Exceptional service in the national interest

SAND2016-2539C

Sandia National Laboratories

Parallel Preconditioners and Solvers for Modern Architectures

Erik Boman

Mehmet Deveci, Siva Rajamanickam

Sandia National Labs

March 21, 2016

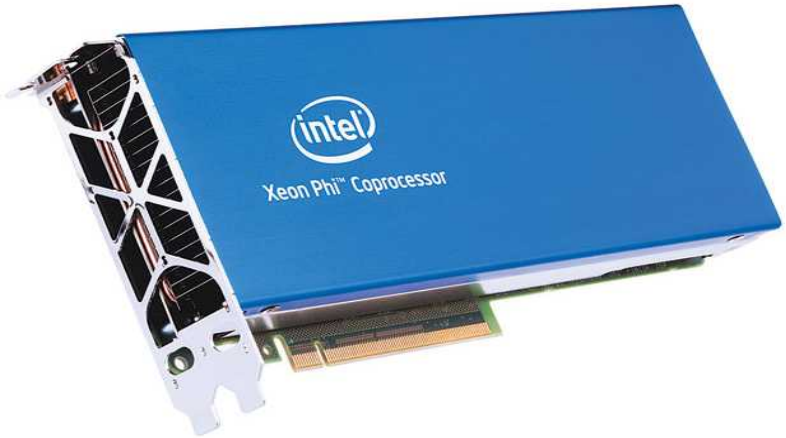
U.S. DEPARTMENT OF ENERGY | NNSA National Nuclear Security Administration | CCR Center for Computing Research

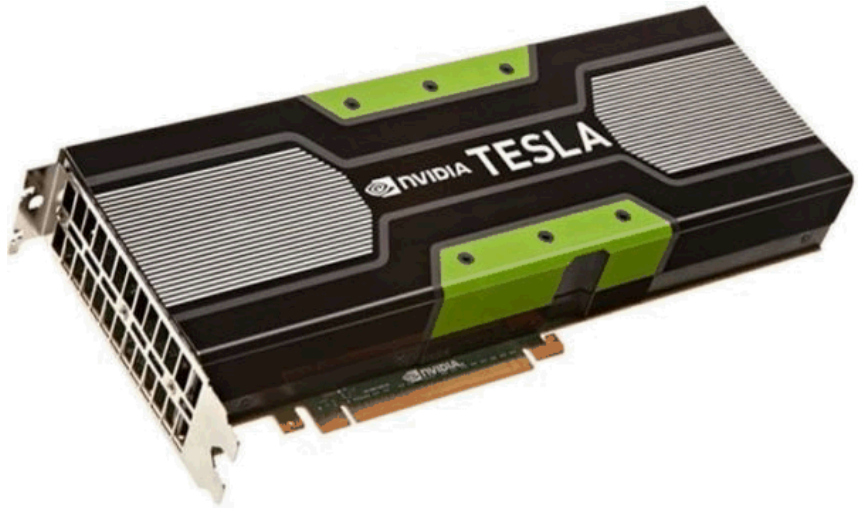
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Extreme-Scale Computing Challenges

Sandia National Laboratories

- Current parallel solvers on distributed memory systems are quite successful in many areas
  - Challenge: Reduce global communication (not in this talk)
- Architecture trend: Node count does not grow
- Main challenge is greater concurrency on the node
  - Accelerators: GPU, MIC
  - Concurrency on node is now O(100-1000)

