SAND2017-1079PE

Domain Decomposition Solver Status and Future Plans

Applied Computer Science Meeting
Lawrence Livermore National Laboratory
February 6-10, 2017

COMPSIM
STRUCTURAL DYNAMICS

Ryan G. Coleman, Clark R. Dohrmann
Sierra Structural Dynamics Team

U.S. DEPARTMENT OF ENERGY | NNSA

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Acknowledgements

- Clark Dohrmann
- Sierra Structural Dynamics Team
  - Garth Reese, David Day, Tim Walsh, Scott Miller, Lynn Munday, Greg Bunting, Zeke Racca, Julia Seymour
- Andrew Bradley – HTS Solver
- Mark Merewether, Chris Forster, Mike Glass, Victor Brunini & Dave Glaze – Mutrino Help & Tips
- Kendall Pierson, Julia Plews, San Le – Solver integration & testing
- Garth Reese & David Day – Pardiso integration
- Paul Wolfenbarger & Rich Drake – OpenMP testrun & DevOps support

Outline

- Intro to Structural Dynamics / Linear solver
- Performance of Solvers
- Performance for Structural Dynamics problem
- Ongoing & Future Work

Domain Decomposition Solvers

Sandia National Laboratories

- **Computational Kernels:**
  - **Sparse matrix-vector multiplication**
    - **Apply operator/coarse interpolations**
    - **Tpetra/Kokkos**
  - **Sparse Linear Solvers**
    - **Now: Threaded factorizations and solves**
      - **MKL Pardiso**
      - **Sandia efforts (Trilinos, Sierra)**
    - **In progress: Inexact subdomain solves**
      - **Reduced memory, smaller coarse problems, …**
  - **Dense linear algebra**
    - **Iterative solution acceleration**
      - **Subspace recycling (projections)**
    - **Sparse direct solvers (supernodal variants)**

Sparse Linear Solvers

- **MKL Pardiso:**
  - **Threaded factorization and solve phases**
  - **Earlier disappointments with solve phase (better now!)**
- **Recent Sandia Efforts:**
  - **Hybrid Triangular Solver (HTS, Bradley)**
    - **Solve phase only**
    - **OpenMP**
  - **Task Based Cholesky/LDL (Tacho, Kim and Rajamanickam)**
    - **Factorization and solve phases**
    - **Kokkos/Pthreads, should be a good GPU possibility**
    - **FY17/Q3 integration/testing to be completed**
  - **Ng-Peyton* Threaded (NPT, Dohrmann)**
    - **Factorization and solve phases**
    - **OpenMP Tasks**

*Esmond G. Ng and Barry W. Peyton, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Comput., Vol. 14, No. 5, pp. 1034-1056, 1993.

Domain Decomposition Parallelism

Sandia National Laboratories

- **MPI:**
  - **Historically a single subdomain per MPI rank**
  - **Multiple subdomains per MPI rank offers another level of parallelism, e.g. via threads (in progress)**
- **Threaded Sparse Solvers:**
  - **No changes to basic DD algorithm**
  - **Reporting on this today**
- **Vectorization:**
  - **Level 2 and 3 BLAS used extensively**
  - **Rely on optimized BLAS libraries**
- **Concurrency across levels:**
  - **Fine and coarse problems historically solved sequentially**
  - **Asynchronous computations across DD levels (future)**

Some MPI versus Threads Tradeoffs

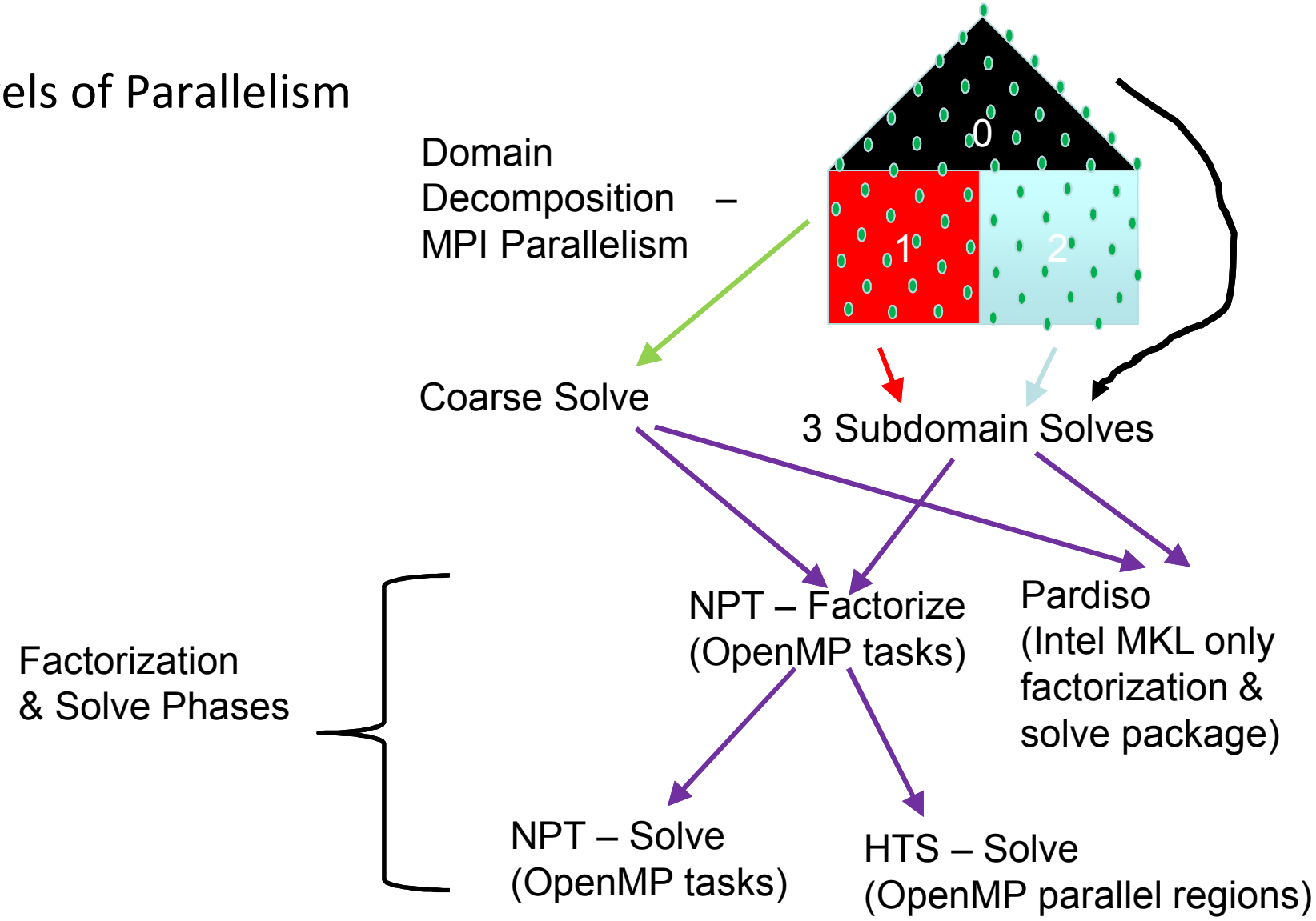
Sandia National Laboratories

- **Fewer MPI ranks & more threads:**
  - **+ Fewer subdomains often leads to better DD convergence**
  - **+ Fewer coarse levels can have convergence/comm benefit**
  - **+ Less memory & communication needed for MPI**
  - **- Less isolation from NUMA & shared memory effects**
  - **- More work at subdomain level (order N^2 flops for 3D)***
  - **- More memory at subdomain level (order $N^{4/3}$ for 3D)***
- **More MPI ranks & fewer threads:**
  - **+ Greater isolation from NUMA & shared memory effects**
  - **+ Less computation and memory for each subdomain**
  - **+ Fewer changes to existing algorithms, history of success**
  - **- More subdomains/coarse levels may slow convergence**
  - **- More memory & communication needed for MPI**

***N = subdomain size**

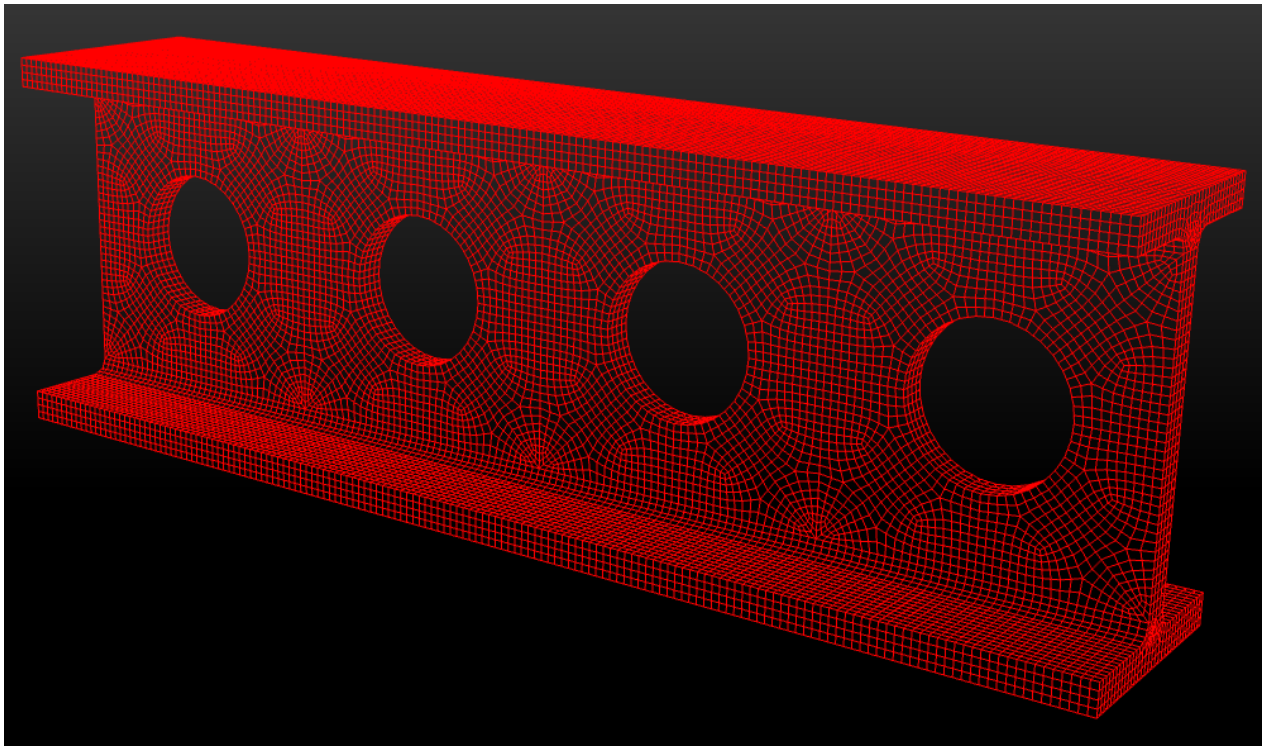
Sierra Structural Dynamics & Solver

- Levels of Parallelism



Sparse Linear Solvers

- **Test Matrices:**
  - **2 subdomain matrices from test suite (models3-4)**
  - **2 I-beam models of interest**



# of unknowns
model3: 57,201
model4: 36,195
Ibeam_r0: 39,411
Ibeam_r1: 259,431

Notes: Metis nested dissection and symbolic factorization not threaded.

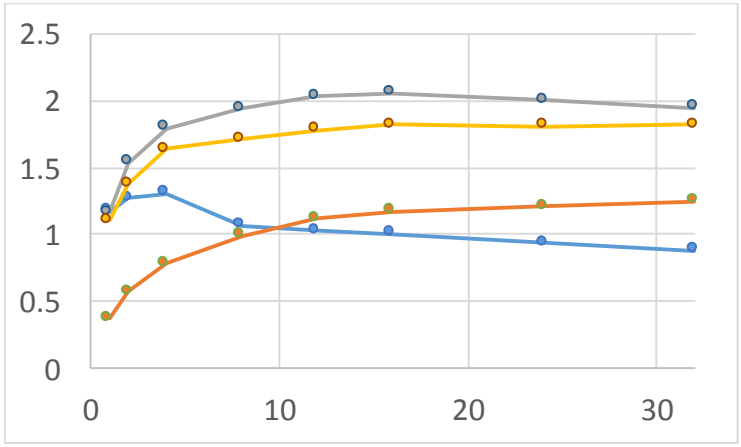
Intel17 OpenMP

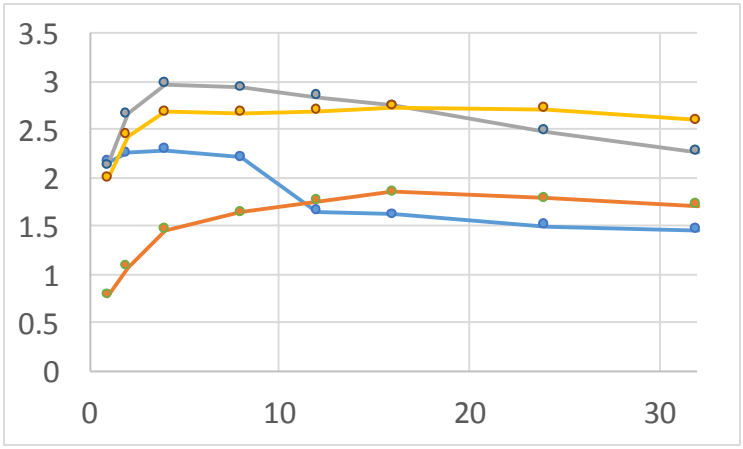
Sparse Linear Solvers

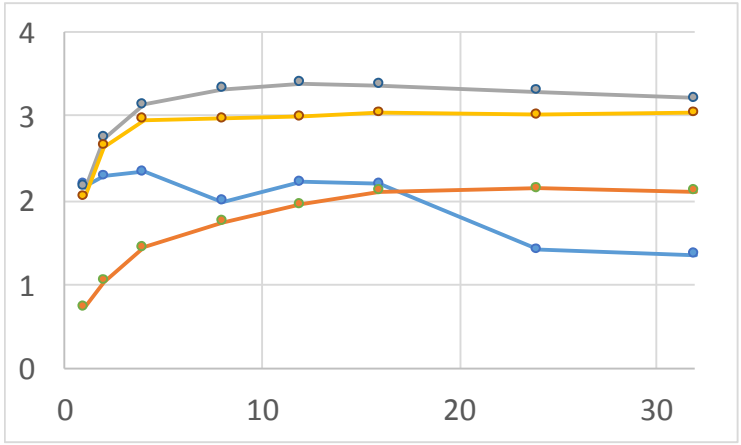
Sandia National Laboratories

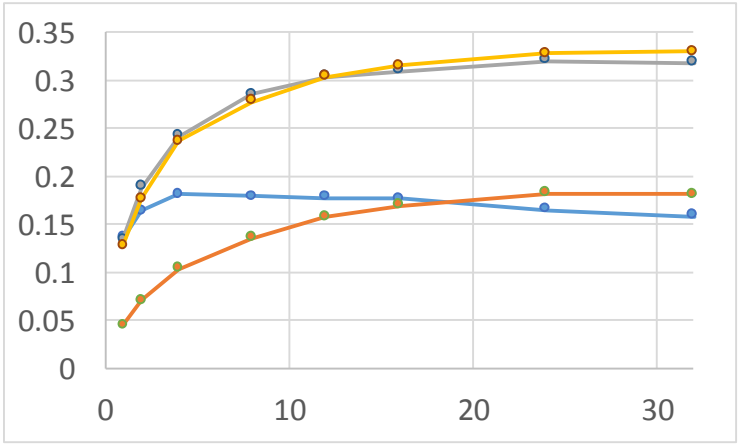
- **Two different architectures on Mutrino tested:**
  - **Haswell, 32 cores on 2 sockets, 2 hardware threads/core**
  - **KNL, 68 cores, 4 hardware threads/core**

