Exceptional service in the national interest

Sandia National Laboratories

A Hybrid Multithreaded Direct Sparse Triangular Solver

Andrew M. Bradley

Thanks: E. Boman, C. Dohrmann, S. Hammond, W. Held, M. Heroux, M. Hoemmen, K. Kim, S. Olivier, A. Prokopenko, S. Rajamanickam

SIAM CSC16

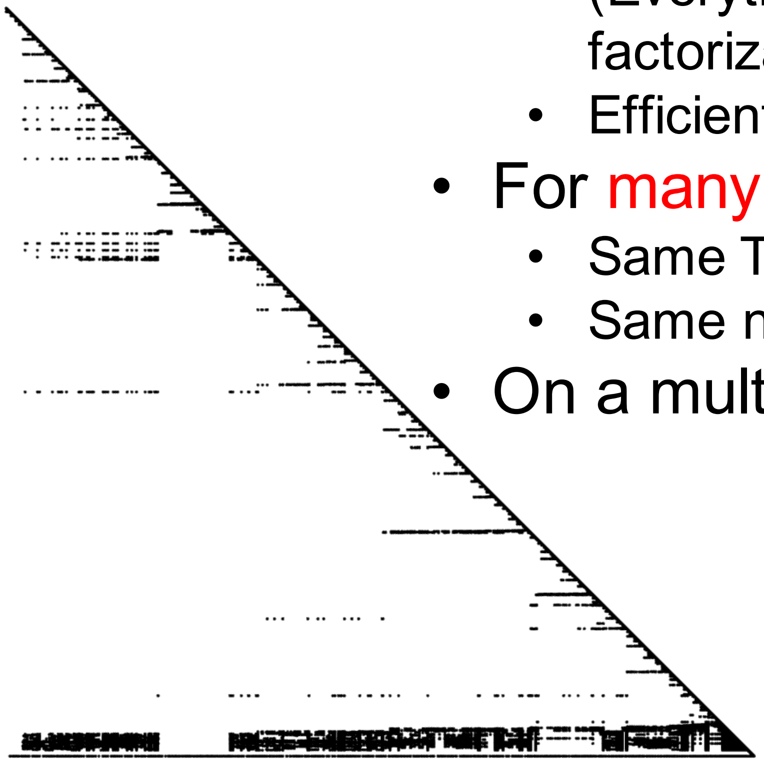
SAND2016-10150 C

U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration CCR Center for Computing Research

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.

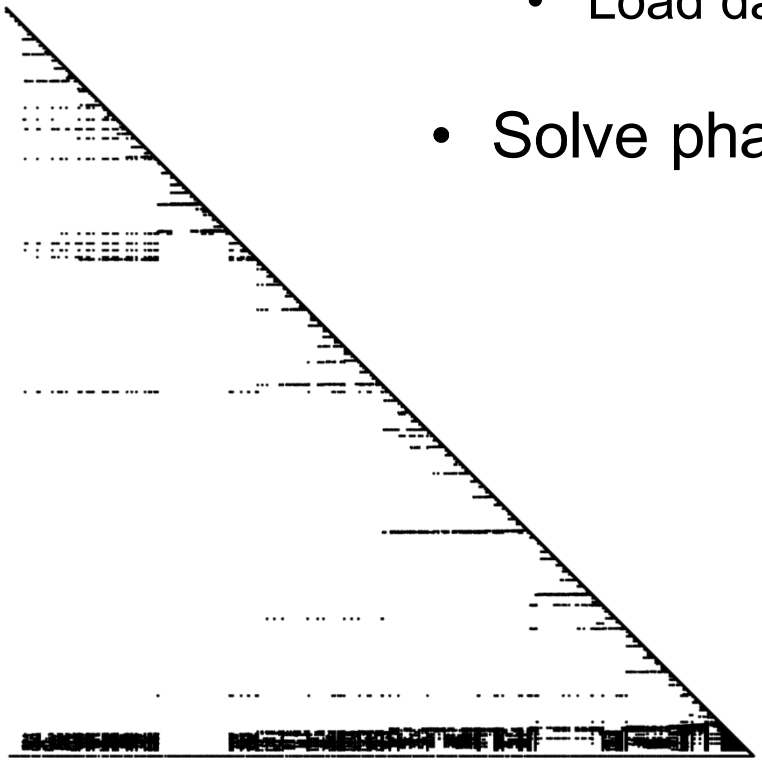
Problem Statement

- Solve R P T Q x = b
  - Upper or lower sparse triangular matrix T
  - Row scaling R
  - Permutations P, Q
  - Solution and RHS x, b
  - (Everything that is needed for LDL, LU, incomplete factorizations, etc.)
  - Efficient to absorb user data
- For many sequential RHS with
  - Same T or
  - Same nonzero pattern pat(T)
- On a multi/many-core node

