Exploring KNL Performance of the MILC Code

Steven Gottlieb, Ruizi Li
Indiana University

MILC Collaboration

QCDNA 2016
Edinburgh
August 3, 2016

Introduction

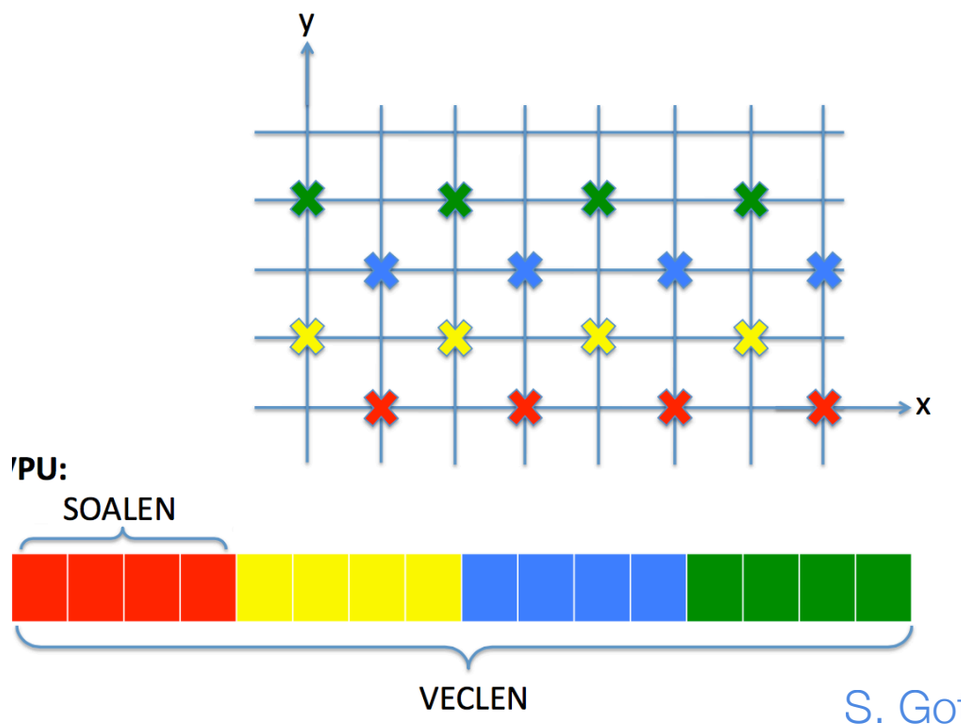
- ✦ BEACON project & KNC
- ✦ QPhiX
- ✦ Indiana Intel Parallel Computer Center
- ✦ NESAP project
- ✦ LQCD Acquisition Committee
  - What performance can we expect on untuned code?
  - How does that compare with ordinary (multi-core) cluster?
- ✦ How well will Xeon Phi perform for gauge configuration generation?
- ✦ I will concentrate on single-node performance
- ✦ Ruizi Li will discuss multi-node results

BEACON & KNC

- ✦ BEACON project at Oak Ridge National Lab provided support during Ruizi Li's Ph.D. studies and access to first generation Intel Xeon Phi processor (Knights Corner)
- ✦ We tried MILC with both MPI and OpenMP
  - in double precision about 24 GF/s was the best we could do
- ✦ QPhiX was generalized to support staggered quarks
  - greatly improved single node performance (140 GF/s SP Dslash w/ o compression)
  - MPI performance was not good enough for multi-node running.
  - Unsuccessfully tried Intel's MPI proxy that worked with Chroma
- ✦ See R. Li & S.G., PoS Lattice 14 (2015) 034; arXiv:1411.2087

