SAND2016-7034C

Application of Task Parallel Direct Solvers in Domain Decomposition Preconditioners

Clark Dohrmann

Computational Solid Mechanics &
Structural Dynamics Department

ICCM2016
Berkeley, CA
August 1-4, 2016

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.

Thanks

Sandia National Laboratories

- Andrew Bradley
- Kyungjoo Kim & Siva Rajamanickam

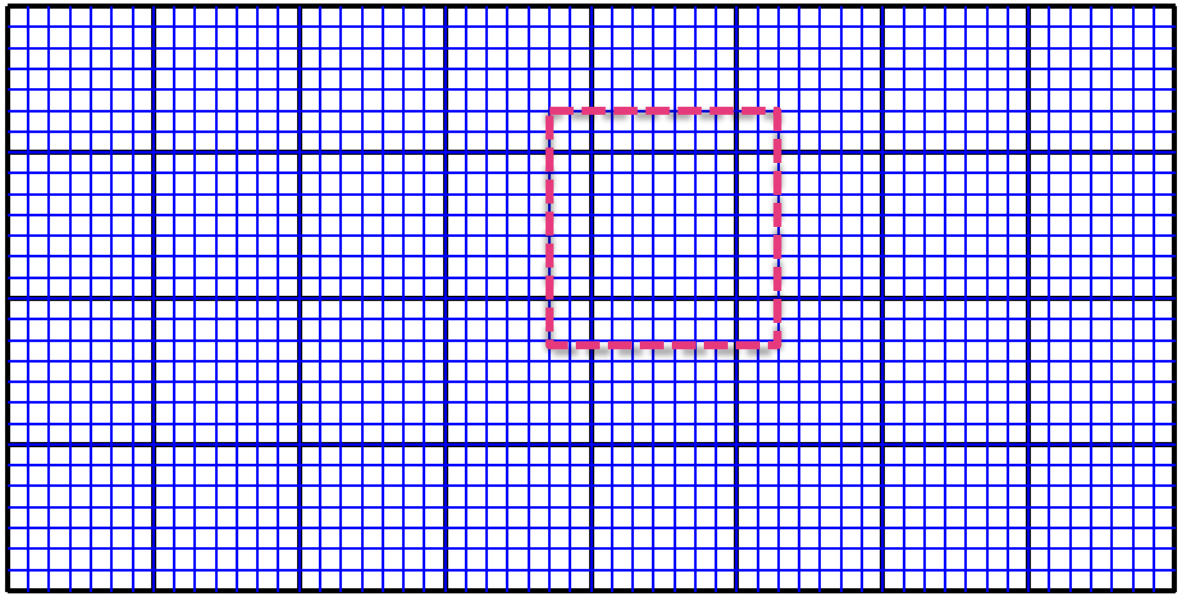
Outline

- **Domain Decomposition Preconditioners:**
  - Introduction
  - Motivation
  - Computational Kernels
- **Sparse Linear Solvers:**
  - Shared Memory (Threading) Options
  - Performance Results
- **Target Applications:**
  - Structural Dynamics
  - Solid Mechanics
  - Early Results
- **Closing Remarks**

Domain Decomposition Preconditioners

Sandia National Laboratories

Introduction: Two-level Additive Schwarz



$$Ax = b$$

$$AM^{-1}y = b$$

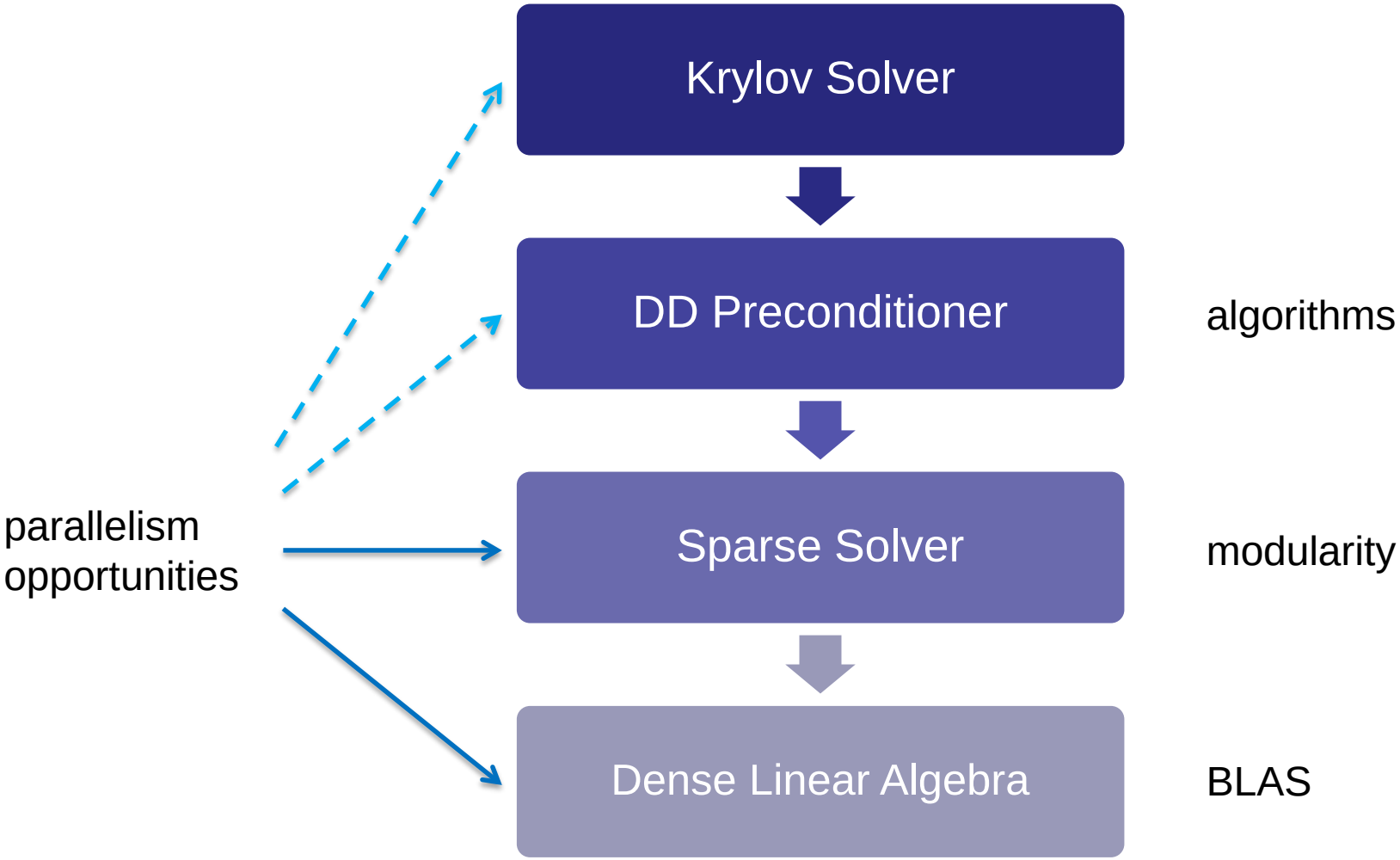
$$M^{-1}r = \sum_{i=1}^{N} R_i^T (R_i A R_i^T)^{-1} R_i r + \Phi(\Phi^T A \Phi)^{-1}\Phi^T r$$

