Staggered Dslash Performance on Intel® Xeon Phi™ Architecture

Ruizi Li & Steven Gottlieb

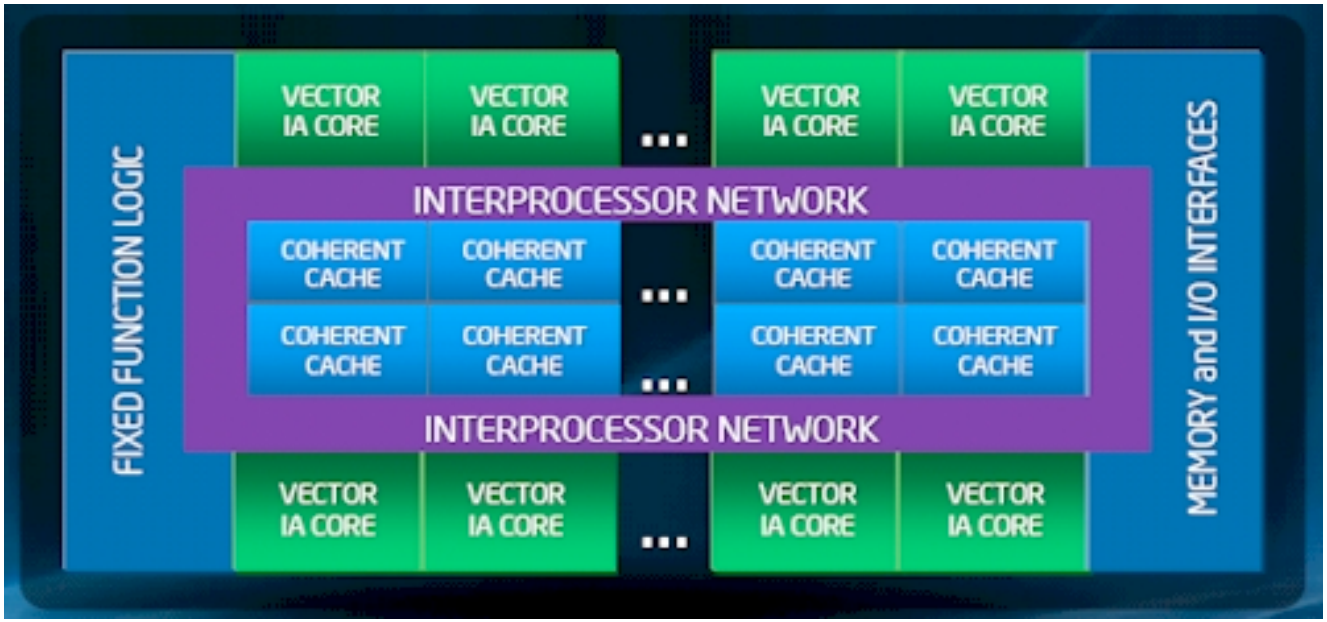
Indiana University

Outline

- Background on Intel® Xeon Phi™ (MIC) ‘Knights Corner’ architecture
- Staggered CG performance on MIC with MPI and OpenMP
- Vectorized staggered dslash on MIC
  1. Data layout
  2. Dslash performance on one MIC card
- Conclusion
- Work in progress and to do
- The future

Background on Intel® Xeon Phi™ (MIC) 'Knights Corner' architecture

- Symmetric multiprocessor (SMP) on-a-chip



	Intel® Xeon Phi™ KNC (5110p)
Number of cores	60 (4 threads per core)
CPU speed	1.05 GHz
Cache size	32KB each of L1 Icache and L1 Dcache 512KB L2 cache
Vector processing unit (VPU)	512bit – 16 float or 8 double
Maximum memory bandwidth	320Gbytes/sec
PCI express bus	5.5GT/s

Staggered CG performance on MIC with MPI and OpenMP

- MILC code 7.7.8
- Native mode on MIC
- Memory usage (in bytes for double precision) in CG routine:
  (28+2*num_qmasses)*48*nx*ny*nz*nt
- With staggered HISQ quarks, CG Flops per iteration:

  (1205 + 15*num_qmasses)*nx*ny*nz*nt

Staggered CG performance on MIC with MPI and OpenMP

- CG speed on one MIC card:
- Multiple mass inverter (9 or 11 masses), and around 50 iterations:

  MPI : 12 ~ 16GFLOPS (peak at L = 6, 8)

  OpenMP : 15 ~ 20GFLOPS (peak at L = 6)
- Single mass inverter with OpenMP (from Carleton DeTar) (L ~ 8) :

  around 600 iterations: ~ 42GFLOPS

  around or more than 1000 iterations: ~ 45GFLOPS
- Does not weak scale well on multiple MIC cards with Intel MPI: performance gain < 20% (with a few exceptions).

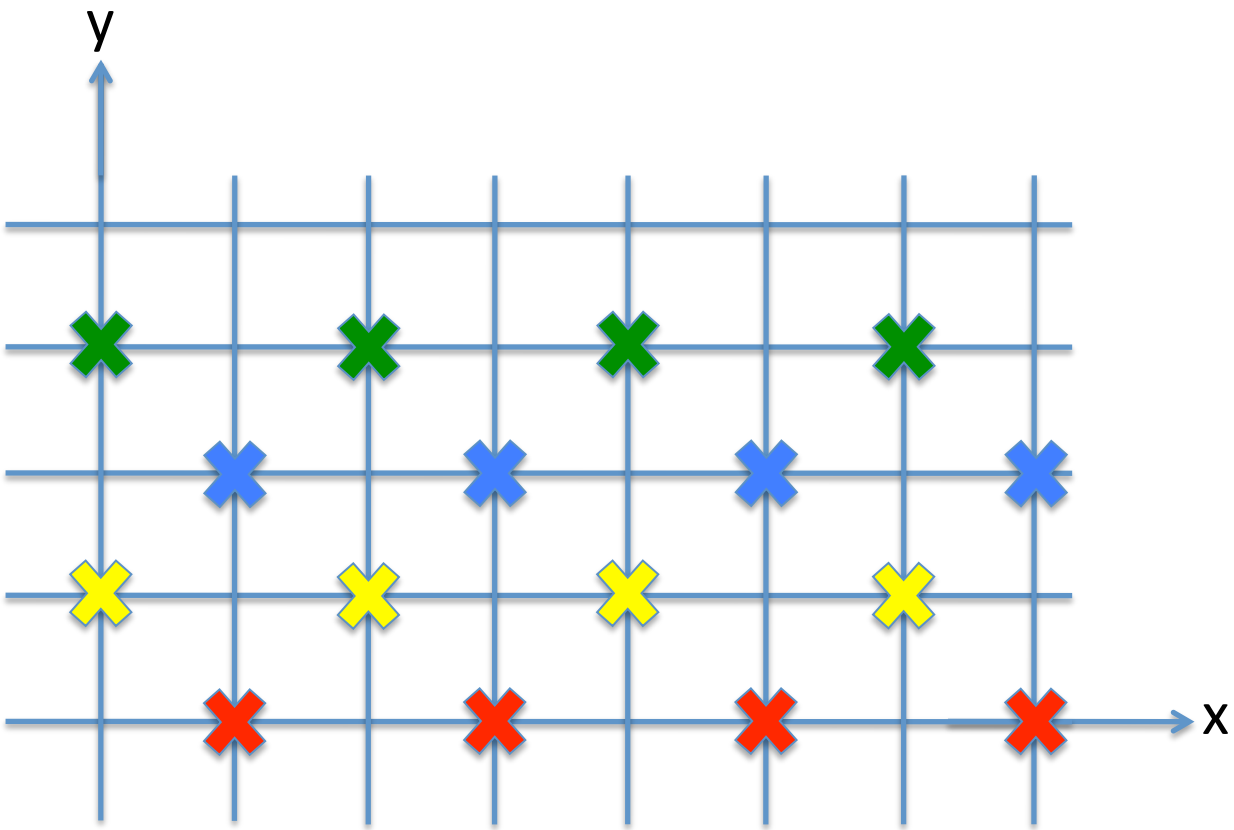
Vectorized staggered dslash on MIC

- Based on Wilson dslash code from Bálint Joó and Intel
- Naive dslash & dslash with the Naik term
- Run on one MIC card, 59 cores with OpenMP
- Options that affect performance:

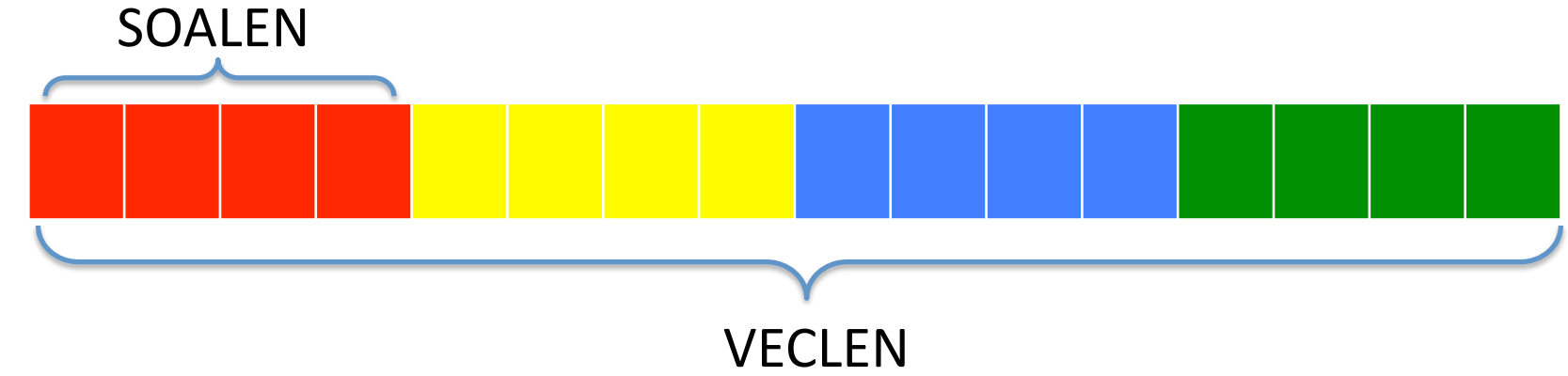
1. Compressed gauge fields storage
2. Streaming store for K-S fermion fields
3. Change of SOALEN, the way lattice being assigned to threads and cores, etc.

Data layout

- For single precision VECLEN = 16
- “Structure of arrays” (SoA) layout along x direction with length SOALEN (= 4, 8, 16)
- Data in cache & VPU also aggregate in y direction with length (VECLEN/SOALEN)
- Lattice size has to be (SOALEN*n) in x direction and (VECLEN/SOALEN*m) in y direction



VPU:



Data layout

- Storage of K-S fermion fields

  in memory : float KS[3][2][SOALEN]

  in cache : float KS[3][2][VECLEN]

- Compare to MILC data layout: complex KS[3]

- Streaming store : SOALEN = 8, 16

Data layout

- Storage of gauge fields

  in memory :

  ```
  #ifdef USE_PACKED_GAUGES
    float  Gauge[8][GROWS][3][2][VECLEN]
  #else
    float  Gauge[8][GROWS][3][2][SOALEN]
  #endif
  ```

  in cache :

  ```
    float  Gauge[8][GROWS][3][2][VECLEN]
  ```

  where GROWS = 3 or 2 for un/compressed gauges
- Compare to MILC data layout: complex Gauge[4][3][3]

Naive vectorized dslash performance on one MIC card

- Naive staggered dslash performance: with OpenMP
  1. Compressed gauges : 184 ~ 193 GFLOPS
     bandwidth : 142 ~ 153 Gbytes/sec
     with staggered phase : 179 ~ 196 GFLOPS
     bandwidth : 142 ~ 154 Gbytes/sec
  2. Uncompressed gauges : 138 ~ 141 GFLOPS
     bandwidth : 152 ~ 159 Gbytes/sec
- Compare to the vectorized naive dslash code with MILC code data layout: ~ 50 GFLOPS
- Speed increases slightly with increased SOALEN
- Rephase (calculate naive links with K-S phase) performance: 21GFLOPS; bandwidth 174Gbytes/sec.