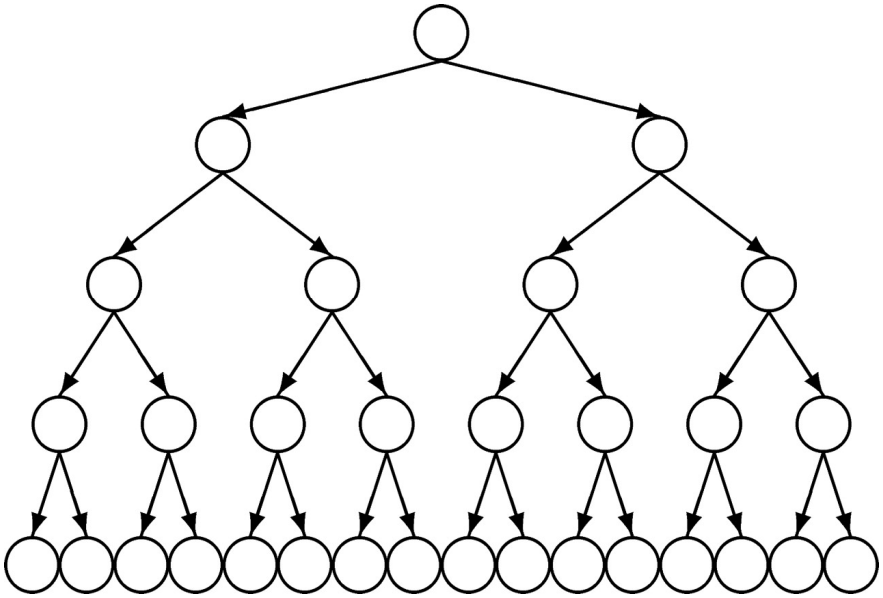




Common Types of Parallelism

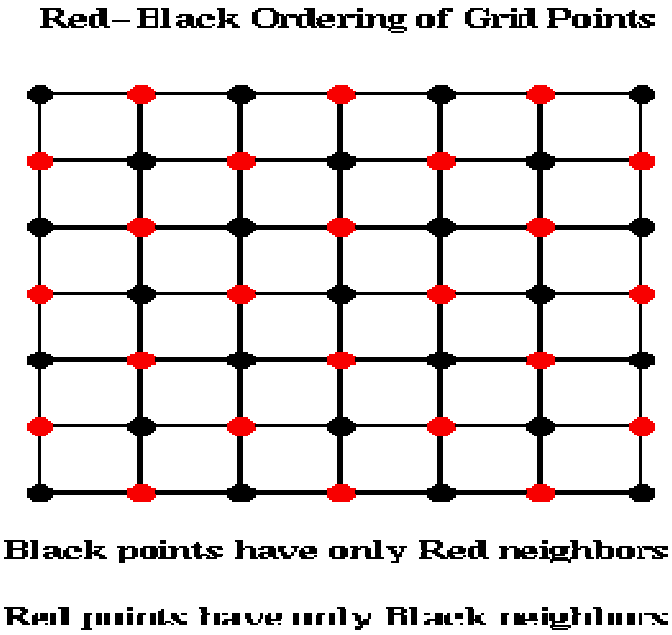
Coarse-grained:

- Divide-and-conquer
- Nested Dissection



Fine-grained:

- Independent Sets
- Graph Coloring



Concurrency vs. convergence

Sandia National Laboratories

Trade-off: Reordering for greater concurrency often gives slower convergence.

Example: Multicolor Gauss-Seidel.

- Update all vertices within each color class in parallel.
- #synchronization points = #colors
- Slow convergence, use as preconditioner in Krylov.

Matrix	Method	Natural	Multicolor
Wathen120	CG/SGS	20 iter.	23 iter.
Bcsstk18	CG/SGS	323 iter.	322 iter.
Torso2	GMRES/GS	22 iter.	19 iter.
Audikw1	CG/SGS	2512 iter.	2780 iter.

Multicolor Gauss-Seidel

Sandia National Laboratories

CPU:

Compare serial vs. parallel precondioner.

SGS	1 thr.	4 thr.	16 thr.
Serial	2.84	1.31	1.07
Parallel	4.66	1.43	0.73

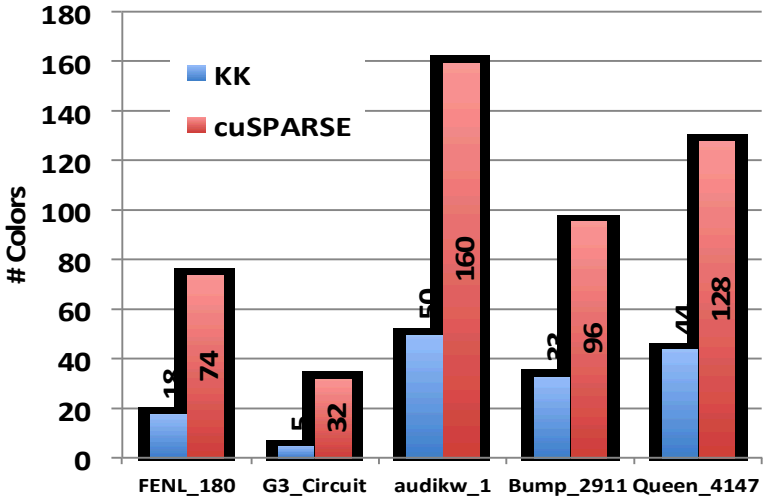
Multithreading pays off after only ~4 threads.

