MILC Staggered CG Performance on Intel KNL

Carleton DeTar, Douglas Doerfler,
Steven Gottlieb, Ashish Jha, Bálint Joó,
Dhiraj Kalamkar, Ruizi Li*, Doug Toussaint

* : Presenter

Intel Parallel Computing Centers

INDIANA UNIVERSITY

Outline

- Staggered QPhiX library
- Benchmarks and performance results
- Conclusions and outlook

Staggered QPhiX library

- Developed from standard open source QPhiX library for Wilson quarks. Joint work by Intel, Jlab, and MILC collaboration
- Portable to various IA ISA's: SSE, AVX2, KNC, AVX512
- Supports MPI, OpenMP
- Current implementation on CG, supports both single and double precision
- Extending to gauge force

Staggered QPhiX library

- Data structure AoSoA :

  **KS_Color_Vector[3][2][VECLEN],*

  **Gauge[8][3][3][2][VECLEN],* etc.

- Data layout: EO checkerboarded, 4D or 3D block for SP or DP, with size 2 along each dimension to fill up the inner *VECLEN* array, i.e., data at (x, y, z, t), (x+Nx/2, y, z, t), (x, y +Ny/2, z, t), (x+Nx/2, y+Ny/2, z, t), … are stored consecutively.

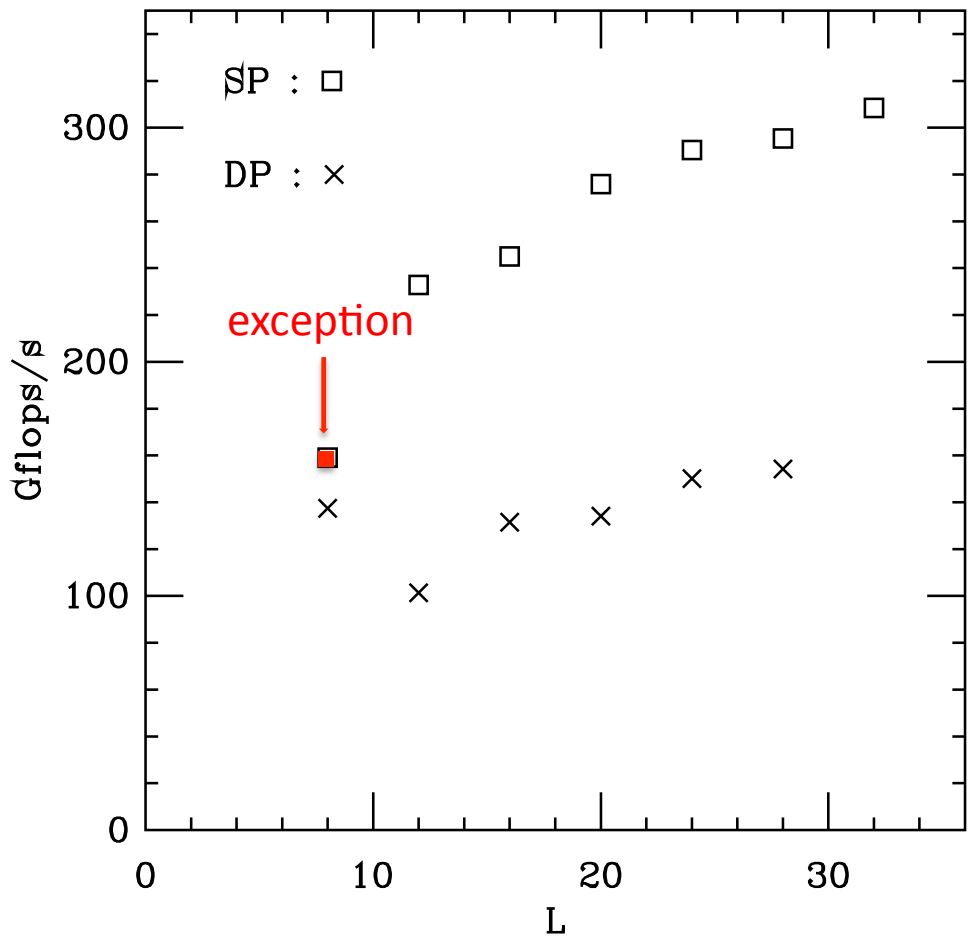
Hardware configuration

- Intel Endeavor Cluster:
  - Intel Xeon Phi™ processor 7210 & 7250 (KNL): 64 & 68 cores @ 1.3 & 1.4 GHz, 8 or 16 GB MCDRAM, 6x16 GB DDR4 @ 2.1 & 2.4 GHz and 115 GB/s peak BW
  - Intel Broadwell(BDW) multi-core processor: Intel Xeon Dual Socket processor E5-2697 v4, 18 Cores/Socket, 36 Cores @ 2.3 GHz, 128GB DDR4 @ 2.4 GHz
- NERSC Cori Cluster:
  - Intel Haswell(HSW) multi-core processor, based on Cray's XC40 architecture: 2 sockets, 16 cores/socket @ 2.3 GHz, 128 GB DDR4 @ 2.1 GHz, 1.92 Pflops (theoretical peak)

Benchmarks

- Multi-mass CG using 9 or 11 masses
- Benchmarks :
  - QPhiX Dslash one KNL node
  - MILC+QPhiX vs. baseline MILC code, one and multi-node

QPhiX HISQ Dslash Performance



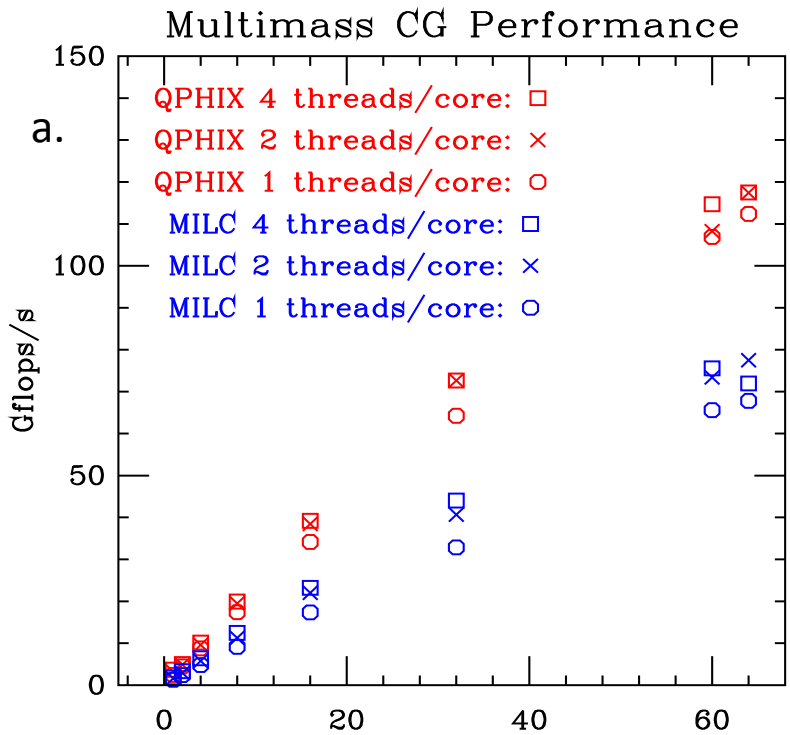
One KNL 7250:
64 cores;
2 threads/core
except L = 8

