPA high-speed interconnect
- Cori - NERSC
  - Cray XC-40 Architecture
  - Intel Haswell & KNL processors
- Edison - NERSC
  - Cray XC-30 Architecture
  - Intel Ivy Bridge processors
- Theta – Argonne National Lab
  - Cray XC-40 with KNL nodes
- Stampede – TACC
  - Dell, Intel and Seagate
  - Intel KNL with OPA interconnect



MILC computational phases: Optimizations & Performance

Where does MILC spend its time?

- Representative run time breakdown (single node)
- su3_rhmd_hisq application



Profiling Tools and Methods

- Intel VTune and Advisor
- Good ol' Linux prof and gprof
- Roofline analysis
  - e.g. used to look how close to the KNL MCDRAM BW roofline
- Integrated Performance Monitoring (IPM) tool
- And, MILC has extensive timing support, in particular functions known to be time consuming (time in sec. and GF/s reports) …
  - However, we found some timings didn't add up
    - Needing additional timing for code not represented
    - and this led to routines getting OpenMP that were assumed to be trivial
  - E.g. Update_u() calls from the "trajectory update"
    - Function speedup after threading was 42.5x
    - resulting in an overall 12% improvement to the trajectory update

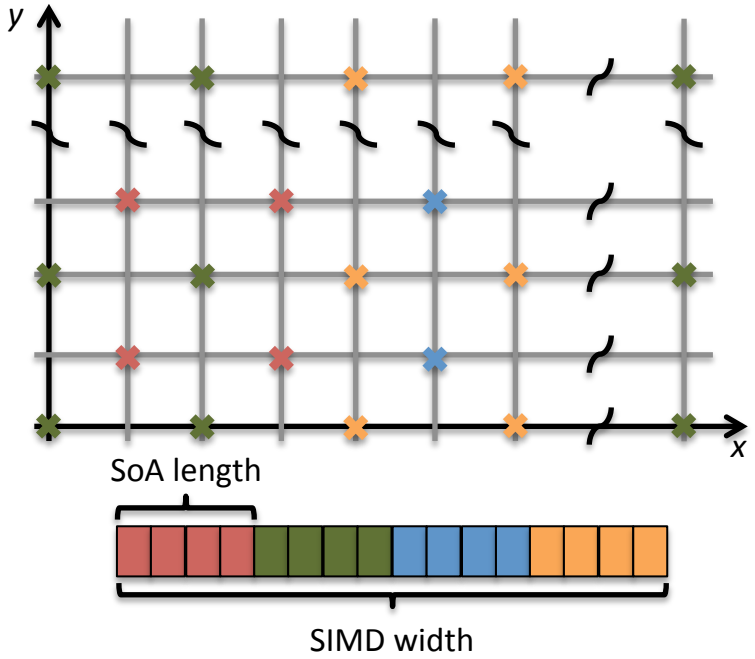
OpenMP Enhancements in Baseline MILC

Directory	# of files total	# files with candidate loops	# of loops	# of loops modified	% of loops rewritten
generic	102	43	229	71	31%
generic_ks	185	42	529	17	3.2%
ks_imp_rhmc	20	8	36	4	11%

- MILC abstracts loops with FORALLSITES and FORALLFIELDSITES.
  - Convert to FORALLxxx_OMP macros
  - Candidate OpenMP loops need to be examined for OMP private variables and reductions
- Examination of all loops would be tedious, so we used the tools mentioned in the slide above to help us identify loops with potentially the most impact

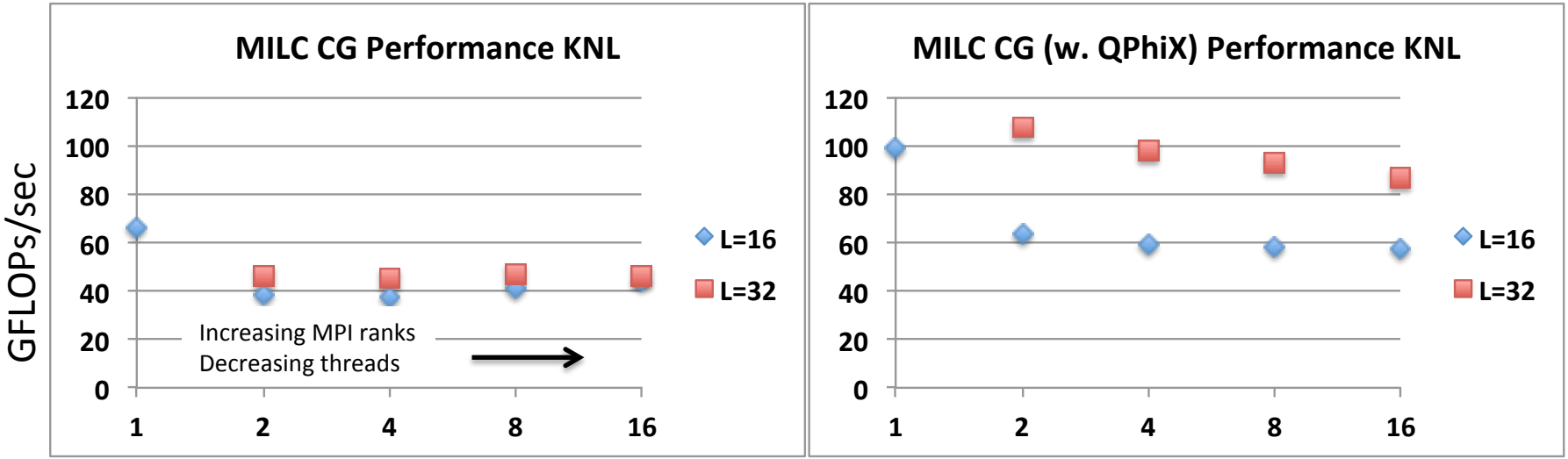
Integrating QPhiX Solver for Staggered Fermions (aka The Big Win)

- QPhiX, Staggered Fermion version, developed to improve vectorization, and threading performance
- Staggered fermions vs. Wilson/Clover
  - Looking at a different "action" than Wilson/Clover
  - Multi-mass CG solver implementation
  - Uses a single right hand side
    → limits reuse and decreases arithmetic intensity
  - 3 complex values per grid point vs. 12
  - Primarily used with double-precision variables
- X-Y blocking for SIMD vectorization
  - Data stored as arrays of structures of arrays
  - SoA in the X dimension
  - SIMD_width/SOA_length in the Y dimension
  - Enables efficient cache blocking of X-Z



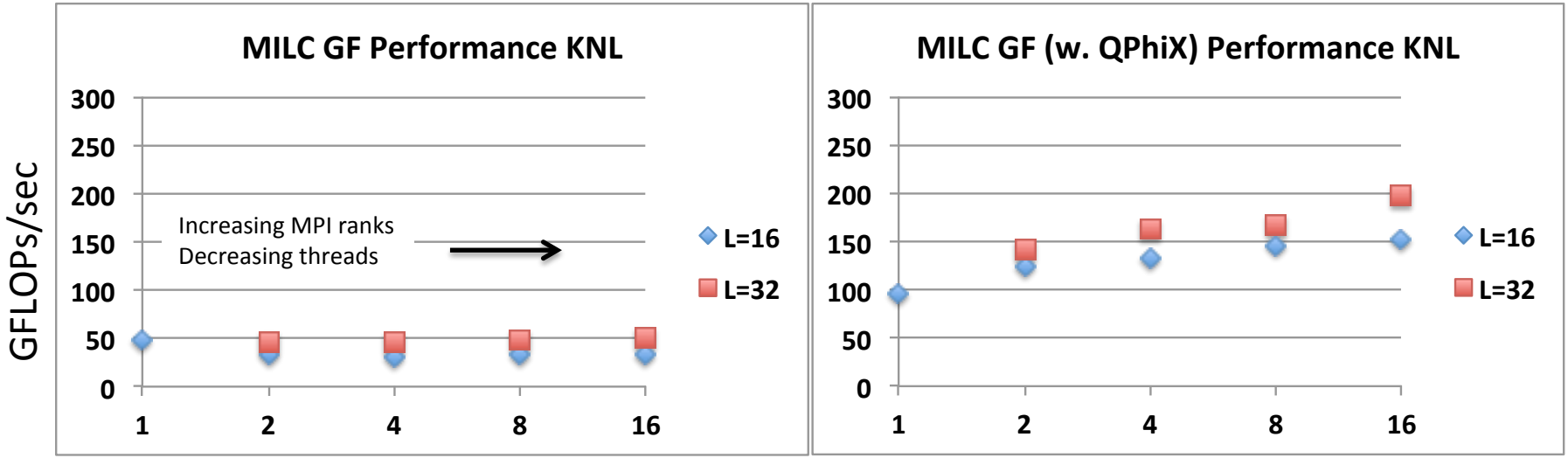
https://github.com/JeffersonLab/qphix

Multi-mass CG Solver (Single Node)