Init Phase, 1 MPI rank, Haswell

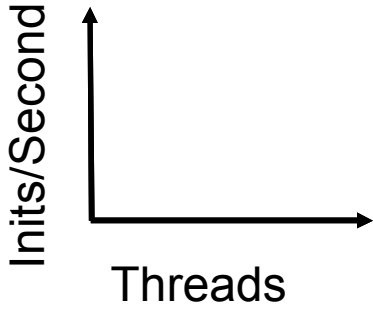




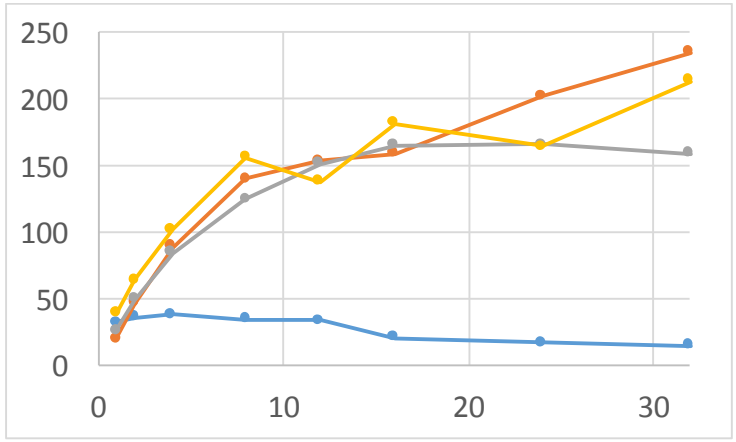


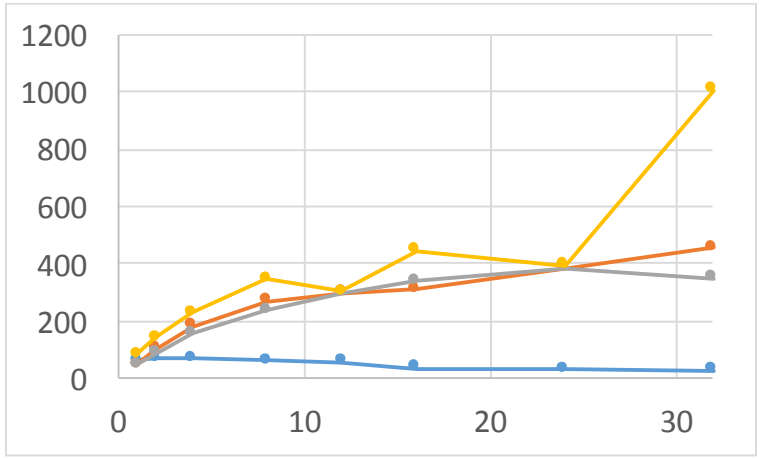


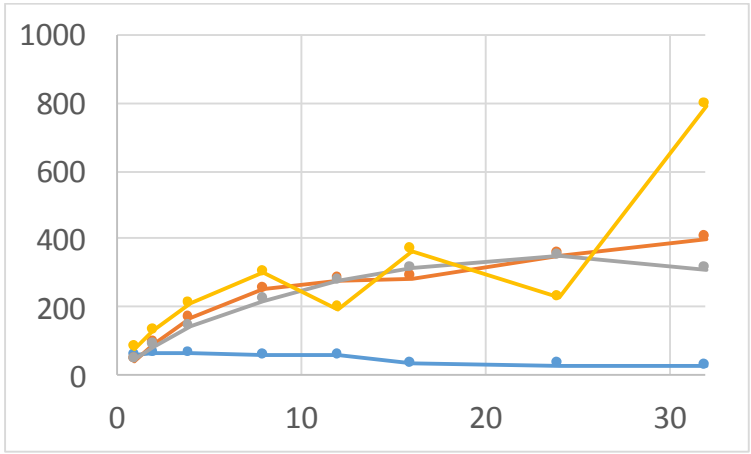
Model3 Model4
IbeamR0 IbeamR1

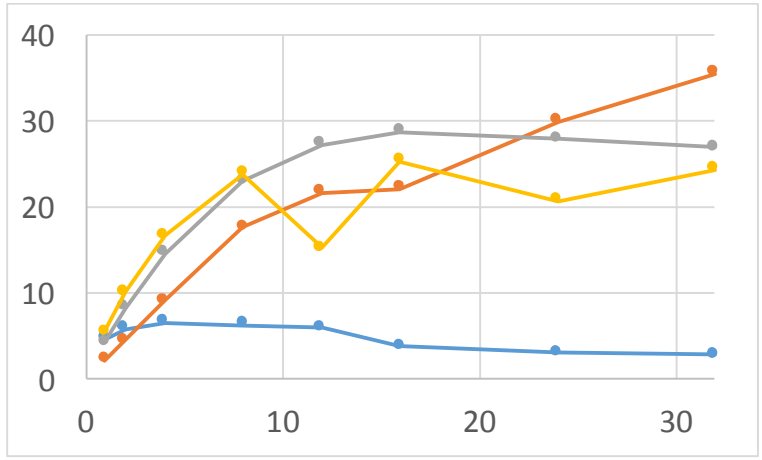


Solve Phase, 1 MPI rank, Haswell

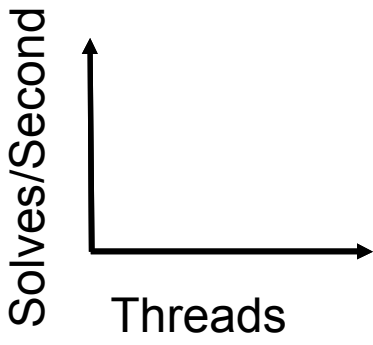




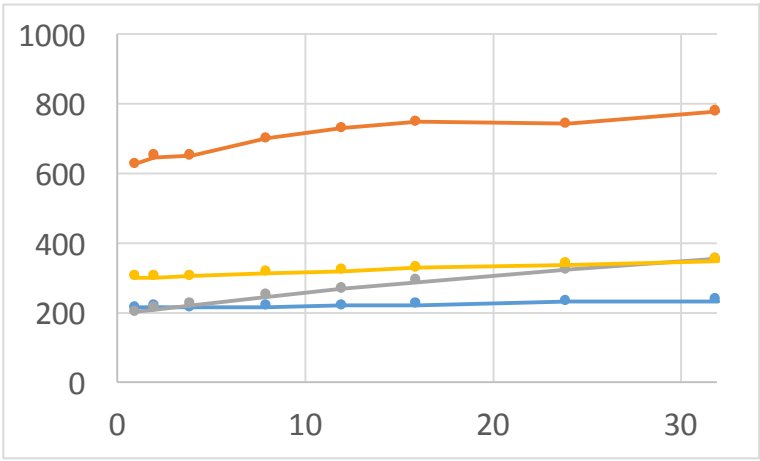


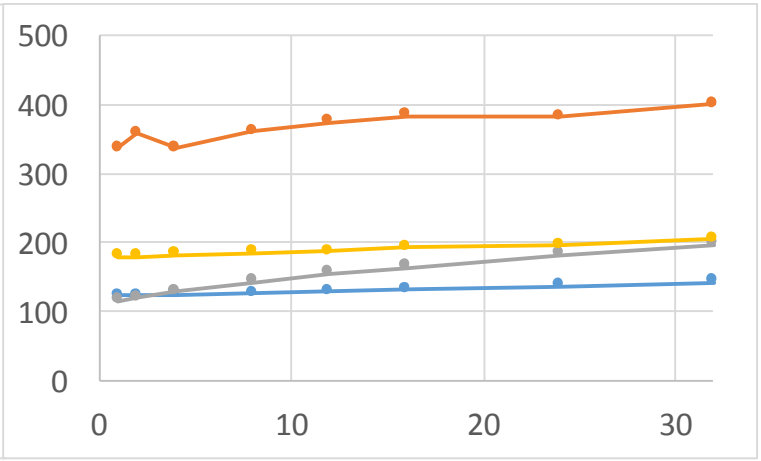


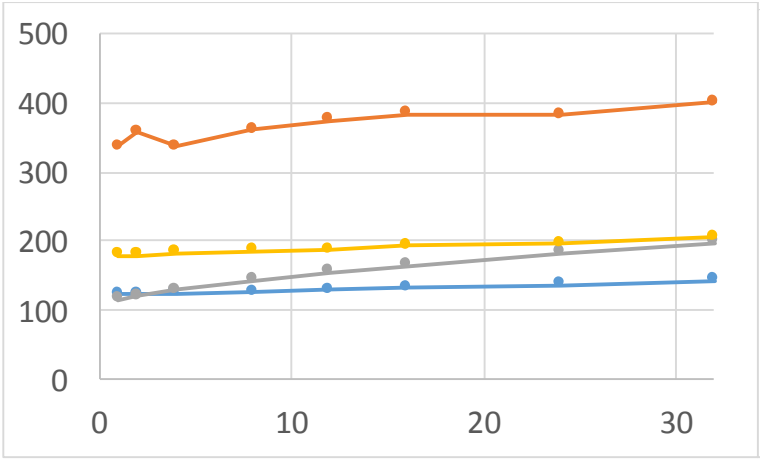
Model3 Model4
IbeamR0 IbeamR1

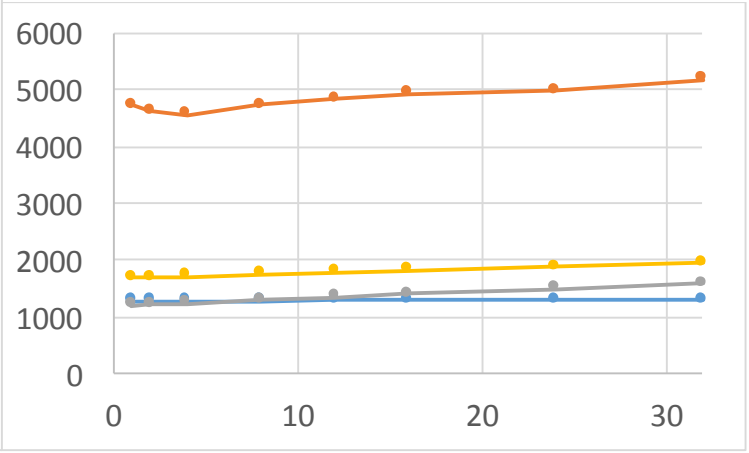


Memory, 1 MPI rank, Haswell

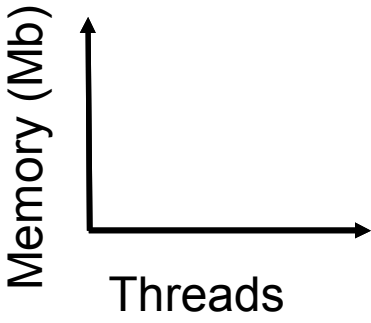




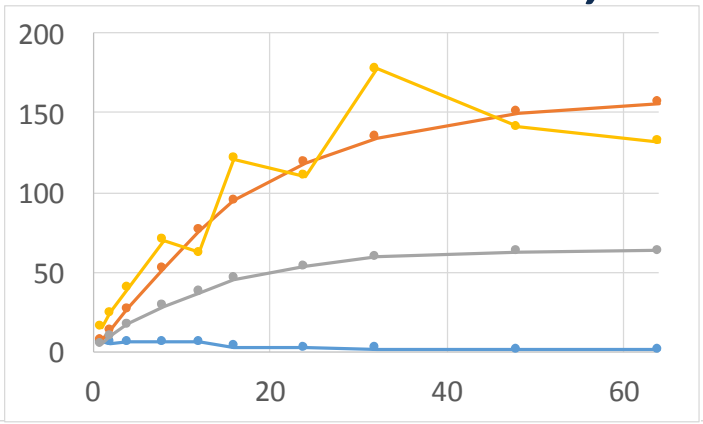


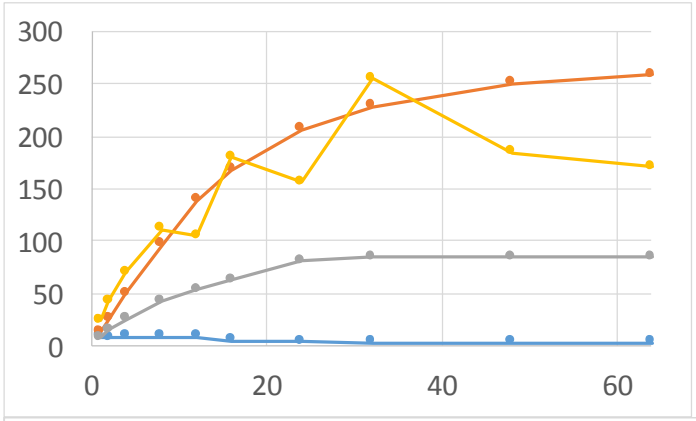


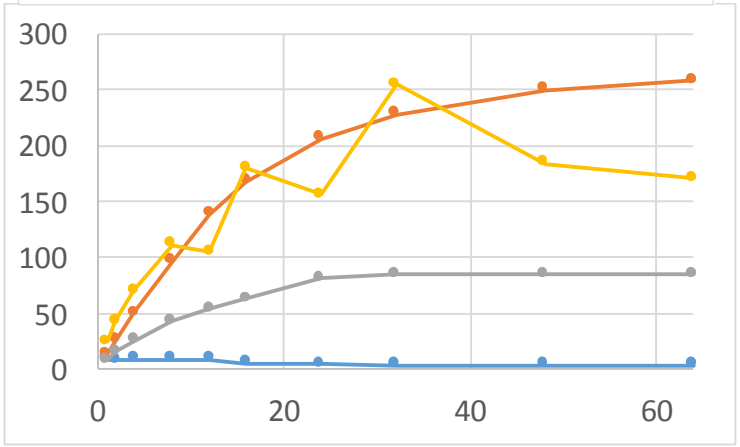
Model3 Model4
IbeamR0 IbeamR1

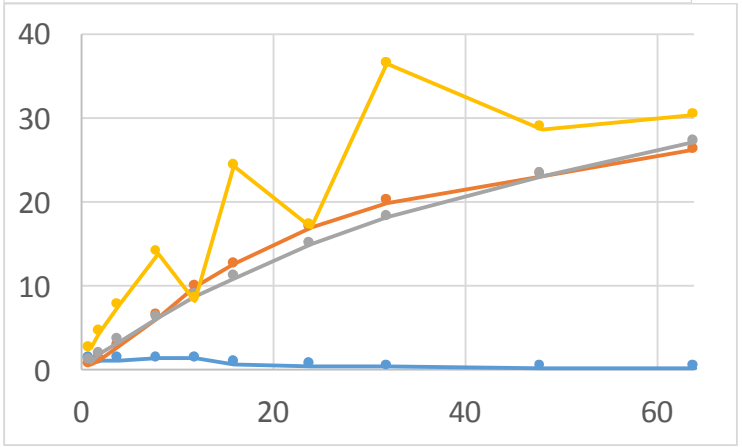


Solve Phase, KNL

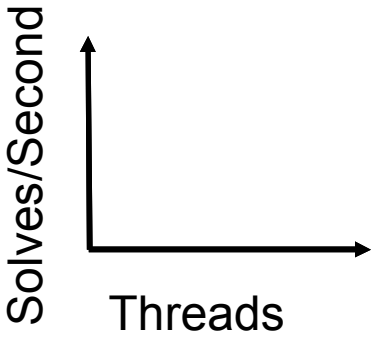








Model3 Model4
IbeamR0 IbeamR1