$R_i =$ Boolean matrix $\qquad \Phi =$ interpolation matrix

Domain Decomposition Preconditioners

Sandia National Laboratories

Motivation: Expected need for more parallelism



Domain Decomposition Preconditioners

Sandia National Laboratories

Computational Kernels:

- Sparse matrix-vector multiplication
  - Apply operator/coarse interpolations
- Sparse Linear Solvers
  - Now: Threaded factorizations and solves
    - MKL Pardiso
    - Sandia efforts (Trilinos)
    - Literature focus much on factorization
  - Future: Inexact subdomain solves
    - Algorithmic changes for reduced memory/computation
    - Hierarchical solvers or other preconditioners
- Dense linear algebra
  - Iterative solution acceleration
    - Subspace recycling (projections)
  - Sparse direct solvers (supernodal variants)
  - Vectorization via BLAS

Sparse Linear Solvers

Sandia National Laboratories

- **MKL Pardiso:**
  - Threaded factorization and solve phases
  - Earlier disappointments with solve phase
- **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-based
  - Threaded Ng-Peyton* (NPT, D)
    - Factorization and solve phases
    - OpenMP/Tasks
- **Others: WSMP, SuperLU_MT, … modularity key**

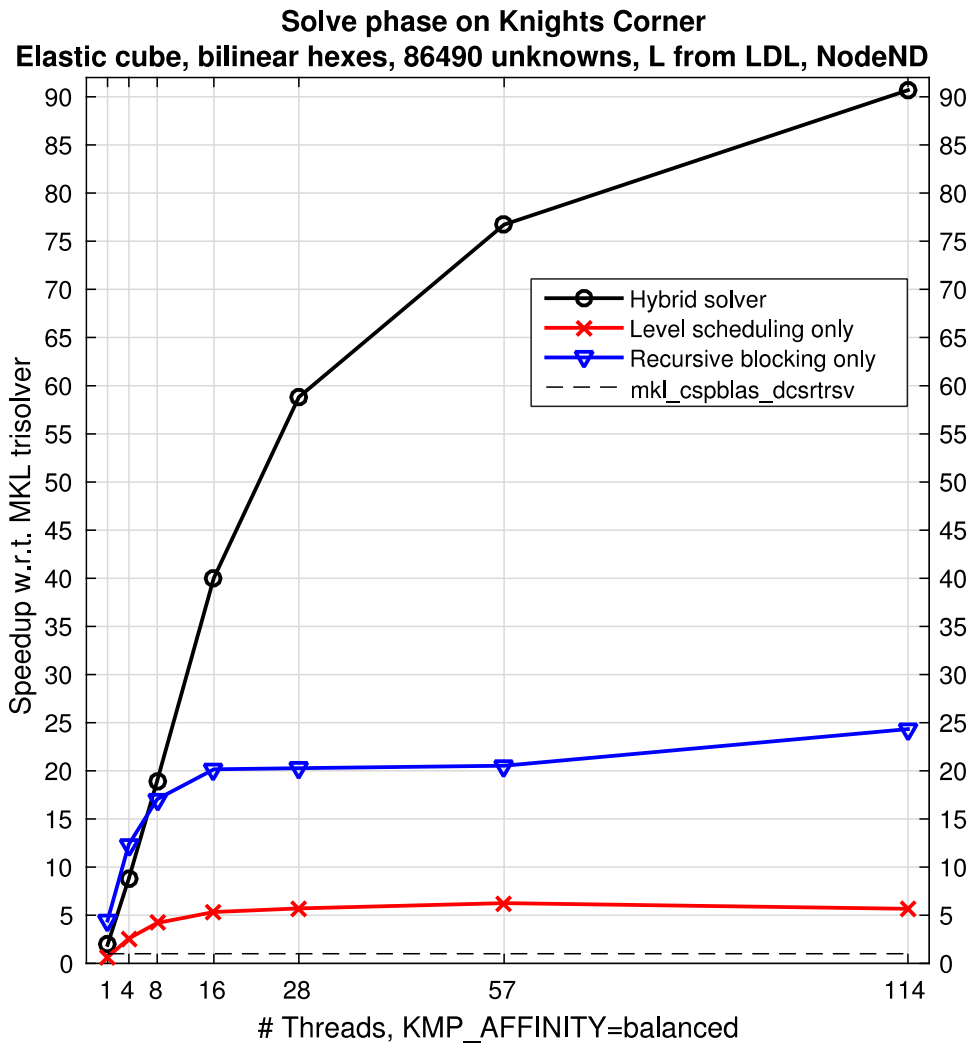
*Esmond G. Ng and Barry W. Peyton, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Comput., Vol. 14, No. 5, pp. 1034-1056, 1993.