QPhiX

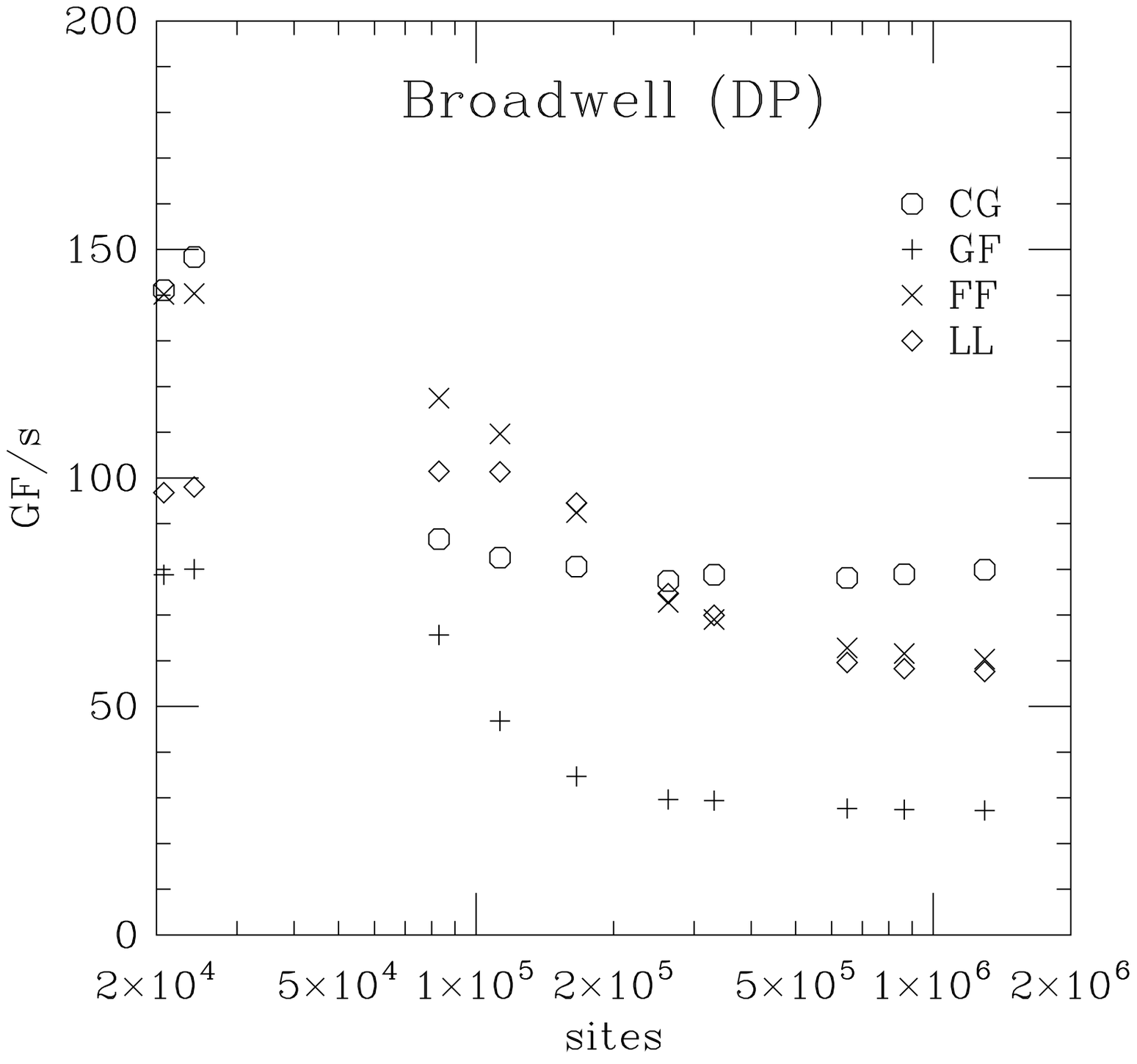
- A solver library for Intel Xeon Phi developed by JLab and Intel
- Lattice QCD on Intel Xeon Phi Processors, B. Joó, D.D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnany, V.W. Lee, P. Dubey, and W. Watson III, ISC 2013, Lecture Notes in Computer Science, Vol. 7905, 40 (2013), J.M Kunkel, T. Ludwid, and WH.W. Meuer (Eds.).



Fermibench

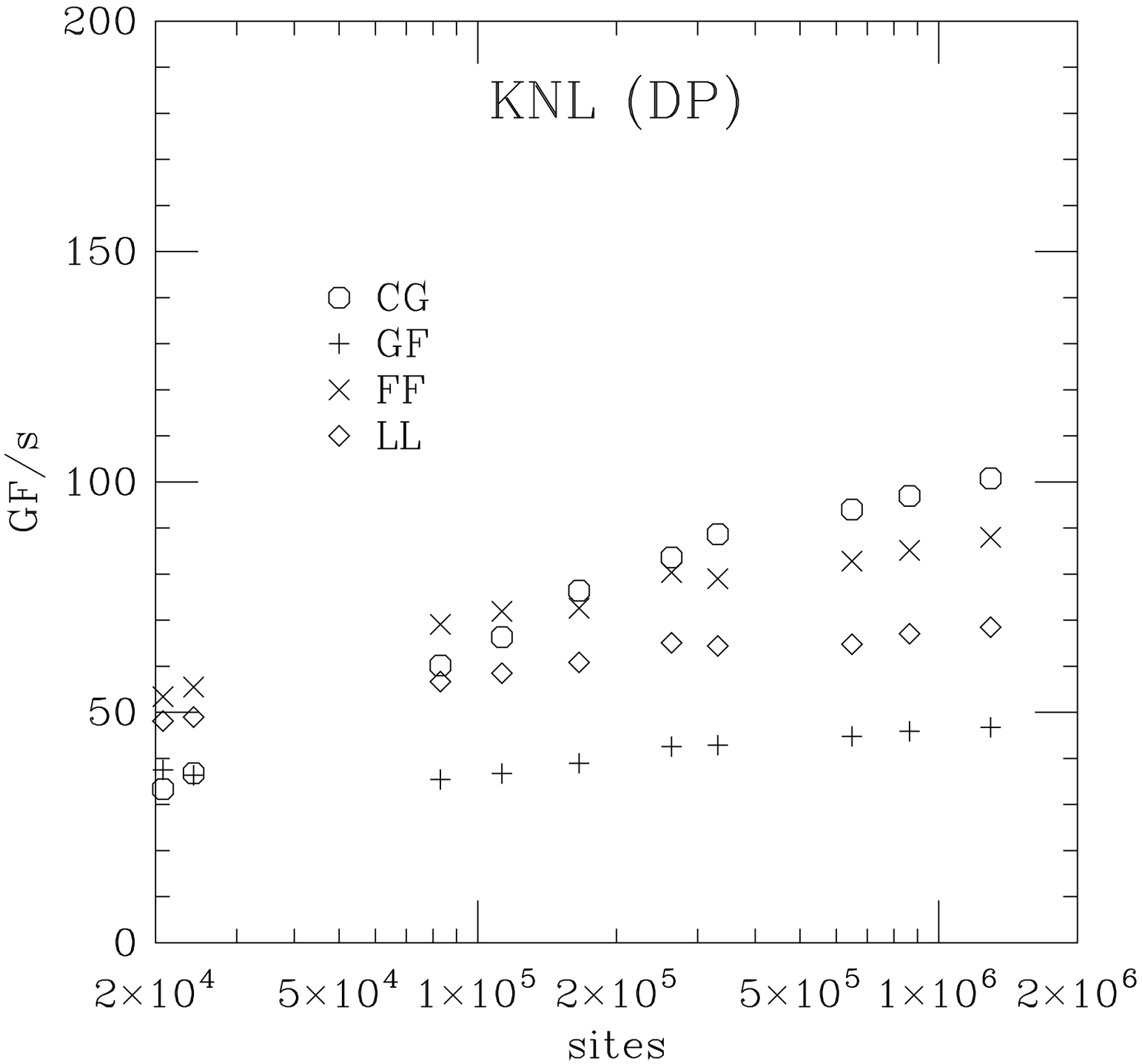
- ✦ USQCD uses solver performance for domain wall, staggered, and Wilson/clover to rate its clusters and for acquisition benchmarks.
- ✦ Fermibench contains MPI executables and scripts that can be sent to vendors.
- ✦ Note that KNL Fermibench results use an executable precompiled for generic x86 hardware
  - uses SSE
  - Broadwell E5-2690 2.6 GHz, dual socket 16 core part
  - 24 core benchmarks from Amitoj Singh & Don Holmgren
- ✦ KNL results obtained at IU on *chess* an Intel Xeon Phi 7250 development node (68 cores, 1.2 GHz)