Caveat: NPT & HTS performance may vary depending on compiler flags (fma & fp-model)

Sierra Integration (Target Apps)

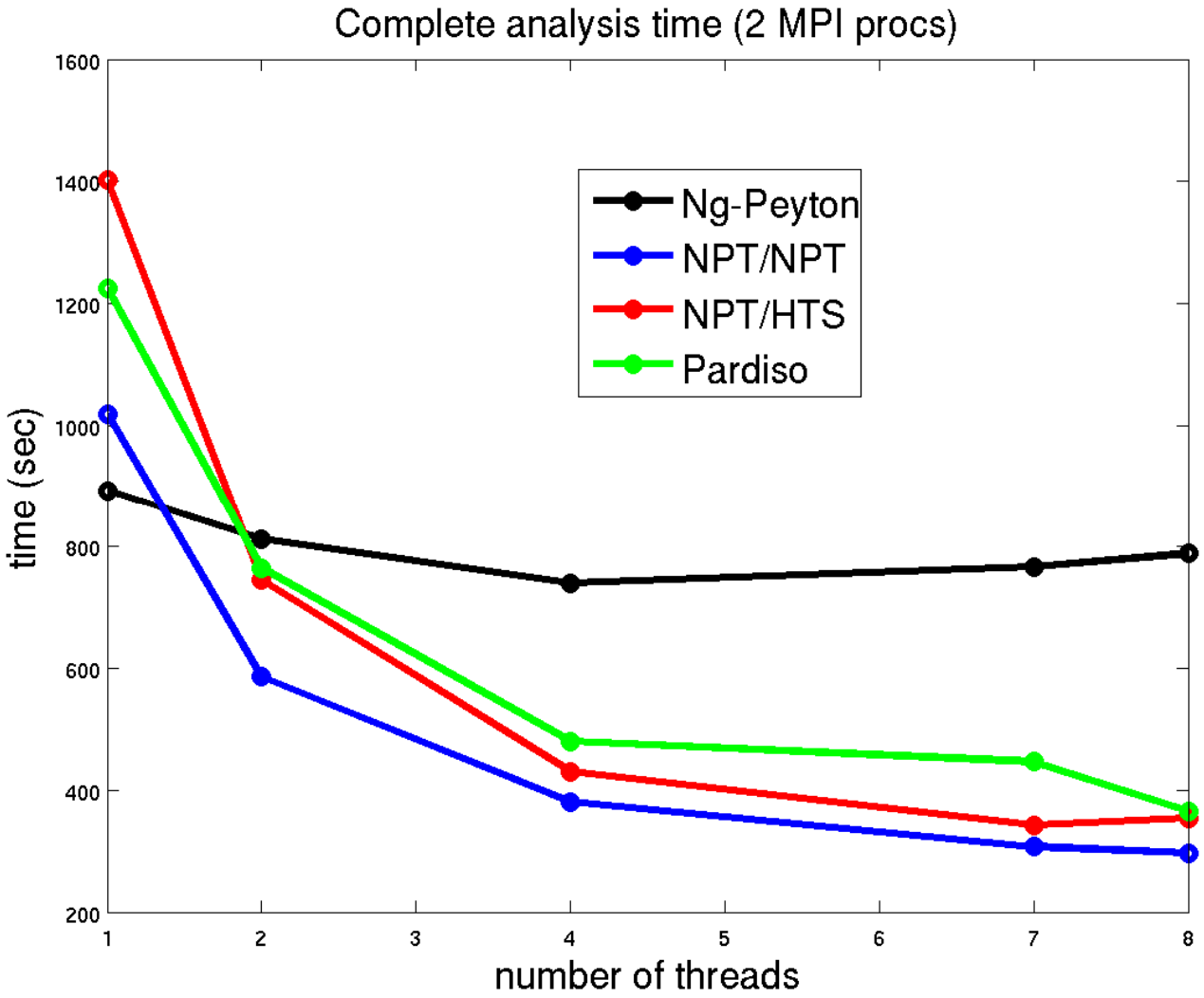
Sandia National Laboratories

- **Sierra/SD (Structural Dynamics):**
  - **Modal, transient, frequency response, static, inverse, … analyses (primarily linear)**
  - **Operator matrix often constant ⇒ many solves/factorization**
  - **GDSW* iterative solver**
- **Sierra/SM (Solid Mechanics):**
  - **Nonlinear explicit & implicit structural analysis**
  - **Tangent matrix changing ⇒ fewer solves/factorization**
  - **FETI-DP** used as preconditioner**

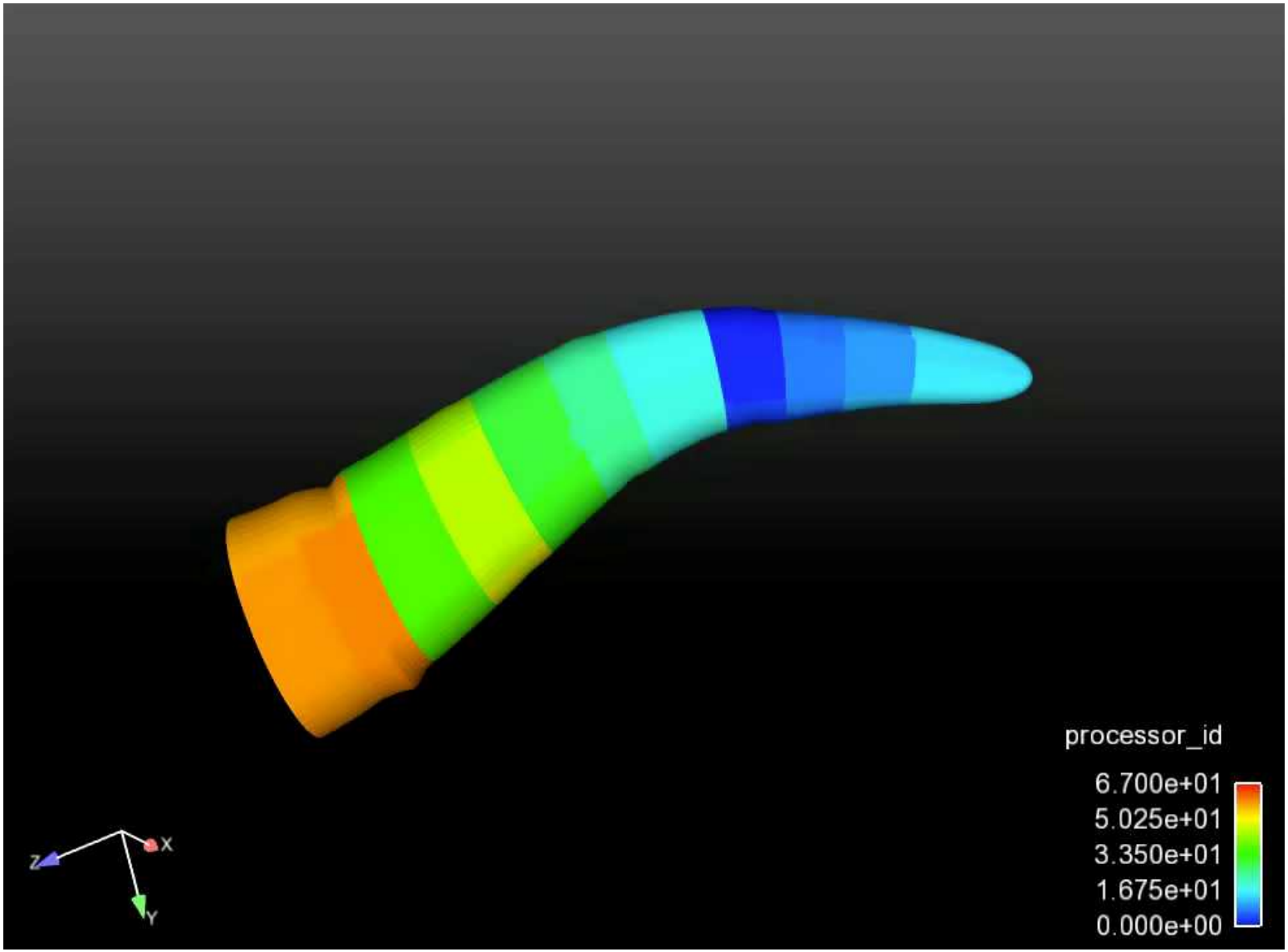
**Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity*, Int. J. Numer. Meth. Engng, Vol. 82, pp. 157-183, 2010.

***FETI-DP: A dual-primal unified FETI method – part I: A faster alternative to the two-level FETI method,* Int. J. Numer. Meth. Engng, Vol. 50, pp. 1523-1544, 2001.

