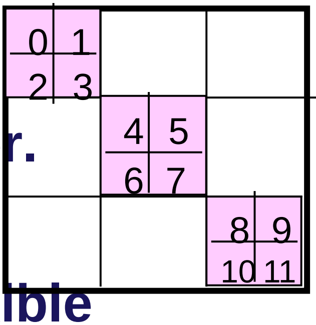
Sparse solvers & preconditioners

Sherry Li, LBNL

- SuperLU direct solver used in M3D-C1 and NIMROD for many years. In 3D, used as Block-Jacobi preconditioner.



- STRUMPACK also based on factorization, but more flexible
  - Direct solver without approximation
  - “Inexact” direct solver with small low-rank compression
    - One factorization, one triangular solve.
    - May ask for a few steps of iterative refinement to get higher accuracy.
  - Preconditioner with large compression
    - Can use PETSc’s GMRES (or any iterative solver), specify STRUMPACK as preconditioner.
- May be worthwhile trying STRUMPACK preconditioner for the entire 3D problem, not just block diagonals.

SuperLU factorization optimization on Cori KNL

Work with Sam Williams, Jack Deslippe, Steve Leak, Thanh Phung (Intel)
(New release SuperLU_DIST version 5.2.0)

- **Overall, factorization is up to 80% speedup on single node.**
  - **Jin Chen’s experiments at Intel Dungeon: M3D-C1 10% speedup.**
  - **Total time in SuperLU 50-80% : LU 43%, Triangular solve 57%**

- Replace small independent single-threaded MKL GEMMs by large multithreaded MKL GEMMs: **15-20% faster.**
- Use new OpenMP features: **10-15% faster.**
  - “task parallel” to reduce load imbalance
  - “nested parallel for” to increase parallelism
- Vectorizing Gather/Scatter: **10-20% faster.**
  - **Hardware support: Load Vector Indexed / Store Vector Indexed**

```
#pragma omp simd     // vectorized Scatter
    for (i = 0; i < b; ++i) {
            nzval[ indirect2[i] ] = nzval[ indirect[i] ] - tempv[i];
    }
```

- Several techniques to reduce cache misses.

Future work – next 0.5 year

SuperLU:

1. **Improve triangular solve.**

   As preconditioner, factorization needs only once, but each iterative step needs a triangular solve. Several techniques will be explored: asynchronous tree broadcast, selective inversion.

2. **Improve factorization with 3D algorithm: replicate data structure, while reducing communication.**
3. **Suggestion for M3D-C1 configuration: use more than one 2D plane as diagonal block.**

STRUMPACK:

- **More detailed performance profiling to identify bottlenecks, understand various solver/preconditioner configurations.**

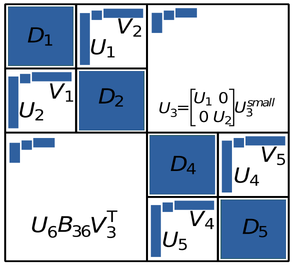
Optimization: with Sam and postdoc

- **Autotuning of the solvers parameter space.**

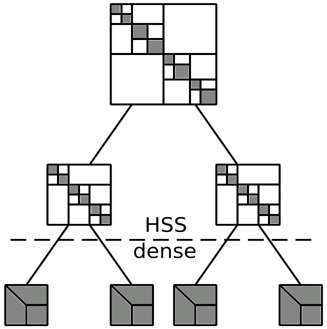
“Inexact” direct solver – STRUMPACK

- Algebraic generalization to FMM, independent of Green’s function.
  - Apply low-rank compression to off-diagonal blocks (“far-field interaction”), use hierarchical matrix algebra to reduce asymptotic complexity.
- Provides flexibility with inexact factorization via tolerance-controlled compression (save FLOPS and memory)

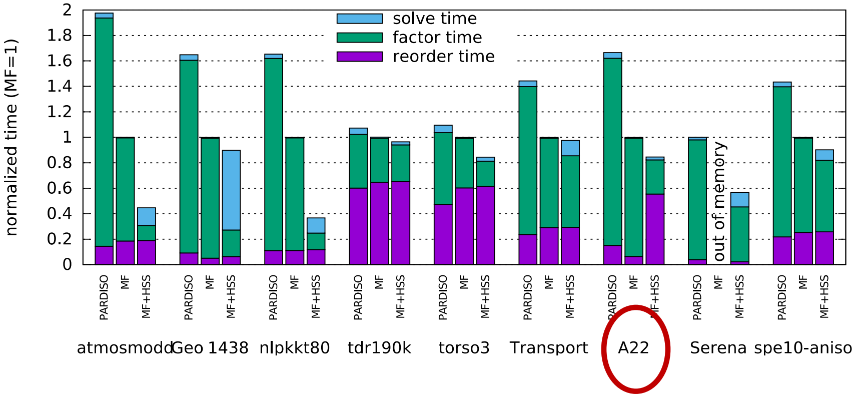
dense



Sparse embedding in MF solver



OpenMP task parallelism, 12-core Intel Ivy Bridge



Status:

- 5x faster than dense LU (ACM TOMS 2016)
- 7x faster than traditional sparse MF solver (SIAM SISC 2016)

Future tasks:

1. Strumpack already in PETSc, will incorporate in Trillinos soon.
2. Tested Jin's recent matrices, found some inefficiency in the ordering phase, we fixed it.
3. Code is very new, less mature than SuperLU, will do lots of algorithm optimization, architecture-otiended code optimization …