Sparse Linear Solvers

Sandia National Laboratories

HTS:

- Multithreaded Direct Sparse Triangular Solver Combining Level Scheduling and Recursive Blocking (Andrew Bradley)
- Currently OpenMP and C++98
- See link below for talk and presentation



https://trilinos.org/community/events/trilinos-user-group-meeting-2015/

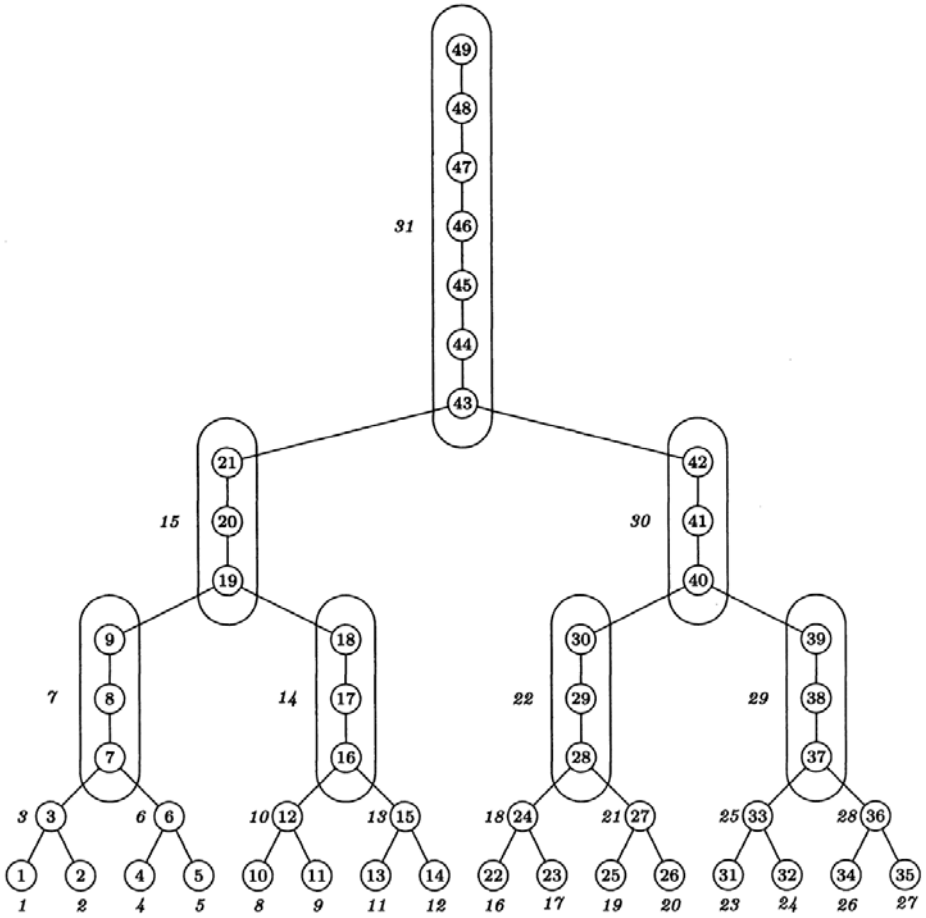
SIAM CSC '16 paper also accepted (final draft to be submitted later this month)

Sparse Linear Solvers

Sandia National Laboratories

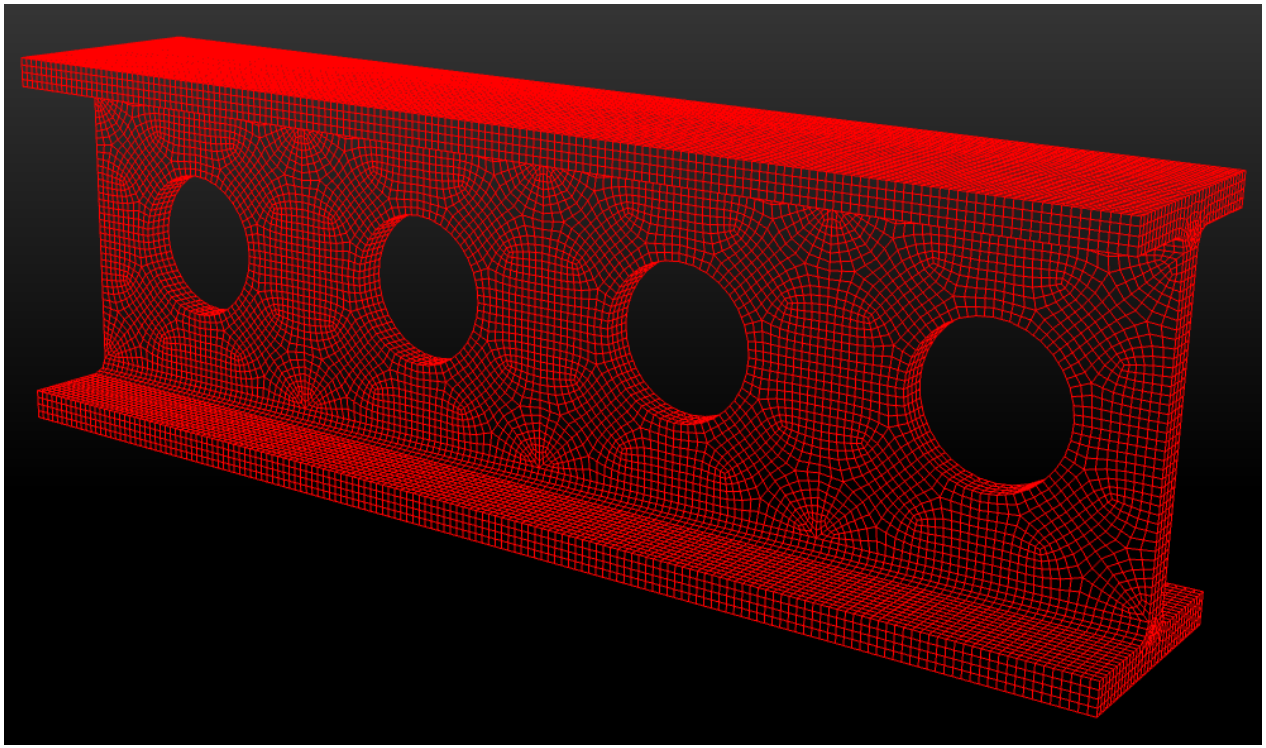
NPT:

- Threaded version of the Ng-Peyton solver using OpenMP (D).
- Uses elimination tree for task parallelism in factor and solve phases
- Tasks for dense linear algebra.
- Left looking supernodal
- BLAS for vectorization



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.