Status from Early 2016 – Xeon



Eigen RV problem



~330,000 Nodes and Elements

10 eigen modes

Mutrino KNL or Haswell

Intel-17 OpenMP

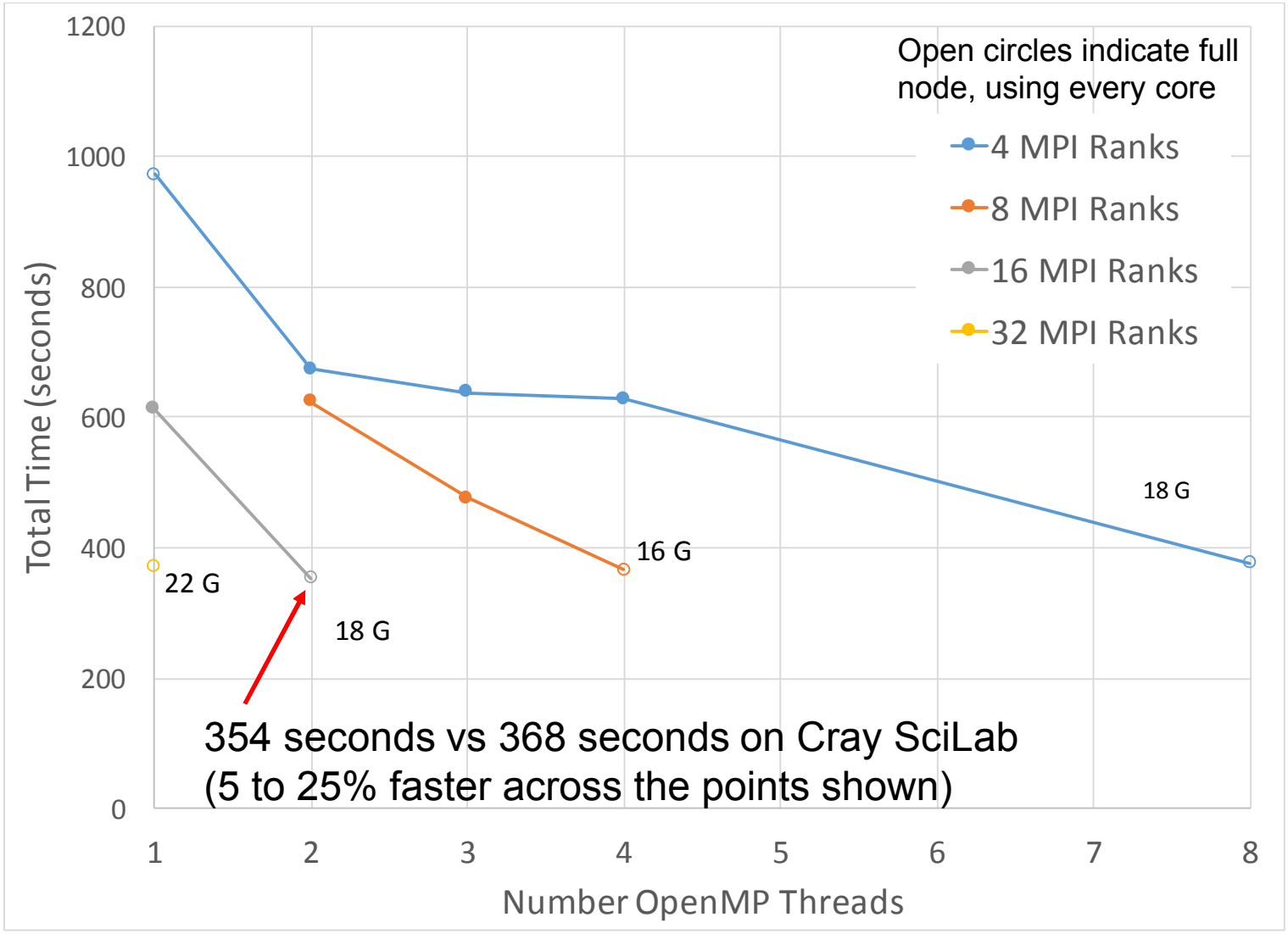
All results minimum over 5 runs

Mutrino Standard Deviations

- For identical runs, standard deviations about 2 to 10% of total runtime

- For this reason, data shown is minimum across 5 runs

NPT on Haswell (Intel MKL)

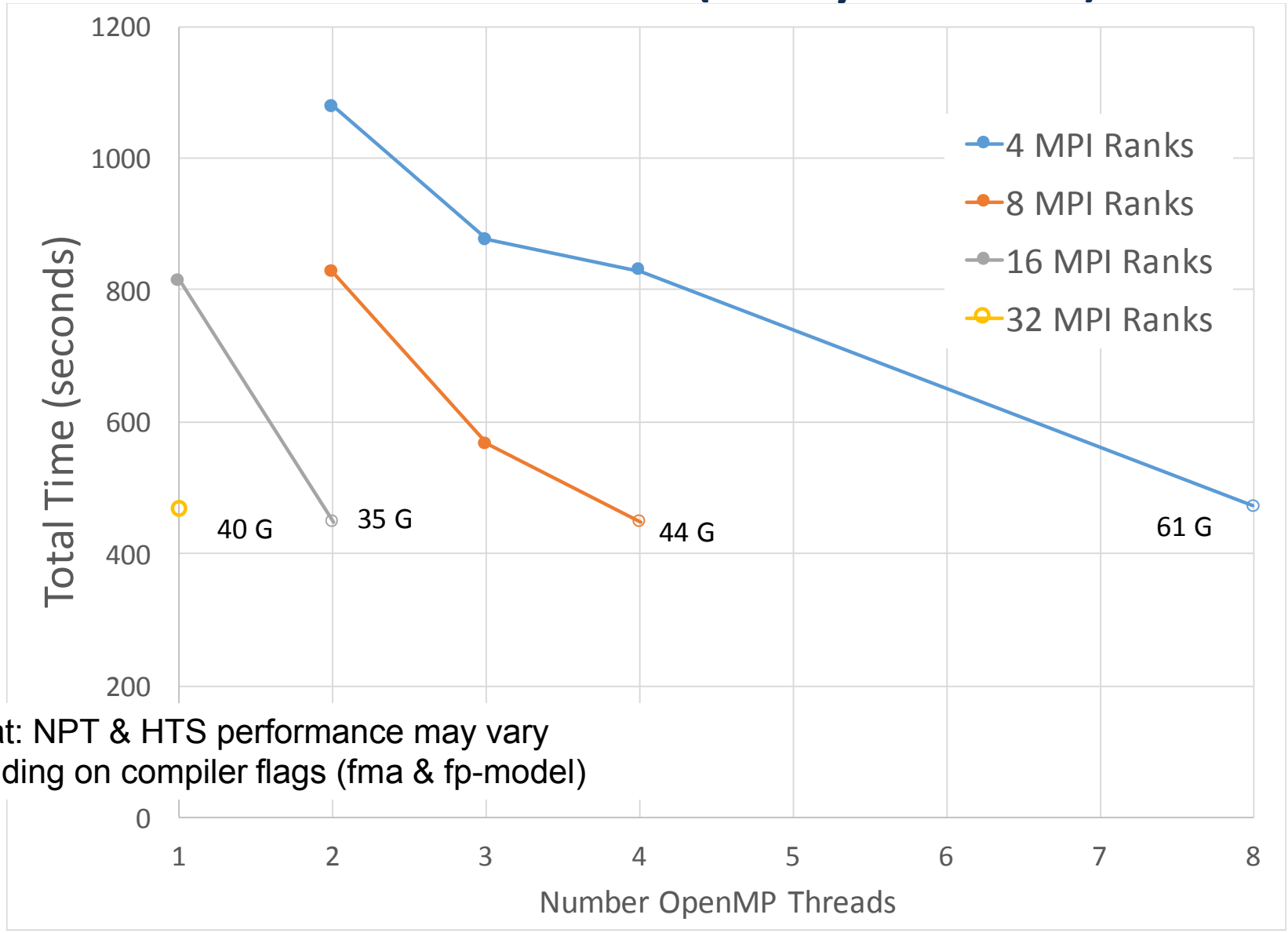


Solver Init vs Solve Phase

Sandia National Laboratories

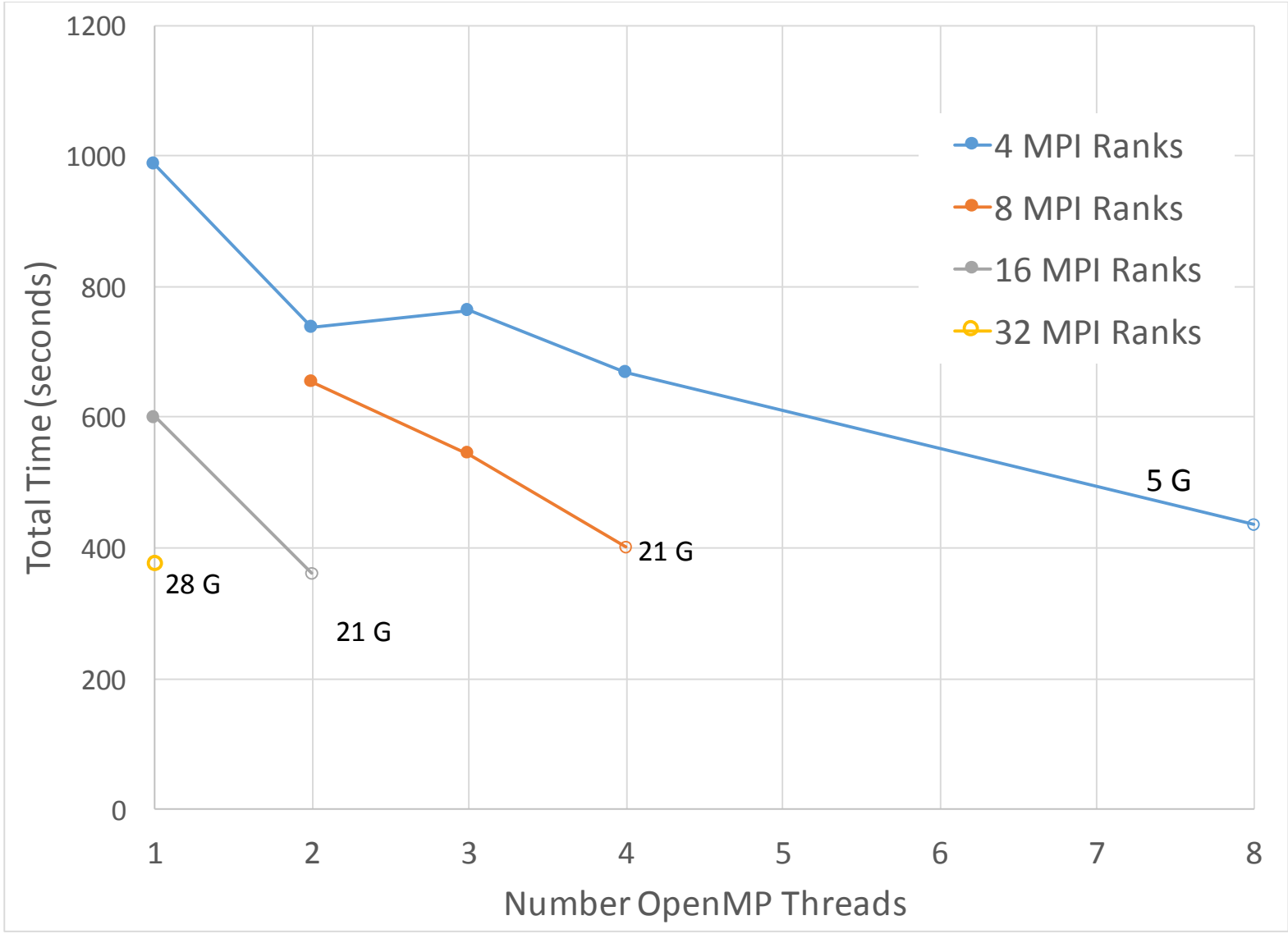
- About 70% solve phase, 10% init phase, 10% coarse solve, 3% orthogonalization time for either NPT, NPT-HTS or Pardiso (> 70% for more modes)
- Eigensolver (P-ARPACK) does not seem to be a major factor in runtime
- About 90 linear solves (more solves if more modes)
- Varies somewhat, coarse solve longer if more MPI ranks (more subdomains).

NPT-HTS on Haswell (Cray Scilab)

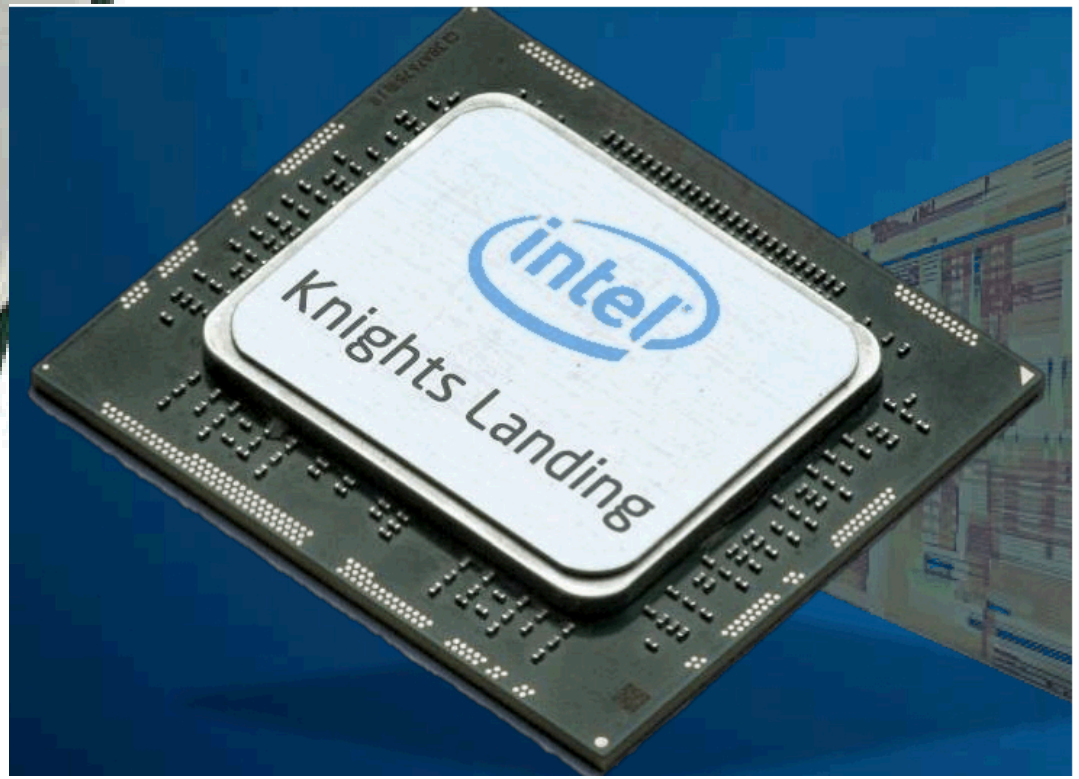


Caveat: NPT & HTS performance may vary depending on compiler flags (fma & fp-model)