Broadwell Fermibench



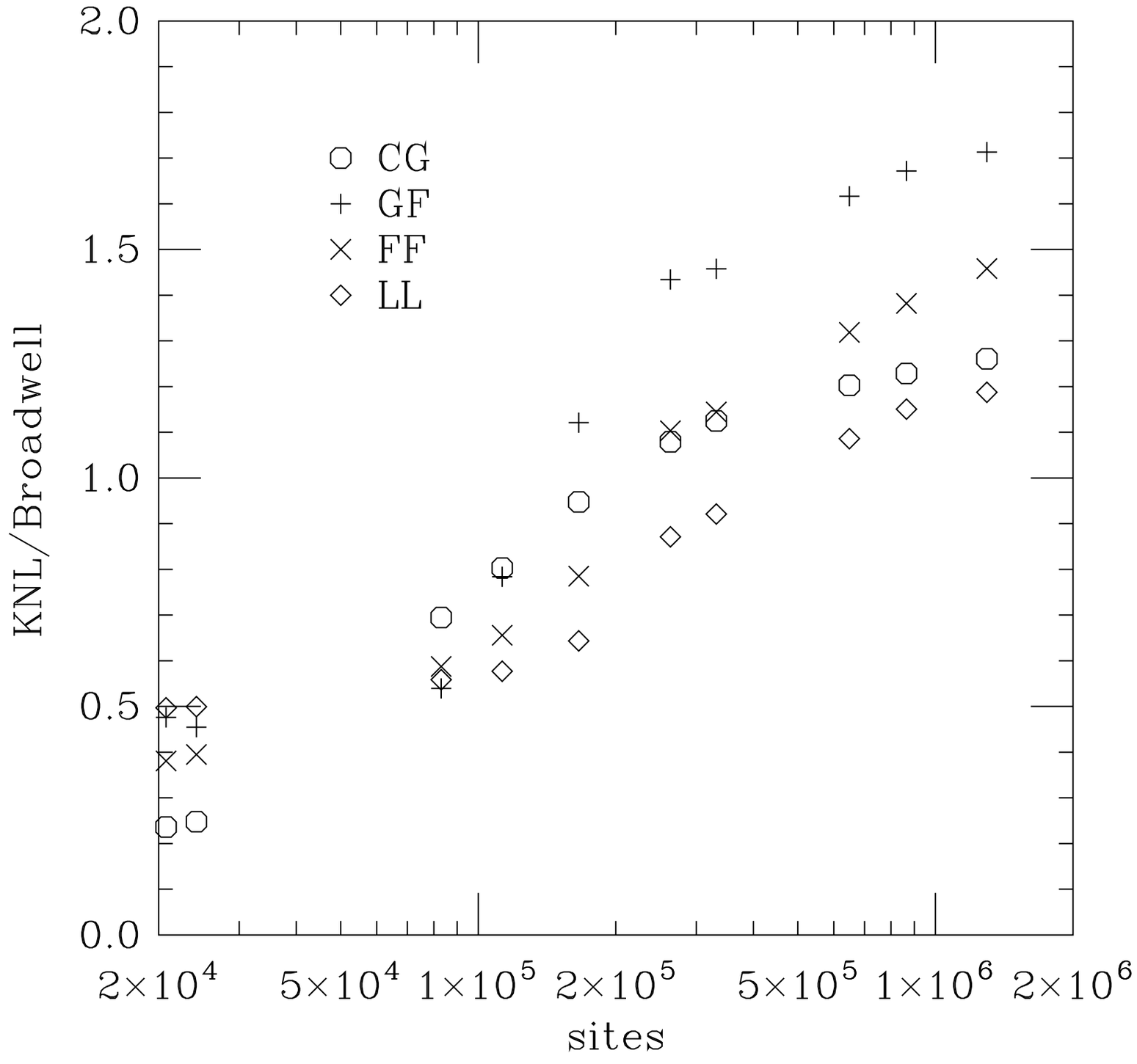
- Uses 24 of available cores
- performance for conjugated gradient (CG), gauge force (GF), fermion force (FF), and link fattening (LL)
- noticeable cache effect, with performance decreasing as volume increases

KNL Fermibench



- Uses 64 of available cores
- Volume dependence is very different from Broadwell
- Global sums probably slow performance of CG at small volume

KNL vs. Broadwell



- Ratio of DP performance on KNL versus Broadwell shows KNL with better performance at large volume, presumably because of better MCDRAM speed than DDR memory on Broadwell node