Naive dslash code performance on one MIC card

- Lattice size : 32^3*128

	SOALEN	Compressed gauges (GFLOPS / Gbytes/sec)	Uncompressed gauges (GFLOPS / Gbytes/sec)
Disabled Streaming Store	4	184 / 147	139 / 158
	8	188 / 150	140 / 159
	16	189 / 152	139 / 158
Enabled Streaming Store	8	187 / 142	139 / 153
	16	190 / 144	140 / 154

Naive dslash code performance on one MIC card

- Lattice size 32*40*24*96

	SOALEN	Compressed gauges (GFLOPS / Gbytes/sec)	Uncompressed gauges (GFLOPS / Gbytes/sec)
Disabled Streaming Store	4	186 / 149	139 / 158
	8	187 / 150	140 / 159
	16	191 / 153	138 / 157
Enabled Streaming Store	8	189 / 143	141 / 154
	16	193 / 146	138 / 152

Naive dslash performance with K-S phase

- lattice size 32^3*128, compressed gauges

SOALEN	Streaming Store (GFLOPS / Gbytes/sec)	No Streaming Store (GFLOPS / Gbytes/sec)
4	/	178 / 142
8	193 / 146	189 / 151
16	193 / 147	185 / 149

Naive dslash performance with K-S phase

- lattice size 32*40*24*96, compressed gauges

SOALEN	Streaming Store (GFLOPS / Gbytes/sec)	No Streaming Store (GFLOPS / Gbytes/sec)
4	/	179 / 143
8	196 / 148	193 / 154
16	195 / 148	185 / 148

Dslash code with the Naik term on one MIC card

- Staggered dslash with the Naik term performance: with OpenMP
  1. Compressed gauges : 178 ~ 193 GFLOPS
     bandwidth : 130 ~ 141 Gbytes/sec
  2. Uncompressed gauges : 136 ~ 140 GFLOPS
     bandwidth : 144 ~ 149 Gbytes/sec
- Performance of calculating the 'long links' (production of gauge fields in the Naik term) from naive links:
  1. Compressed gauge : ~ 304 GFLOPS; bandwidth ~ 147 Gbytes/sec
  2. Uncompressed gauge : ~ 184 GFLOPS; bandwidth ~ 134 Gbytes/sec

Dslash code with the Naik term on one MIC card

- Lattice size : 32^3*128

	SOALEN	Compressed gauge (GFLOPS / Gbytes/sec)	Uncompressed gauge (GFLOPS / Gbytes/sec)
Disabled Streaming Store	4	179 / 131	136 / 145
	8	192 / 141	139 / 149
	16	193 / 142	135 / 144
Enabled Streaming Store	8	193 / 137	140 / 146
	16	193 / 137	140 / 147

Dslash code with the Naik term on one MIC card

- Lattice size 32*40*24*96

	SOALEN	Compressed gauge (GFLOPS / Gbytes/sec)	Uncompressed gauge (GFLOPS / Gbytes/sec)
Disabled Streaming Store	4	177 / 132	136 / 145
	8	196 / 143	141 / 151
	16	192 / 137	136 / 145
Enabled Streaming Store	8	197 / 142	143 / 150
	16	196 / 140	132 / 149

Conclusion

- Staggered CG speed with MPI and OpenMP goes up to 45GFLOPS for the single mass inverter and decreases with the multiple mass inverter and fewer iterations.
- Communications between coprocessors seem to be the bottleneck for code performance.
- With SoA data layout, vectorized staggered dslash performance:

  with compressed gauges: 180 ~ 195GFLOPS,

  with uncompressed gauges: 138 ~ 140GFLOPS;

  bandwidth: 140 ~ 160Gbytes/sec.
- Data layout is crucial to improve code performance.
- Streaming store, change of SOALEN, or the Naik term doesn't affect code speed much.