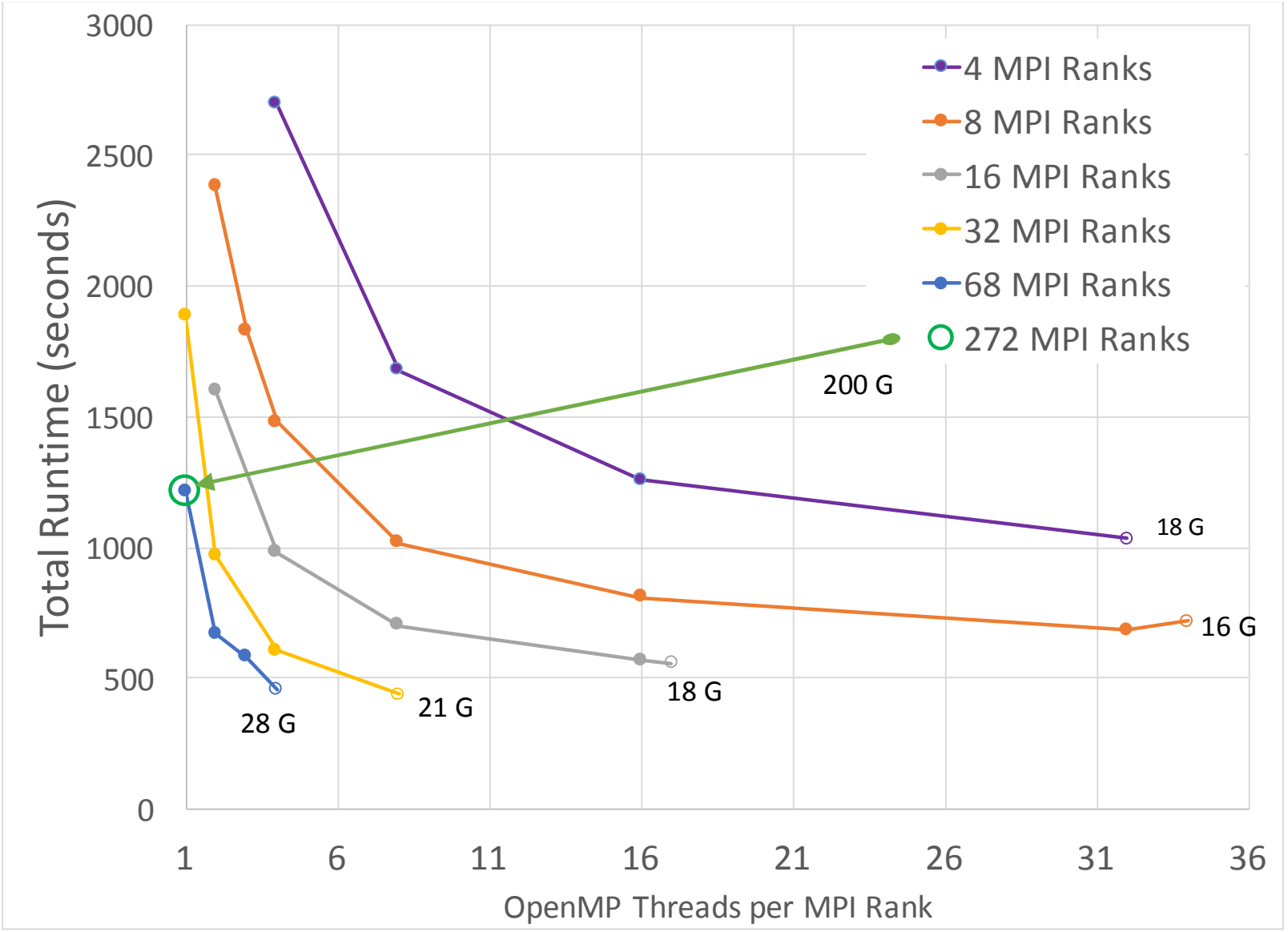
Pardiso on Haswell (MKL Library)



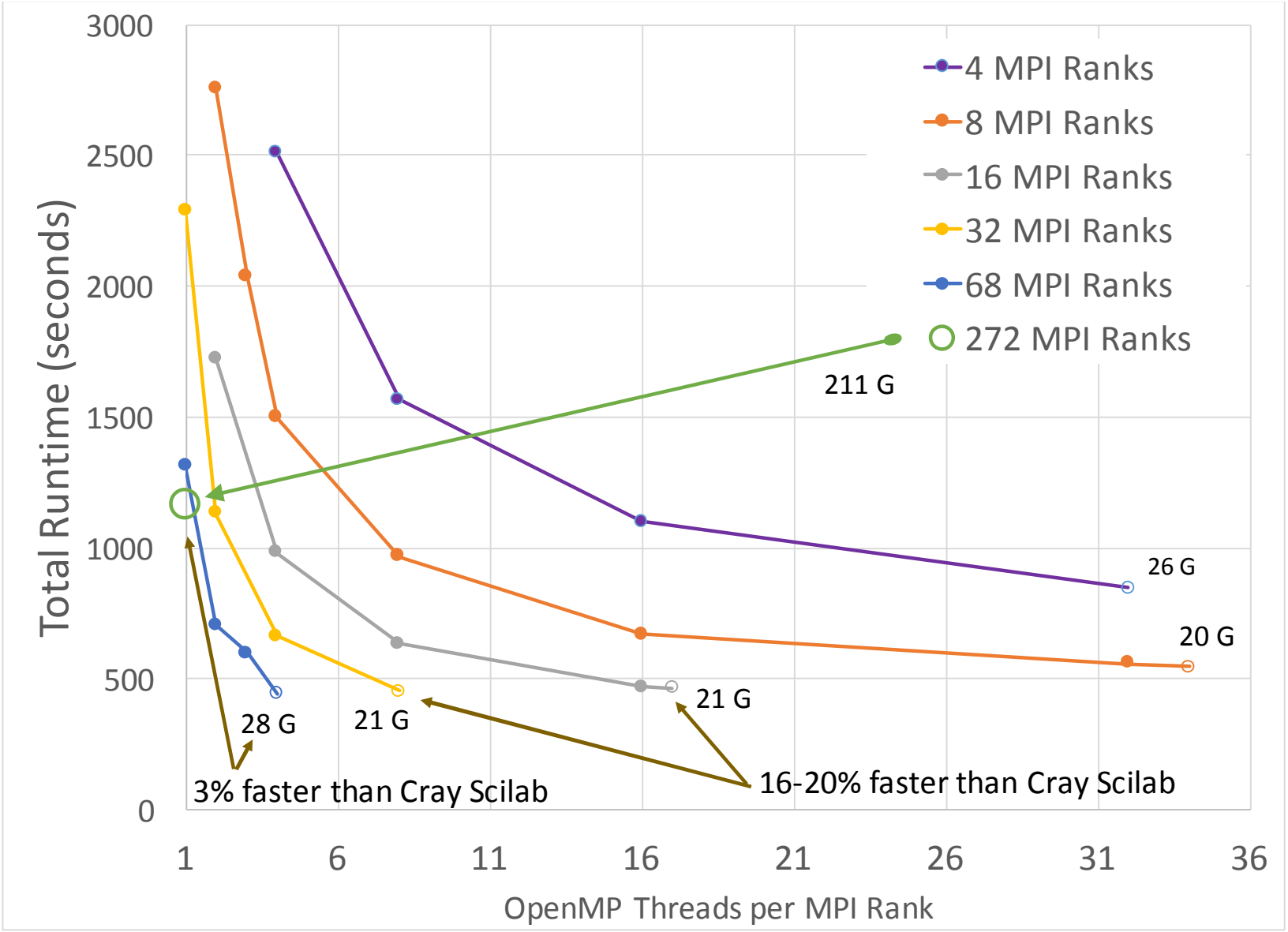
Switch Gears to KNL



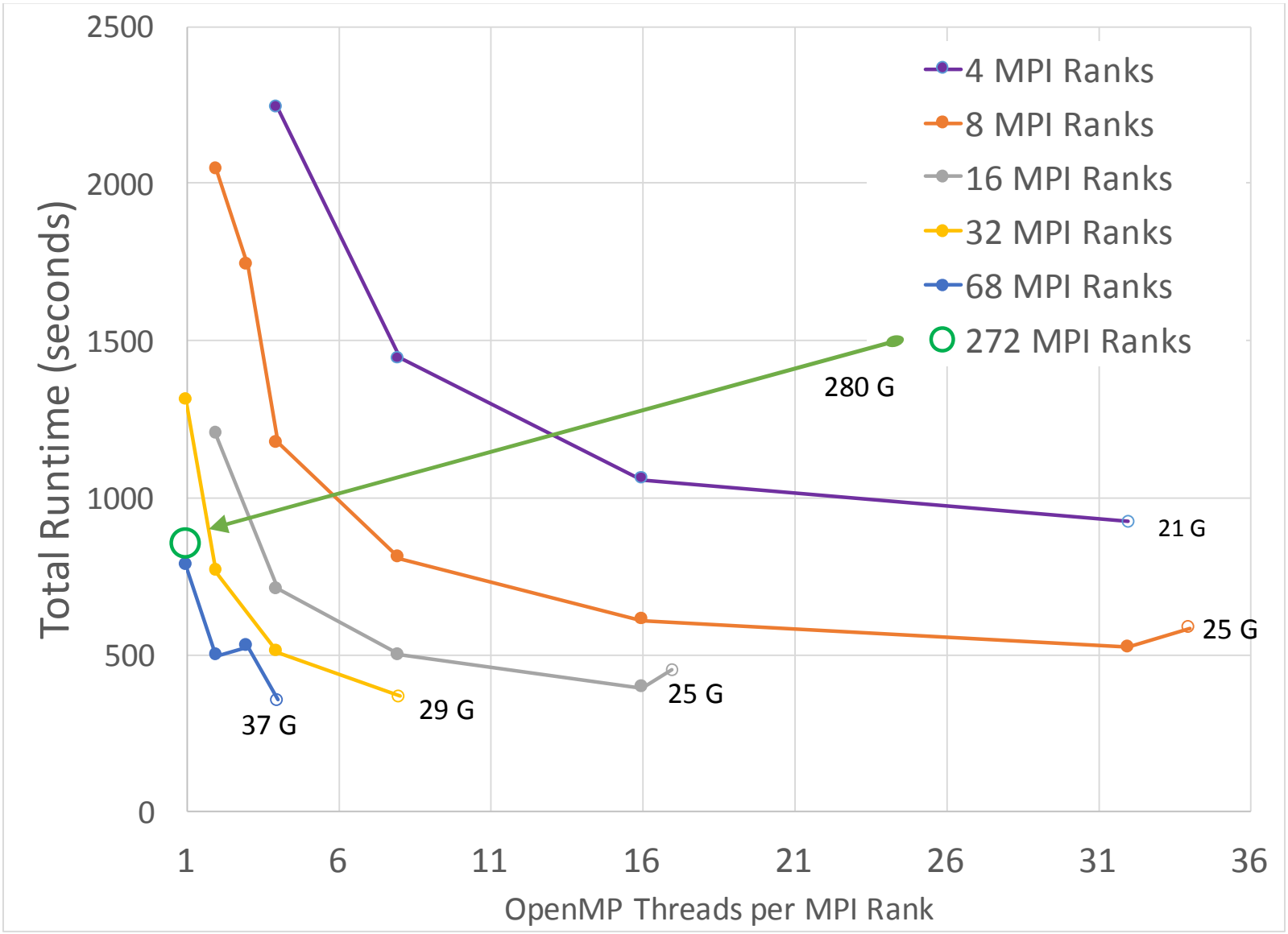
NPT on KNL (Cray SciLab)



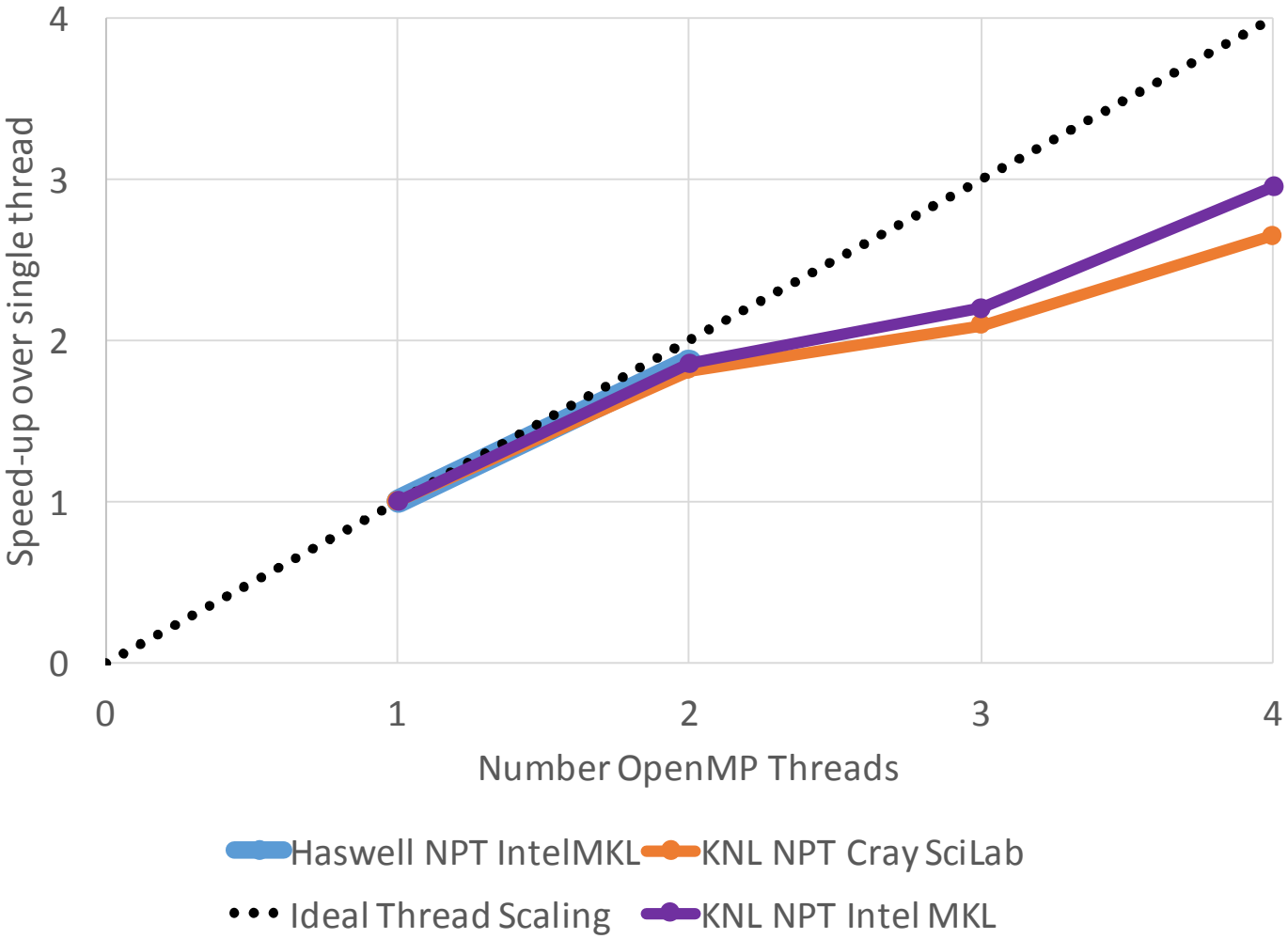
NPT on KNL (Intel MKL)