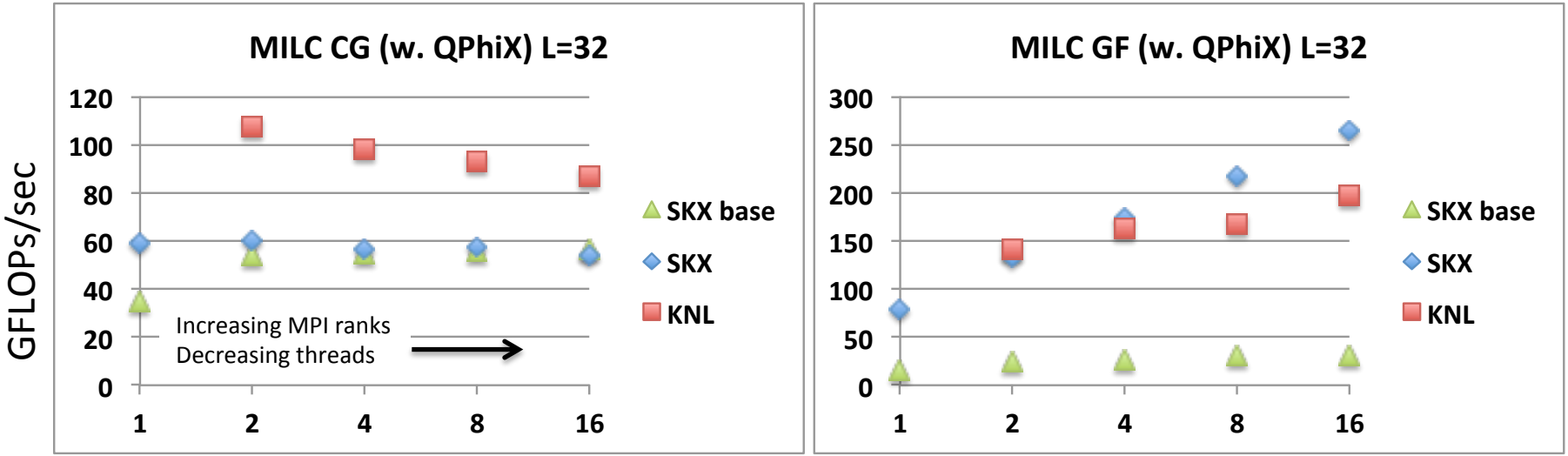
- The QPhiX solver provides a 1.5x for L=32 lattice
- For small lattice sizes, performance is limited by data remapping time

Gauge Force (Single Node)



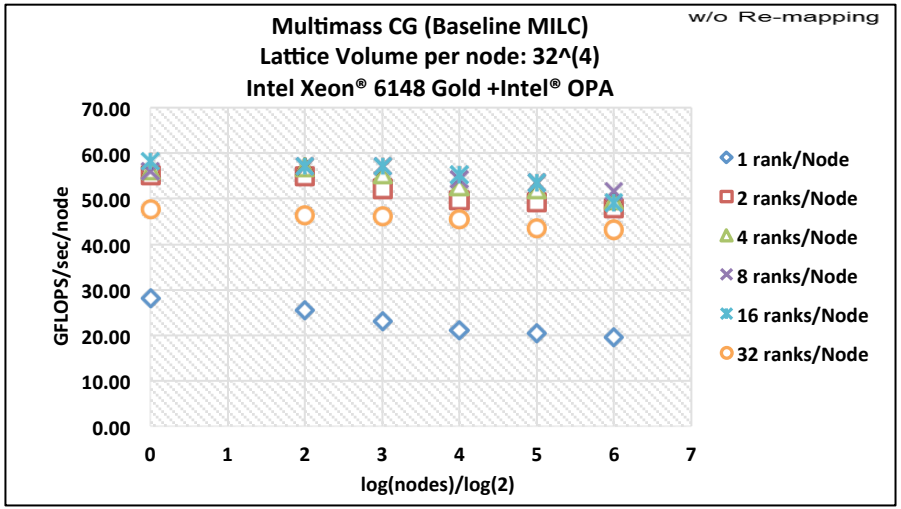
- Gauge Force performance improvements are up to 4x for L=32 lattice size

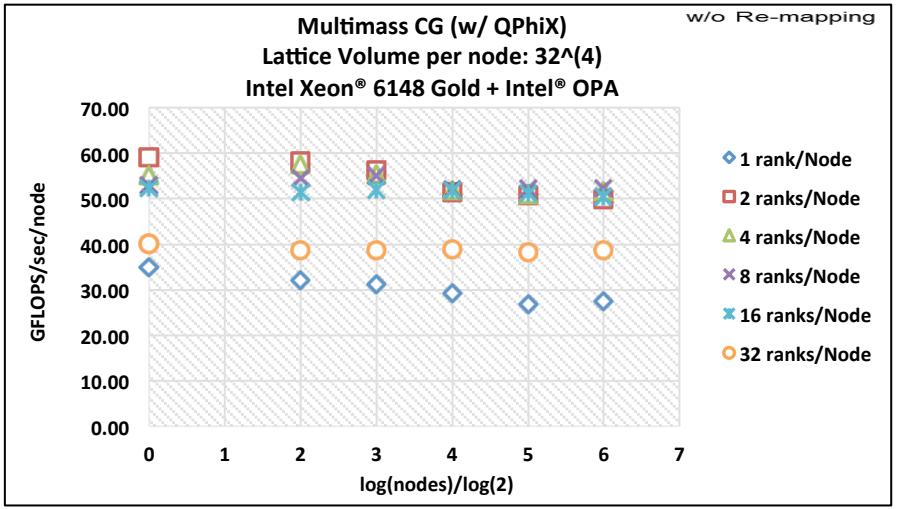
Sky Lake vs. KNL



- Improvements translate to latest Xeon processor with AVX512 as well
- Architectures behave differently wrt rank/thread tradeoffs

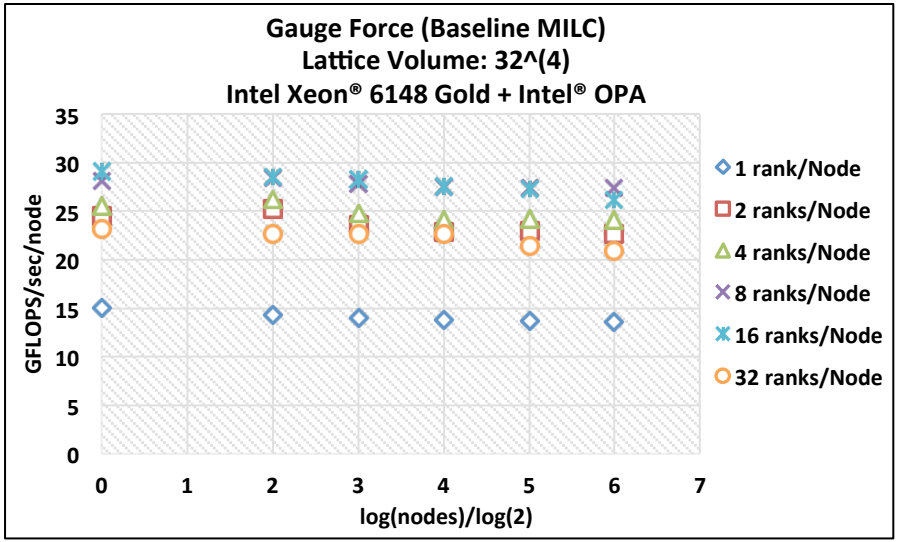
CG Performance on SKX + OPA (MultiNode)

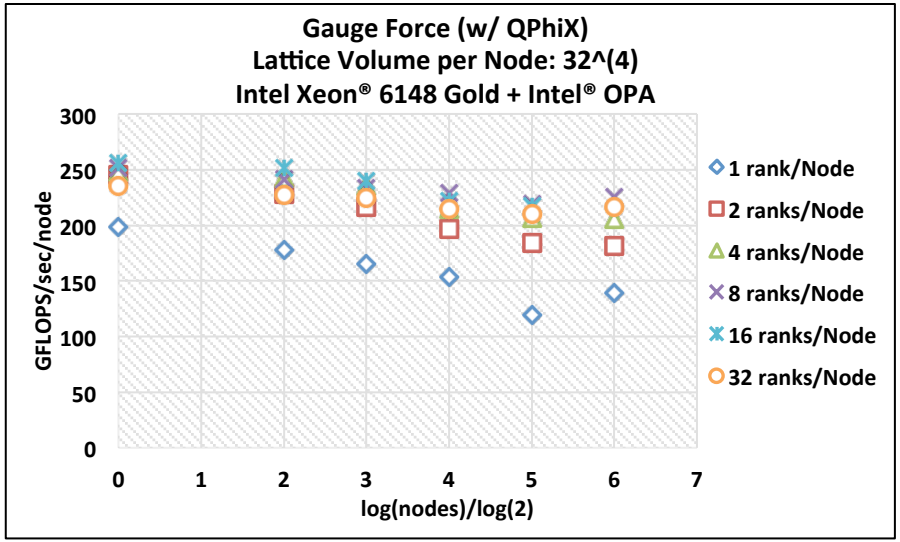




- Weak Scaling Runs on Intel Xeon® 6148 Gold + Intel® OPA on Intel's Endeavor Cluster
- Requires minimum 2 Ranks/Node for best performance (1 rank per NUMA node)
- **CG Solver Performance limited by memory bandwidth on Xeons. Hence, no improvement with QPhiX**
- Parallel Efficiency :
  - ~99% @64 Nodes for 32^(4) Lattice Volume per Node

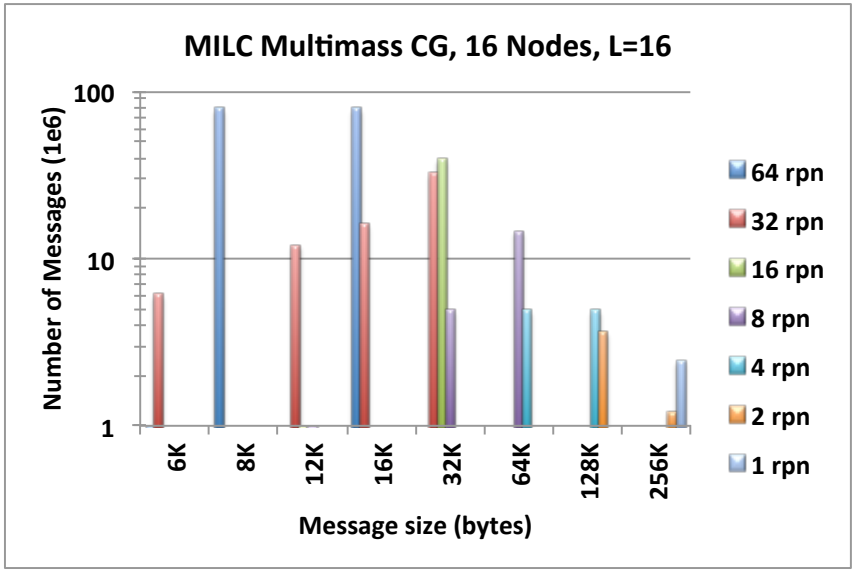
GF Performance on SKX + OPA (MultiNode)