Morgan KNC*

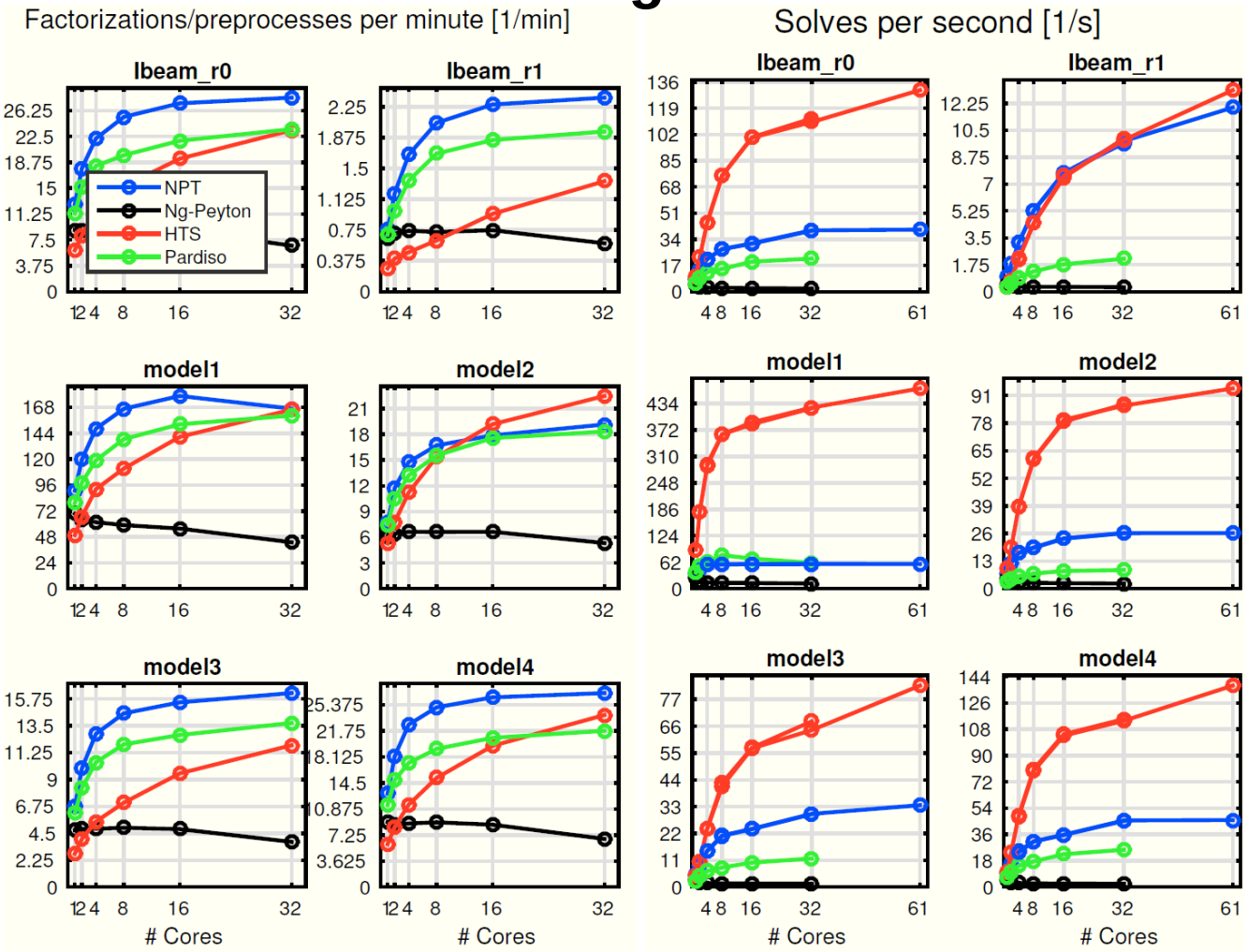


Figure 4: KNC, 61 cores, 4 hardware threads/core. Results are from two runs. NPT and HTS solvers were run at higher thread counts in a separate run. Runs were done with KMP_AFFINITY=BALANCED (1 thread/core until all cores used, then add more threads round robin) and KMP_AFFINITY=COMPACT (fill a core 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

Target Applications

Sandia National Laboratories

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

**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.

Target Applications

Sandia National Laboratories

- **Sierra/SD (Structural Dynamics):**
  - **Modal Analysis:** $(K - \sigma M)x = b$
    - Each eigen-mode requires multiple linear solves
    - Each linear solve requires 10s to 100s of iterations
    - Each iteration requires subdomain sparse triangular solves
  - **Implicit Transient Dynamics:** $(a_1 M + a_2 C + K)x = b$
    - Simulations with > 1000 time steps not uncommon
    - Each time step requires a linear solve, …
  - **Message:**
    - Triangular solves very important for structural dynamics
    - Factorization costs can be amortized over several linear solves
    - Factorization costs more important for solid mechanics applications where matrices are changing

Sierra/SD Integration (Early Results)

