SAND2016-0695PE

Domain Decomposition Solver Preparations for Trinity

Clark Dohrmann

Computational Solid Mechanics & Structural Dynamics Department

Applied Computer Science Meeting
Los Alamos National Laboratory
February 1-5, 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

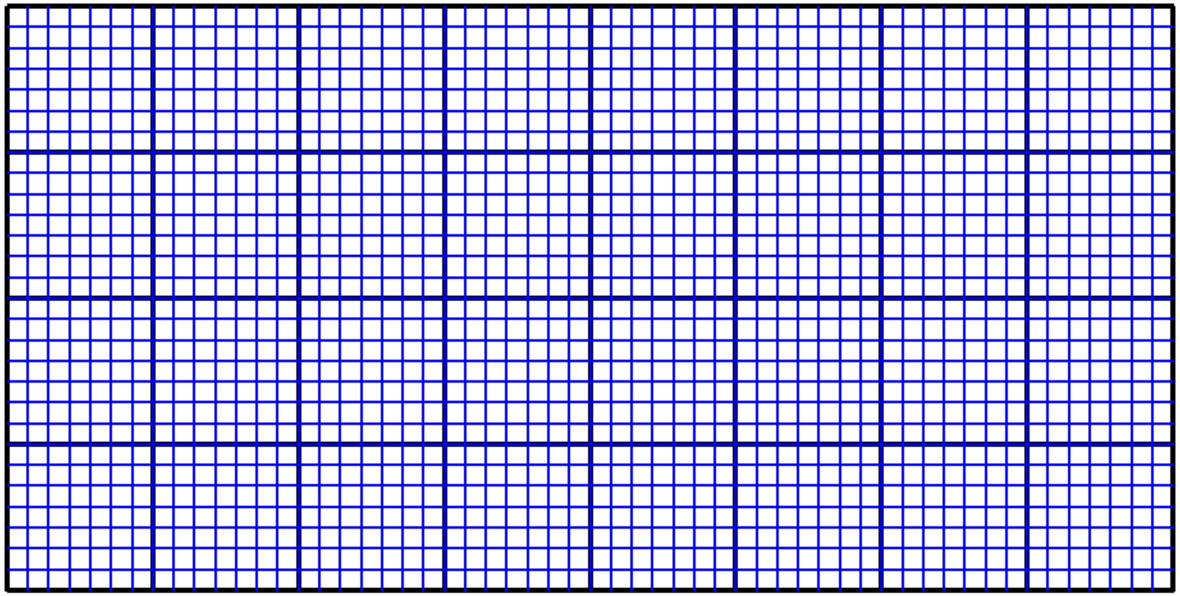
- Andrew Bradley
- Siva Rajamanickam
- Erik Boman

Outline

- **Domain Decomposition Solvers:**
  - Introduction
  - Computational Kernels
- **Sparse Linear Solvers:**
  - Options & Threading Approaches
  - Recent Performance Results
- **Integration Efforts:**
  - Target Applications
  - Some Early Results
- **Ongoing Work:**
  - Intel Interactions
  - Algorithms, …

Domain Decomposition Solvers

Two-level Additive Schwarz Preconditioner:



$$Ax = b$$

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

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

$R_i = \text{Boolean matrix}$ $\quad\quad$ $\Phi = \text{interpolation matrix}$

Domain Decomposition Solvers

- **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)**
    - **Future: Inexact subdomain solves**
      - **Reduced memory, smaller coarse problems, …**
  - **Dense linear algebra**
    - **Iterative solution acceleration**
      - **Subspace recycling (projections)**
    - **Sparse direct solvers (supernodal variants)**