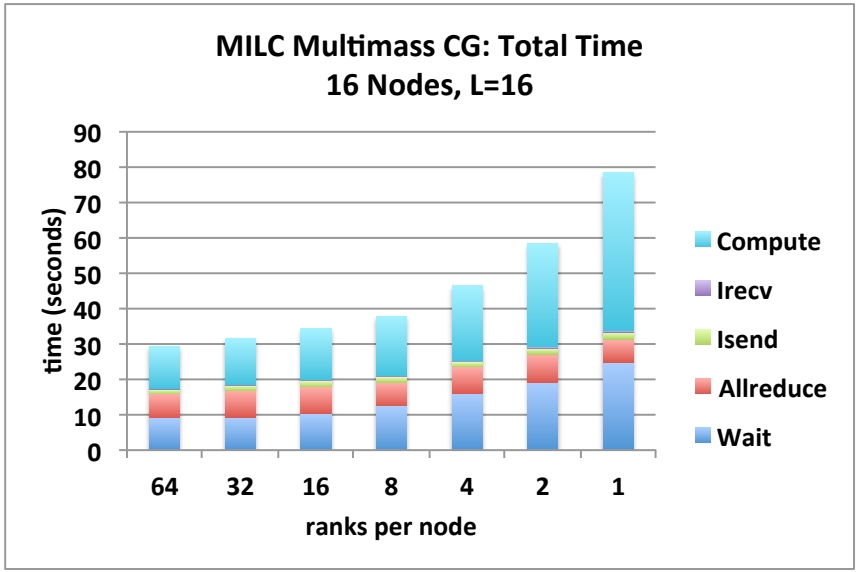




- Weak Scaling Runs on Intel Xeon® 6148 Gold + Intel® OPA on Intel's Endeavor Cluster
- ~3.5x Node Level Performance improvement with QPhiX
- Multiple Ranks/Node gives better performance for Gauge Force
- Parallel Efficiency :
  - ~85% @64 Nodes for 32^(4) Lattice Volume per Node

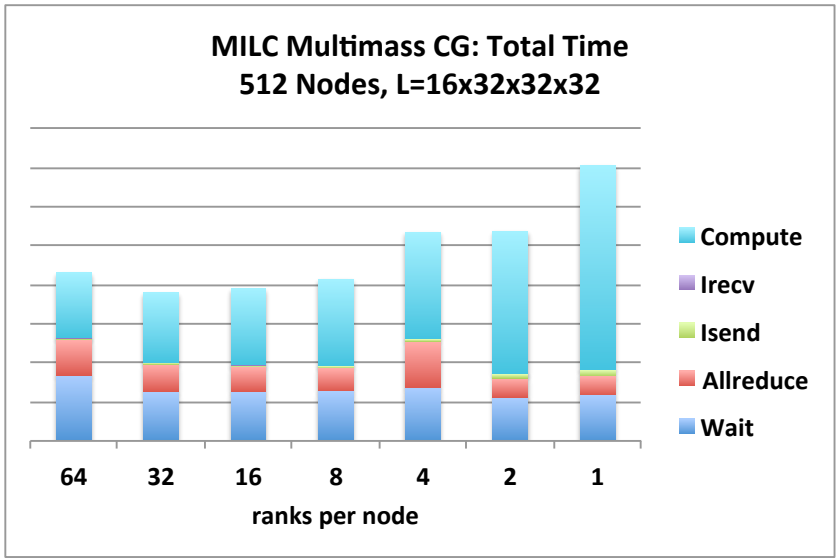
MPI Messaging Characteristics, 16 KNL Nodes

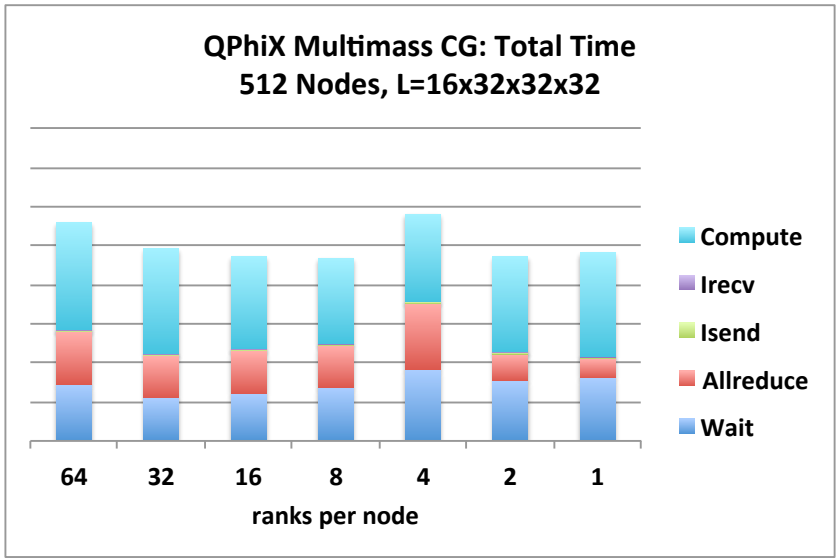




- Message sizes vary with number of ranks per node (RPN)
- As RPN decreases
  - Lattice size (volume) per rank increases -> message sizes increase
  - Surface-to-volume decreases -> total amount of data decreases (proportional to $1/N^4$)
  - Same analogy with increasing lattice size (volume) per node

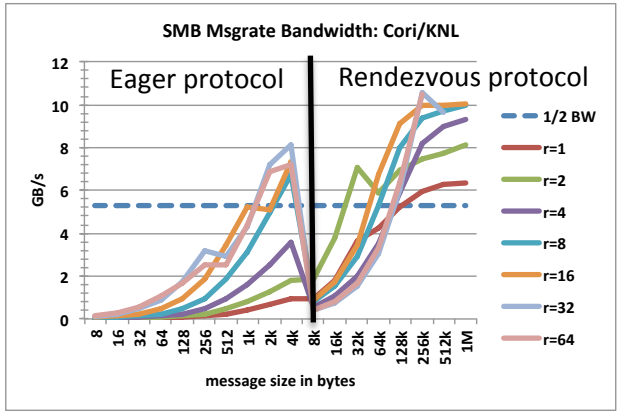
MPI Characteristics, 512 Nodes

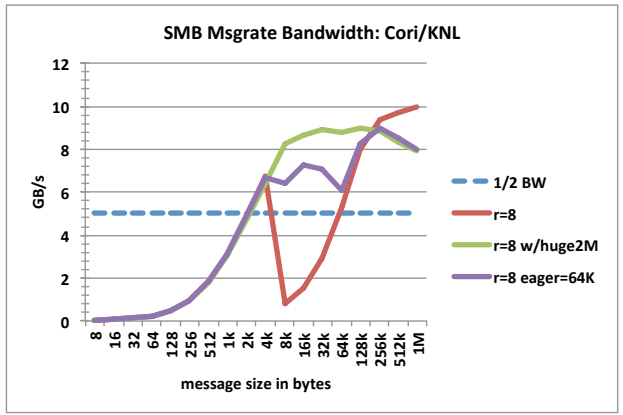


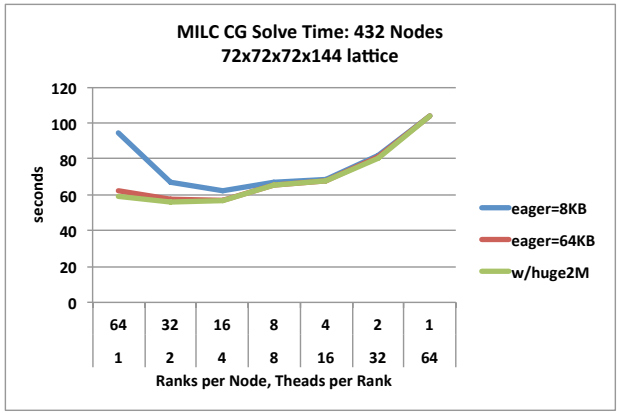


- Caveat 1: at this scale, the "full" IPM instrumentation impacts absolute time, but relative times "should" be representative (I need more confidence that this is true)
- Caveat 2: Results are from a single trial, could be up to 10% variability

Cori (Cray Aries) Huge Pages Optimization







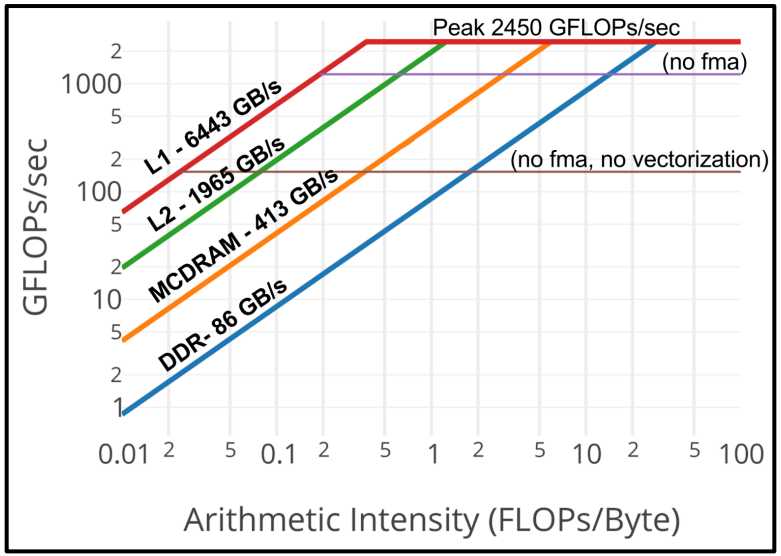
- MPI message rate micro-benchmark identified an issue where BW drops significantly when transitioning to the Rendezvous protocol