MILC Baseline

- Challenge of Xeon Phi is to make use of multiple levels of parallelism:
  - SIMD vector unit
  - on-chip many-core architecture
  - multi-node
- MILC has supported
  - SSE2 SIMD parallelism
  - MPI
  - OpenMP
- The OpenMP support started in 2000, but was never found that useful, so was dropped until recently

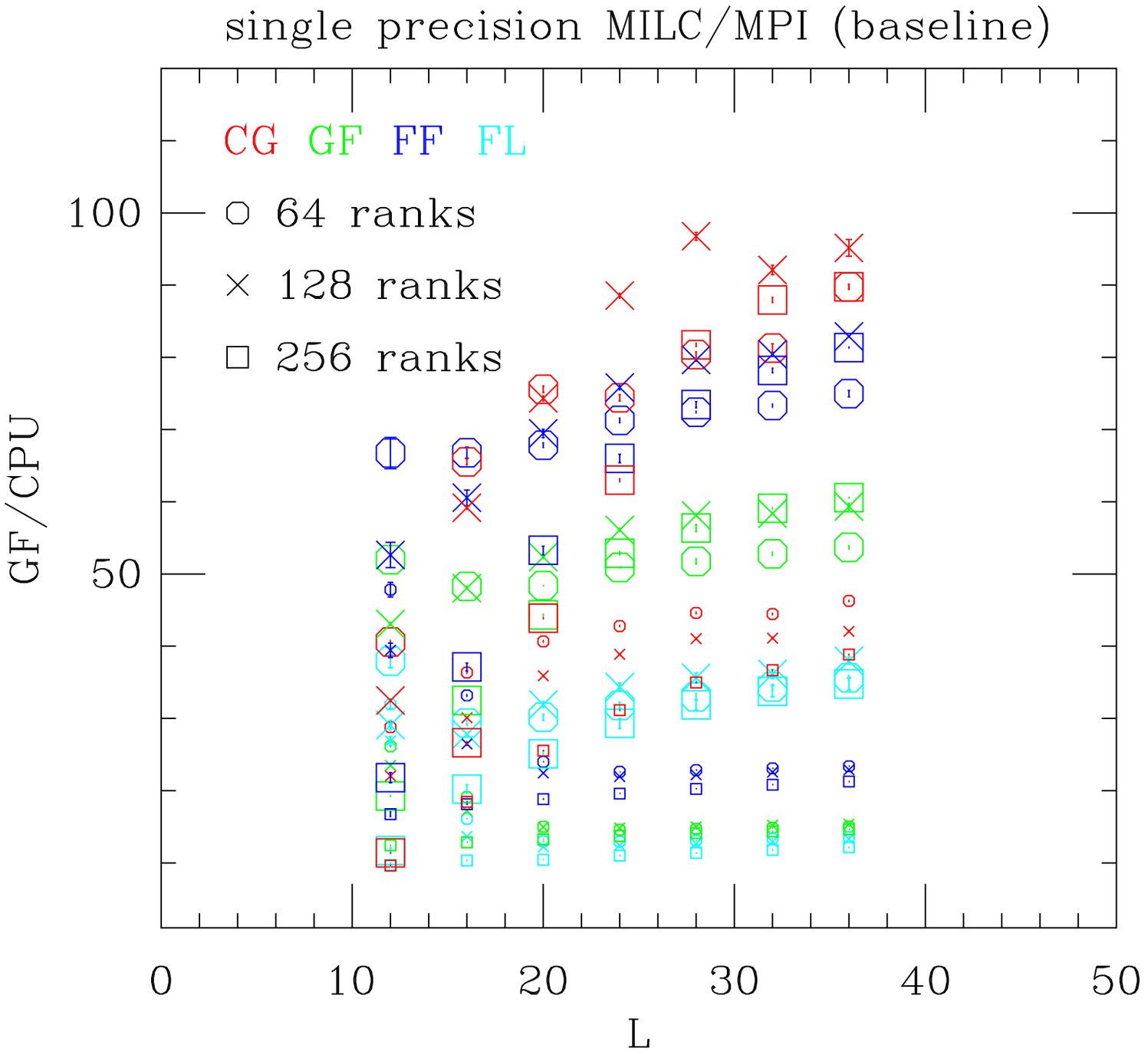
SIMD Vectorization

- SSE2 vectorization works well for optimization of basic SU(3) library of operations
- However, 8 or 16 way parallelism in double or single precision, respectively, requires a new approach,
  - except for operations like vector addition, scalar multiply and add, etc. on an entire field.
- Similar to challenge posed by GPU programming
  - Requires a new layout of data
  - We are adopting QPhiX approach invented by Balint Joo at JLab and several Intel software engineers
  - They supported solver for Wilson/Clover solver
  - Ruizi Li, as Ph.D. student, developed staggered dslash code on KNC.