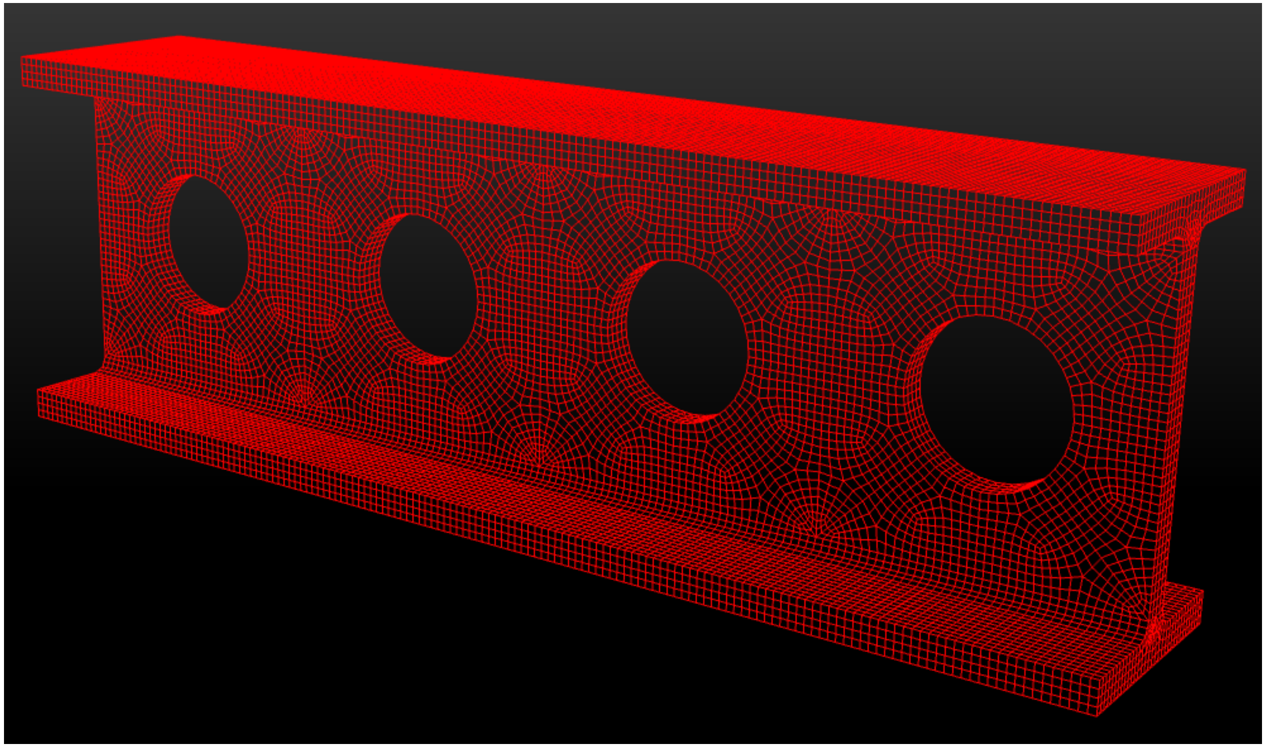
Sparse Linear Solvers

- **MKL Pardiso:**
  - **Threaded factorization and solve phases**
  - **Earlier disappointments with solve phase**
- **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**
    - **Coming soon**
  - **Threaded Ng-Peyton* (NPT, D)**
    - **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.

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. Intel 15 compiler used

Sparse Linear Solvers (Recent Results*)