Work in progress and to do

- Include vectorized dslash code in the MILC code staggered CG routine, in progress.
- Routine for loading fat links.
- MPI Proxy for code running on multiple MIC cards. Try new version of Intel MPI Library 5.0 beta.
- Benchmarks for vectorized MILC CG code.

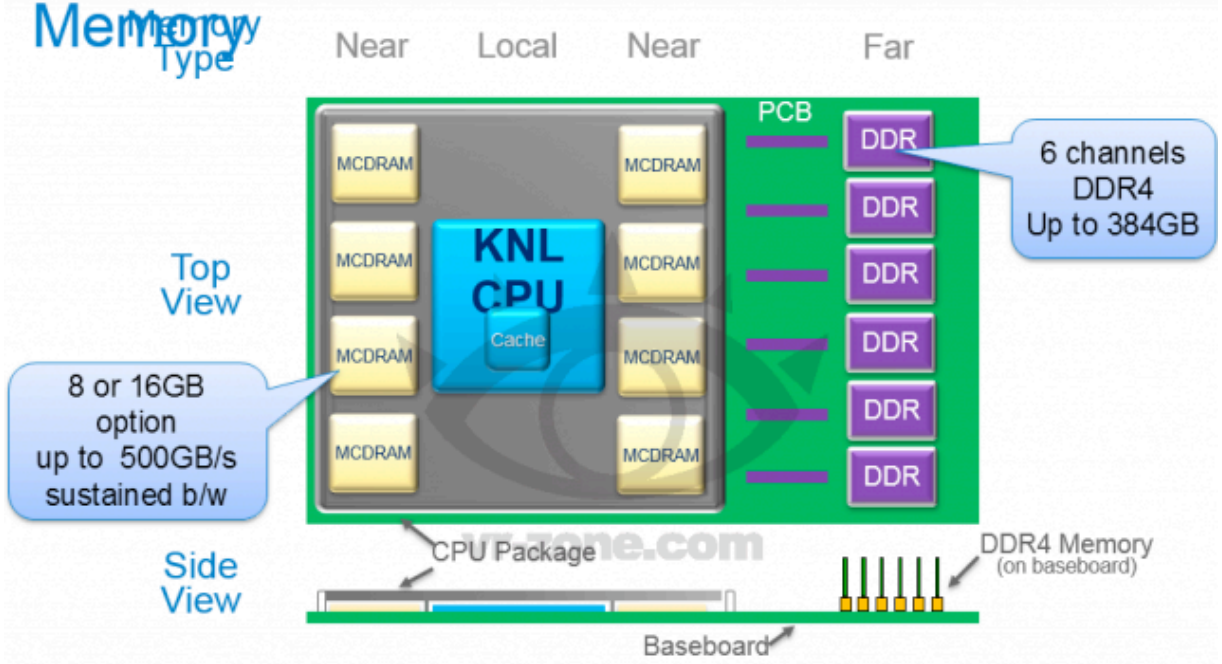
The future

- NERSC-8: Cori (2015/2016)
  1. Ten times increase in sustained performance than NERSC-6 Hopper system.
  2. More energy-efficient many-core architectures: next-generation Intel® MIC architectures.

The future

- Next generation of the MIC architecture in 2015: ‘Knights Landing’ (KNL)
  1. Standalone CPU.
  2. Including near and far memory, using High-Bandwidth In-Package Memory (HBW IPM or MCDRAM) with the bandwidth up to 500GB/s.
  3. Two VPUs per core.