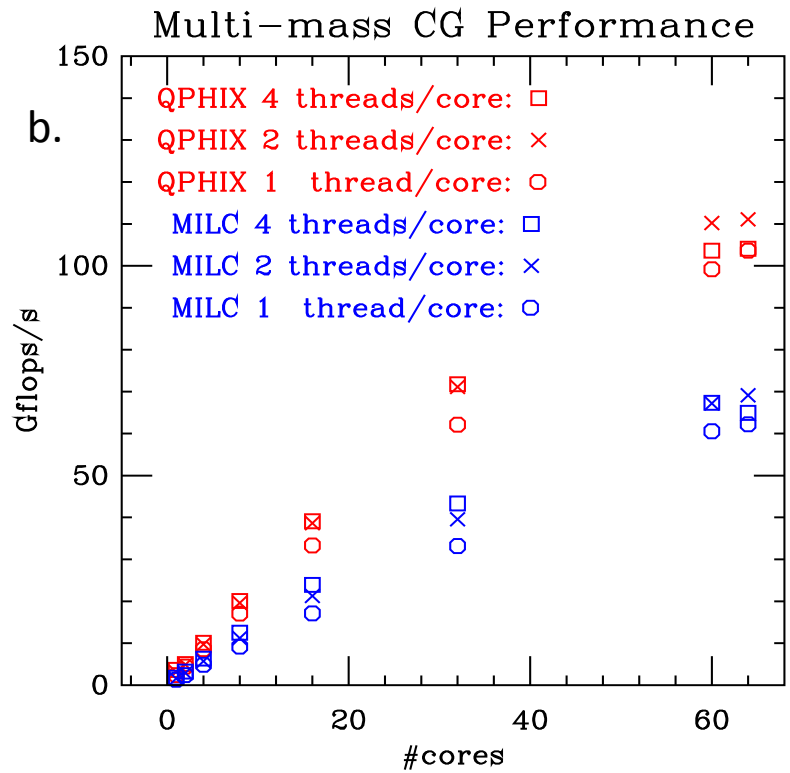
Further improve:
Compressed gauge;
Software prefetches

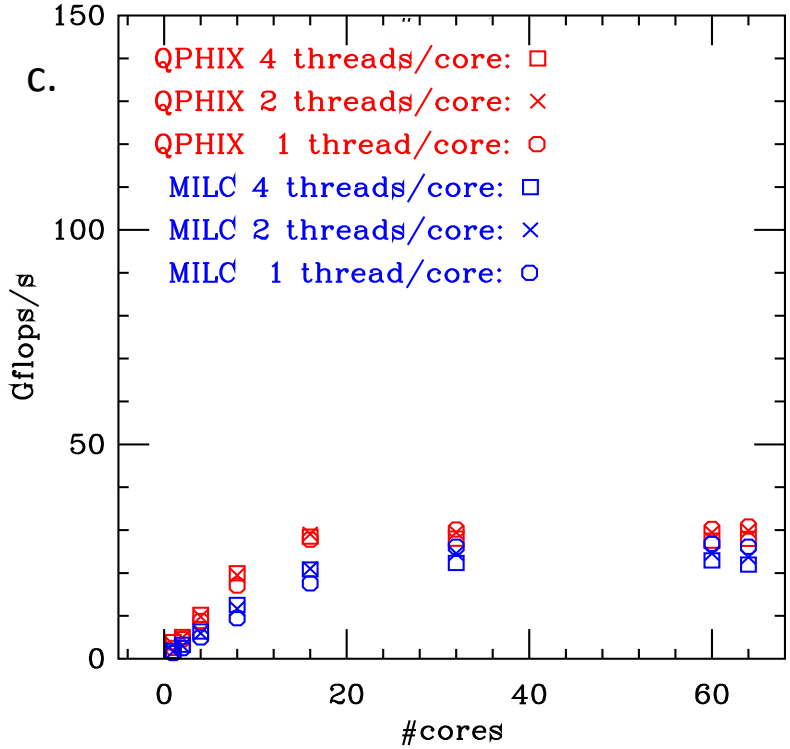
MILC+QPhiX performance

- QPhiX staggered Dslash bandwidth on single KNL:
  - Increases w. increasing lattice volume(L > 12);
  - (Model BW) Hits 80% peak read bandwidth w. hardware prefetches.