- **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*

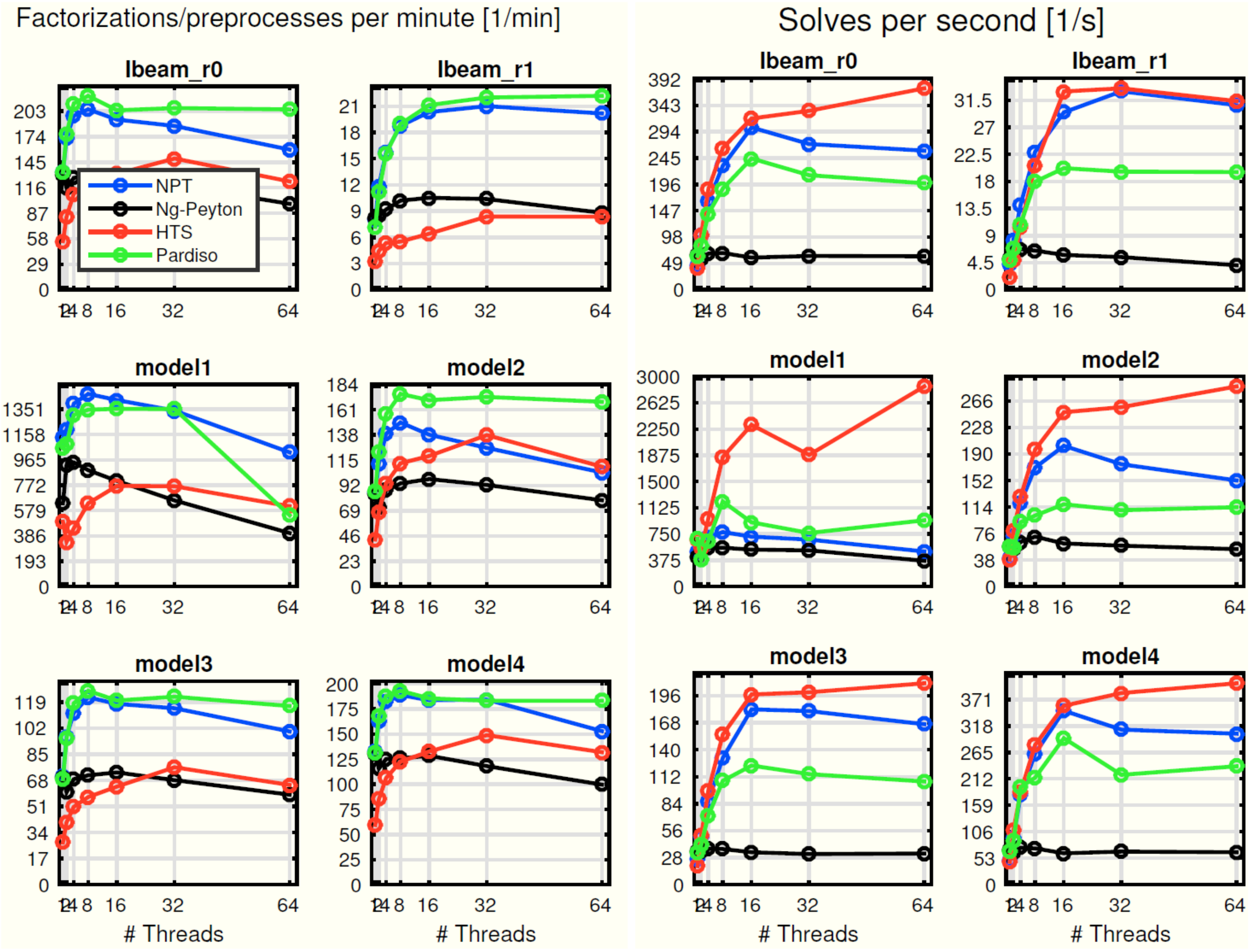


Figure 3: Haswell, 32 cores on 2 sockets, 2 hardware threads/core. Runs were done the same as before.