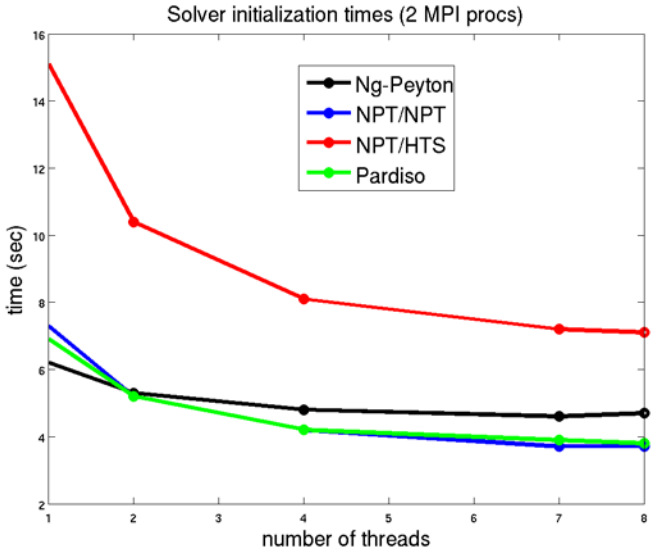
Sandia National Laboratories

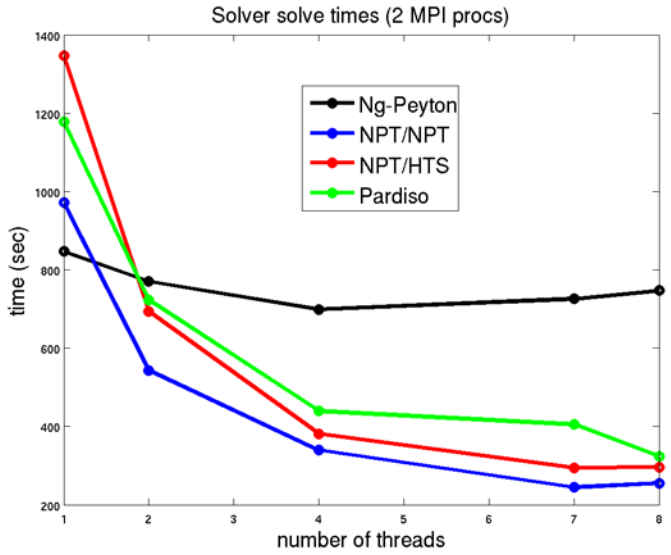
Run using 2 MPI processes (Sandy Bridge, 2 sockets, 8 cores/socket)

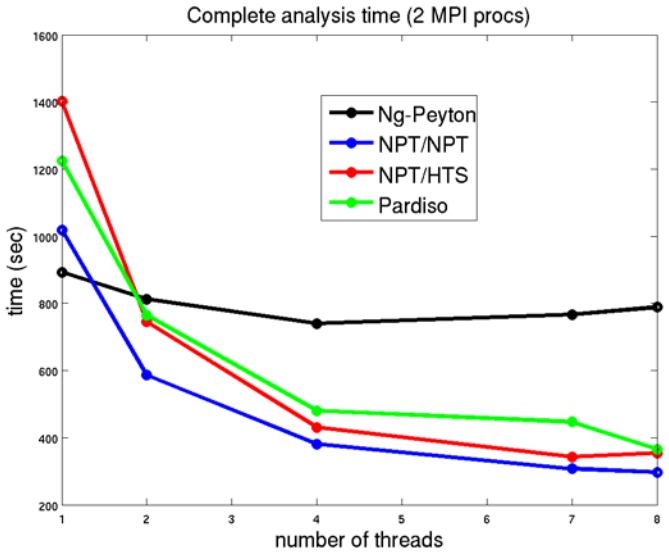
Problem too easy using default GDSW solver parameters (2 iters/solve average)

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)

Closing Remarks

- **Threaded Sparse Solvers:**
  - **Convenient way to introduce more parallelism**
    - **Small changes to existing DD preconditioners**
    - **Modularity is key**
  - **Early results encouraging**
    - **HTS very promising for larger thread counts**
    - **Clear benefits for structural dynamics application**
- **Future Work:**
  - **Determine suitable number of threads per subdomain on newer architectures (e.g. Knights Landing)**
  - **Explore over-decomposition concepts**
    - **Multiple subdomains per MPI rank**
    - **small number of threads per subdomain**
  - **Explore inexact methods for DD preconditioners, …**

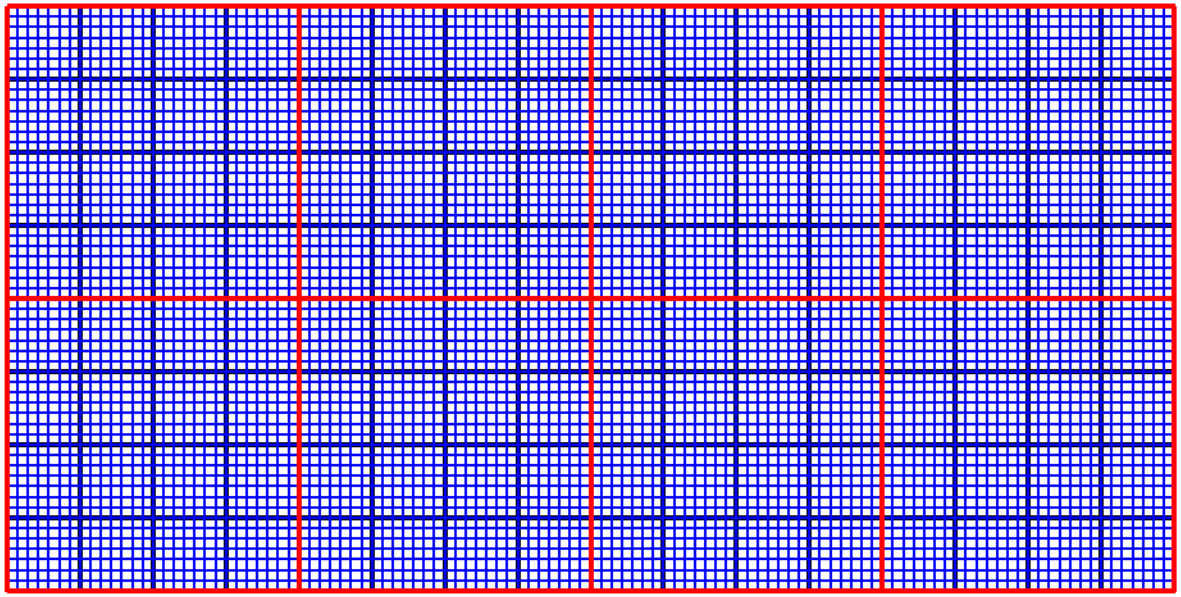
Extra Slides

Sandia National Laboratories

Domain Decomposition

Sandia National Laboratories

Multi-level Additive Schwarz Preconditioner:



$$Ax = b$$

$$AM^{-1}y = b$$

$$M^{-1}r = \sum_{j=1}^{M-1} \sum_{i=1}^{N_j} R_{ij}^T (R_{ij} A_j R_{ij}^T)^{-1} R_{ij} r_j + \Phi_M (\Phi_M^T A \Phi_M)^{-1} \Phi_M^T r$$

$$r_j = \Phi_j^T r, \quad \Phi_1 = I \qquad A_j = \Phi_j A \Phi_j^T$$

Sparse Linear Solvers

Sandia National Laboratories

Tacho:

- Task-based Cholesky.
- Currently uses Kokkos /Pthreads for tasks
- Moving to OpenMP backend
- Not integrated yet in structural mechanics codes
- See link below for report on incomplete factorizations

https://arxiv.org/pdf/1601.05871.pdf

Morgan Haswell*

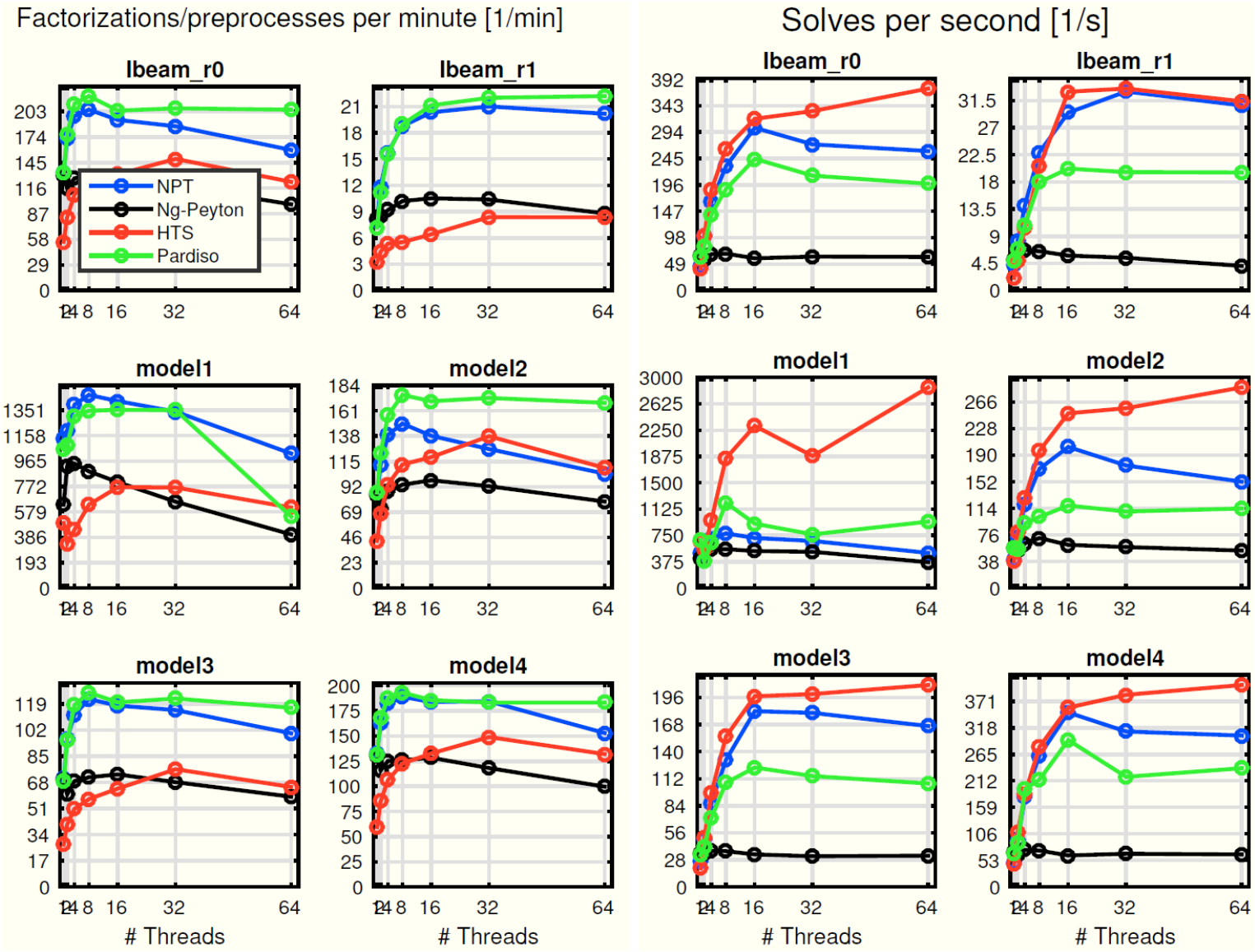


Figure 3: Haswell, 32 cores on 2 sockets, 2 hardware threads/core. Runs were done the same as before.