- The next two slides compare MILC+QPhiX and Baseline MILC code on:
  - KNL 7250 one node;
  - KNL 7210 and Broadwell multi-node.



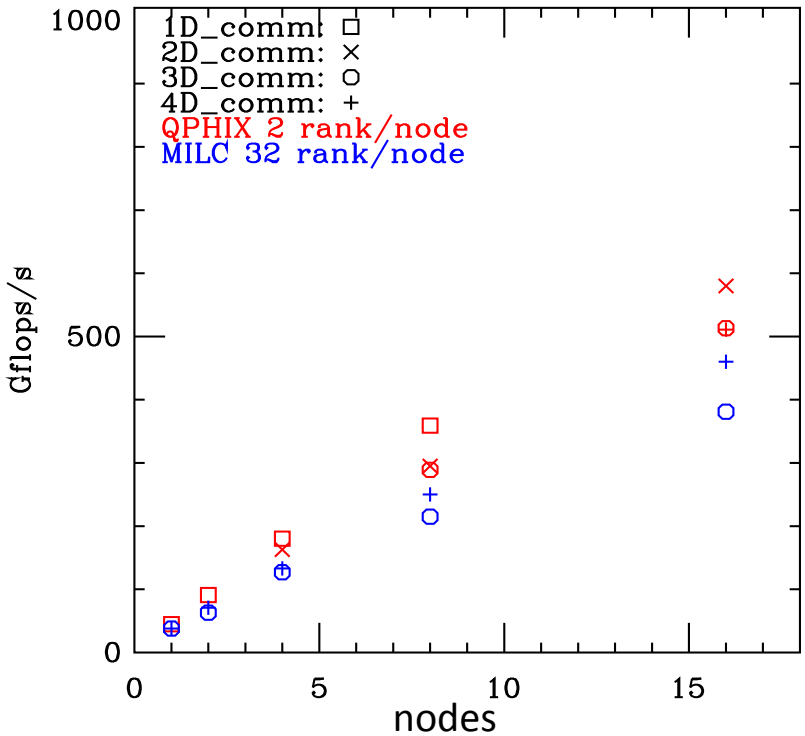




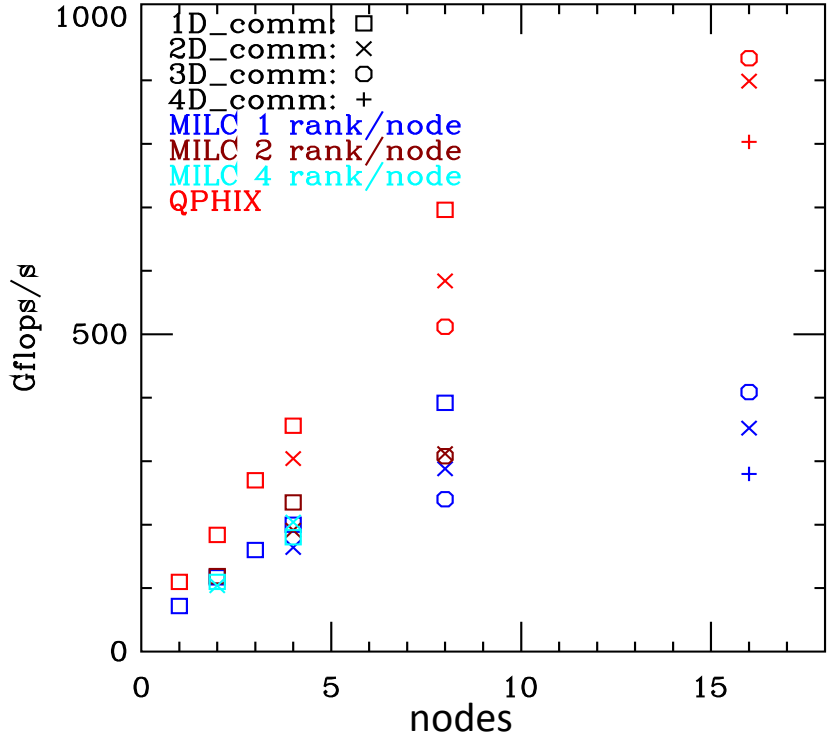
MILC+QPhiX vs. baseline MILC,
single KNL 7250 weak scaling,
up to 64 cores w. OpenMP
Lattice volume 8^3 x 24 per core