*results courtesy of Andrew Bradley

Morgan KNC*

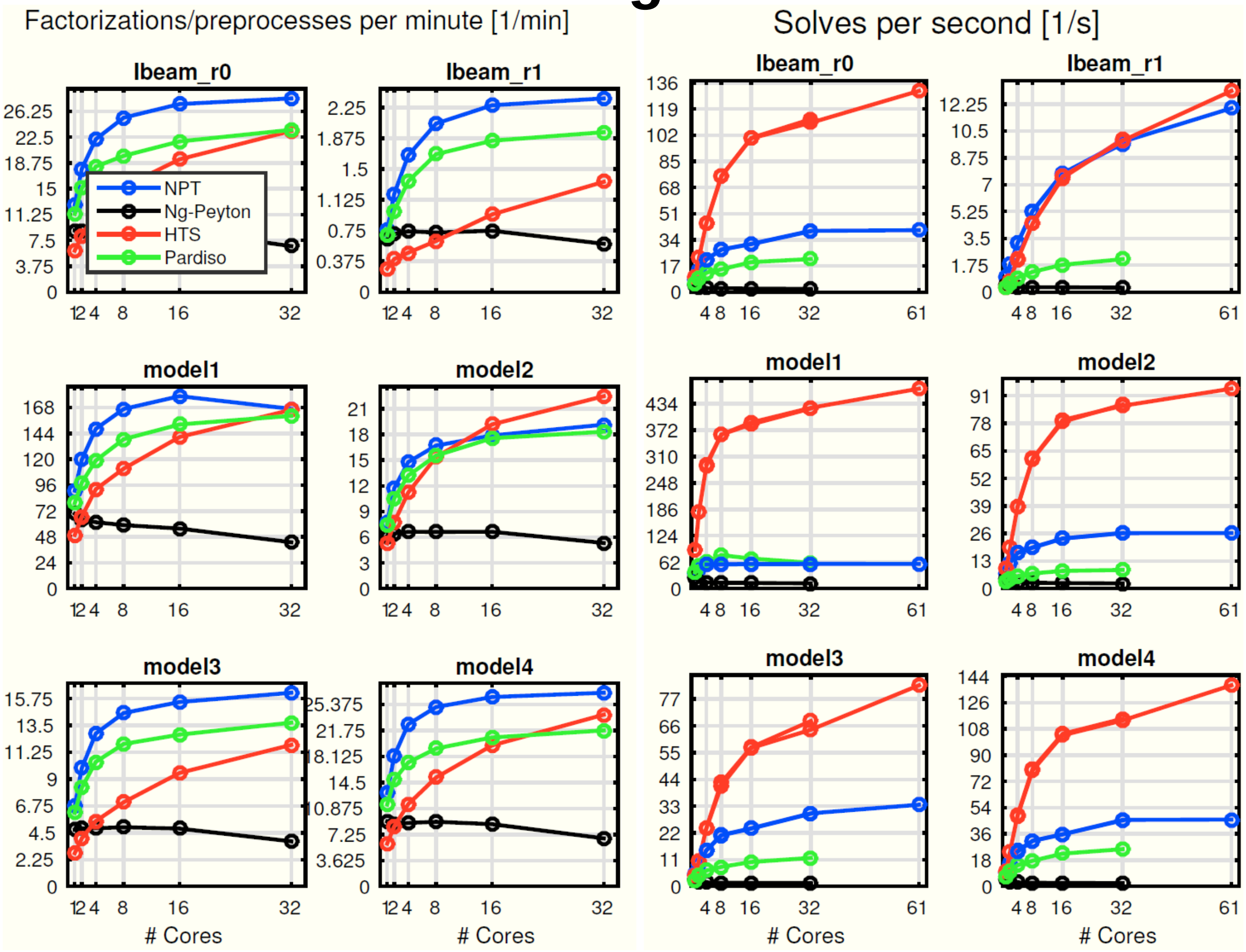


Figure 4: KNC, 61 cores, 4 hardware threads/core. Results are from two runs. NPT and HTS solvers were run at higher thread counts in a separate run. Runs were done with KMP_AFFINITY=BALANCED (1 thread/core until all cores used, then add more threads round robin) and KMP_AFFINITY=COMPACT (fill a core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

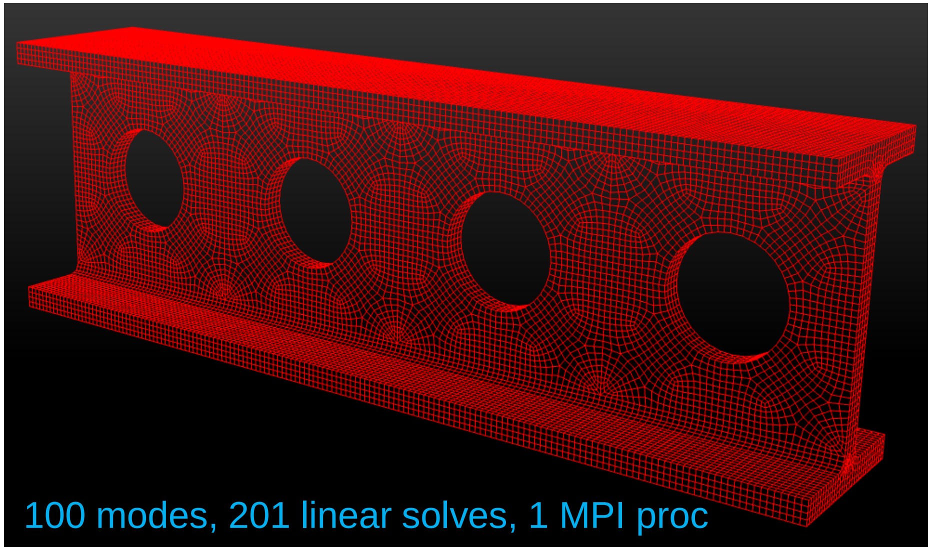
Integration Efforts (Target Applications)

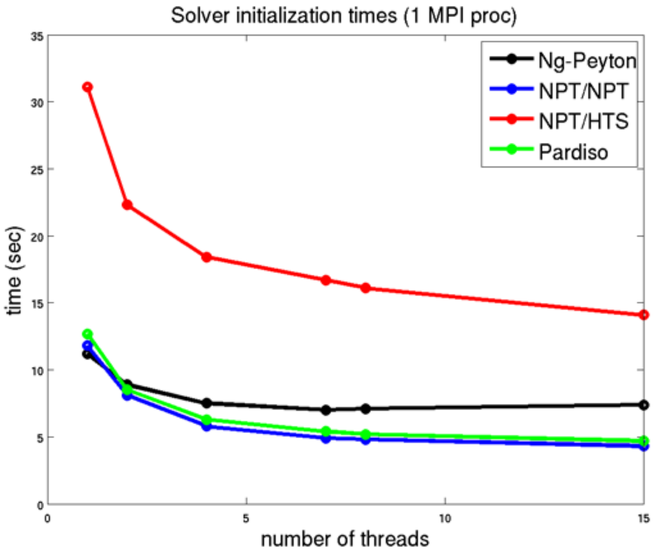
- **Sierra/SD (Structural Dynamics):**
  - **Modal, transient, frequency response, static, inverse, … analyses (primarily linear)**
  - **Operator matrix often constant $\Rightarrow$ many solves/factorization**
  - **GDSW* iterative solver**
- **Sierra/SM (Solid Mechanics):**
  - **Nonlinear explicit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