Knights Landing Integrated On-Package Memory

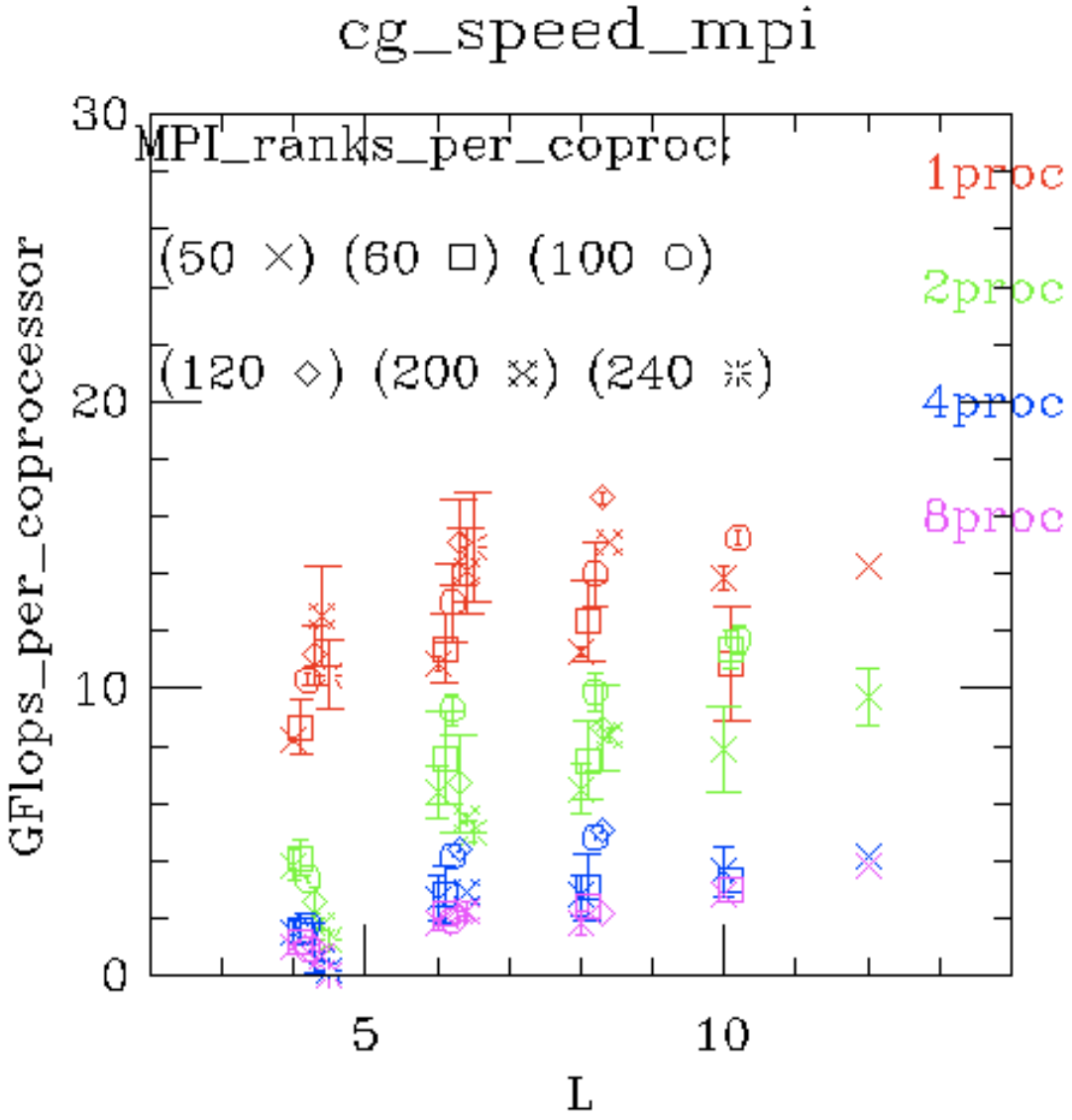


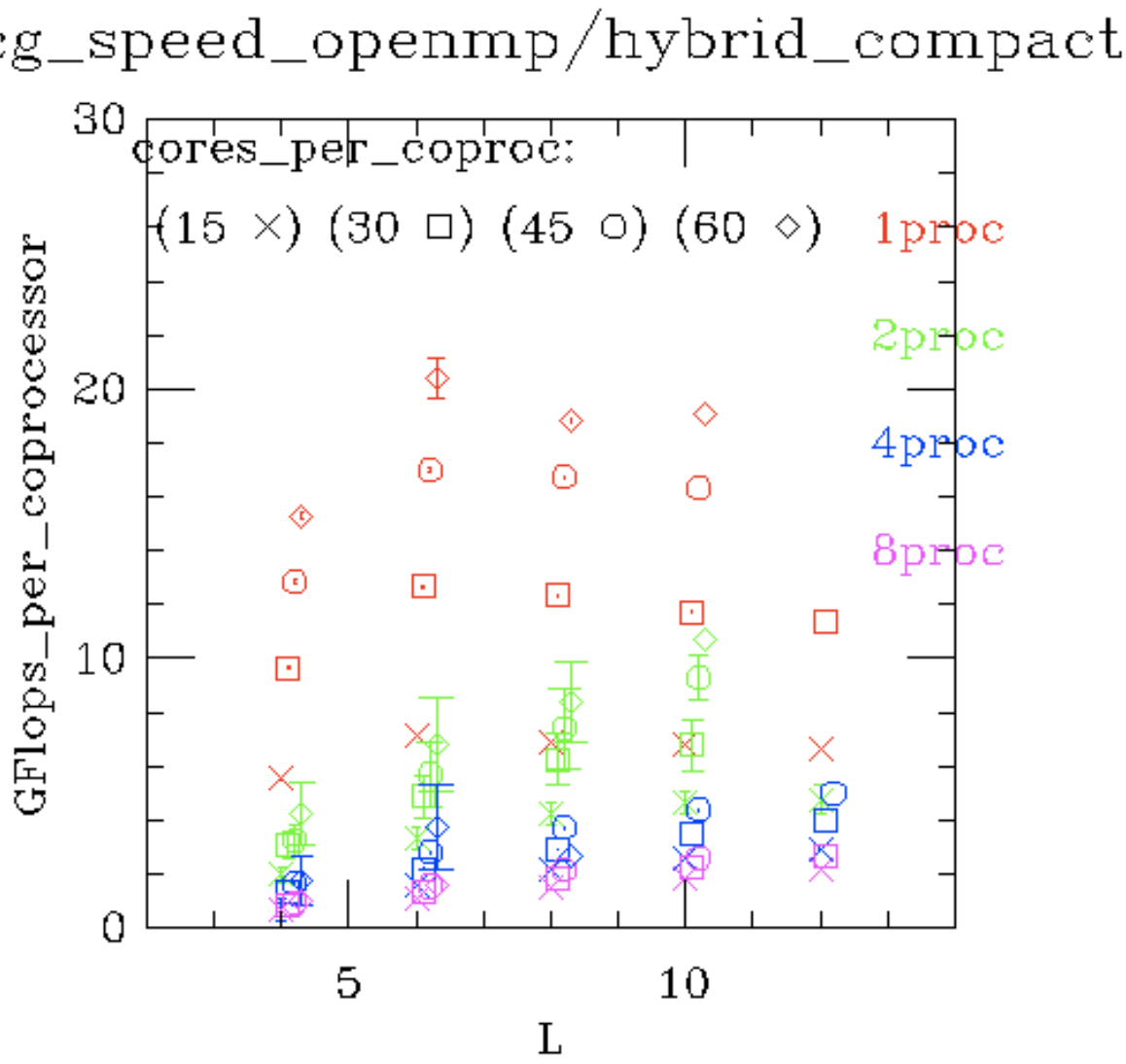
Integrated on-package MCDRAM brings memory nearer to CPU for higher memory bandwidth and better performance