a. All data in MCDRAM.
b. MCDRAM used as cache, all data in DDR4 memory.
c. DDR4 mode (no MCDRAM use).

Broadwell



KNL



1 node = 2 sockets, 16 cores/socket

1 node = 60 cores, 2 threads/core

MILC+QPhiX vs. baseline MILC w. MPI+OpenMP
(baseline MILC code w. MPI on BDW),
multi-node weak scaling, up to 16 nodes & 4D communications
Lattice volume 24^3 x 60 per node

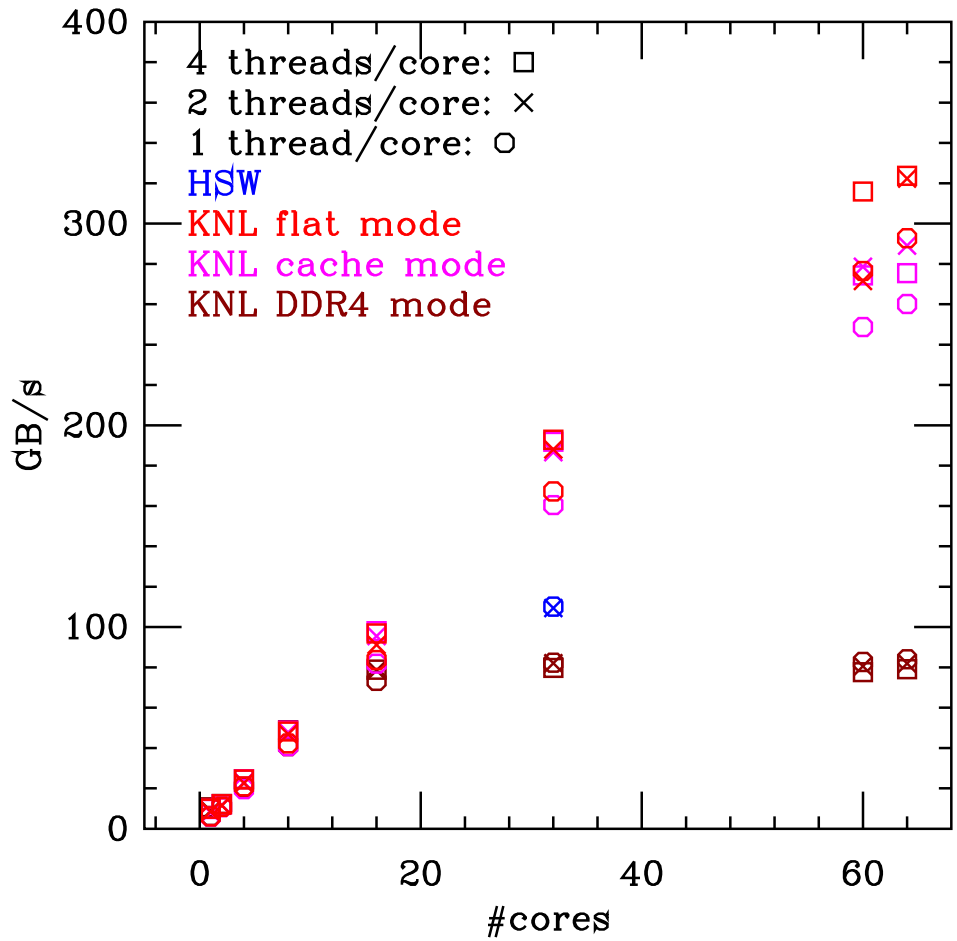
Conclusions and outlook

- Conclusions:
  - Staggered QPhiX improves multi-mass CG performance by a factor of 1.5 ~ 2.5 in DP.
  - Benefits from MCDRAM, hyperthreading.