*results courtesy of Andrew Bradley

Morgan Sandy Bridge*

Sandia National Laboratories

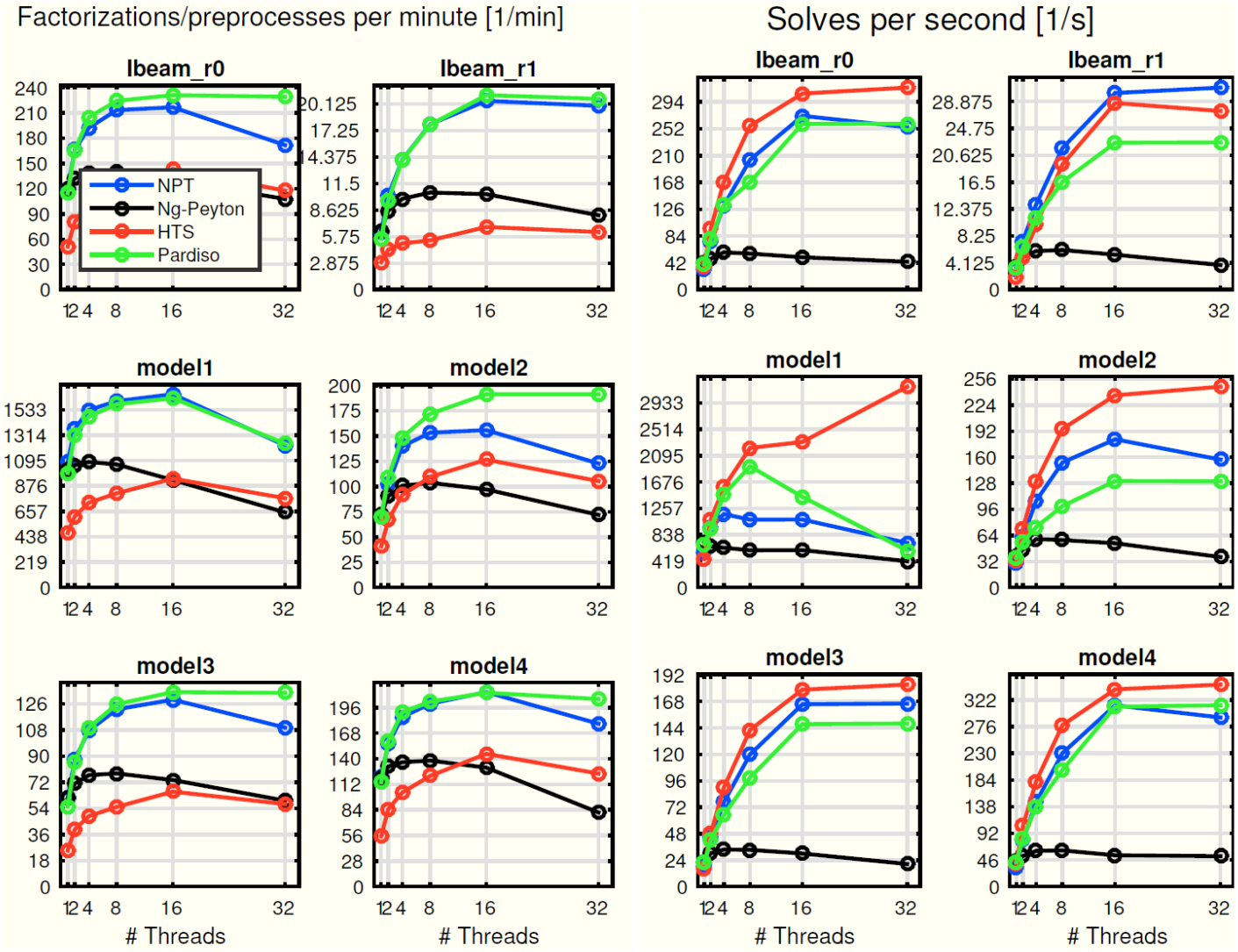


Figure 1: Sandy Bridge, 16 cores on 2 sockets, 2 hardware threads per core. Runs were done with OMP_PROC_BIND=SPREAD and OMP_PROC_BIND=CLOSE, always with OMP_PLACES=CORES, and with OMP_NUM_THREADS set to each number indicated in the x axis. The best time for a given thread count is reported.