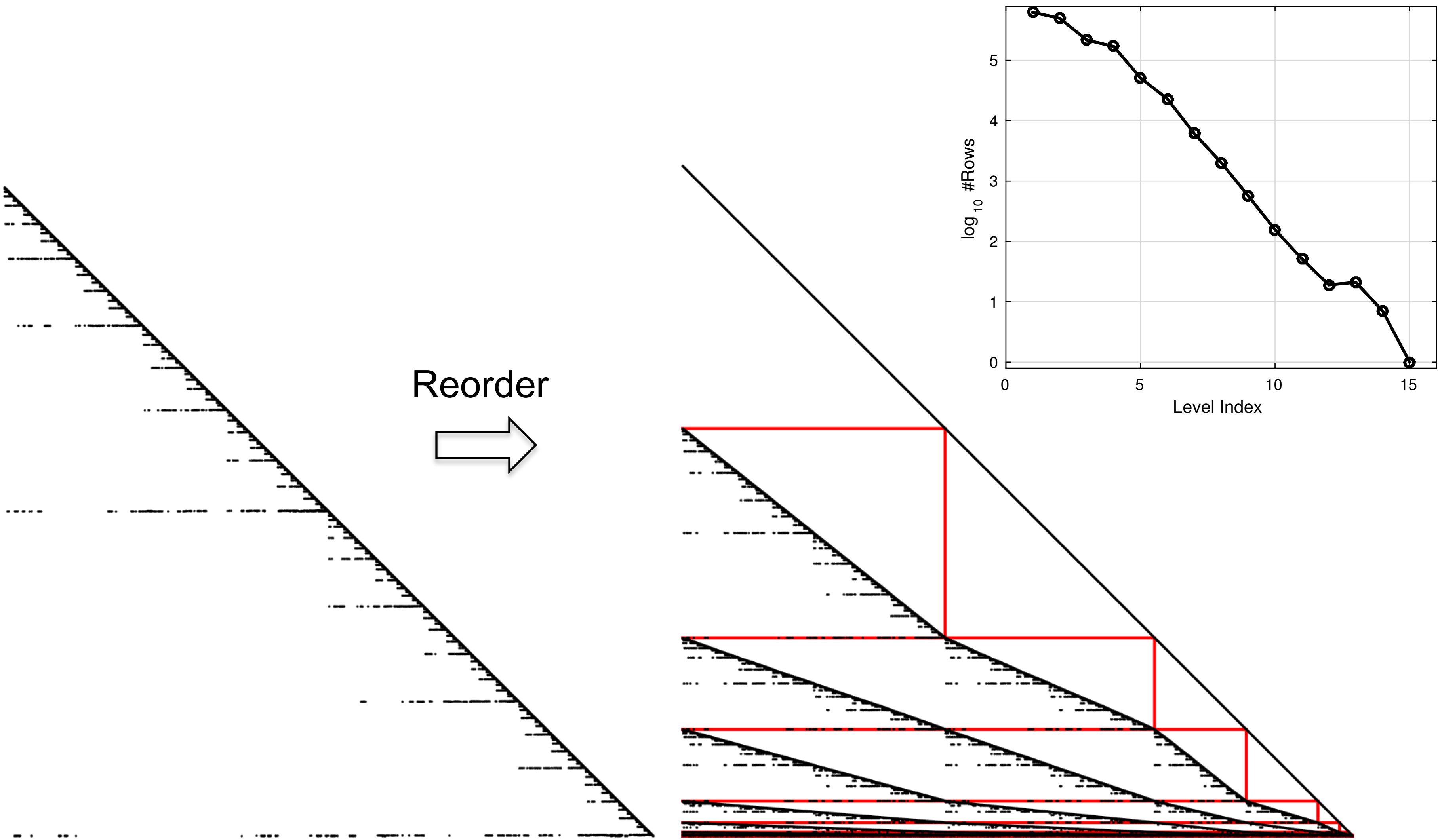


Solution Approach

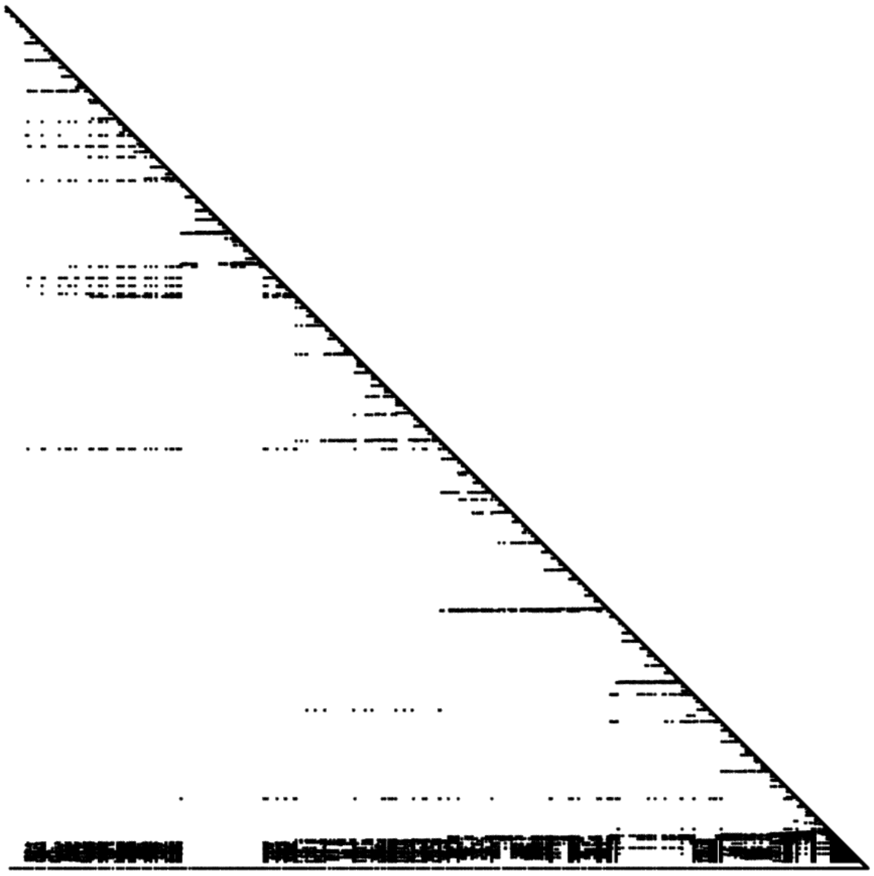
- Symbolic phase
  - Find parallelism in pat(T), the graph of T
- Numeric phase
  - Load data structures with numbers
- Solve phase