Pardiso on KNL (Intel Only)



Thread Scaling

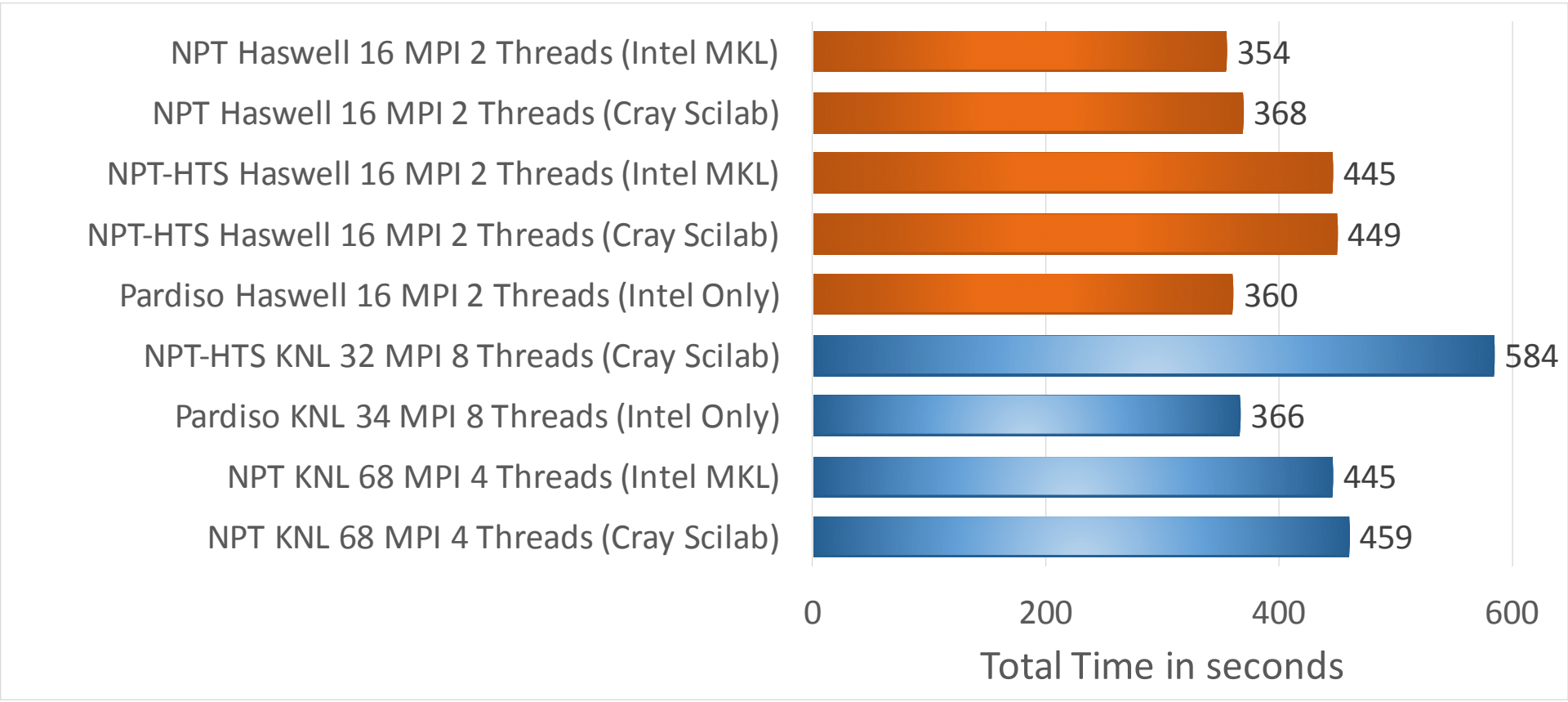


Only threaded the solver!

KNL is 68 MPI ranks and then hardware hyperthreads

Haswell is 16 MPI ranks, 2 threads, across 32 total processors

Best Times


NPT Haswell 16 MPI 2 Threads (Intel MKL)
354
NPT Haswell 16 MPI 2 Threads (Cray Scilab)
368
NPT-HTS Haswell 16 MPI 2 Threads (Intel MKL)
445
NPT-HTS Haswell 16 MPI 2 Threads (Cray Scilab)
449
Pardiso Haswell 16 MPI 2 Threads (Intel Only)
360
NPT-HTS KNL 32 MPI 8 Threads (Cray Scilab)
584
Pardiso KNL 34 MPI 8 Threads (Intel Only)
366
NPT KNL 68 MPI 4 Threads (Intel MKL)
445
NPT KNL 68 MPI 4 Threads (Cray Scilab)
459
0
200
400
600
Total Time in seconds

Take-aways

- Hyperthreads on KNL really do provide an advantage, we lose almost 3X speedups if we don't use them.
- Using the best suited math library can have a moderate performance benefit, especially on KNL (2-20%).
- Intel Pardiso solve phase has improved greatly from last year, and is now competitive if not the fastest solution.
- Compiler flags are important and should be further explored, especially for NPT & HTS.

Ongoing & Future Work

Sandia National Laboratories

- Continue Testing, come up with recommendations for usage on Haswell/KNL, especially Trinity, especially multi-node
- Solver strategies for hybrid CPU/GPU
- Multiple subdomains per MPI rank for threading
- Inexact subdomain solvers for greater parallelism
- Iterative coarse solves to enhance performance and reliability (thanks to Kendall Pierson for idea)
- Asynchronous work across different coarse levels
- Parameter and OpenMP performance enhancements for NPT

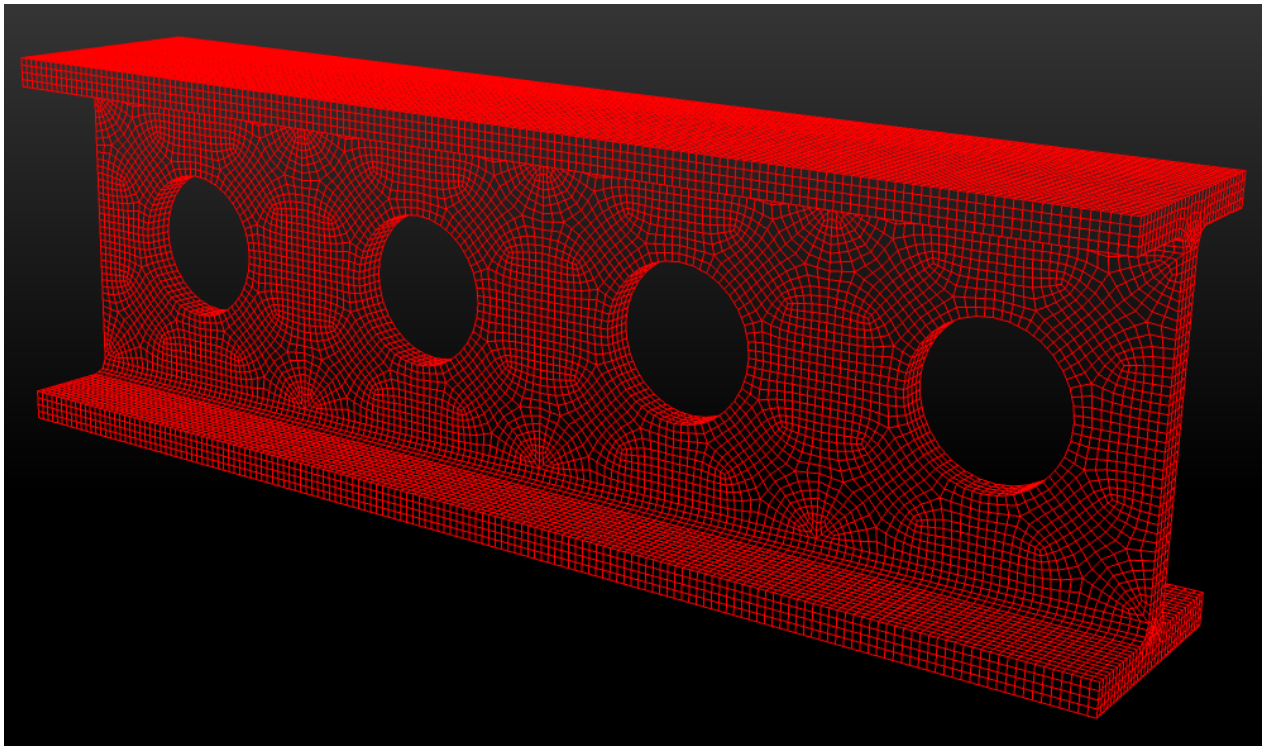
Extra Slides

Data from Last Year's Talk from Clark Dohrmann, also additional data from this year

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sparse Linear Solvers

- **Test Matrices:**
  - **4 subdomain matrices from test suite (models1-4)**
  - **2 I-beam models of interest**



# of unknowns

model1: 7,458
model2: 30,462
model3: 57,201
model4: 36,195
Ibeam_r0: 39,411
Ibeam_r1: 259,431

Notes: Metis nested dissection and symbolic factorization not threaded.

Sparse Linear Solvers (Jan16 Data*)

Sandia National Laboratories

- **Four different architectures on Morgan tested:**
  - **Sandy Bridge, 16 cores on 2 sockets, 2 hardware threads/core**
  - **Ivy Bridge, 20 cores on 2 sockets, 2 threads/core (not used)**
  - **Haswell, 32 cores on 2 sockets, 2 hardware threads/core**
  - **KNC, 61 cores, 4 hardware threads/core**

*courtesy of Andrew Bradley

Morgan Haswell* (Jan16 Data)

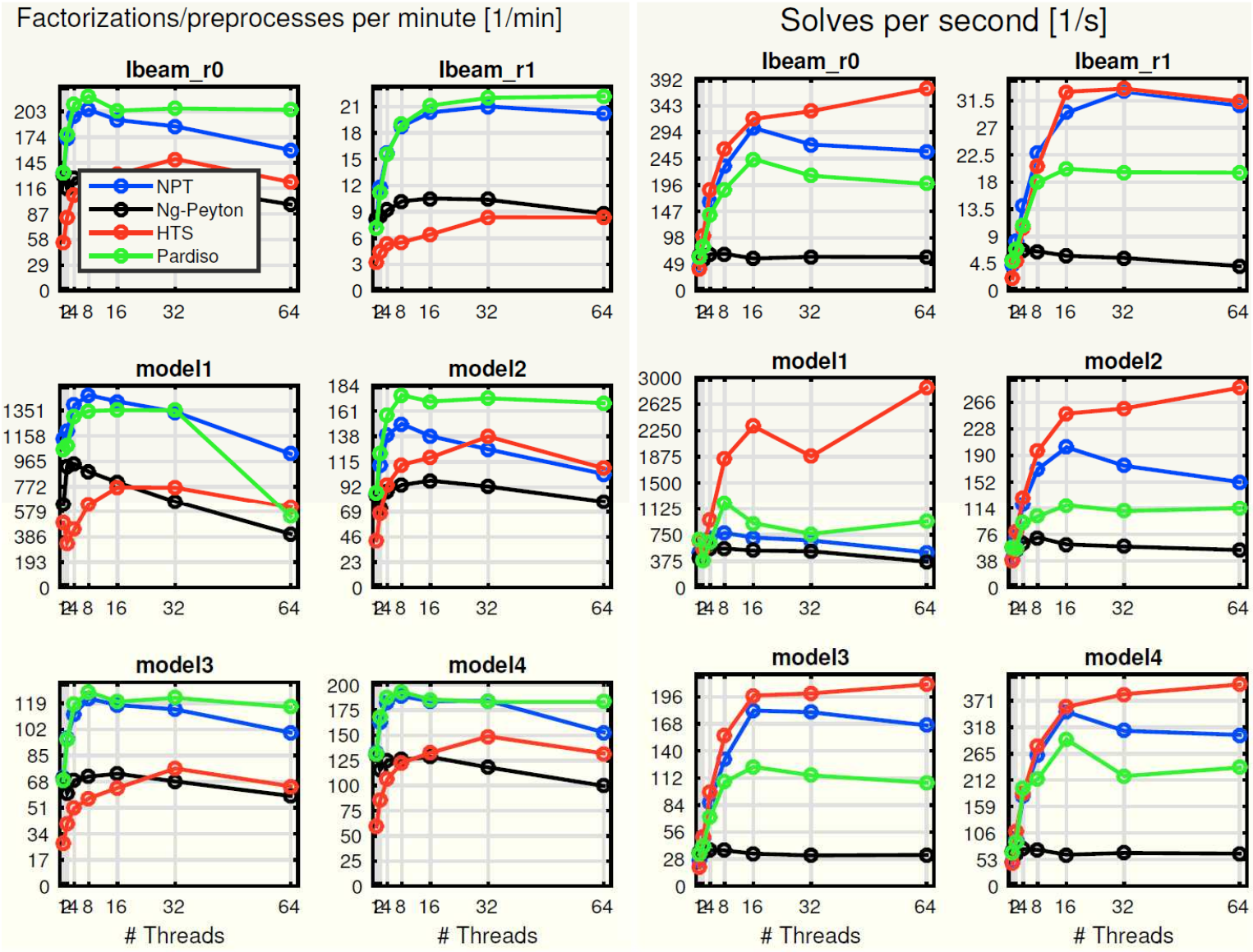


Figure 3: Haswell, 32 cores on 2 sockets, 2 hardware threads/core. Runs were done the same as before.