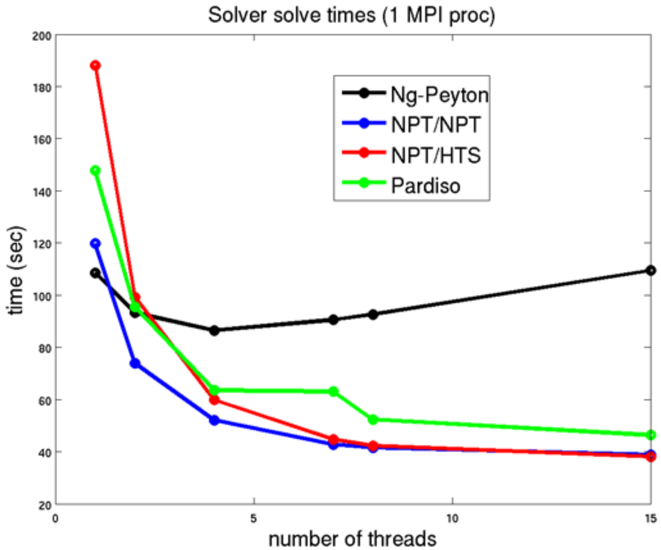
**Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity*, Int. J. Numer. Meth. Engng, Vol. 82, pp. 157-183, 2010.

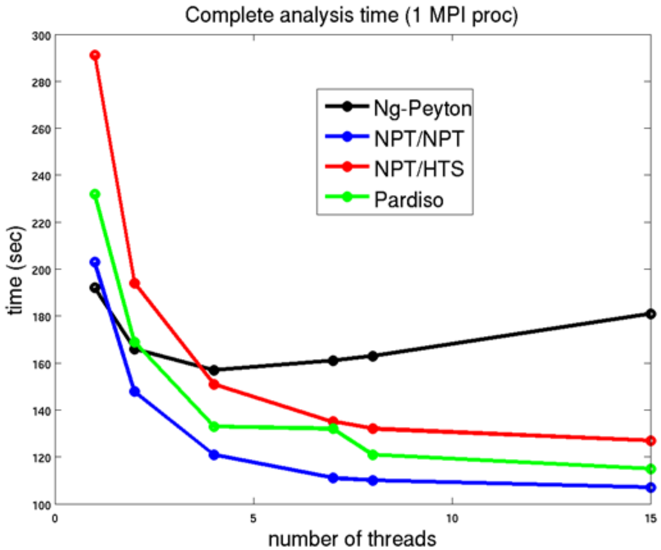
***FETI-DP: A dual-primal unified FETI method – part I: A faster alternative to the two-level FETI method,* Int. J. Numer. Meth. Engng, Vol. 50, pp. 1523-1544, 2001.

Integration Efforts (Early Results)









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

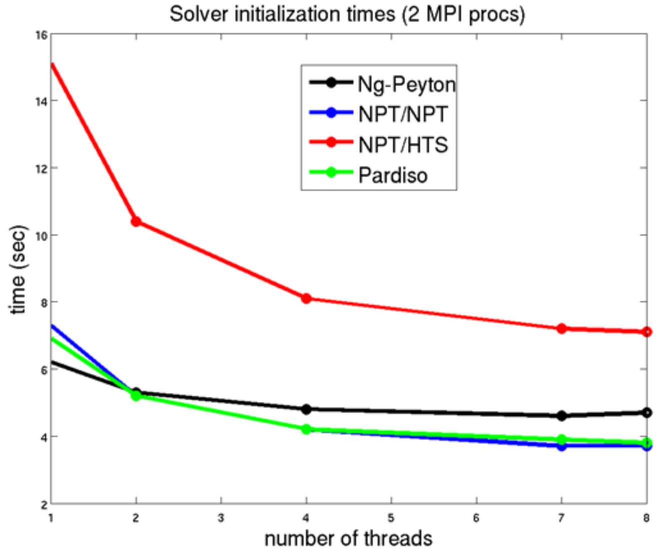
Integration Efforts (Early Results)

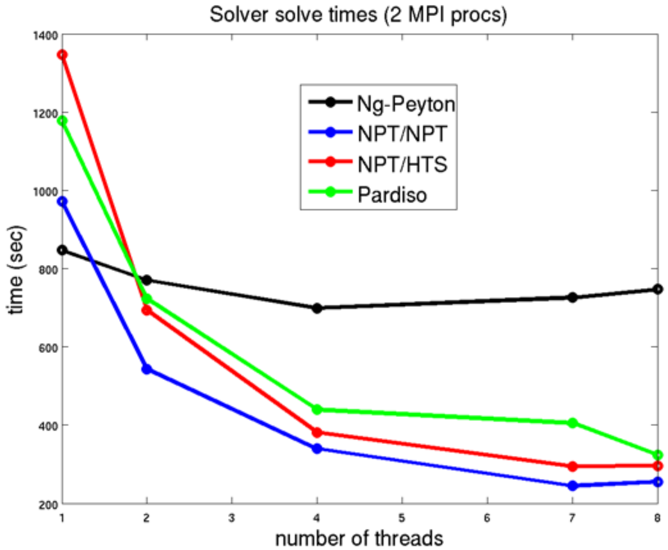
Run using 2 MPI processes on my blade (Sandy Bridge, 2 sockets, 8 cores/socket)