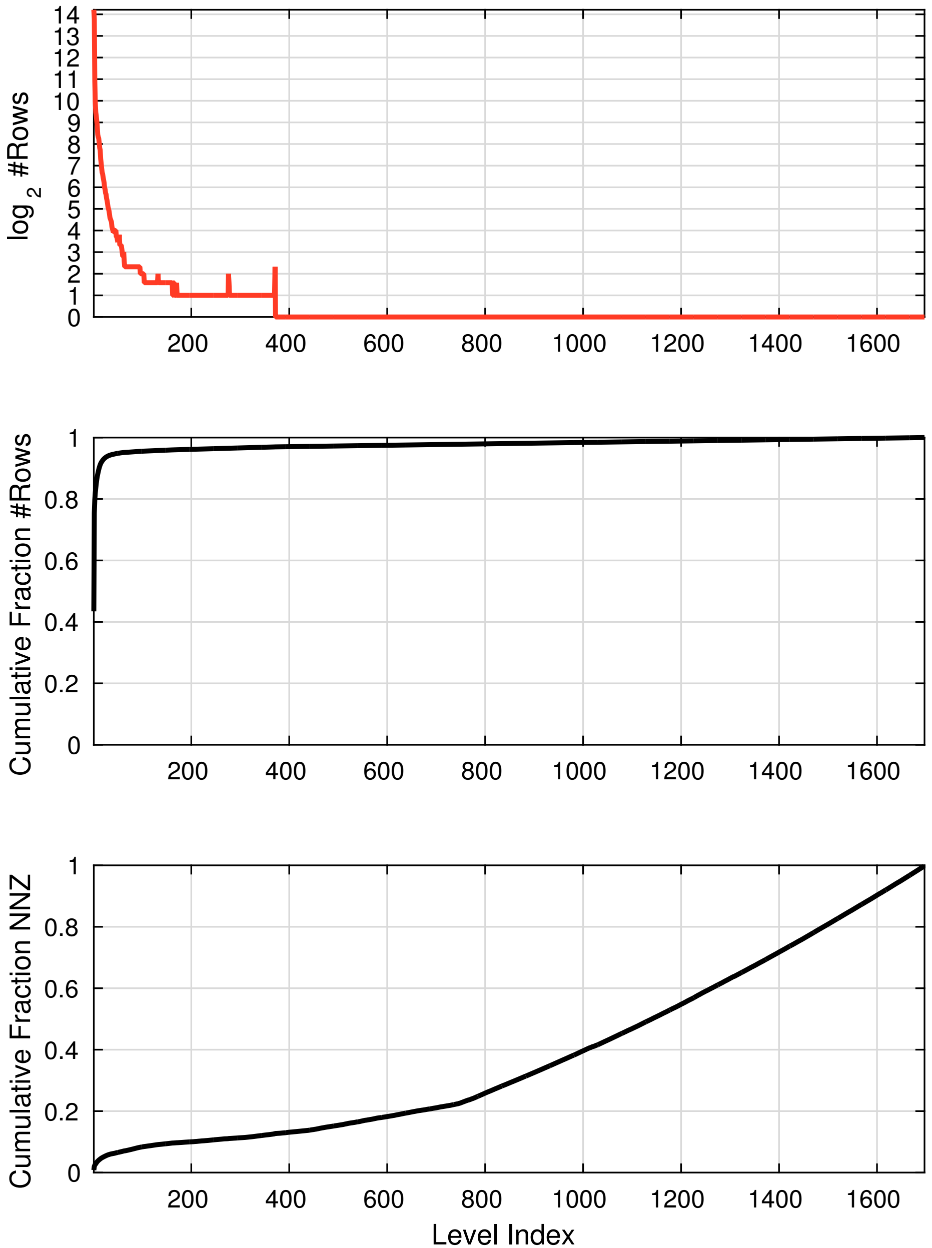


Motivation: Level Scheduling

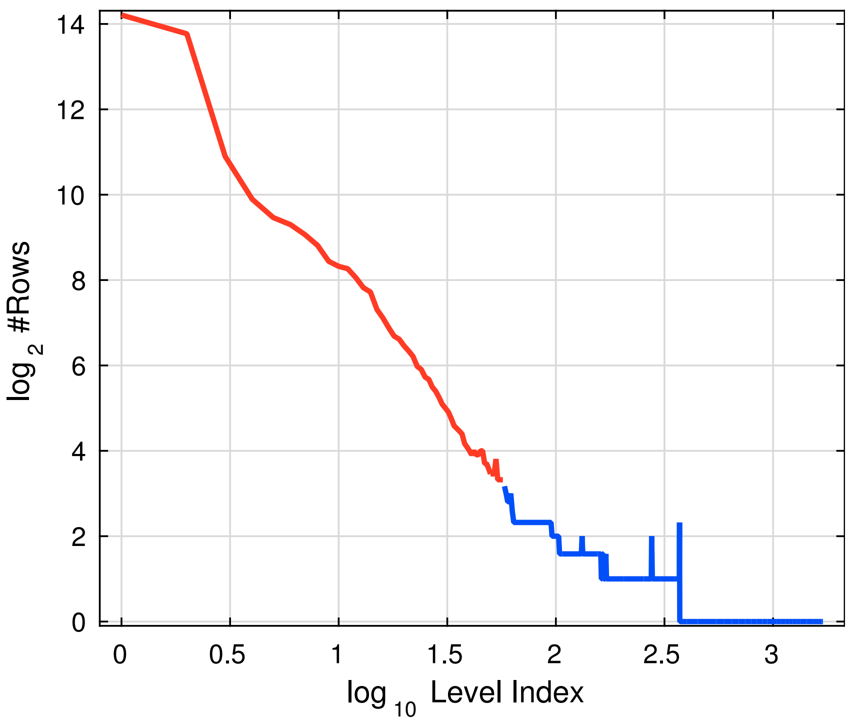


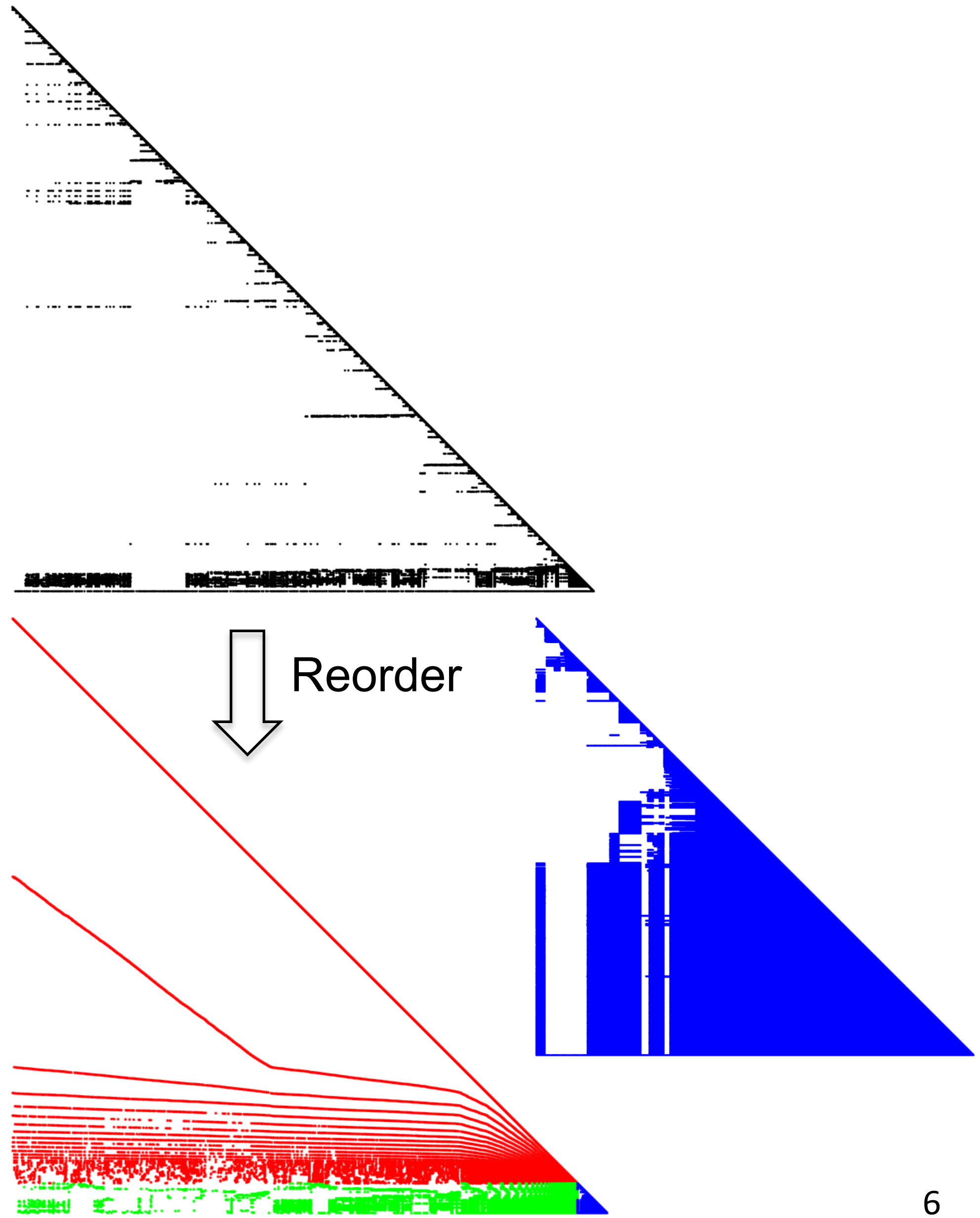
Motivation: Level Scheduling



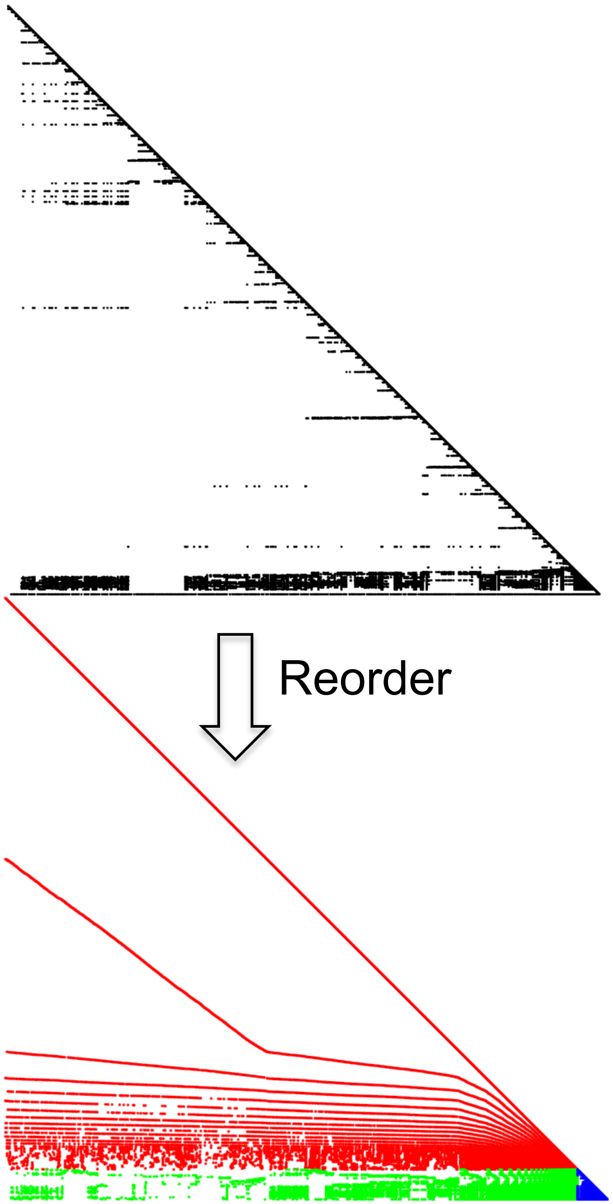


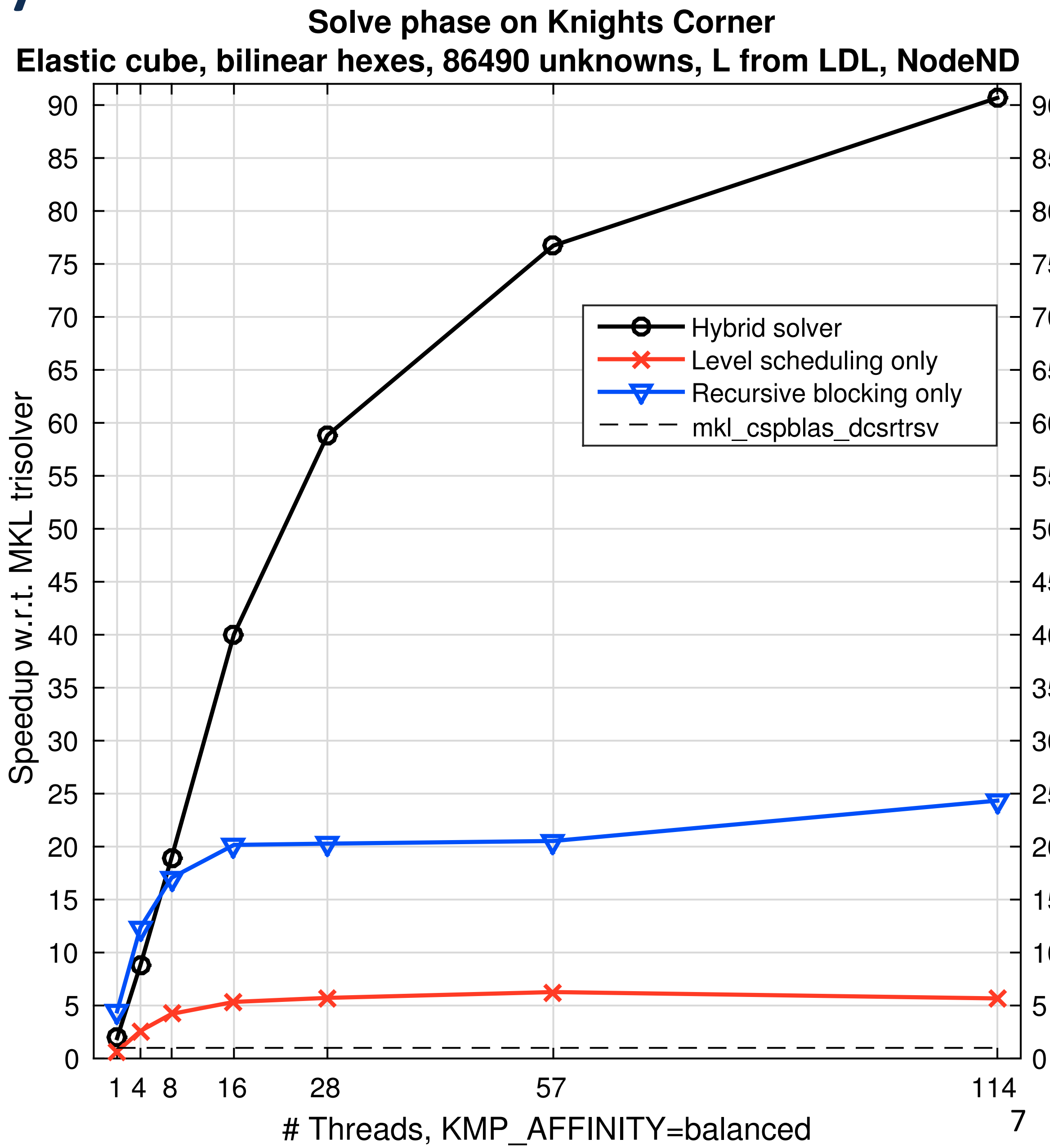
Motivation: Hybrid





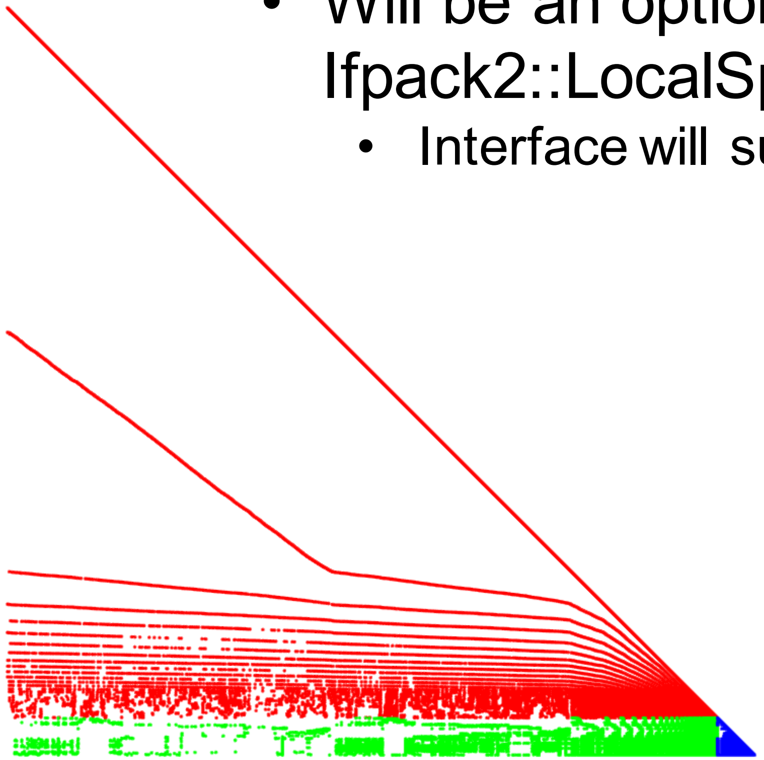
Motivation: Hybrid





Software: HTS

- Trilinos/packages/shylu/hts
- C++ and OpenMP
  - Templated on row pointer, column index, and scalar types