MPI

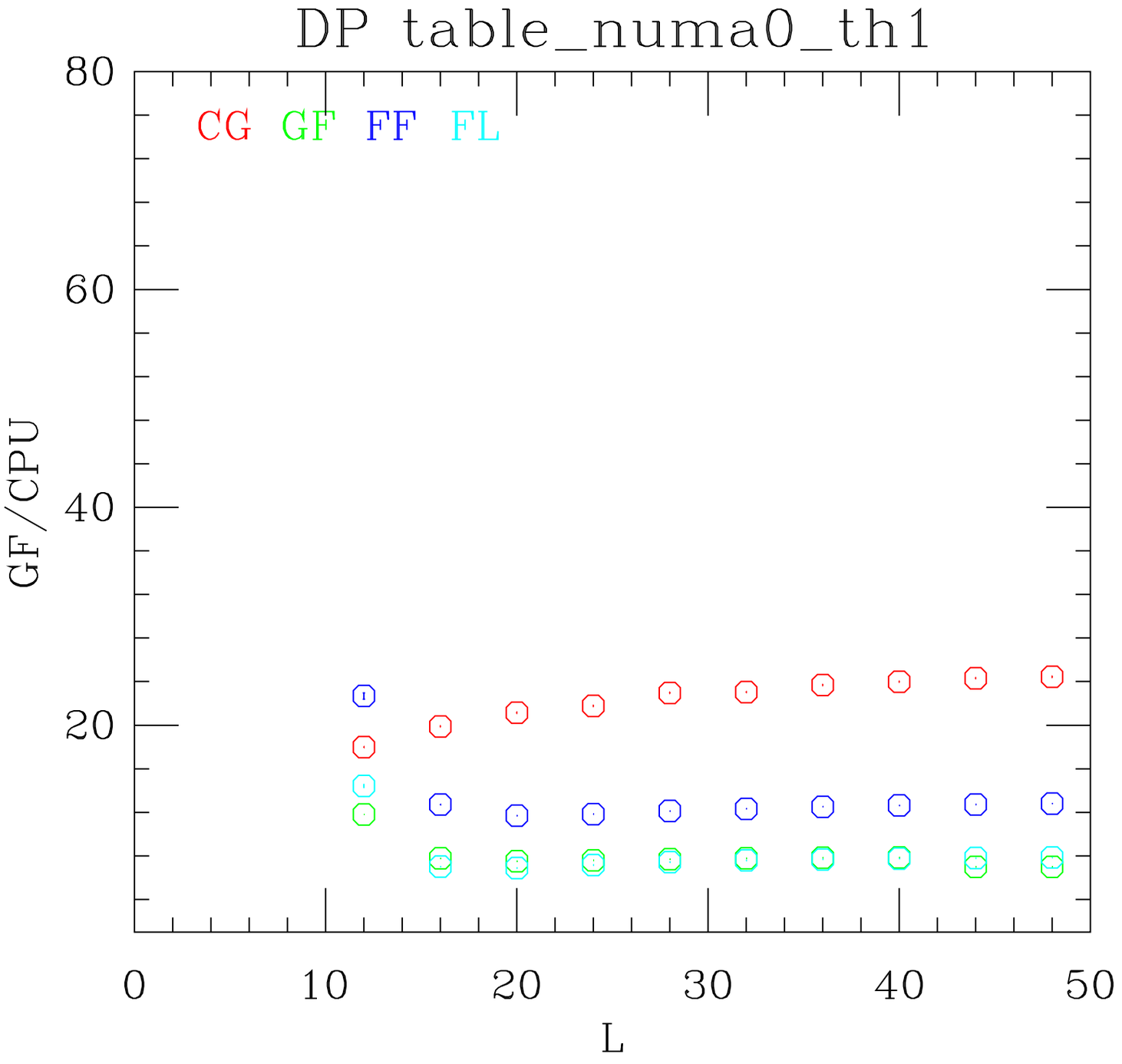
- MILC 7.8.0 was used for pure MPI tests
- Beware of results with >64 ranks as may not have had only 64 cores with 2 or 4 hyperthreads running.
  - need to work on MPI core assignment
- Looked at both single and double precision
- Looked at four major phases of code
  - link smearing denoted FL in the graphs
- Used numactl to allocate memory in MCDRAM or DDR

Single Precision MPI Baseline



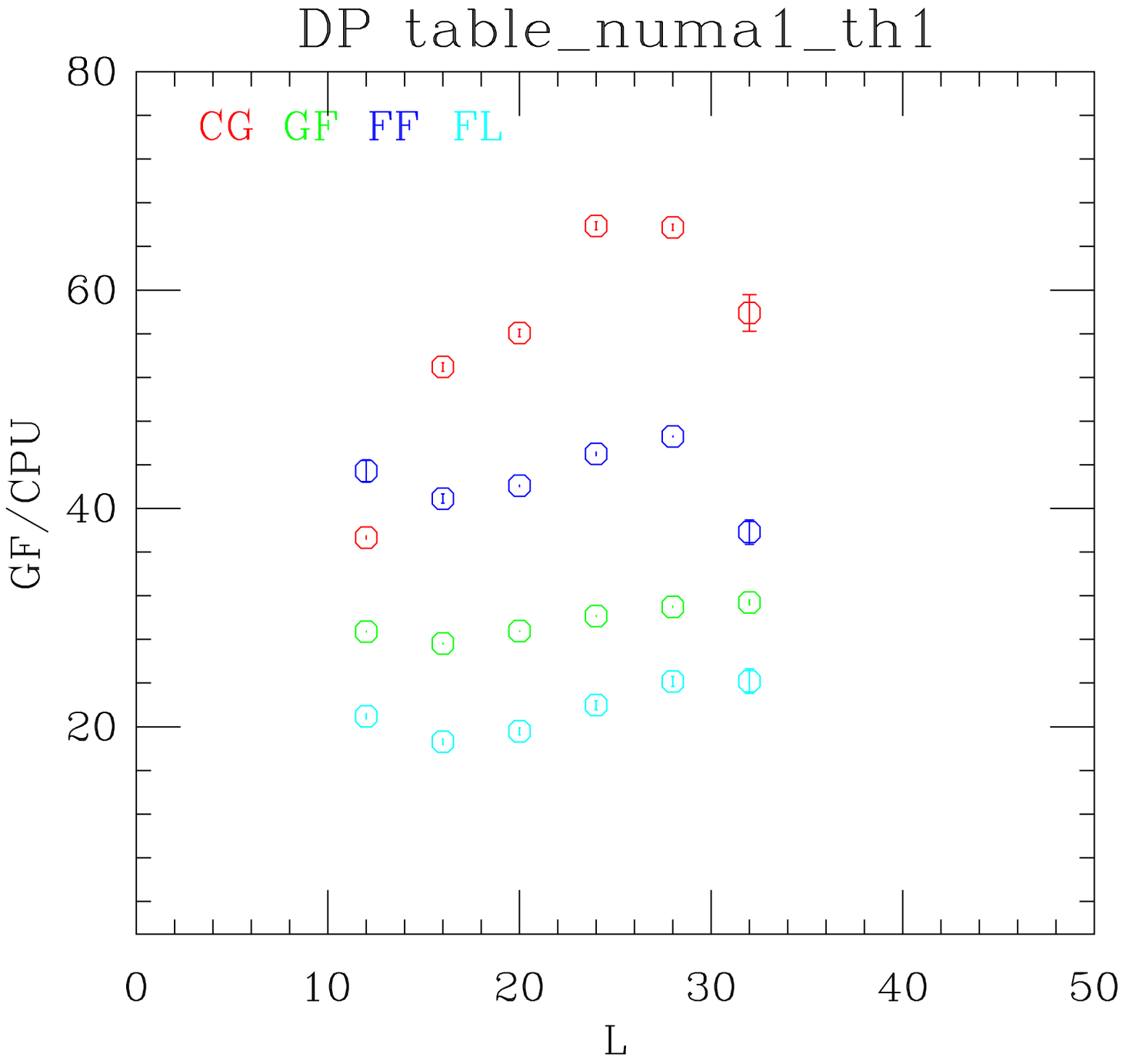
- For large enough volume can reach 60-100 GF/s for CG and fermion force when running from MCDRAM
- Multi-mass CG solver with 9 or 11 masses
- When restricted to DDR, only 10-50 GF/s