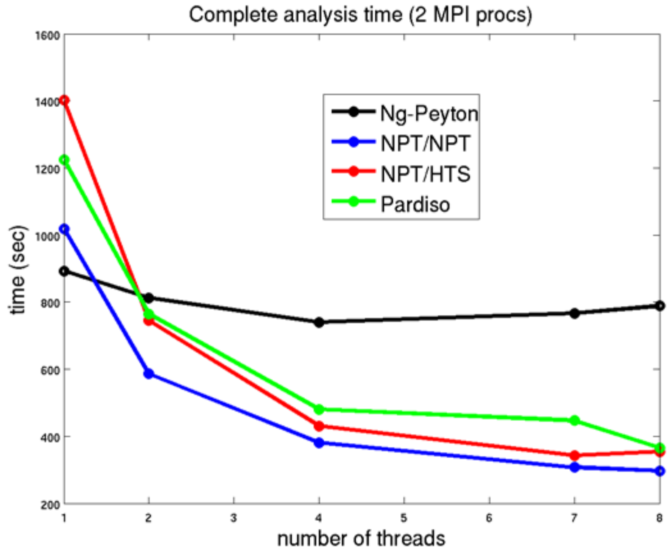
Problem too easy using default GDSW solver parameters (2 iters/solve average)

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)

Ongoing Work

- **Intel Interactions:**
  - **“Dungeon” session in two weeks**
    - **Threaded linear solvers**
    - **BDDC* solver proxy**
- **Algorithms:**
  - **Adapt/tune sparse direct solvers (Haswell, KNL)**
  - **Over-decomposition, inexact solves**
  - **Intra-node focus thus far, inter-node to follow**
- **Integration:**
  - **Initial integration of new sparse solvers in Sierra/SD and Sierra/SM scheduled for Q3 FY16**
  - **Updated domain decomposition algorithms**

*Balancing Domain Decomposition by Constraints, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., Vol. 25, No. 1, pp. 246-258, 2003.

Recap

- **Threaded Sparse Direct Solvers:**
  - **Doing a good job here can help a lot**
  - **Effective threading of solve phase very important**
    - **HTS looks very promising**
- **Domain Decomposition Strategy:**
  - **Push subdomains to larger sizes**
    - **Potential for limited changes to existing algorithms**
    - **Experience shows fewer subdomains $\Rightarrow$ fewer iterations**
  - **Consider over-decomposition/inexact solves only if needed**
    - **Easily parallelized, but may take hit with iteration count**
    - **Additional work on extracting vertex separators needed**
    - **Additional opportunities for ||, but not much experience**
  - **Begin shifting focus to inter-node performance**