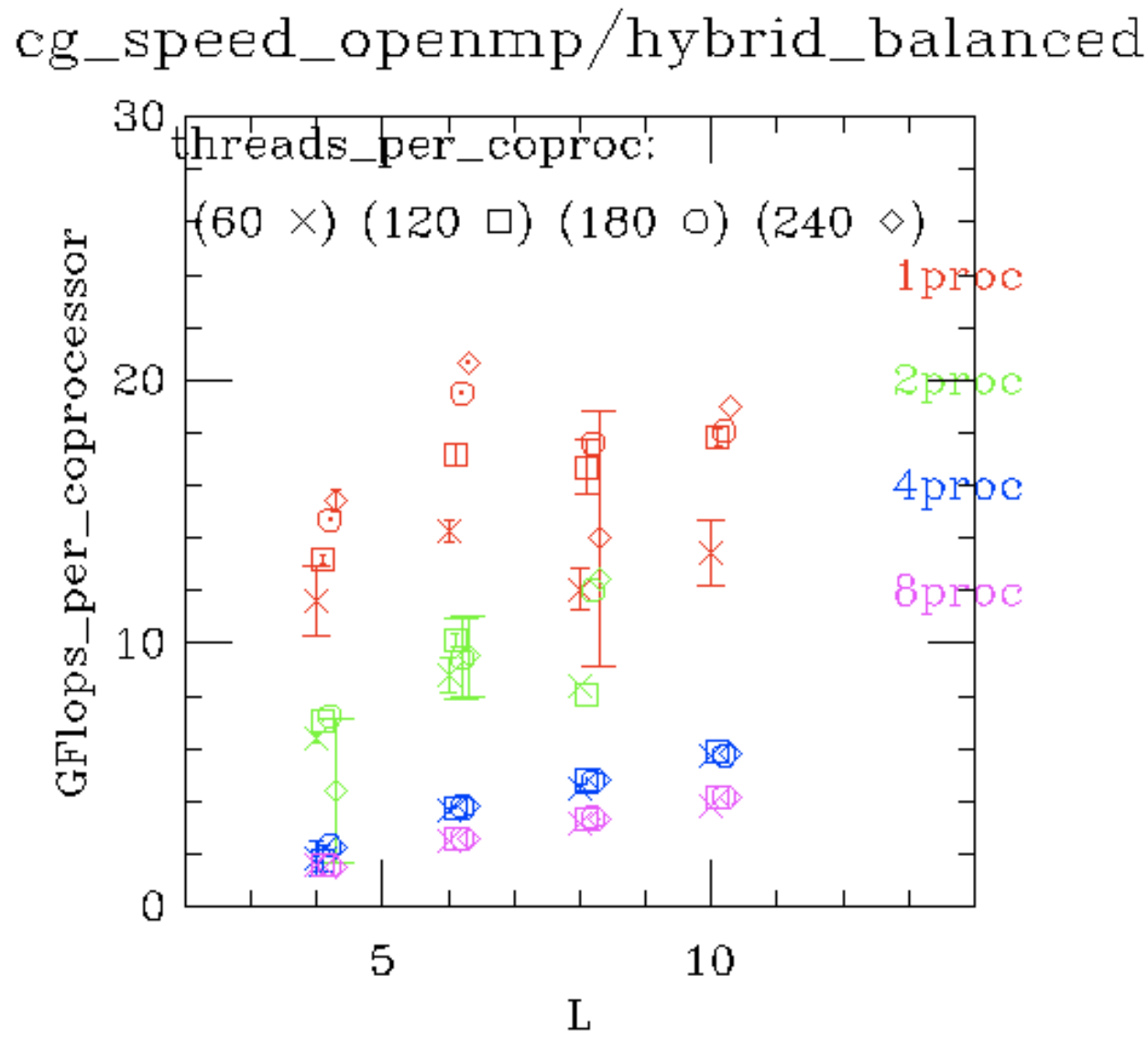
agram is for conceptual purposes only and only illustrates a CPU and memory die
t to scale and does not include all functional areas of the CPU, nor does it represent

Appendix

- Plot of staggered CG performance with MPI
- Plot of staggered CG performance with OpenMP compact distribution
- Plot of staggered CG performance with OpenMP balanced distribution


cg_speed_mpi
MPI_ranks_per_coproc
(50 ×) (60 □) (100 ○)
(120 ◇) (200 ⋆) (240 ※)
1proc
2proc
4proc
8proc
GFlops_per_coprocessor
30
20
10
0
5
10
L


cg_speed_openmp/hybrid_compact
cores_per_coproc:
(15 ×) (30 □) (45 ○) (60 ◇)
1proc
2proc
4proc
8proc
GFlops_per_coprocessor
30
20
10
0
5
10
L


cg_speed_openmp/hybrid_balanced
threads_per_coproc:
(60 ×) (120 □) (180 ○) (240 ◇)
1proc
2proc
4proc
8proc
GFlops_per_coprocessor
30
20
10
0
5
10
L