- CSR, CSC, forward, transpose, conjugate inputs
- Effective nonzero pattern reuse
- Will be an option in Ifpack2::LocalSparseTriangularSolver
  - Interface will support nonzero pattern reuse



Algorithms: Switching Method

- Want robustness to downward and upward spikes in N_i.
- Use levels 1 to *k*:

$$n_{\text{good}} \sim 10, \quad f_{\text{bad}} \sim 1\%$$

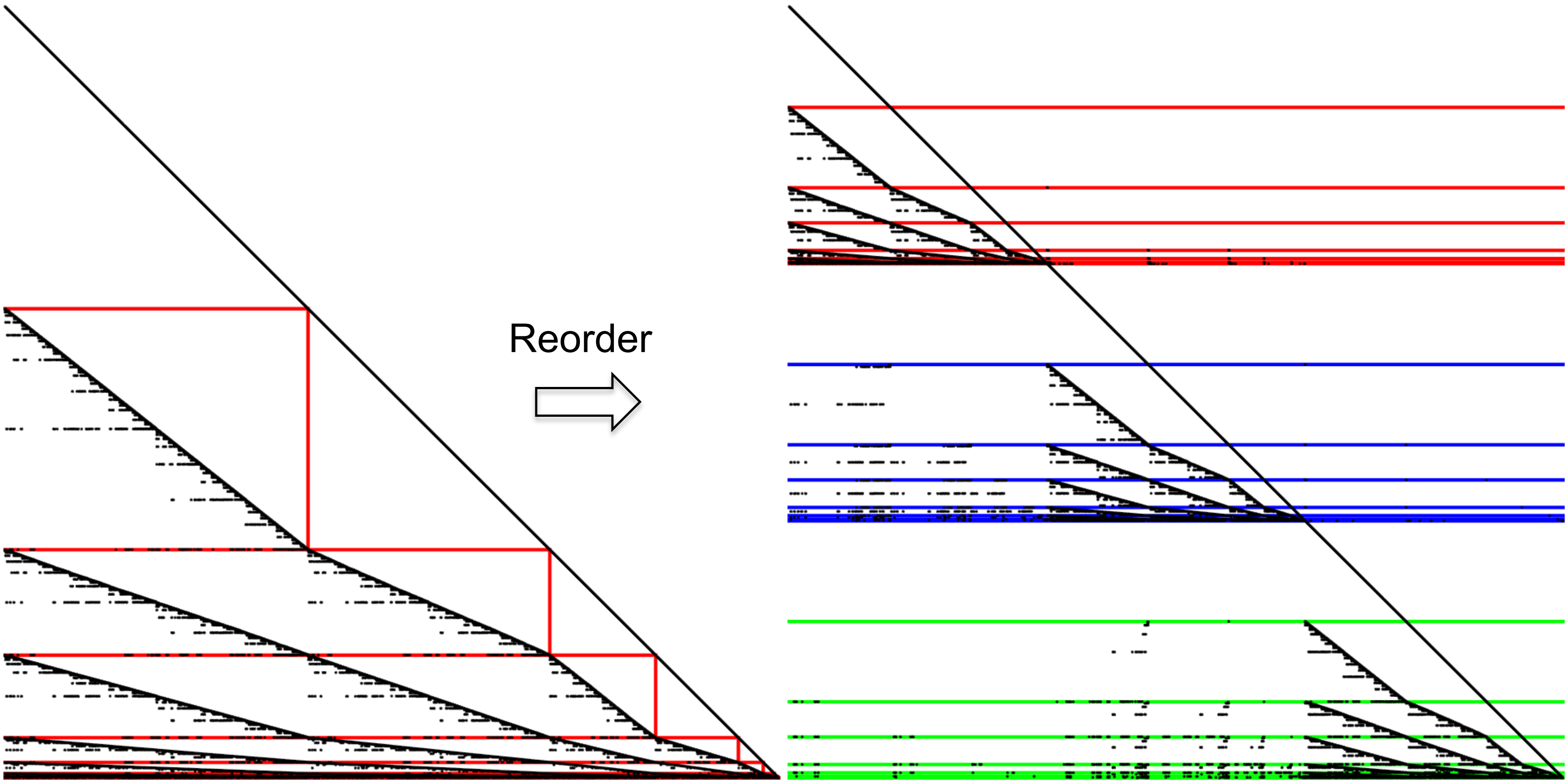
$$N_i \equiv \text{size of level set } i$$