- Two solutions tried:
- Move Rendezvous transition to 64KB
- Per Cray advice, use huge pages, 2M pages in this case

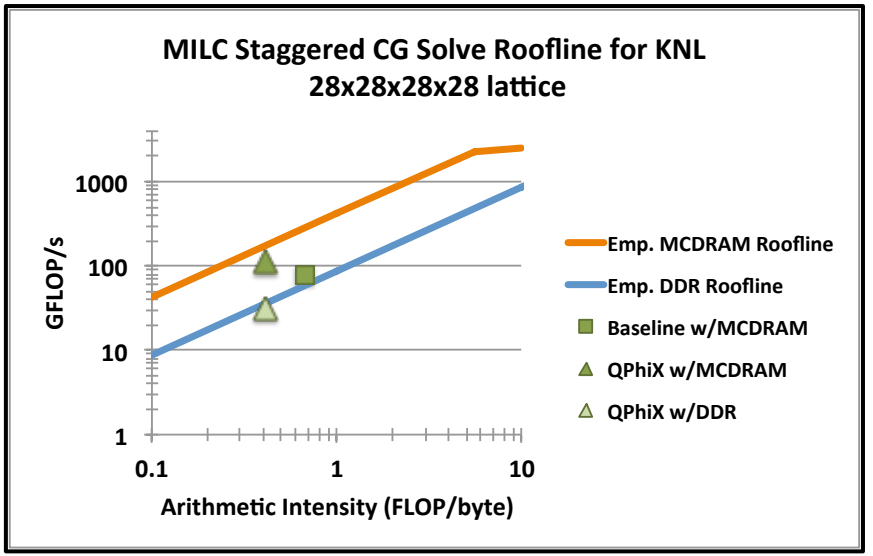
- This has a significant impact on performance when using a large number of MPI ranks per node
- Recommendation → Use huge pages in communication intensive codes with moderate message size

Roofline Analysis

- Helps to focus on areas of code to target for optimization
- MILC is known to memory BW bound, and QPhiX version is near roofline model, but baseline version has higher AI and lower performance?



Roofline Model for KNL



MILC Performance vs. Roofline

Impact on Production Computing

The Rubber Hitting the Road

- Current Cori (KNL) production runs
  - 96x96x96x192 lattice on 128 nodes
  - ks_spectrum_hisq: Calculation of meson and baryon spectra from a wide variety of sources and sinks
  - su3_clov_hisq: generates clover propagators and contracting meson and baryon two point functions
- QPhiX version of the multi-mass staggered fermion solver is being used

	Staggered CG	Clover Bi CG
Standard MILC[1]	40 GF/s	21-69[2] GF/s
QPhiX[3]	52 GF/s	69 GF/s
QPhiX Improvement	1.3x	1x to 3.3x

1. 64 ranks per node, 1 thread per rank, includes OpenMP improvements
2. 256 nodes running a different problem
3. 32 ranks per node, 8 threads per rank

Advantage of using Cori's Burst Buffer for I/O

- I/O overhead is being significantly reduced by using Cori's Burst Buffer (BB) subsystem
- Typical wall clock for 128 node run is ~5 hours
- Using nominal Lustre /scratch, 53 minutes spent writing 61 temporary files
- Using BB, I/O was reduced to 26 minutes
  - 2x improvement

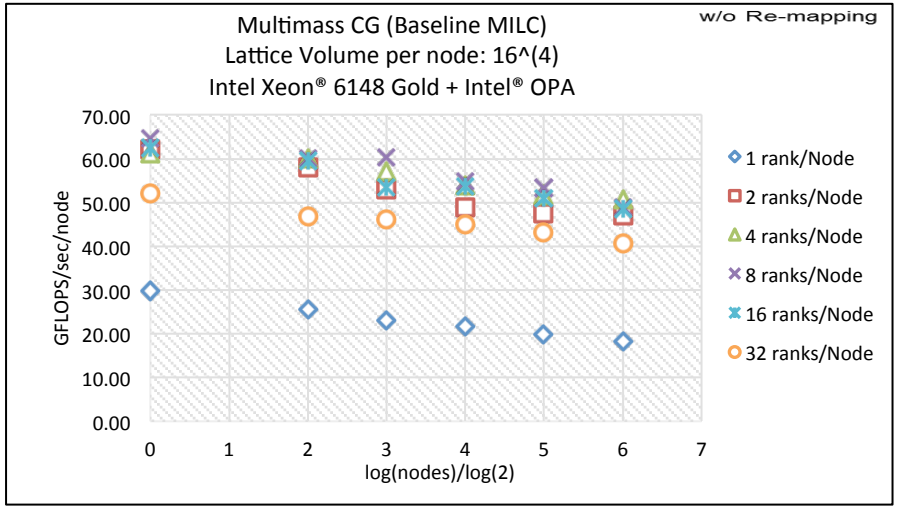
Current and Future Efforts

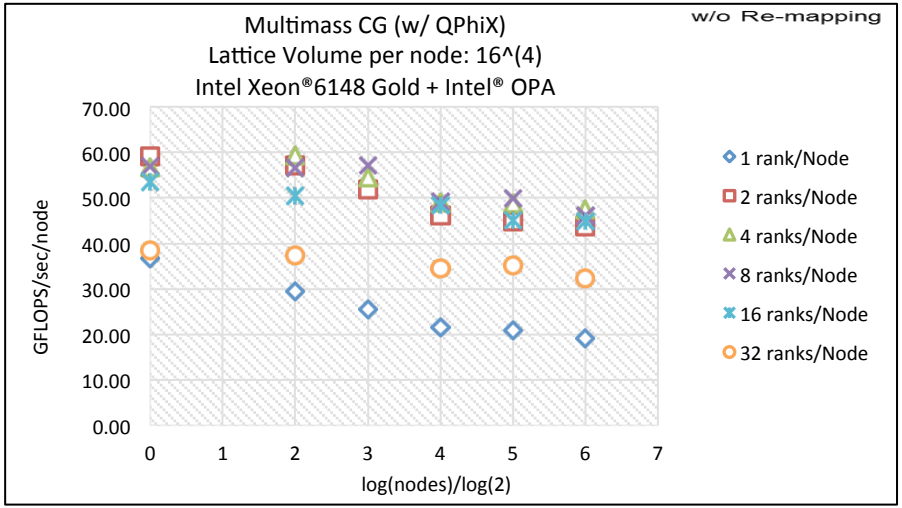
- Fermion Force calculation optimizations
- Fat Links calcuation optimizations
- High-speed interconnect explorations
  - Multiple end points implementation of OPA MPI
- Continue improvements to OpenMP
  - Roofline analysis shows that baseline MILC perhaps not BW bound
- Investigating Grid solver
- Continue working on SciDAC/QOP version of the code

Discussion & Questions

Backup

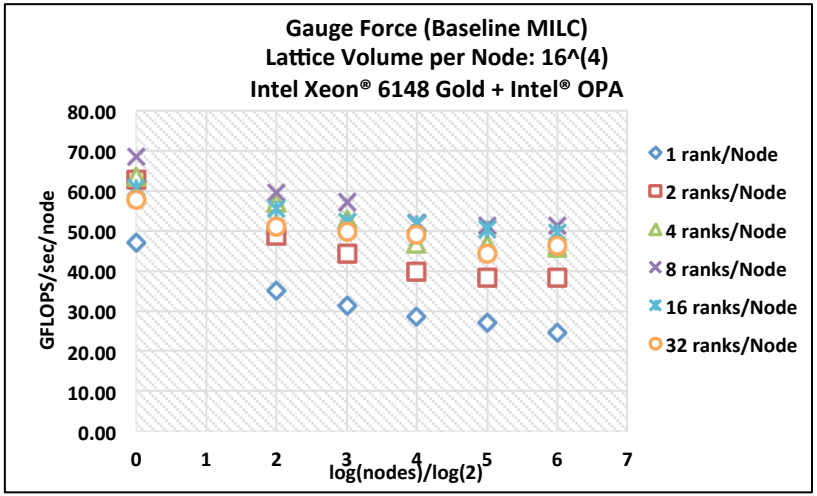
CG Performance on SKX + OPA (MultiNode)

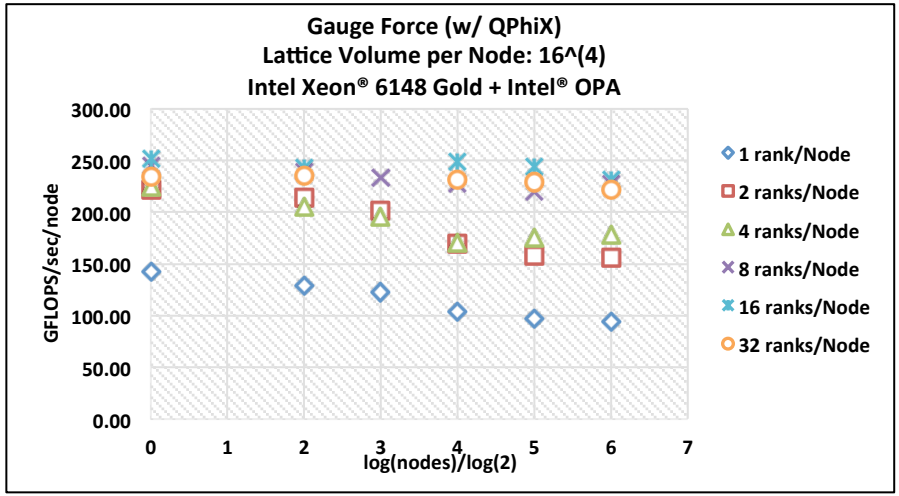




- Weak Scaling Runs on Intel Xeon® 6148 Gold + Intel® OPA on Intel's Endeavor Cluster
- Requires minimum 2 Ranks/Node for best performance (1 rank per NUMA node)
- Smaller Volume [16^(4) per Node] becomes communication sensitive at larger node counts
- Parallel Efficiency :
  - ~80% @64 Nodes for 16^(4) Lattice Volume per Node