Double Precision MPI Baseline I



- Running from DDR memory only
- Volume L^4
- Multi-mass CG solver with 9 or 11 masses
- <24 GF/s
- next slide shows improvement from MCDRAM

Double Precision MPI Baseline II



- Running from MCDRAM memory only
- Volume L4
- Multi-mass CG solver with 9 or 11 masses
- 60+ GF/s for CG, but lower performance for rest of code
- have also run with multiples of 64 ranks

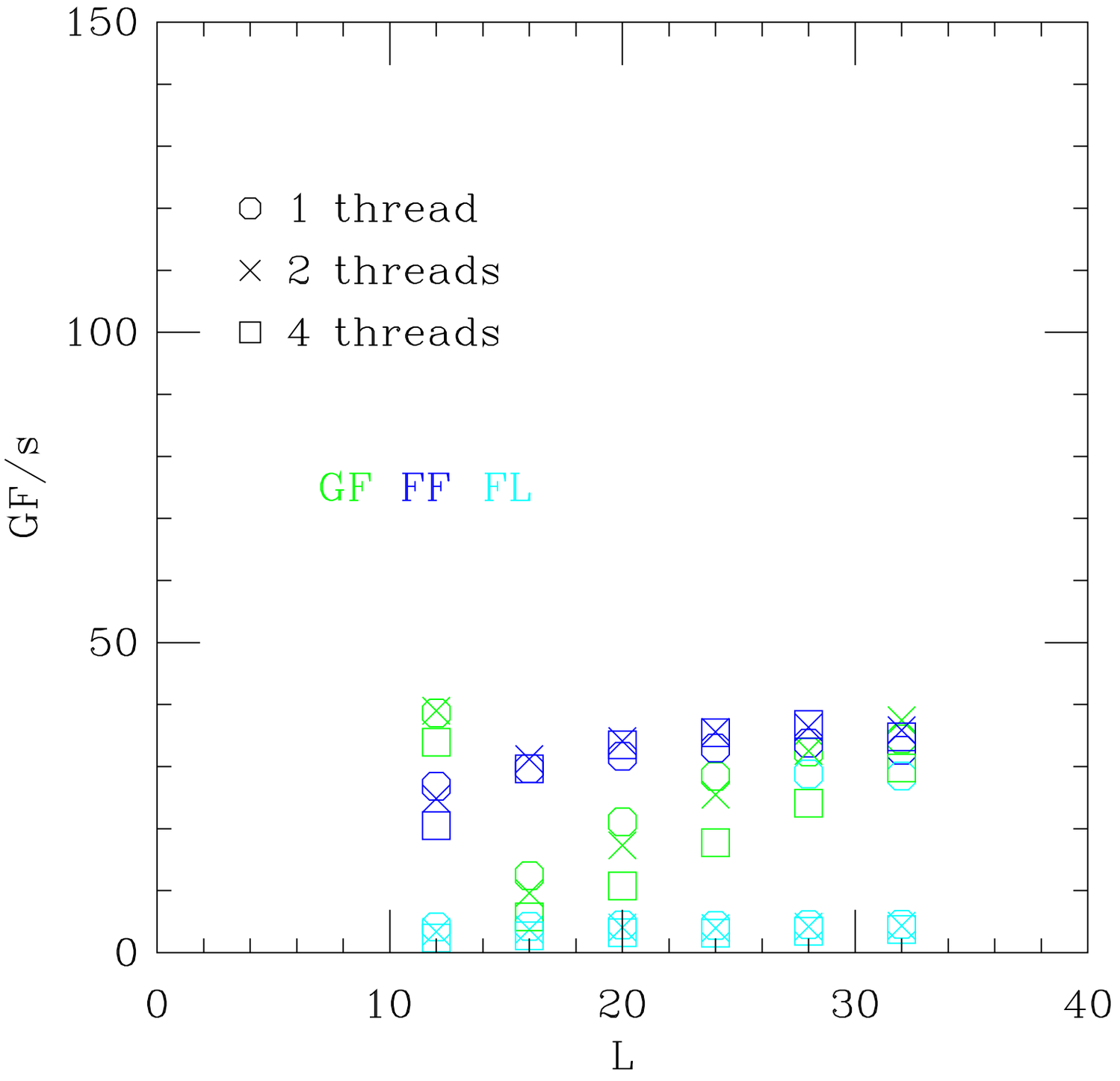
OpenMP I

- ✦ MILC code support for OpenMP started in 2000.
  - SDSC IBM SP
  - Benchmarking MILC Code with OpenMP and MPI, S.G. & S. Tamhankar, Nucl.Phys.Proc.Suppl. 94 (2001) 841-845, arXiv:hep-lat/0011037
- ✦ Replaces FORALLSITES with FORALLSITES_OMP
  - new macro includes ‘omp parallel for’ pragma
  - must examine each loop to identify private variables