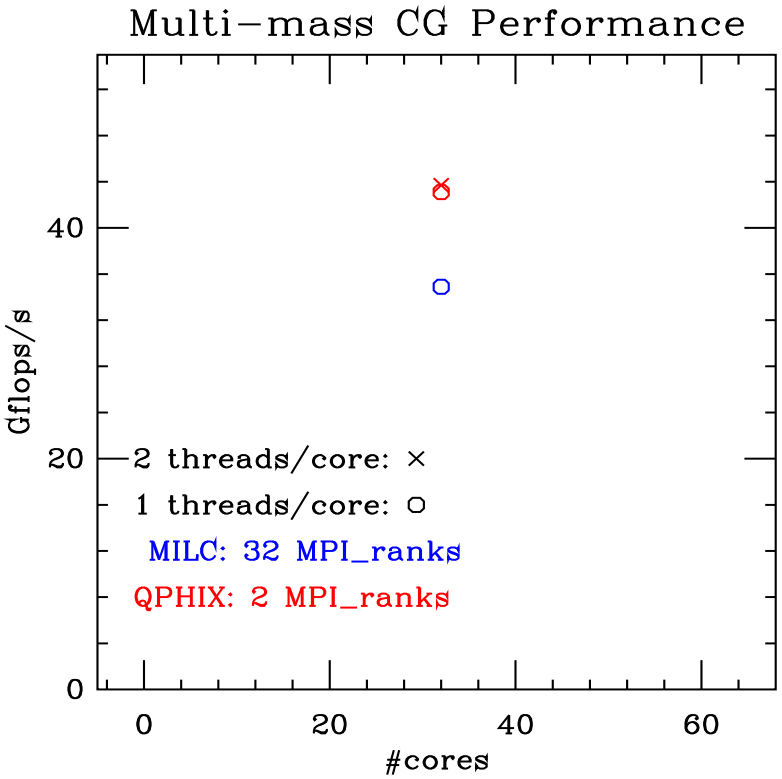
- Outlook:
  - Explore other NUMA modes: hemisphere, quadrant.
  - Omni-path network and various communication strategies for faster across-chip communications
  - Optimize other routines in the code.