*results courtesy of Andrew Bradley

Morgan Ivy Bridge*

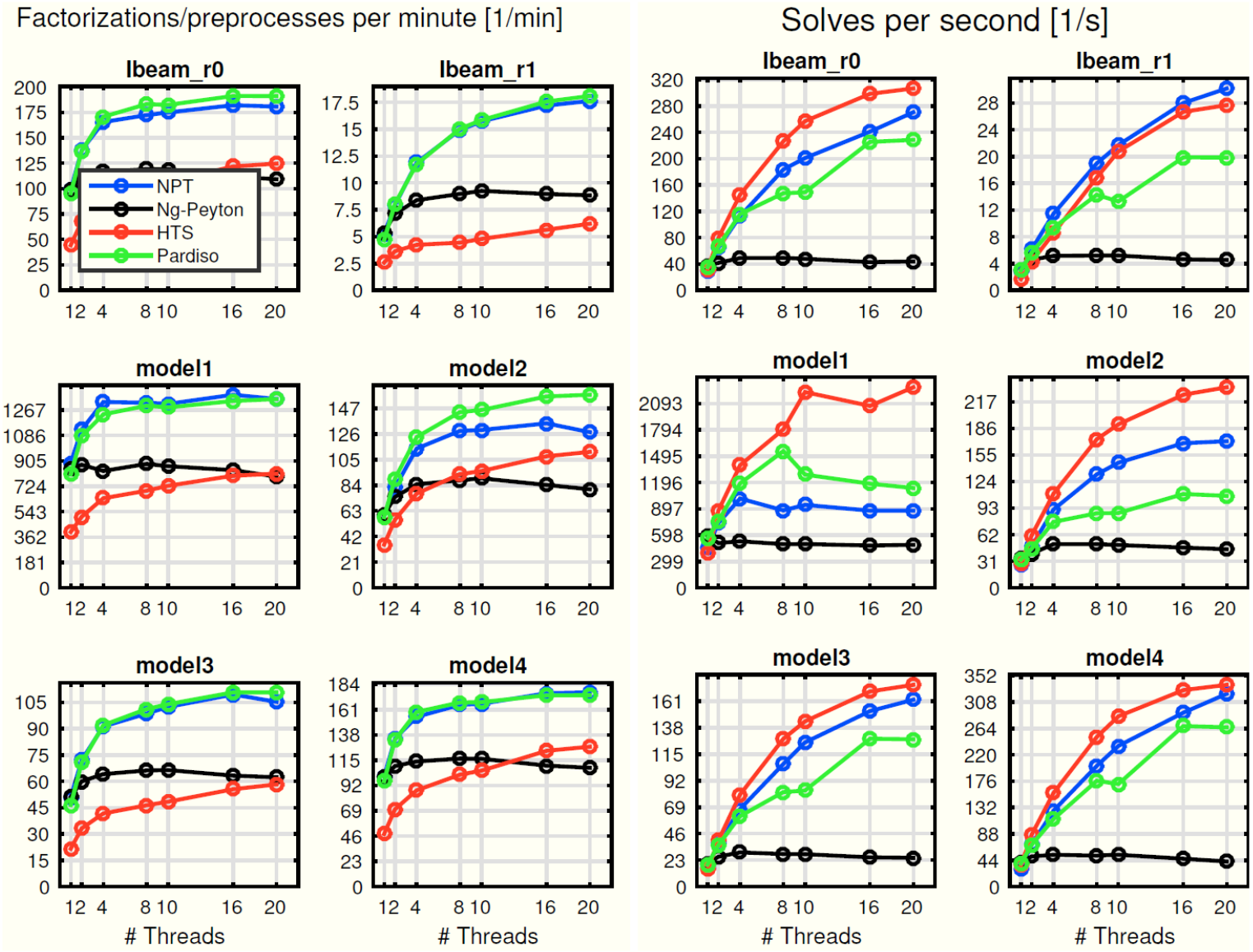
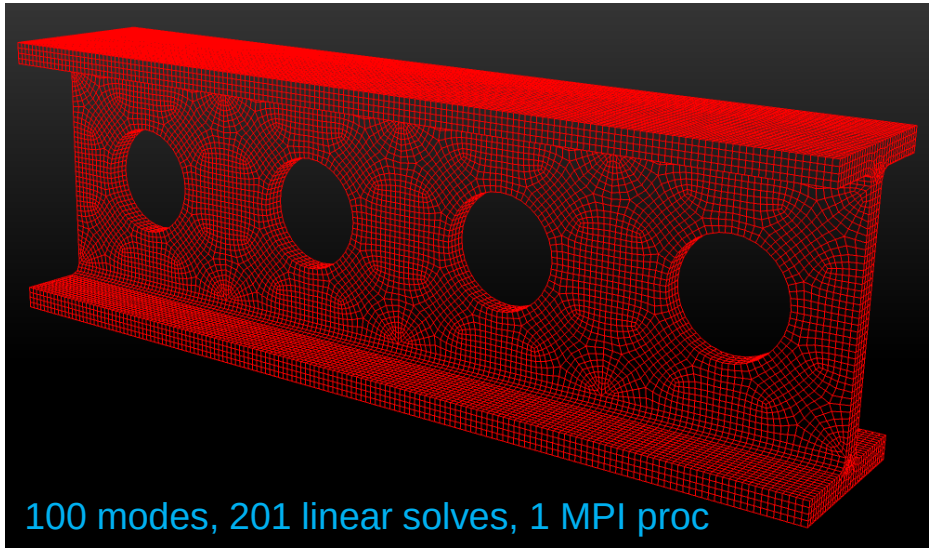
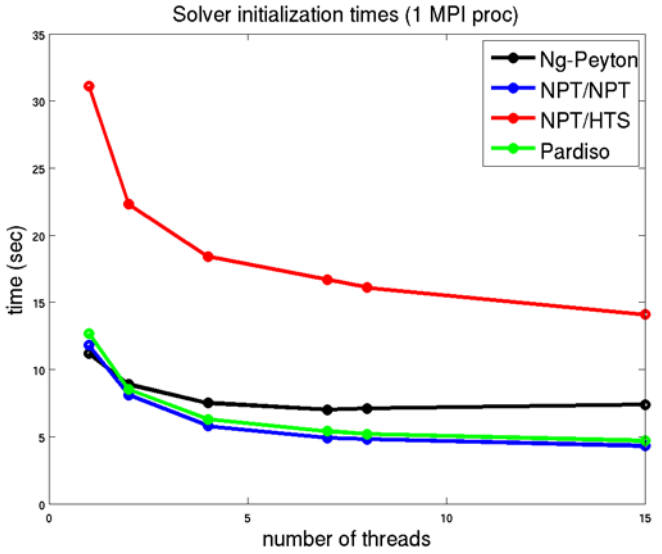


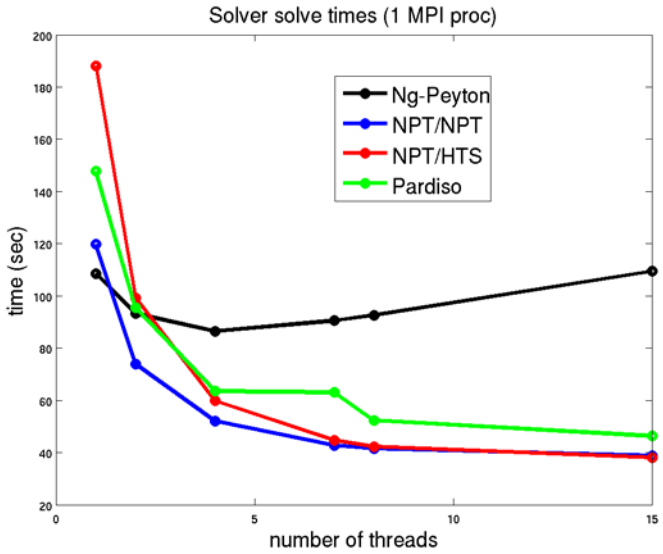
Figure 2: Ivy Bridge, 20 cores on 2 sockets, 2 hardware threads/core (not used). Runs were done the same as before.

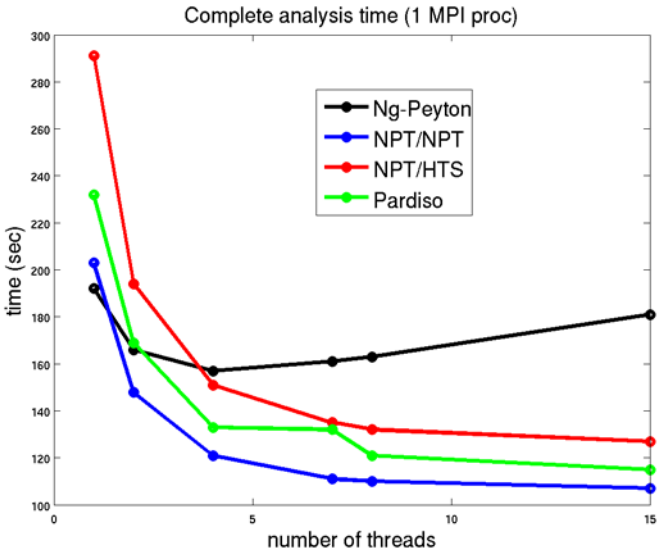
*results courtesy of Andrew Bradley

Sierra/SD Integration (Early Results)









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