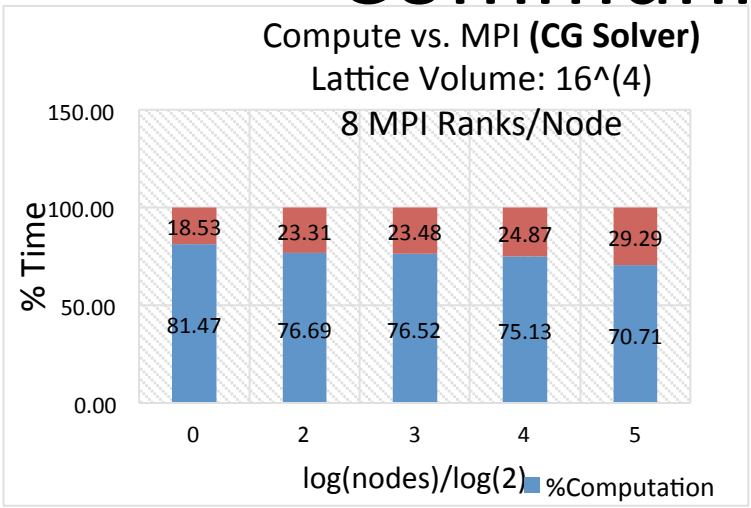
GF Performance on SKX + OPA (MultiNode)

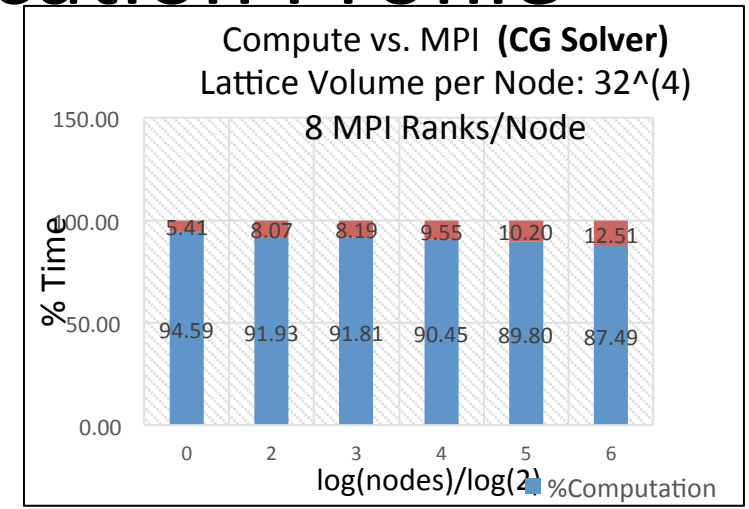




- Weak Scaling Runs on Intel Xeon® 6148 Gold + Intel® OPA on Intel's Endeavor Cluster
- ~3.5x Node Level Performance improvement with QPhiX
  - **Not seeing LLC effects with QPhiX as seen with Baseline MILC (need to investigate)**
- Multiple Ranks/Node gives better performance for Gauge Force
- Parallel Efficiency :
  - ~92% @64 Nodes for 16^(4) Lattice Volume per Node

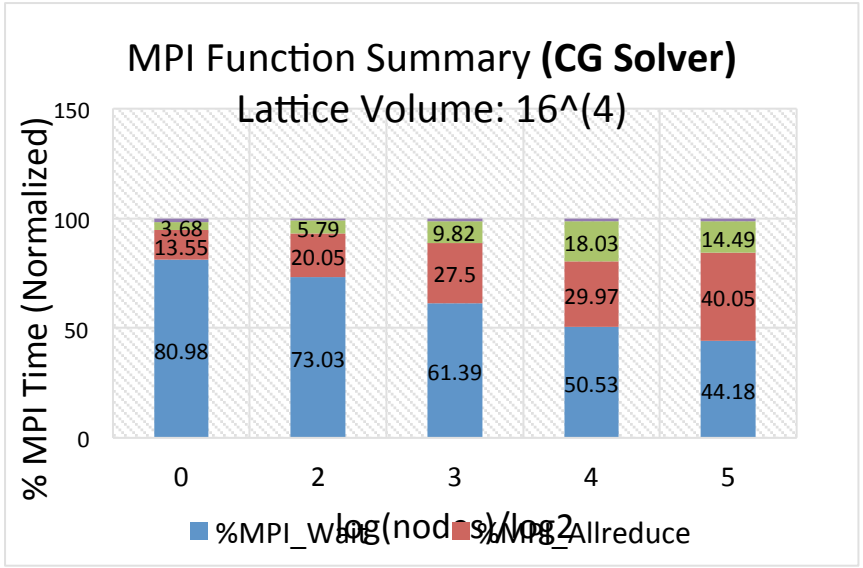
Communication Profile





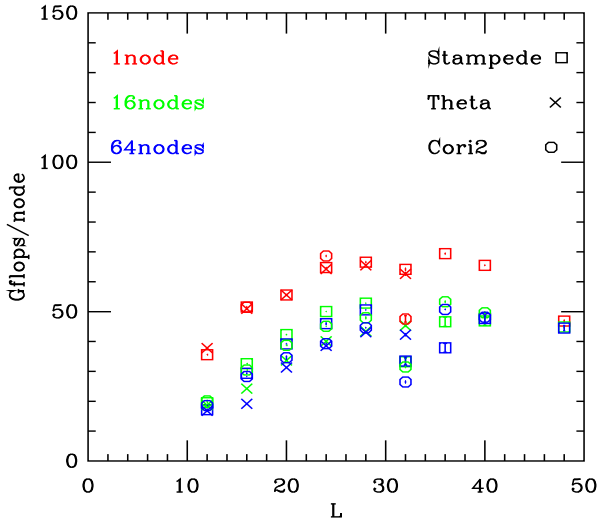
- Profile collected using Intel® MPI Performance Snapshot Tool (part of ITAC)
- Smaller Lattice Volume (16^(4)) spends high % of time in MPI as expected

MPI Function Summary



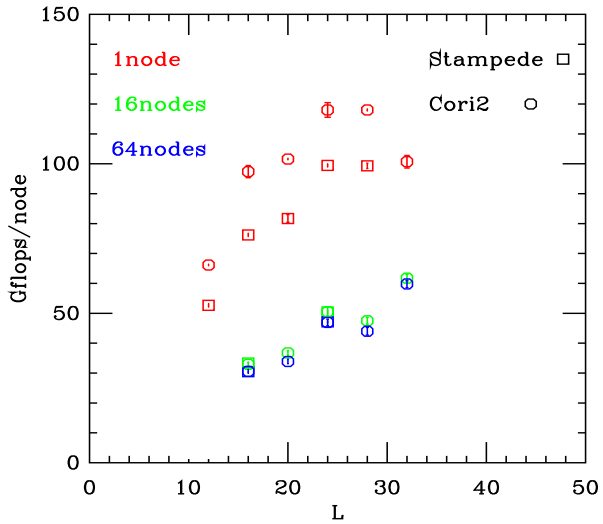
- Most of MPI time spent in MPI Wait (i.e. P2P Send/Recv completion)
- Collective Ops Time (i.e. %MPI_Allreduce) increases with node counts
  - MPI_Allreduce ~40% of MPI Time at 64 Nodes
  - Potential bottleneck at very large nodes

Staggered Multi-mass CG: Lattice Scaling Study



MILC baseline:

- 64 MPI ranks per node

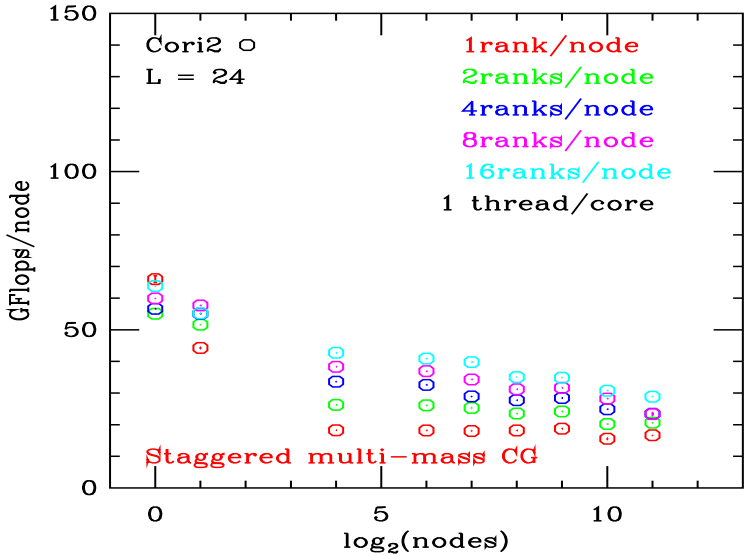


Staggered QPhiX:

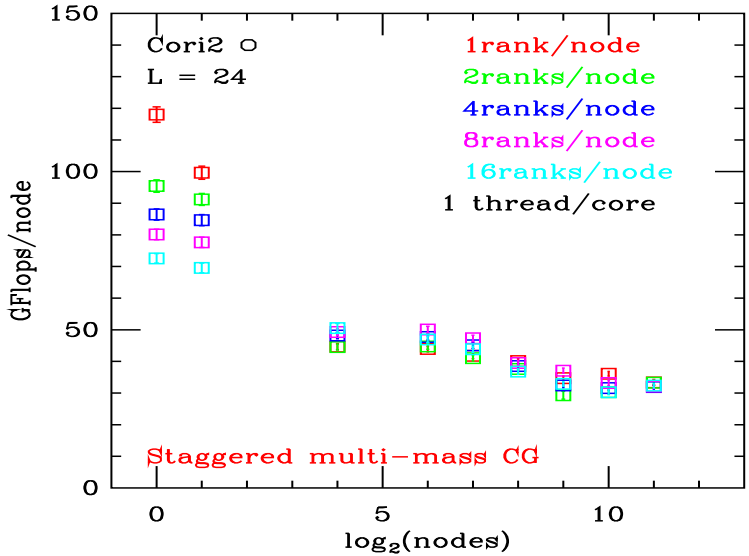
- 1 MPI rank on 1 node
- up to 16 ranks on multiple nodes

Multimass CG: Weak Scaling on Cori

Various MPI rank / OMP thread combinations, L = 24



MILC baseline

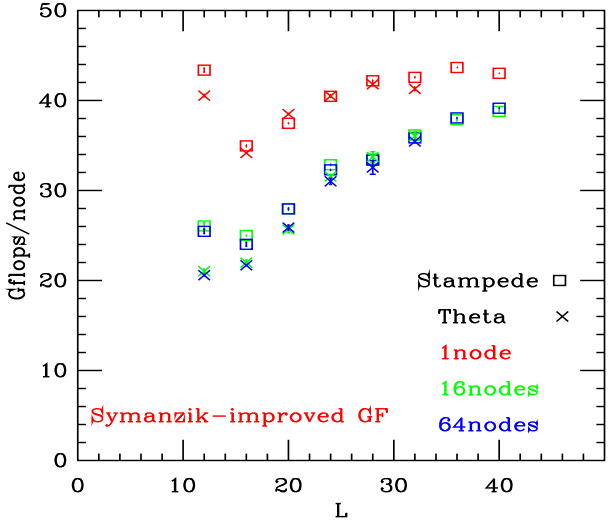


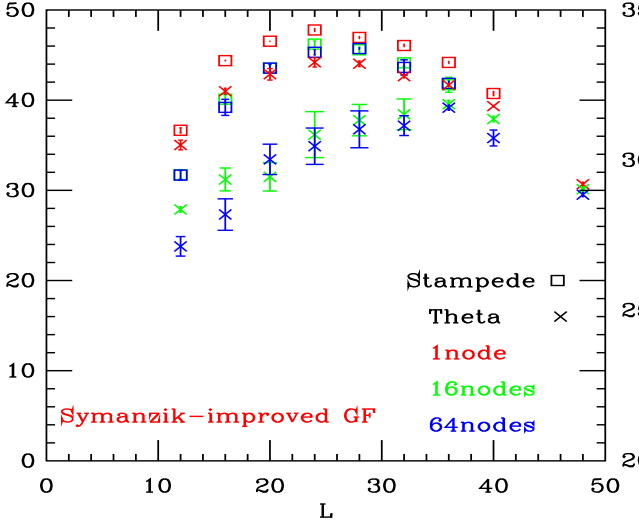
QPhiX

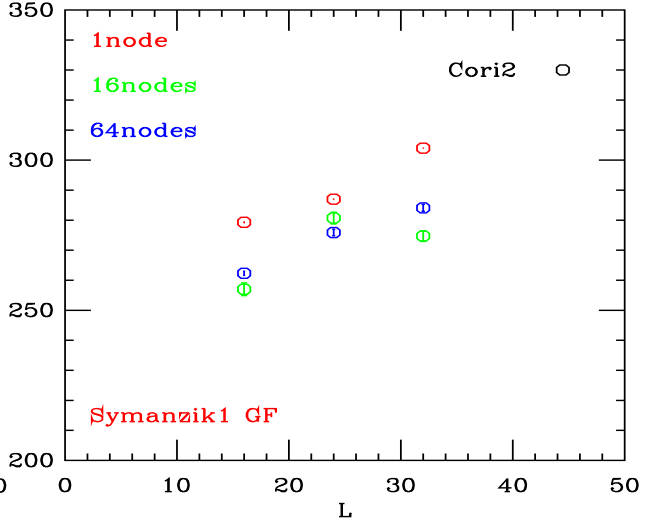
Benchmarks: Symanzik Gauge Force

GF performance in GFLOPS/node, various lattice sizes per node L

1 thread/core







MILC baseline:

- 64 MPI ranks per node

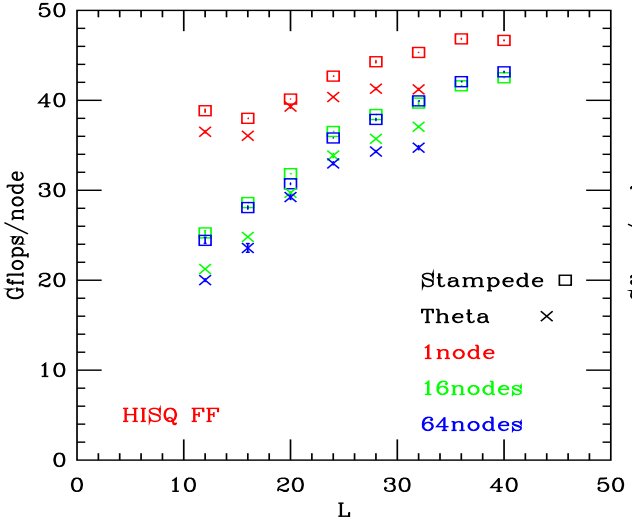
QOPQDP:

- 64 MPI ranks per node

QPhiX:

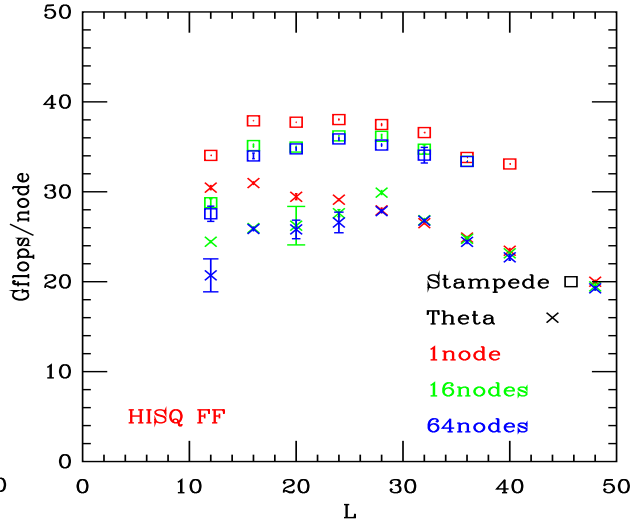
- 1 MPI rank with 1 node
- 16 ranks/node otherwise

Benchmarks: HISQ Fermion Force



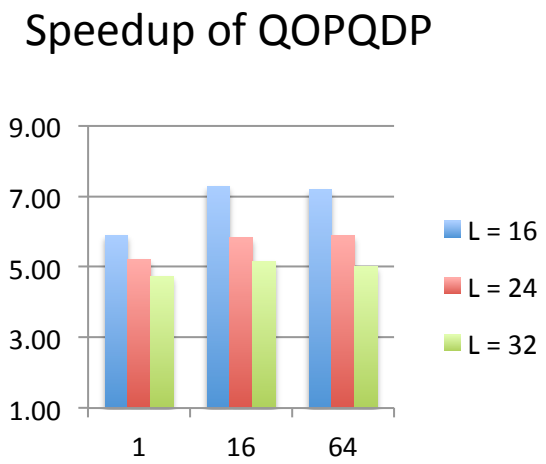
MILC baseline:

- 64 MPI ranks per node



QOPQDP:

- 64 MPI ranks per node



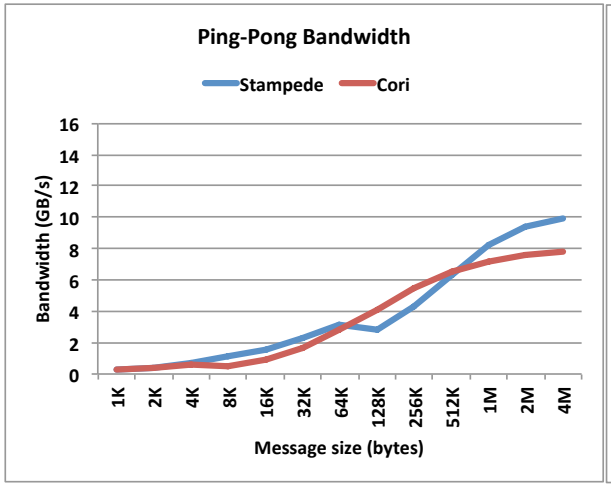
Modified algorithm vs. MILC FF

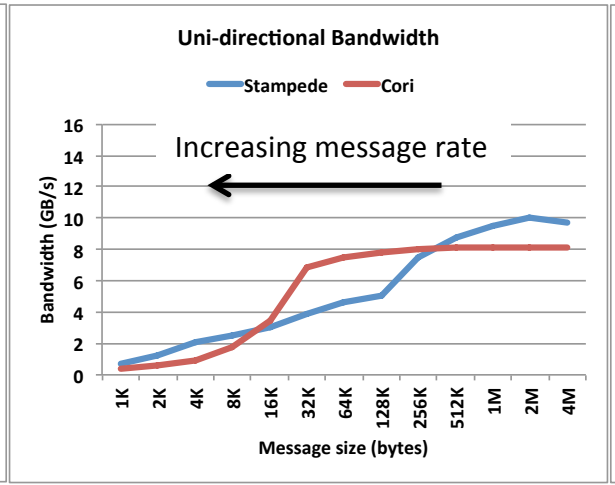
- Speedup due to a reduced number of calculations (FLOPs) to 17% of baseline