  - Was not of much help 16 years ago
    - OpenMP compiler quality?
- ✦ Was soon dropped from distribution, but revived in the past few years for Xeon Phi (MIC)

OpenMP II

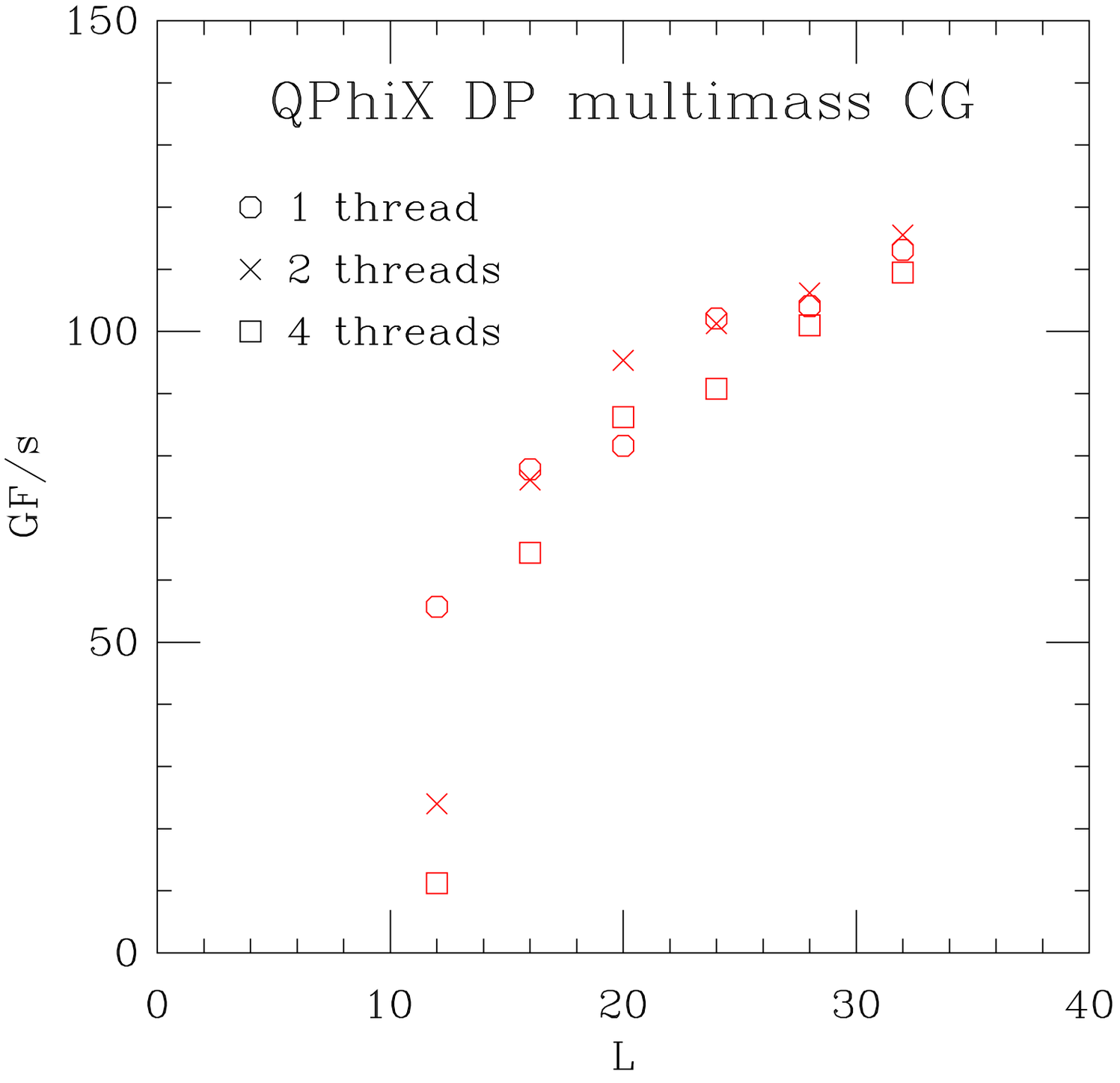
- ✦ During Dungeon session at Intel concentrated on CG & QPhiX
  - However, soon became clear that addition of OpenMP was haphazard
  - E.g.: double precision, 16^4
    - fermion force: 30 GF/s
    - gauge force: 3.5 GF/s
    - link smearing: 4.2 GF/s
- Obviously, big problem with latter two parts of code
  - Rama Malladi of Intel identified two gauge force performance issues
  - On July 18, I incorporated his fixes
  - During Lattice ’16, identified loops missing OMP in link smearing
    - Don’t try to fix code when you are that jet lagged.