*results courtesy of Andrew Bradley

Morgan KNC* (Jan16 Data)

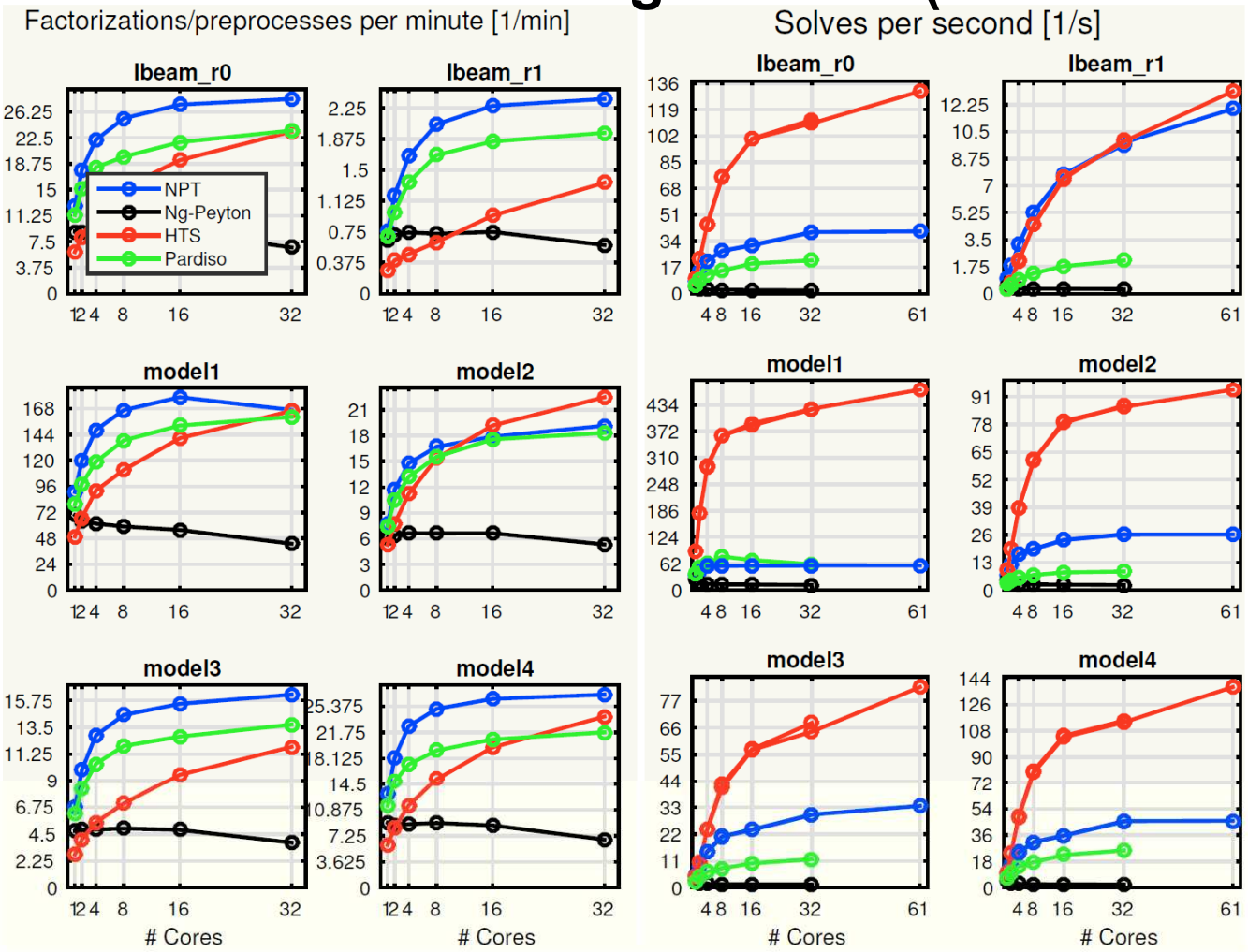
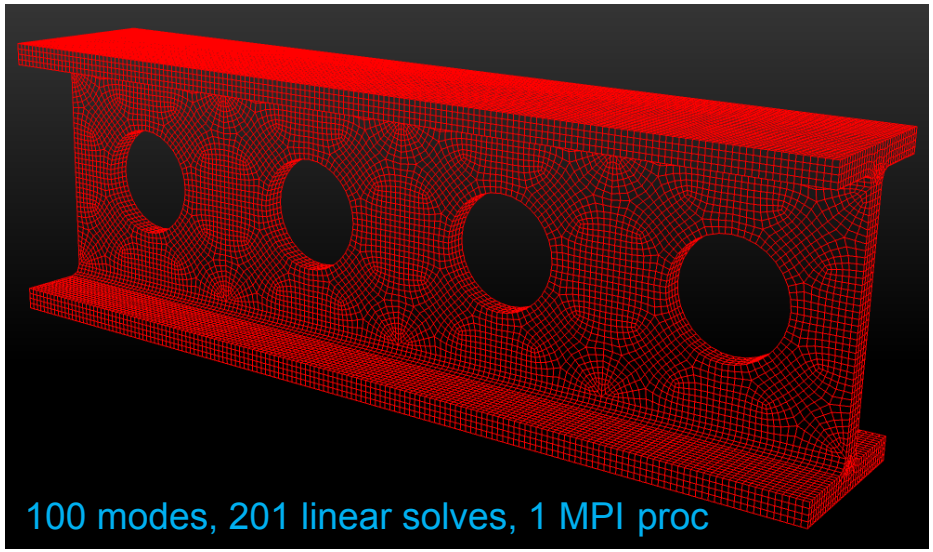
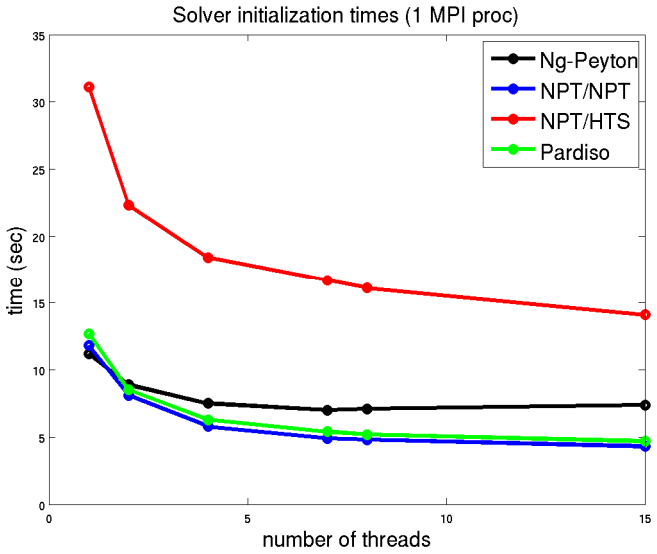


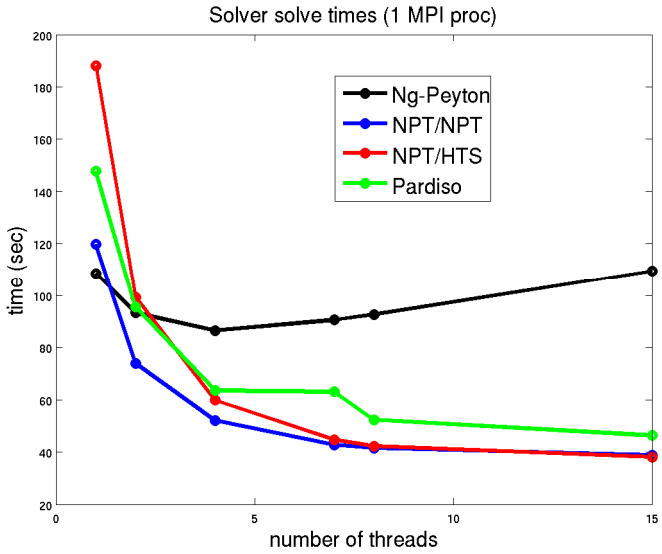
Figure 4: KNC, 61 cores, 4 hardware threads/core. Results are from two runs. NPT and HTS solvers were run at higher thread counts in a separate run. Runs were done with KMP_AFFINITY=BALANCED (1 thread/core until all cores used, then add more threads round robin) and KMP_AFFINITY=COMPACT (fill a core with 4 threads before moving to the next), and with OMP_NUM_THREADS set to a large number of values. The number of cores reported is the number of cores used by the KNC; however, thread affinity affects the number of threads/core. In these tests, 1 and 4 threads/core were tested at a number of core counts, and 2 threads/core was tested at 61 cores.

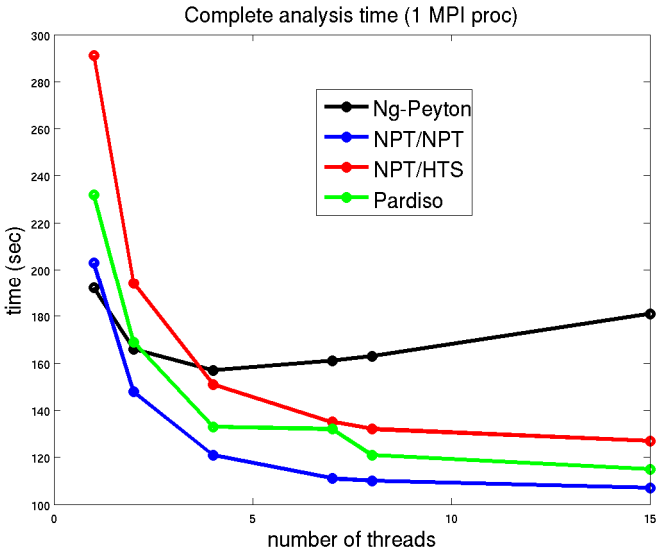
*results courtesy of Andrew Bradley

Integration Efforts (Jan16 Results)









Note: Intel 14 rather 15 compiler used because of Sierra/SD test errors (under investigation)

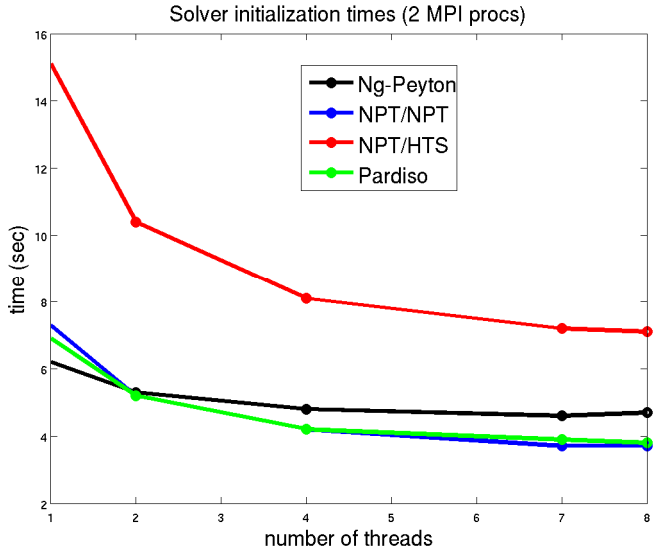
Integration Efforts (Jan16 Results)

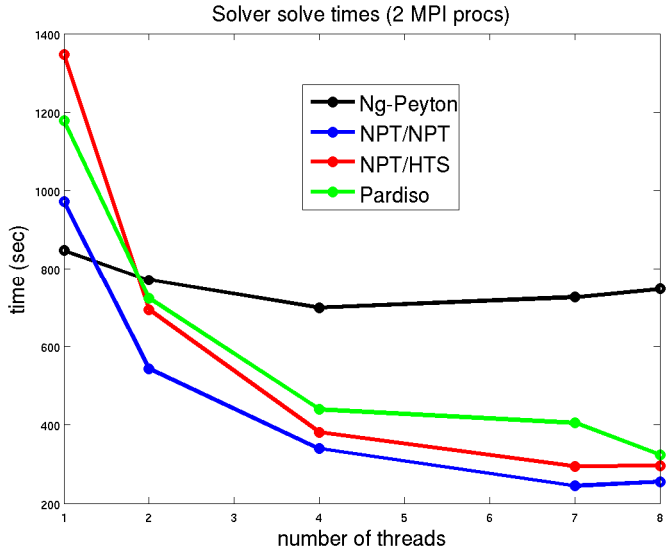
Run using 2 MPI processes on my blade (Sandy Bridge, 2 sockets, 8 cores/socket)

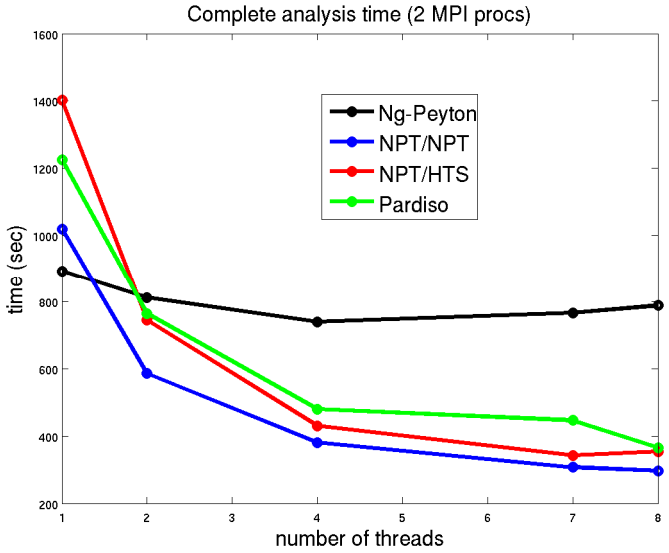
Problem too easy using default GDSW solver parameters (2 iters/solve average)