$$C_i \equiv \sum_{j \le i} N_j$$

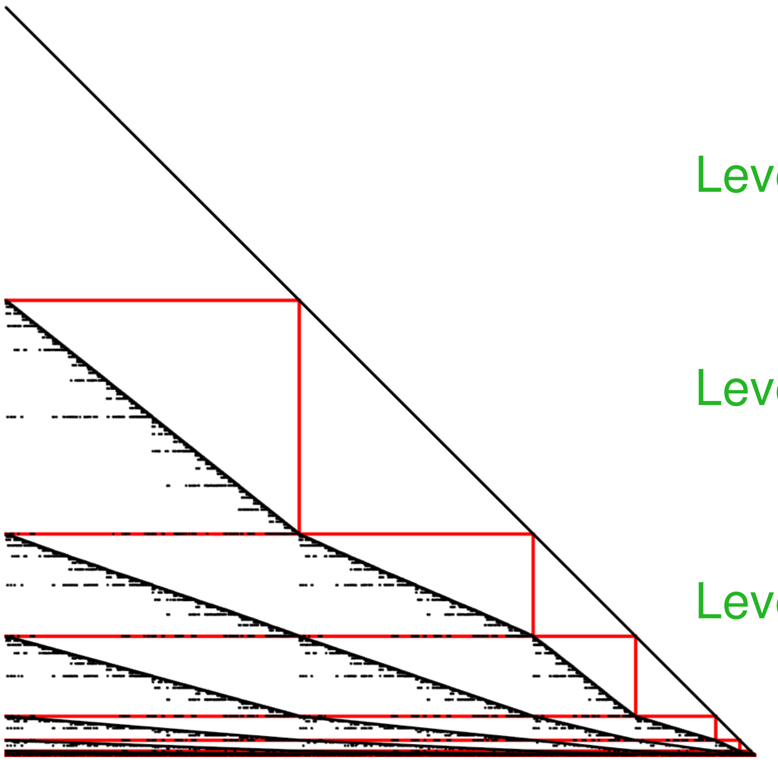
$$C_i^{\text{bad}} \equiv \sum_{j \le i \,\cap\, N_j < n_{\text{good}}} N_j$$

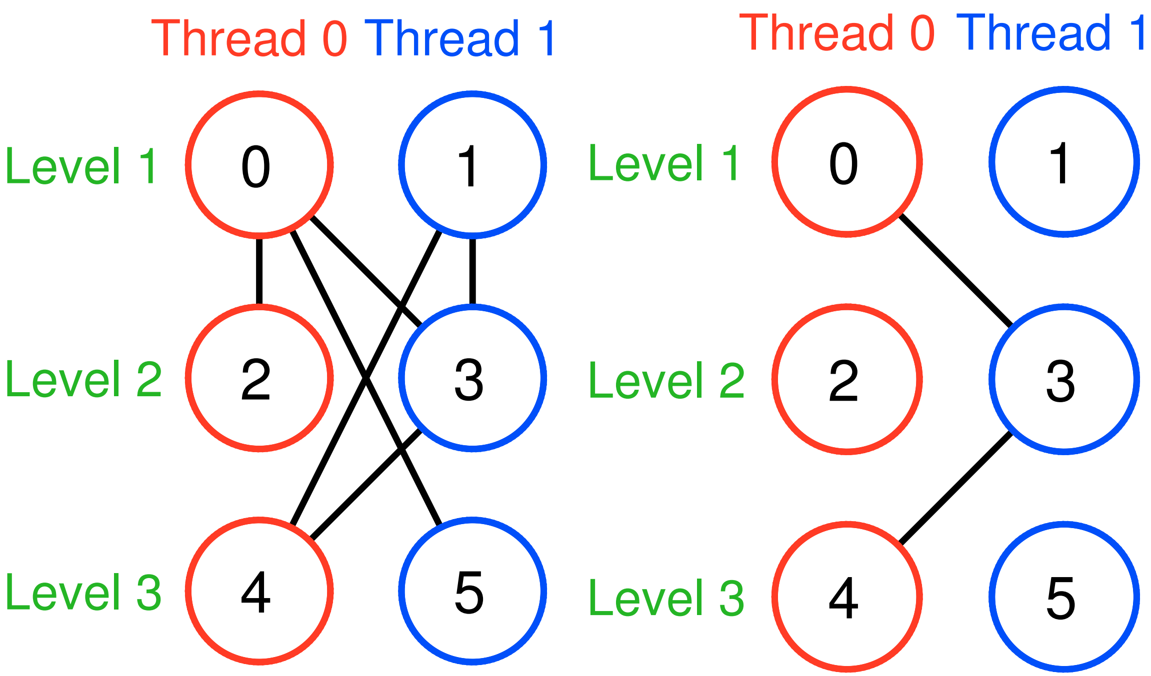
$$k \equiv \arg\max_k \quad N_k \ge n_{\text{good}} \quad \cap \quad C_k^{\text{bad}} \le f_{\text{bad}} C_k$$