OpenMP III



- These modules are not vectorized
- Starting:
  - GF 3.5 GF/s
  - FF 30 GF/s
  - FL 4.2 GF/s
- Rama's suggestion works for GF
- FL needs to be redone!!

OpenMP IV



- CG uses QPhiX
- This does not include remapping time. (Only a few iterations.)

Conclusions

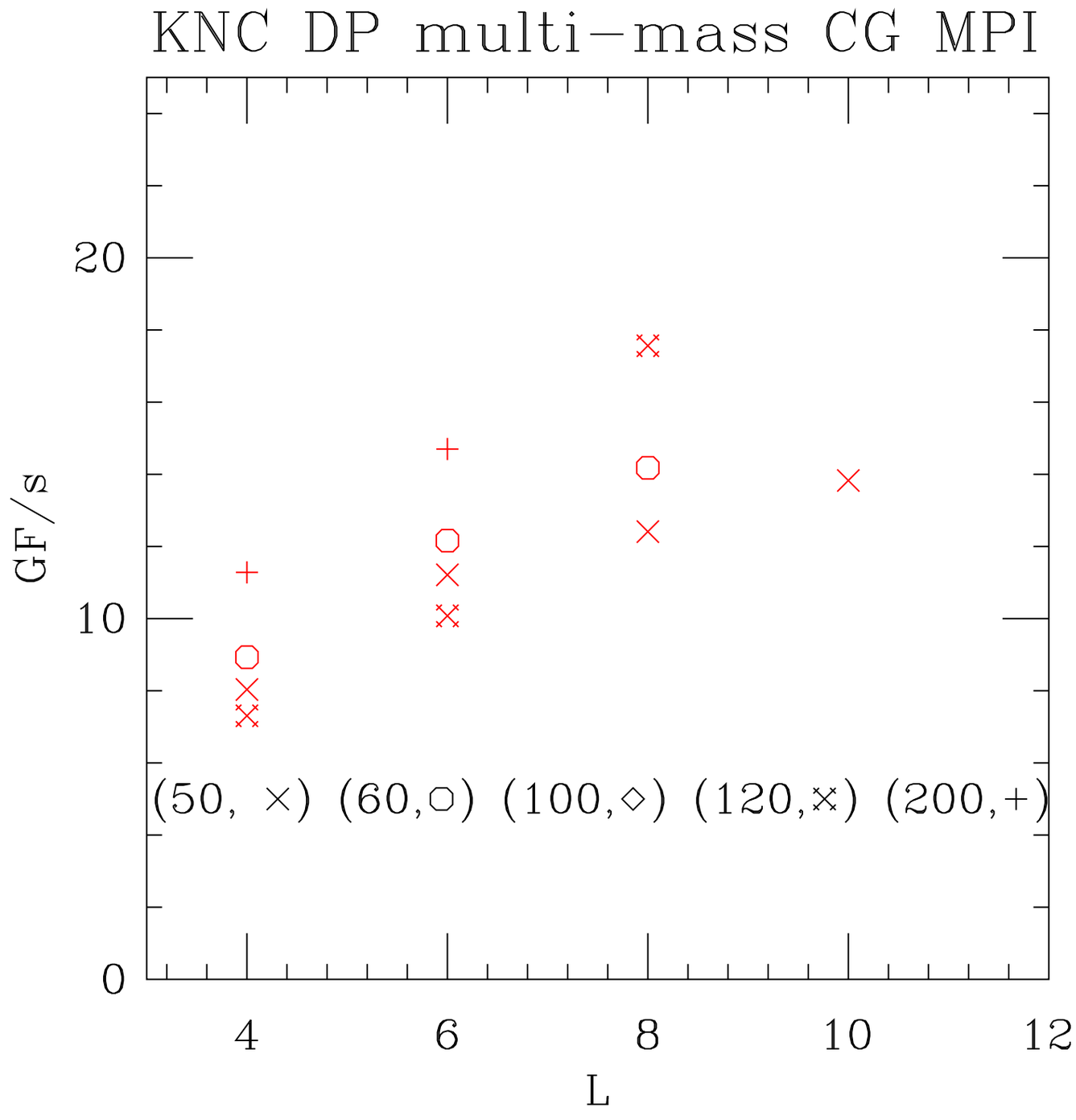
- ✦ QPhiX approach provides a significant speedup for MILC HISQ CG
- ✦ Other parts of code need attention
  - gauge force is next module being ported to QPhiX
- ✦ We are also very interested in Boyle's Grid approach
  - QPhiX now supports the layout used in Grid.
- ✦ We have plenty of work ahead if we wish to use Cori 2 @ NERSC, Theta & Aurora @ ALCF efficiently for gauge generation.
- ✦ Intel has been very generous with its support of our effort to improve performance of MILC code on Xeon Phi.

Thanks

- ✦ Ashish Jha has been organizing weekly calls of our NESAP team
- ✦ MILC: Carleton DeTar, Doug Toussaint
- ✦ Intel: Dhiraj Kalamar, Munara Tolubaeva, Rama Malladi, Thanh Phung
- ✦ NERSC: Doug Doerfer
- ✦ JLab: Balint Joo

KNC MILC Baseline I

- Performance of MILC MPI code on 60 core KNC 5110P coprocessor
- volume is L^4



KNC MILC Baseline II

- Performance of MILC OpenMP code on 60 core KNC 5110P coprocessor
- volume is L^4