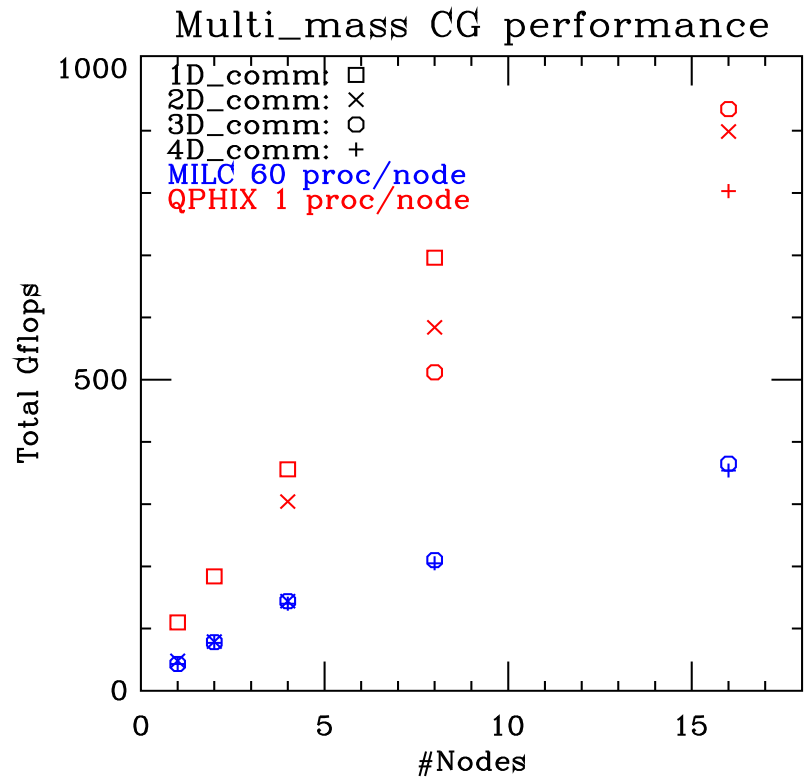
Backup slides

- QPhiX staggered Dslash BW (DP)



16x32x32x48 lattice


Multi−mass CG Performance
Gflops/s
#cores
2 threads/core: ×
1 threads/core: ○
MILC: 32 MPI_ranks
QPHIX: 2 MPI_ranks
0
20
40
60


Multi_mass CG performance
Total Gflops
#Nodes
1D_comm: □
2D_comm: ×
3D_comm: ○
4D_comm: +
MILC 60 proc/node
QPHIX 1 proc/node
0
500
1000
5
10
15

MILC staggered multi-mass Conjugate Gradient(CG)

- Algorithm: shifted polynomials in Krylov space, e.g. *B. Jegerlehner, hep-lat/9612014*

$$(M - \sigma_i) a_i = b$$

where a_i is within a set of vector solutions, each with a bare quark mass σ_l, and b is the source color vector.

Update a_i with Dslash for smallest σ_l, and the rest a_j, $j \neq l$ with local linear algorithm.

Computational requirements

$$Flops = (1205 + 15 \times masses) \times iters \times V$$

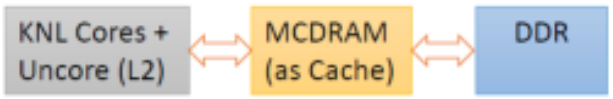
$$Bytes = ((171 + 12 \times masses) \times iters + 9 \times masses) \times V \times sizeof(complex)$$

Where $masses$, $iters$, V are number of quark masses, CG iterations, and total lattice volume. ($masses$ = 9 or 11 in our tests)

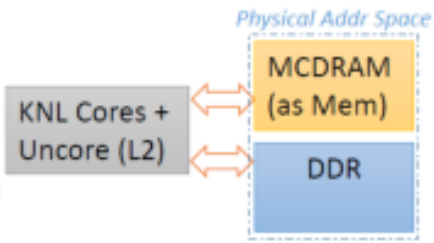
Intel Xeon Phi Knights Landing (KNL) architecture

HBM Modes

- Cache mode
  - No source changes needed to use
  - Misses are expensive (higher latency)
    - Needs HBM access + DDR access



- Flat mode
  - MCDRAM mapped to physical address space
  - Exposed as a NUMA node
    - Use numactl --hardware, lscpu to display configuration
  - Accessed through memkind library or numactl



- Hybrid
  - Combination of the above two
    - E.g., 8 GB in cache + 8 GB in Flat Mode

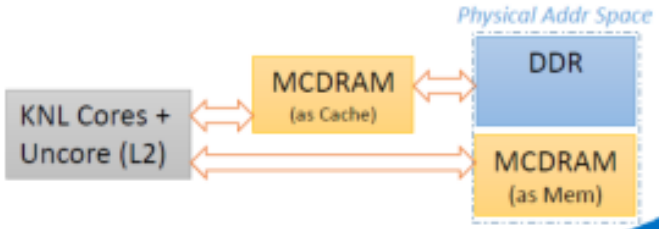


Image quoted from http://www.anandtech.com/show/9794/a-few-notes-on-intels-knights-landing-and-mcdram-modes-from-sc15

Intel Xeon Phi Knights Landing (KNL) architecture