Algorithms: Level Scheduling



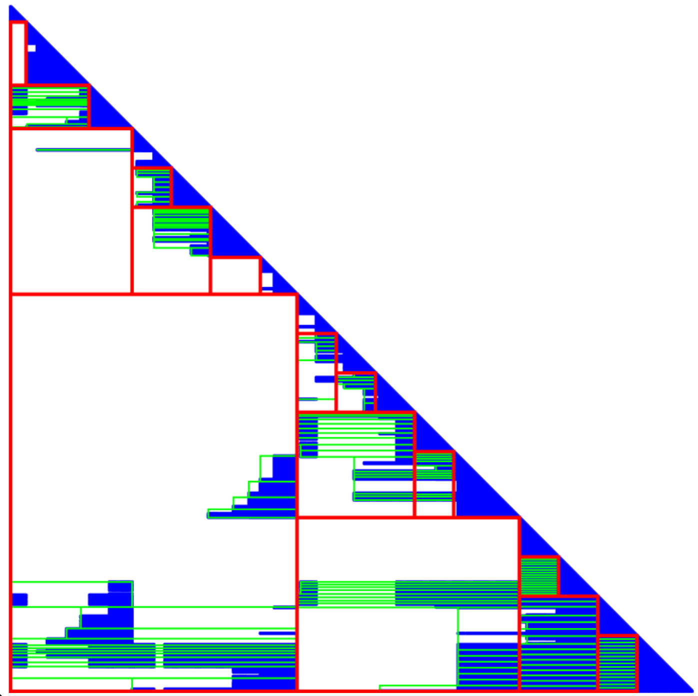
Algorithms: Pruned Point-to-Point

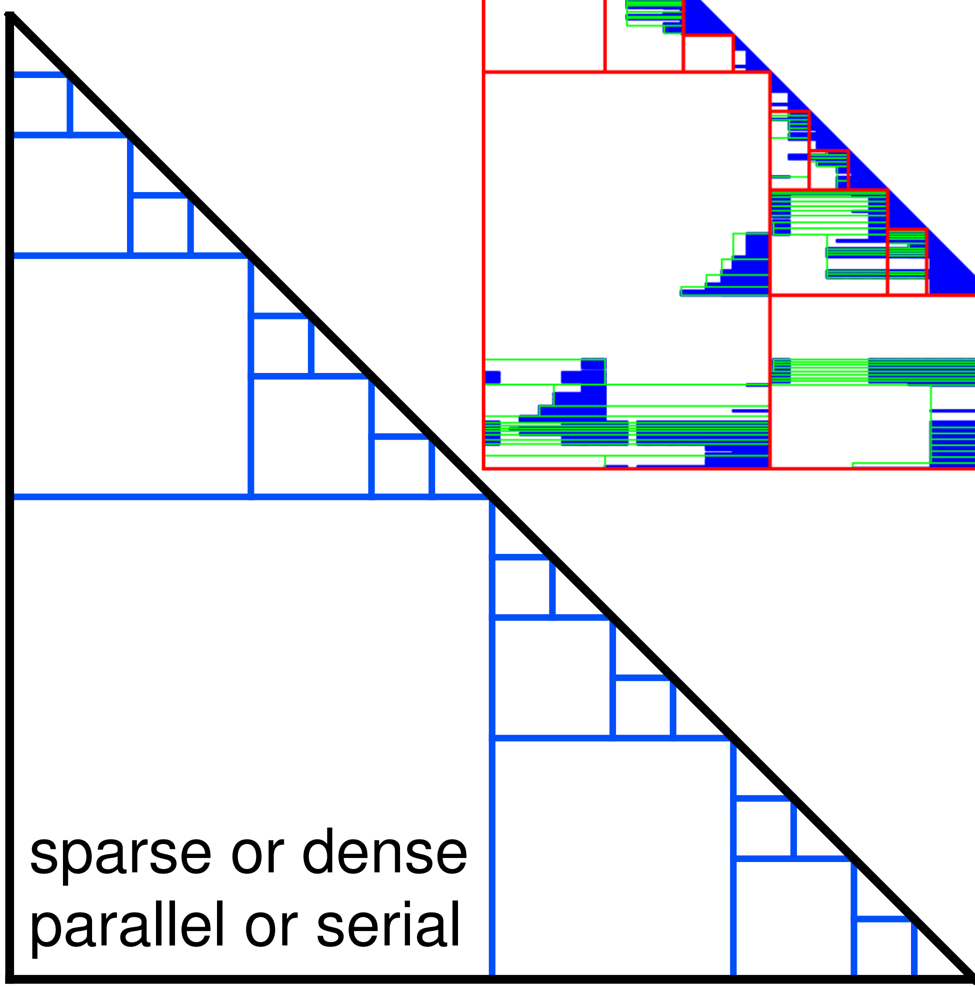


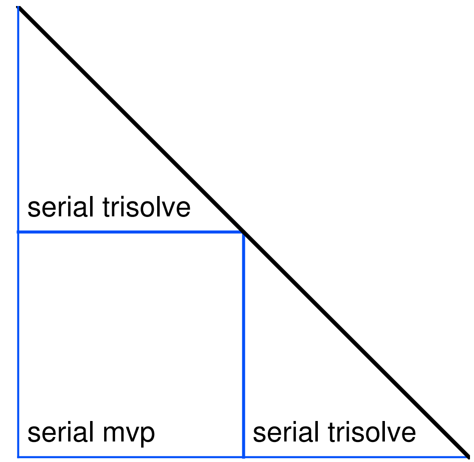


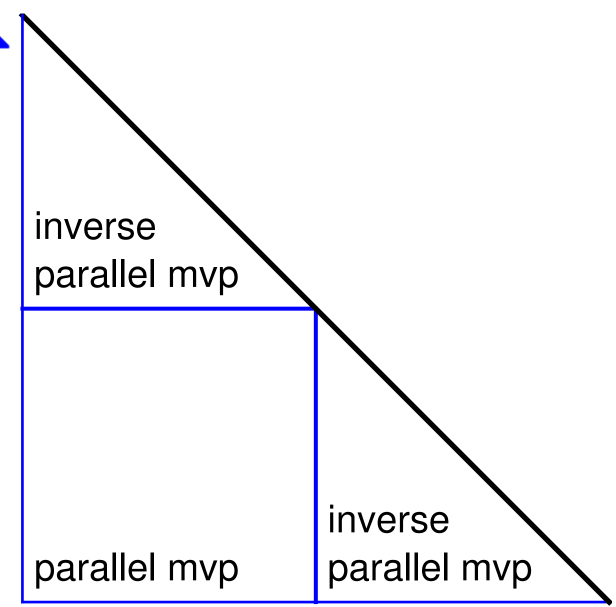
Park, J., M. Smelyanskiy, N. Sundaram, and P. Dubey., "Sparsifying synchronization for high-performance shared-memory sparse triangular solver." In *Supercomputing, pp.* 124-140. Springer International Publishing, 2014.