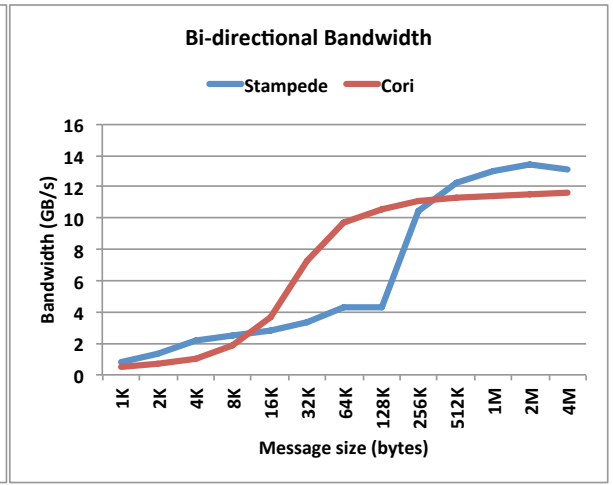
Cori and Stampede

- Both use the same 68-core KNL SKUs
- Cori uses Aries high-speed interconnect, Stampede uses Intel OmniPath (OPA)
  - Cray and Intel MPI respectively
- Cori uses Cray CNL, Stampede is ??? Linux
- Standard (non-QPhiX) version of MILC
  - Primary analysis is high-speed interconnect performance, using relatively small lattice size per node
- Intel C for both, although 17.0.2.x vs. 17.0.0.x
- Limited to maximum job size of 80 nodes on Stampede -> limited scaling study to 64 nodes

OSU Single rank Pt-2-Pt

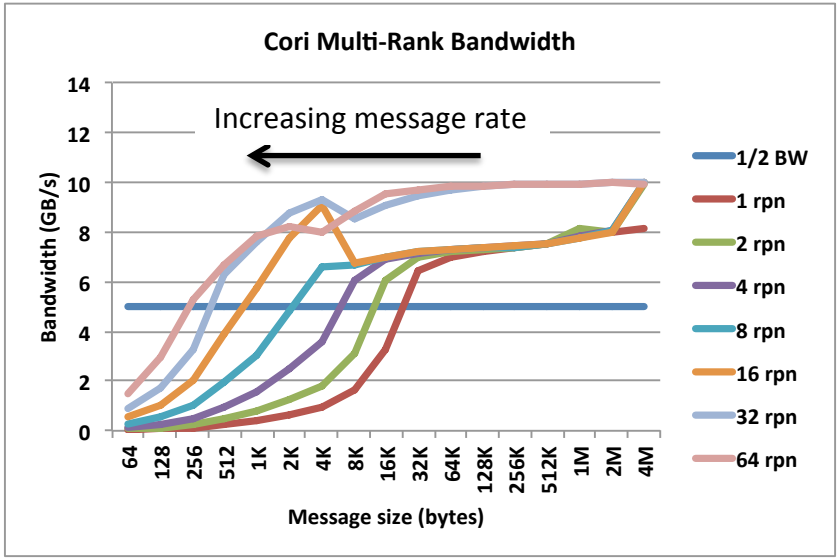


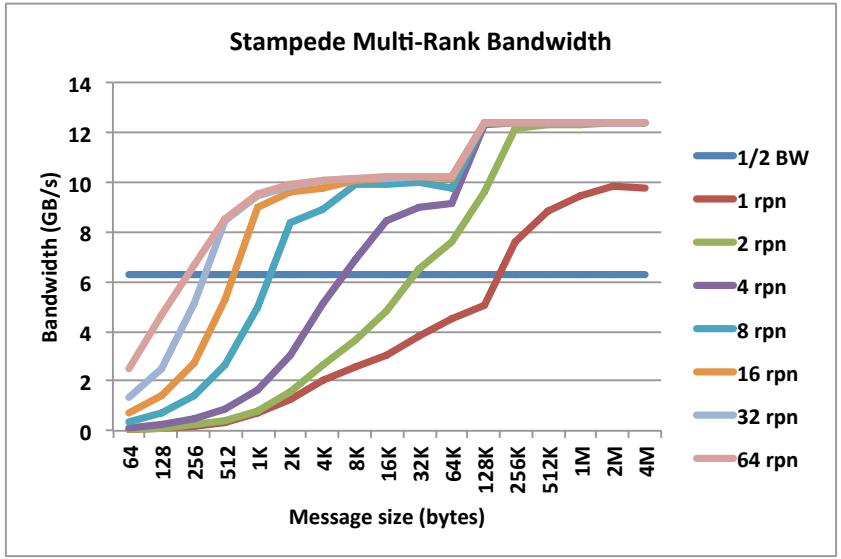




- Single core, point-to-point between 2 nodes
- Latency is comparable at ~3.1 µS (but about 2x that of Haswell)
- Ping-pong is exactly that, single message ping-ponged between 2 nodes
- Uni-directional is a “streaming” exchange with a window of size 64
- Bi-directional is also “streaming”
- Cori shows better small message BW (message rate) and Stampede higher peak BW

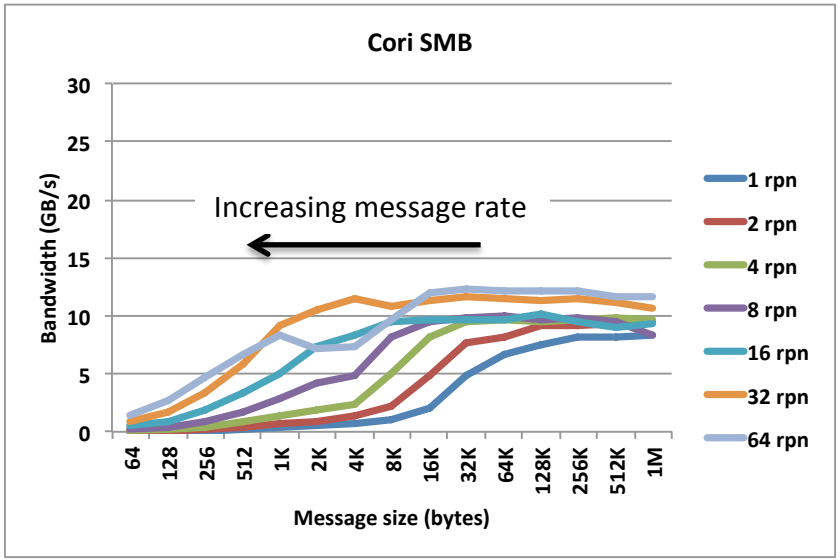
OSU Multi-rank Pt-2-Pt

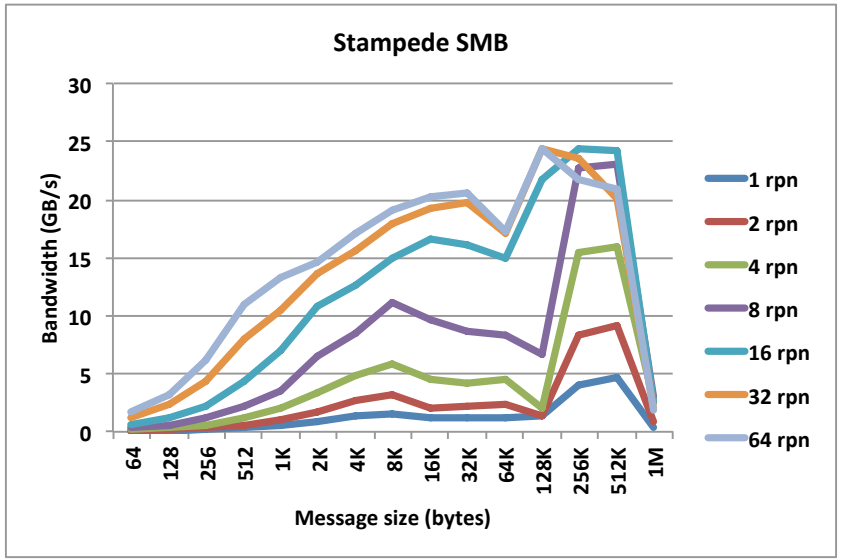




- As ranks per node (RPN) increase in this uni-directional test, Stampede's performance improves.
- Cori has a BW "ceiling" below 32 RPN that limits large message BW that is not observed with Stampede
  - Cray attributes this to a PCI latency issue between KNL and Aries, can be mitigated by moving BTE "put" protocol transition to a smaller message (default is 4 MB)
- It looks as though Stampede could use some tuning in its transition to a large message protocol to take advantage of its higher peak BW