Extra Slides

Morgan Sandy Bridge*

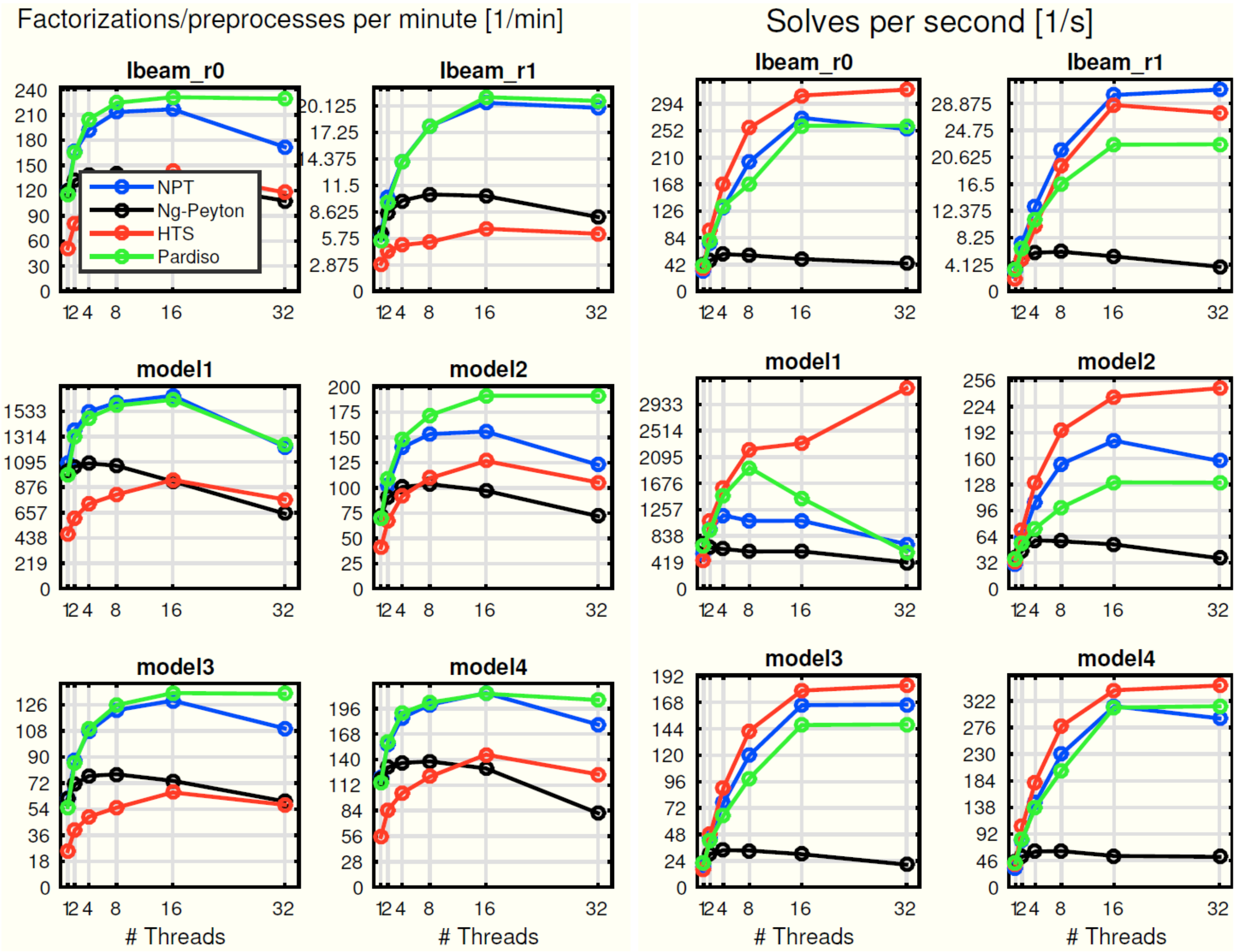


Figure 1: Sandy Bridge, 16 cores on 2 sockets, 2 hardware threads per core. Runs were done with OMP_PROC_BIND=SPREAD and OMP_PROC_BIND=CLOSE, always with OMP_PLACES=CORES, and with OMP_NUM_THREADS set to each number indicated in the x axis. The best time for a given thread count is reported.