Used non-default parameters to be more representative (40 iters/solve average)

krylov_method = gmresClassic, solver_tol = 1e-8, overlap = 1, orthog = 0

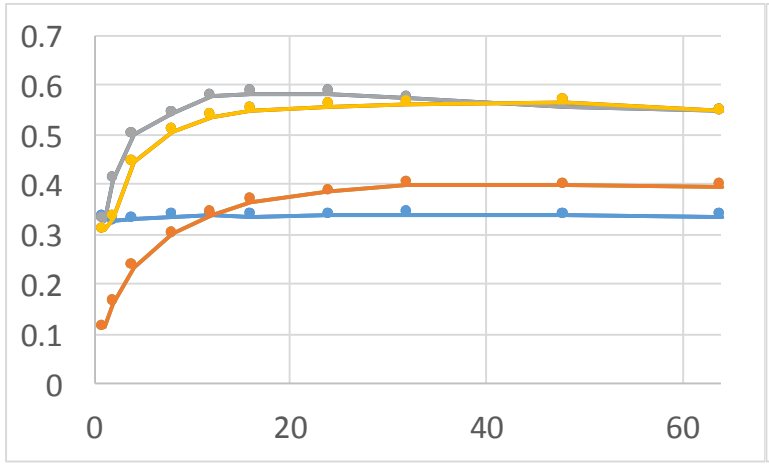


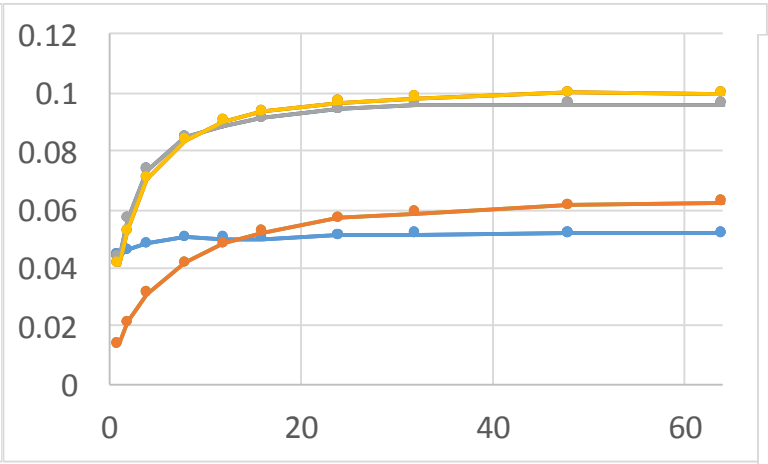


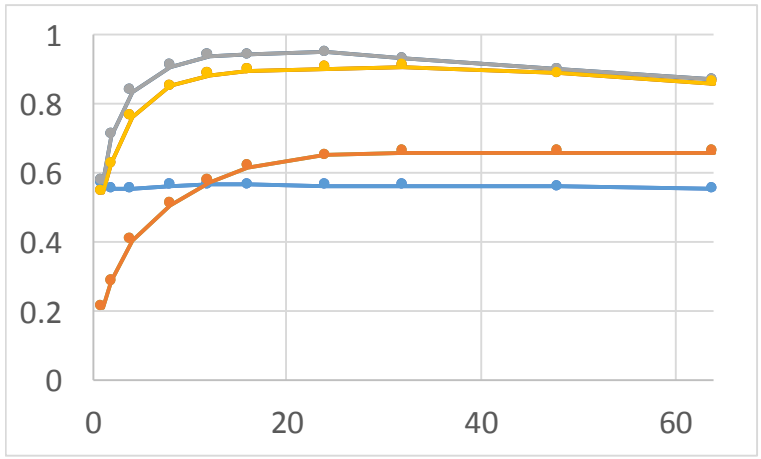


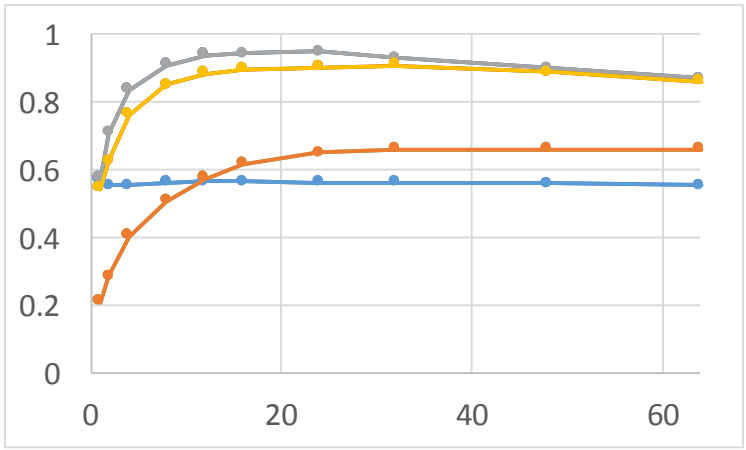
Disclaimer: non-optimal affinity and other settings possible here (lots to keep track of)

Init Phase, 1 MPI rank, KNL

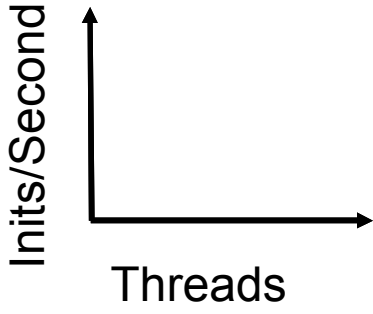




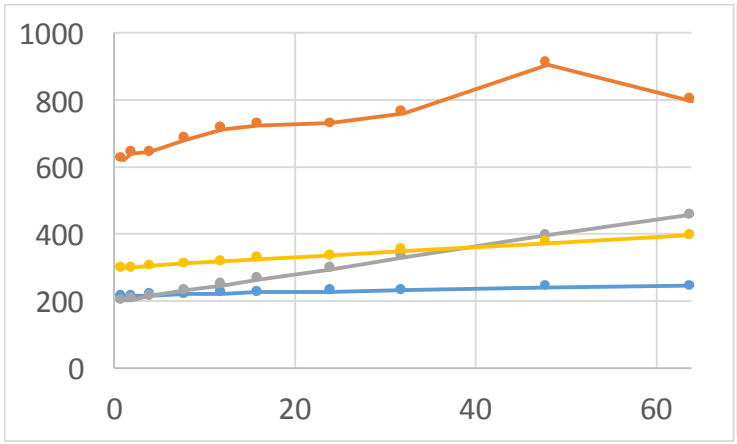


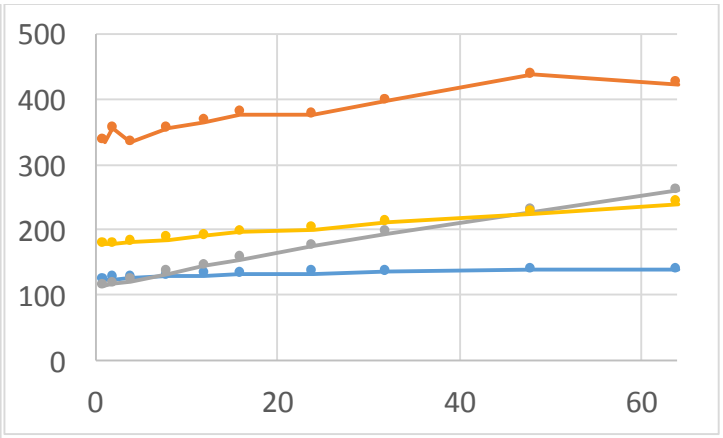


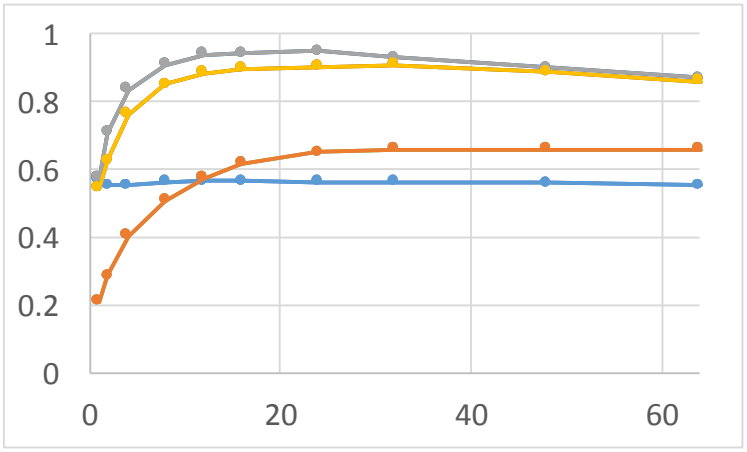
Model3 Model4
IbeamR0 IbeamR1

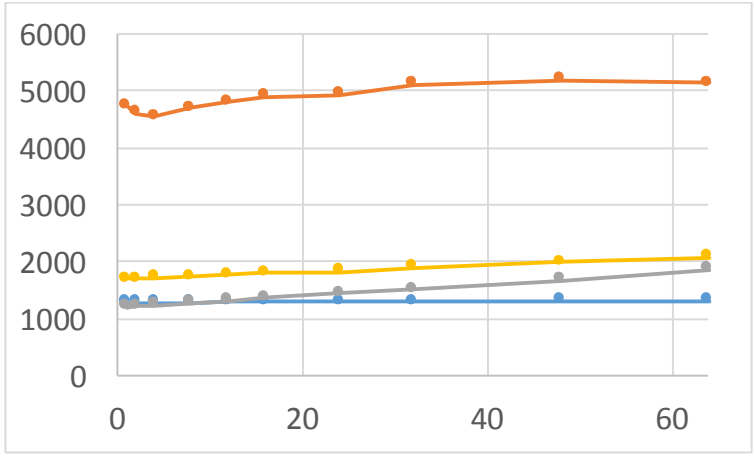


Memory, 1 MPI rank, KNL

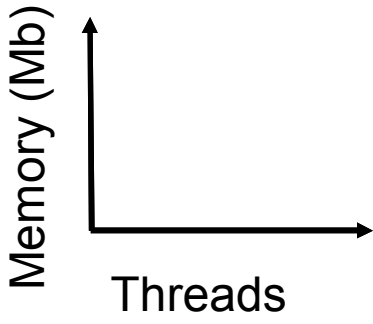




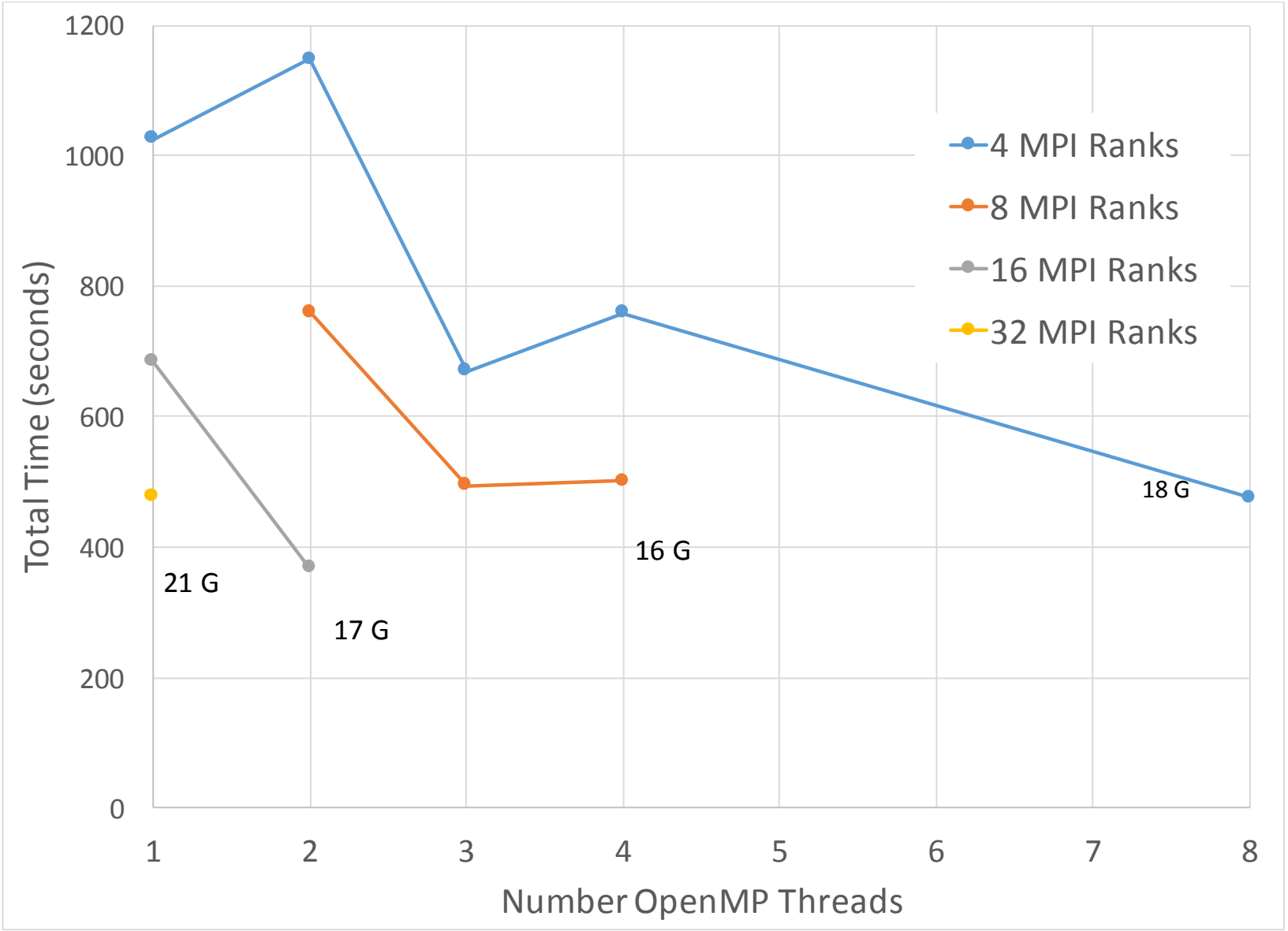




Model3 Model4
IbeamR0 IbeamR1



NPT on Haswell (Cray SciLab)



NPT-HTS on KNL (Cray SciLab)