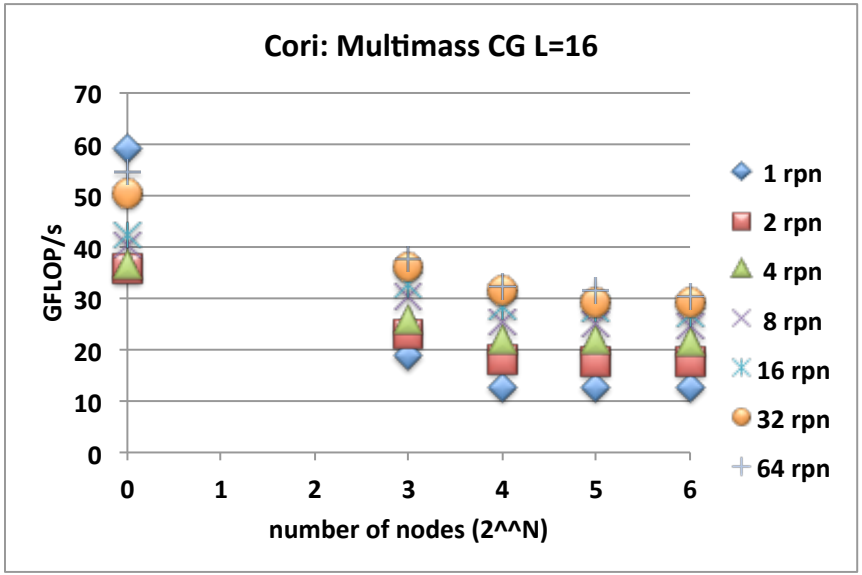
SMB Multi-node, Multi-rank Stencil

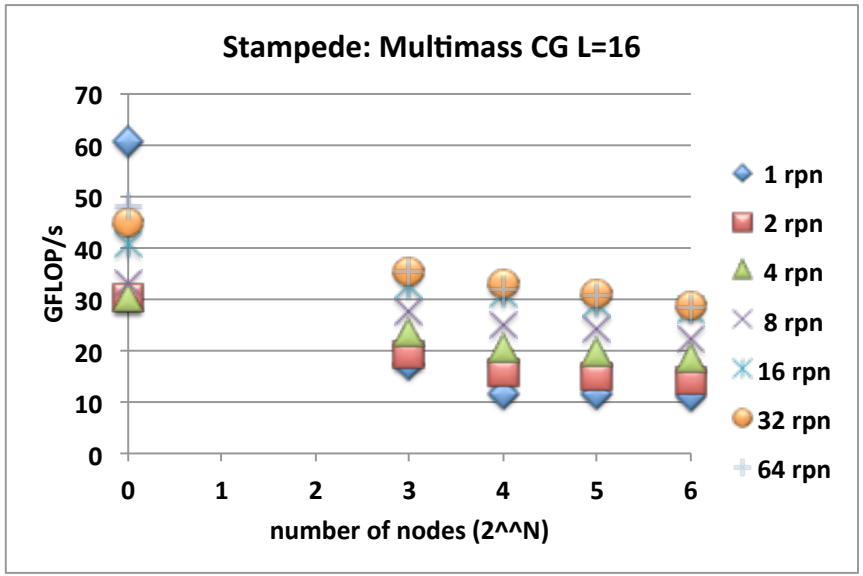




- 16 nodes, 6 neighbors per rank (emulates 3D stencil communication pattern)
- Measures bi-directional message rate (converted to BW)
- Here we see Stampede is able to achieve close to its peak bi-direction BW (25 GB/s)
  - However, something happens at 1 MB message size, to be investigated
- Stampede could also use some tuning in its protocol transition to large messages (> 64 KB)

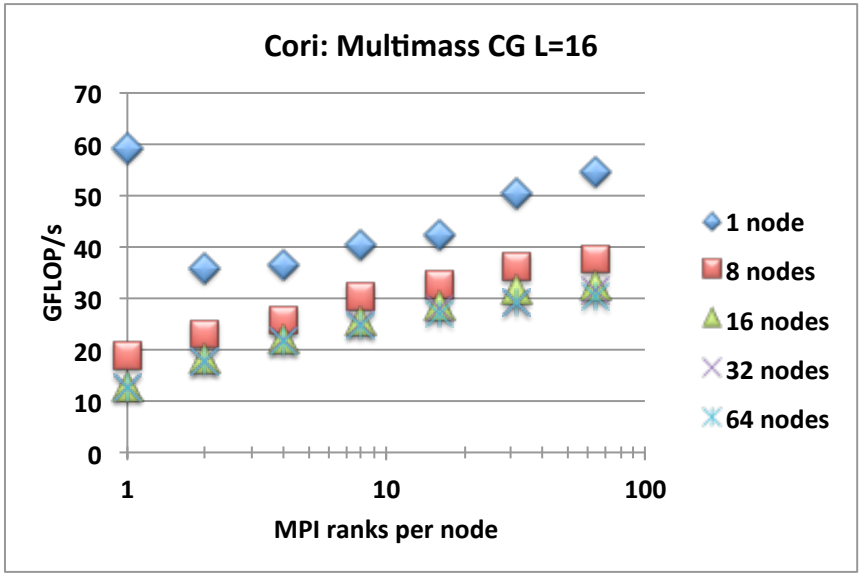
Weak Scaling: Number of Nodes

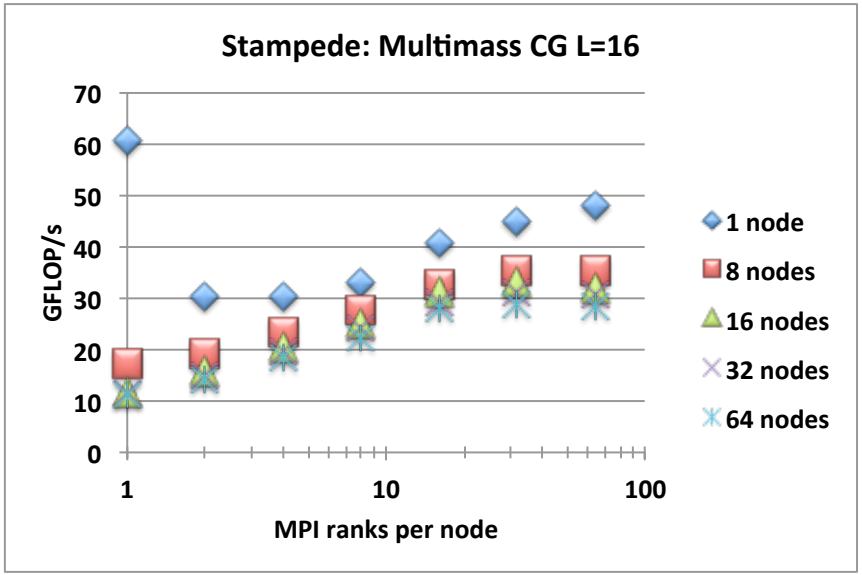




- Weak Scaled: lattice size is 16x16x16x16 per node
- MPI/OpenMP trade off study at 1, 8, 16, 32 and 64 nodes
  - number of cores fixed at 64
  - MPI ranks/node (rpn) varied from 1 to 64
  - All results use 1 thread per core

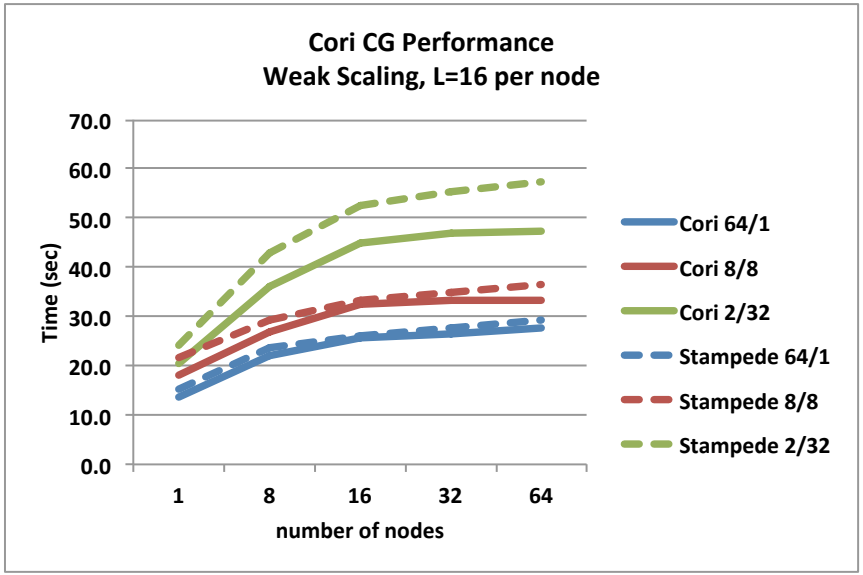
Weak Scaling: Ranks per Node

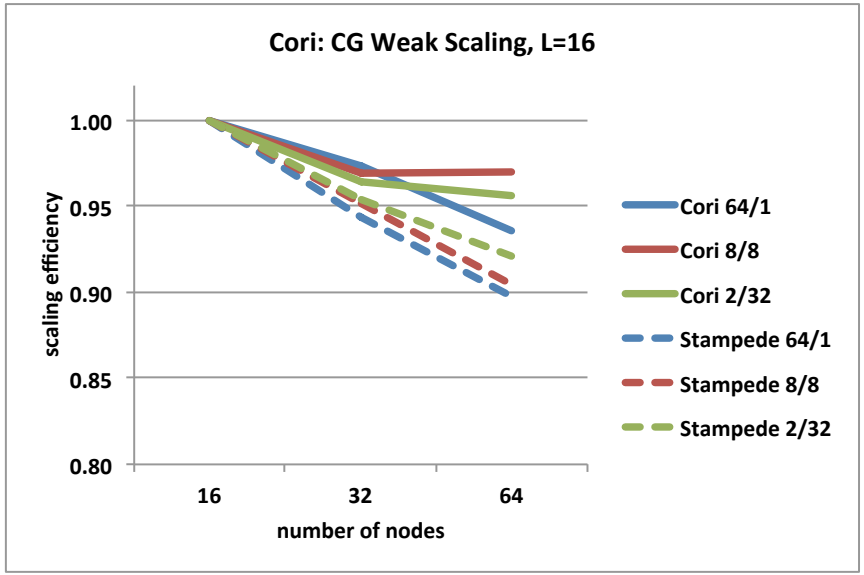




- Weak Scaled: lattice size is 16x16x16x16 per node
- MPI/OpenMP trade off study at 1, 8, 16, 32 and 64 nodes
  - number of cores fixed at 64
  - MPI ranks/node (rpn) varied from 1 to 64

Weak Scaling: Select MPI/OMP





- CG time and CG scaling for selected MPI/OpenMP combinations
  - Ideal scaling would be a flat horizontal line
- Stampede scaling is better with higher RPN
- There is a scaling improvement on Cori going from 32 to 64 RPN, but I wouldn't read to much into it at such as small scale