Algorithms: Recursive Blocking



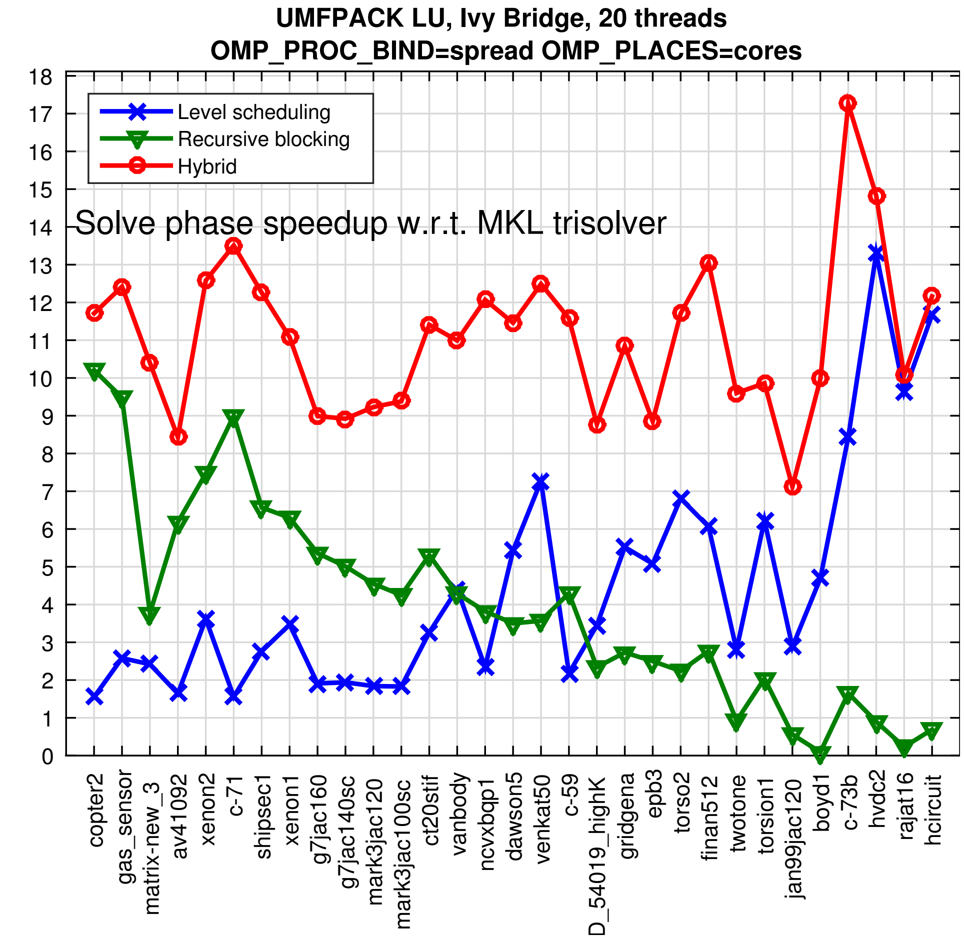


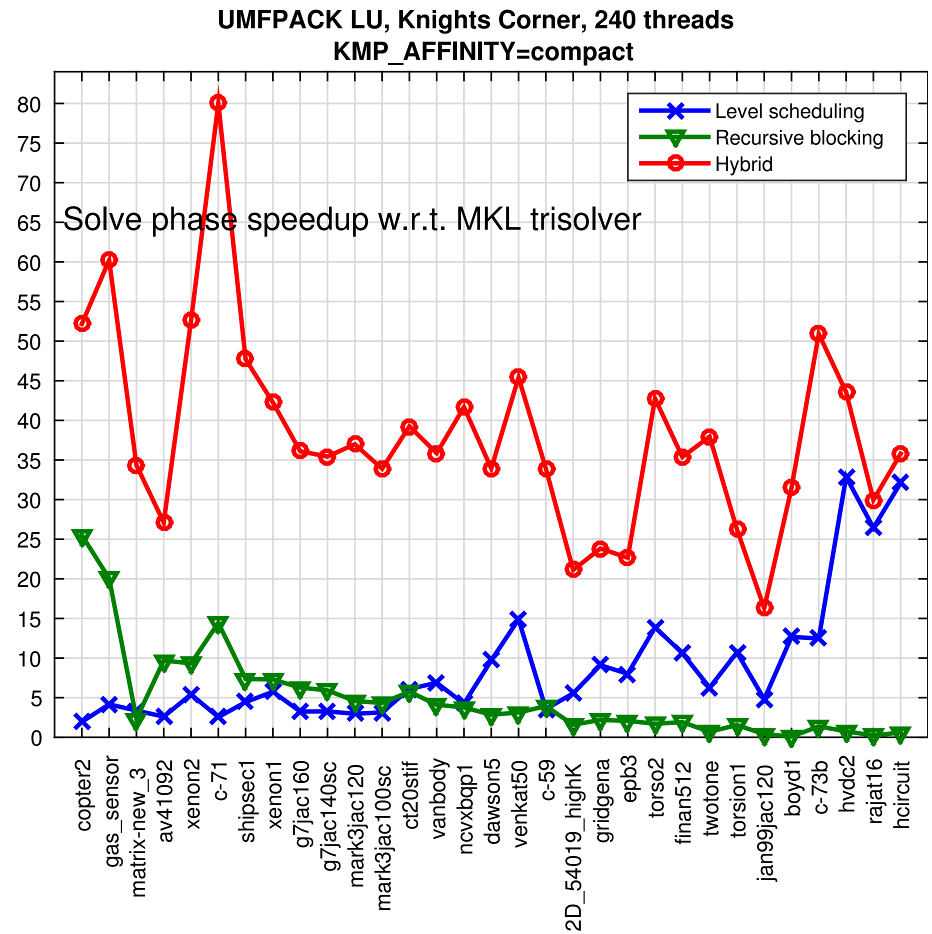


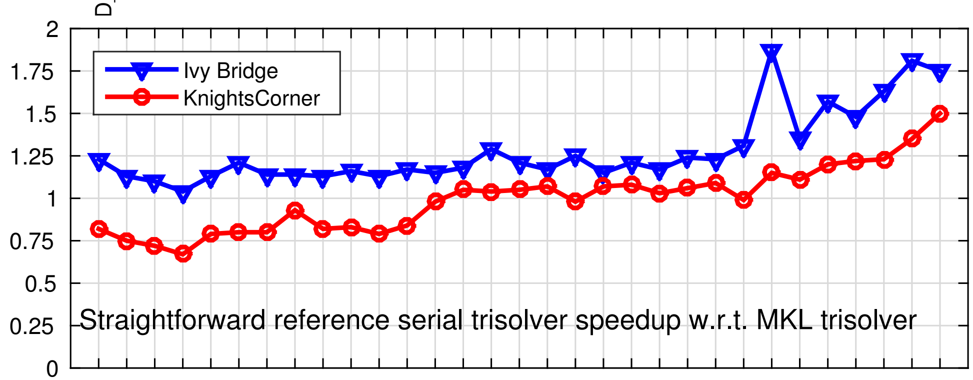


Results: UMFPACK LU on IB and KNC

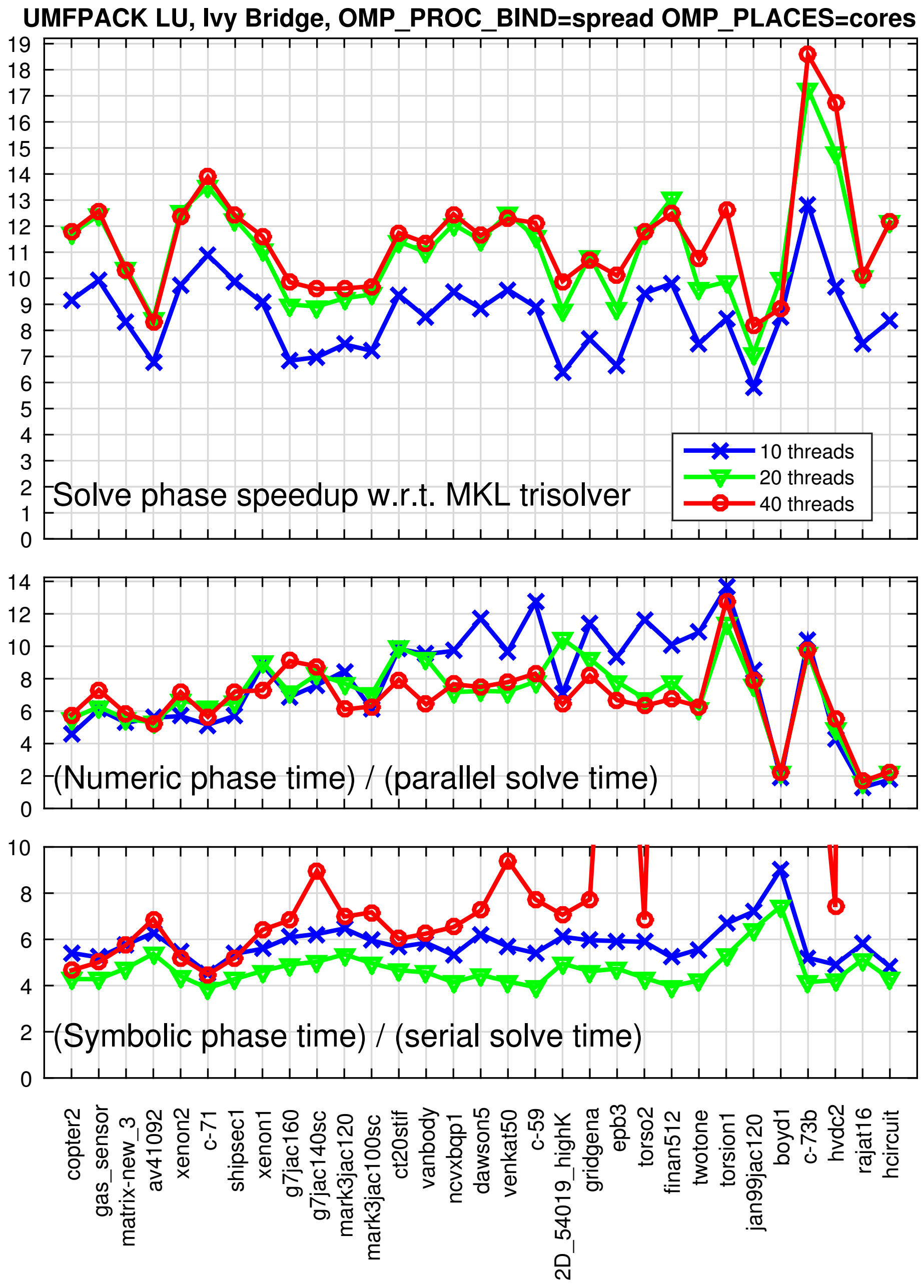
Sandia National Laboratories

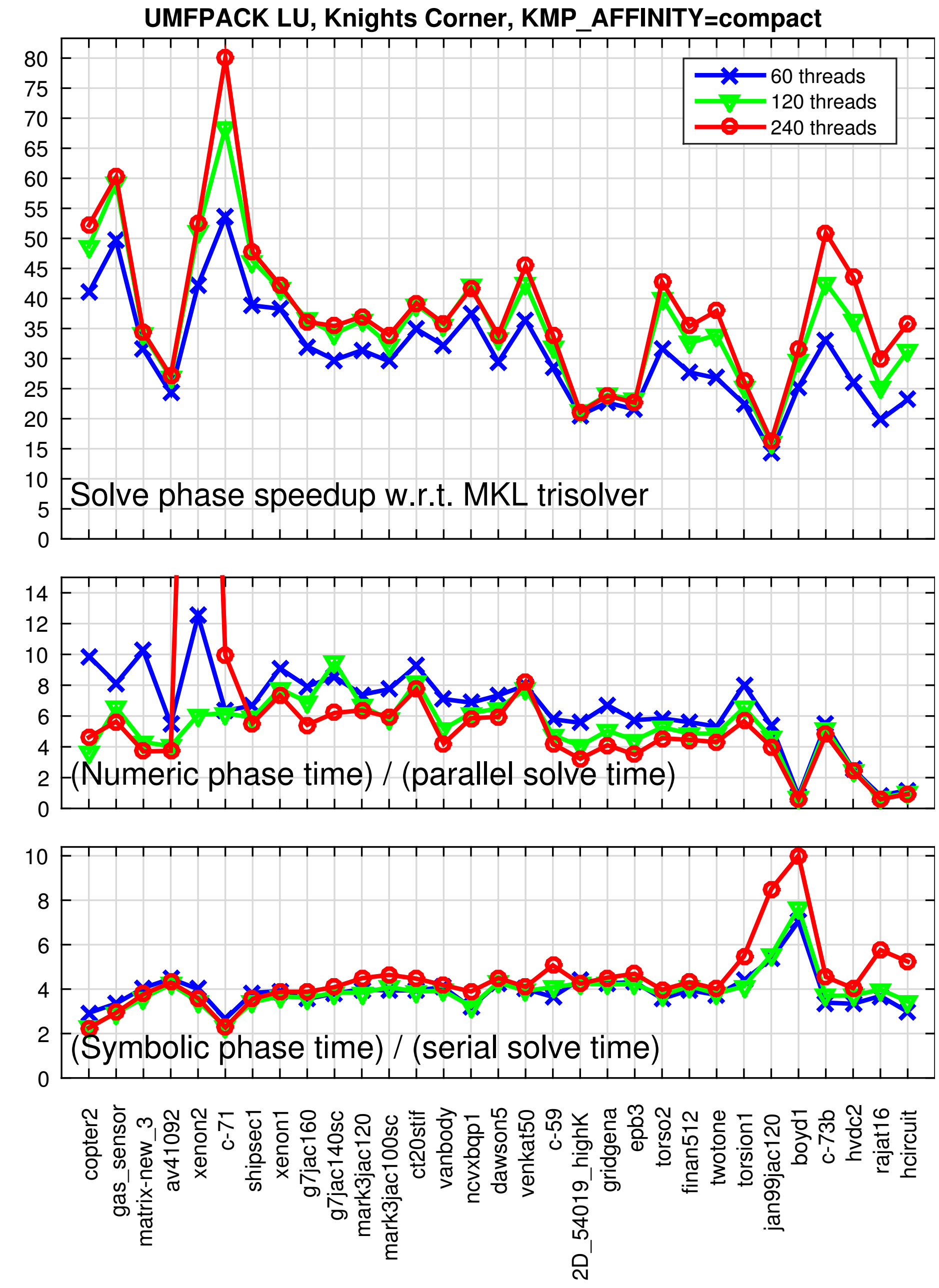


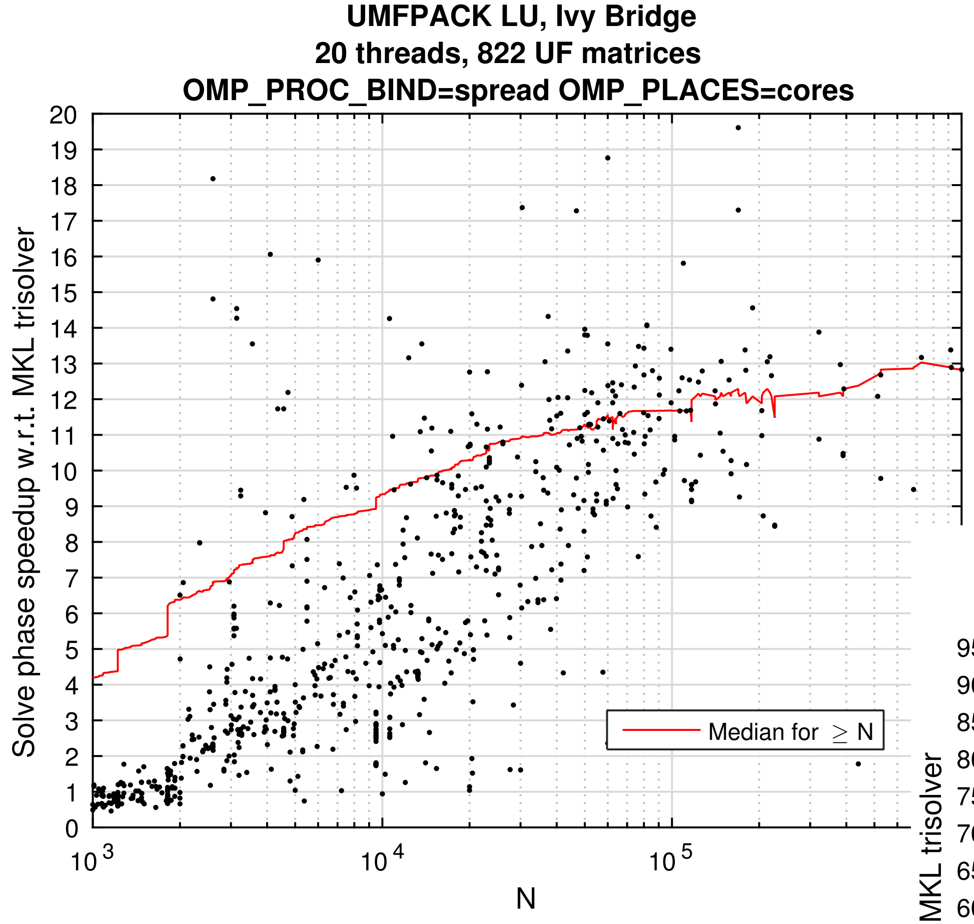


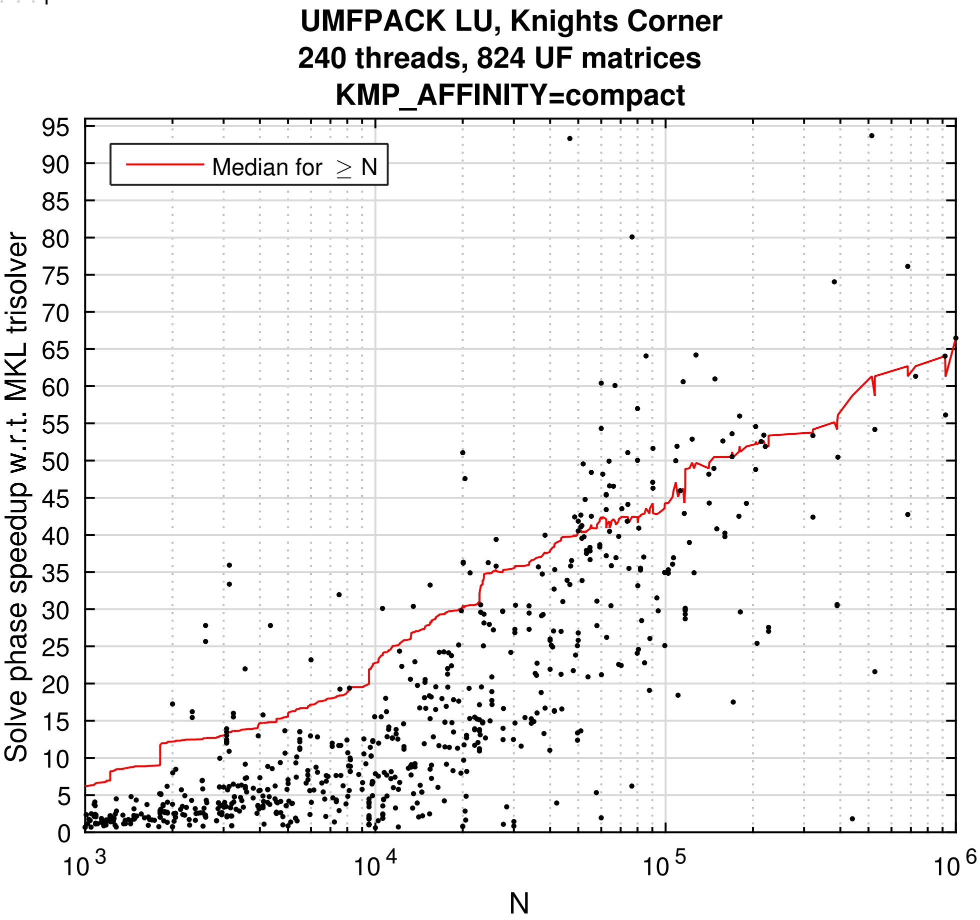


Results: UMFPACK LU on IB and KNC






UMFPACK LU, Ivy Bridge
20 threads, 822 UF matrices
OMP_PROC_BIND=spread OMP_PLACES=cores
Solve phase speedup w.r.t. MKL trisolver
N
Median for ≥ N


UMFPACK LU, Knights Corner
240 threads, 824 UF matrices
KMP_AFFINITY=compact
Solve phase speedup w.r.t. MKL trisolver
N
Median for ≥ N

Future Work

- Point-to-point level scheduling
  - Group rows into tasks to minimize #dependencies
  - Size tasks to reflect level of synchronization
- Hybrid
  - Switching method(s)
  - Does not have to be 3 blocks; alternate
- HTS
  - Improve formatting of recursively blocked part to take further advantage of dense sub-blocks
- Direct sparse methods on GPU?