*results courtesy of Andrew Bradley

Morgan Ivy Bridge*

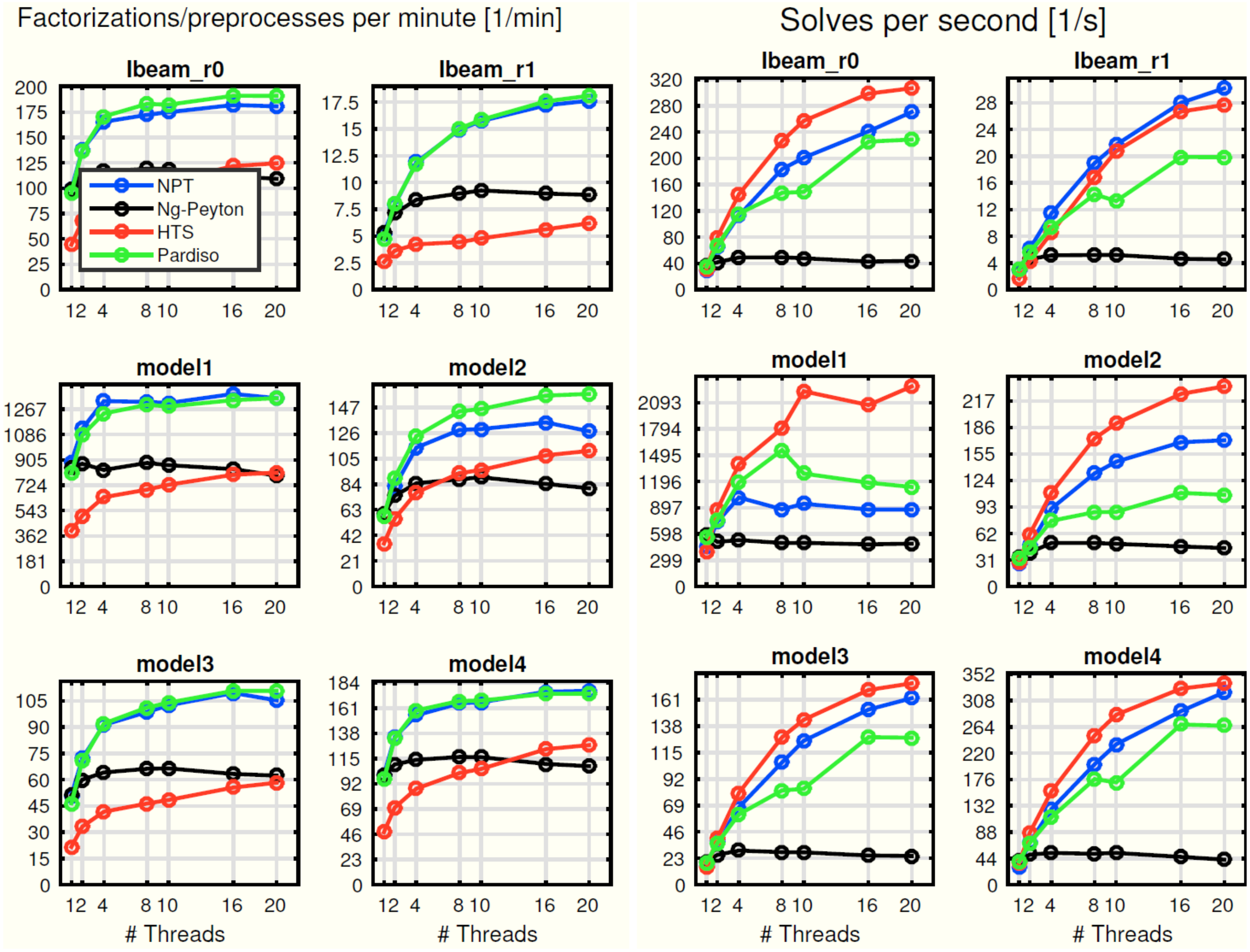
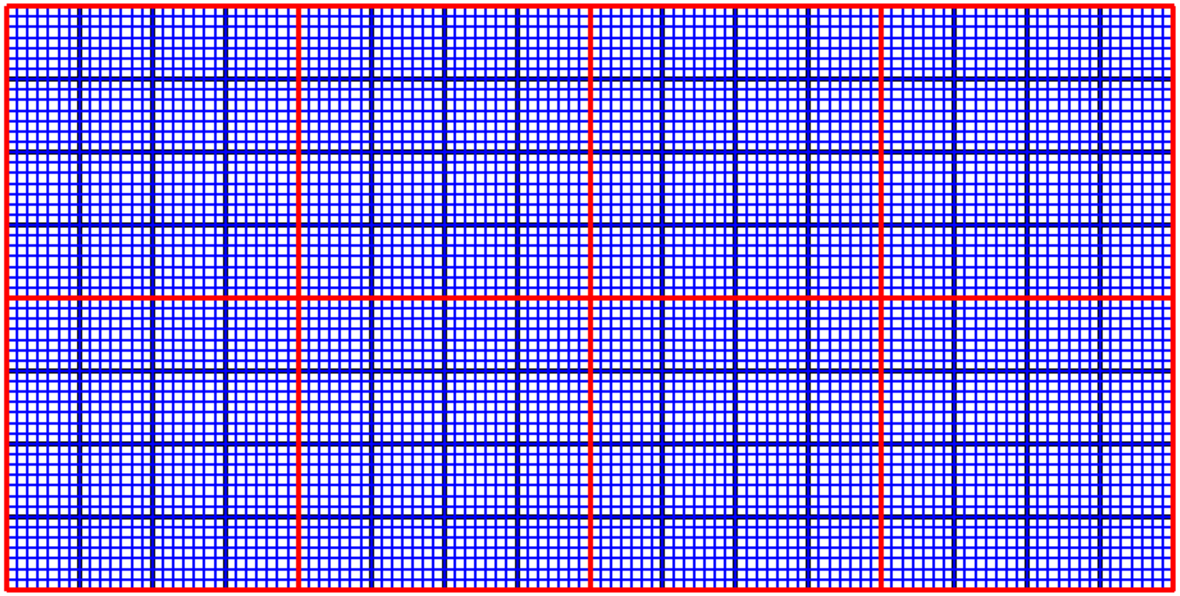


Figure 2: Ivy Bridge, 20 cores on 2 sockets, 2 hardware threads/core (not used). Runs were done the same as before.

*results courtesy of Andrew Bradley

Domain Decomposition

Multi-level Additive Schwarz Preconditioner:



$$Ax = b$$

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

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

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