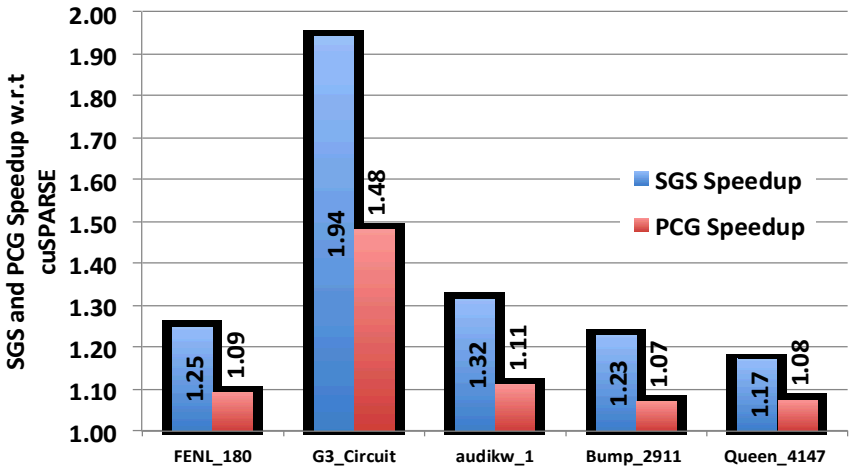
Parallel Graph Coloring

Sandia National Laboratories

- Optimal coloring is NP-hard
- Fast greedy coloring works well in practice
  - But is sequential
- Parallel greedy methods
  - Jones-Plassmann ’90: independent sets
  - Gebremedhin-Manne ‘99: speculative/optimistic method
    - May have conflicts, need several rounds to resolve
- Deveci, B. Devine, Rajamanickam ‘16:
  - Edge-based speculative/optimistic method for manycore
  - Faster and better quality (4X geom. mean) than cuSparse

GPU: Coloring-based Symmetric Gauss-Seidel





- New coloring algorithm and Gauss-Seidel are now available in the KokkosKernels package
- Reduces the overall CG solve time by ~33% compared to cuSparse.
  - Shows that coloring quality matters!

Other Methods: Fine-Grain Parallel ILU

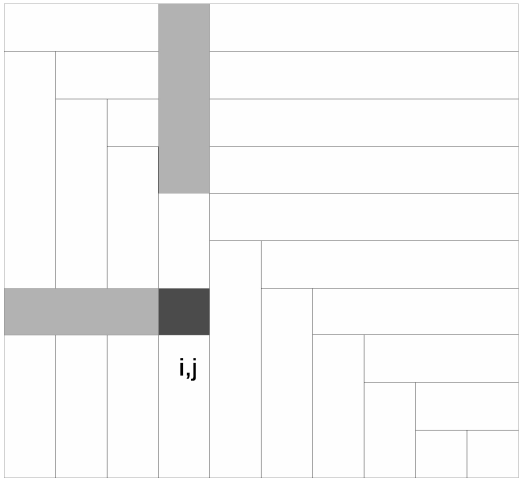
Sandia National Laboratories

Chow & Patel 2015: New way to compute incomplete A~LU.

- Solve larger, bilinear system for each entry in L, U
- "Gauss-Seidel" on factors converges quickly (5-10 sweeps)
  - Asynchronous version works fine, no coloring needed.
  - Highly parallel: Can use one thread per nonzero.
- Triangular solves become the bottleneck, but can solve iteratively.
- Well suited for GPU: Anzt et al. (2015)
- Software implementations in Magma and Trilinos/ShyLU.

Algorithm 1: Fine-Grained Parallel Incomplete Factorization

Set unknowns l_{ij} and u_{ij} to initial values ;
for *sweep* $= 1, 2, \ldots$ *until convergence* **do**
  parallel for $(i, j) \in S$ **do**
    if $i > j$ **then**
      $l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}\right) / u_{jj}$
    else
      $u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$
    end
  end
end



Row Projections: Kaczmarz

Sandia National Laboratories

Row projection methods are very old:

- Kaczmarz (1937)
- Cimmino (1938)

Rediscovered several times, e.g. as ART and CIRT.

Can be used as solver or preconditioner. Several advantages:

- Work on single row at a time, so fine grained.
- No global communication.

Kaczmarz: For each row i

$$x^k = x^k + \lambda \frac{b_i - < a_i, x^k >}{| a_i |} a_i$$

Parallel Kaczmarz

Sandia National Laboratories

Traditional method is sequential, but two options:

1. Multicolored Kaczmarz
   - Update independent rows in parallel.
2. Asynchronous Kaczmarz
   - Use most recent values, changes the algorithm
   - Essentially becomes hybrid Kaczmarz-Cimmino

Randomized Kaczmarz:

- Pick random row to update (no “sweeps”)
- Convergence is linear (“exponential”) [Strohmer & Vershynin]
- Gives hope asynchronous execution will behave similarly.

Software: Kokkos

Sandia National Laboratories

Challenge: How to write portable multi-threaded code for many different platforms?

- CPU, Xeon Phi (MIC), GPU, etc.

Kokkos is a C++ library for performance-portable manycore programming.

- Supports parallel patterns (for, reduce, etc.)
- Optimized layout of multidimensional arrays
- Hierarchical parallelism (leagues, teams, threads)
- Developer productivity: Write once, run anywhere!
- Talk by Christian Trott, Wed. AM

Solvers using Kokkos: Rajamanickam talk, Wed. PM

Future Work: Block Row Projections

Sandia National Laboratories

Many row projection options:

- Kaczmarz, Cimmino, CAV, CARP, …

Block Row Projection Methods are superior:

- Better convergence
- Better memory access?

Challenge: For each block, we need to solve a rectangular system

- Underdetermined system, want min-norm solution
- Options: sparse QR, normal equations, augmented system
- Suitable for 2-level parallelism
  - No sparse QR that works at the “thread team” level
  - ABCD (Zenadi et al.) uses MUMPS on augmented system

Conclusions

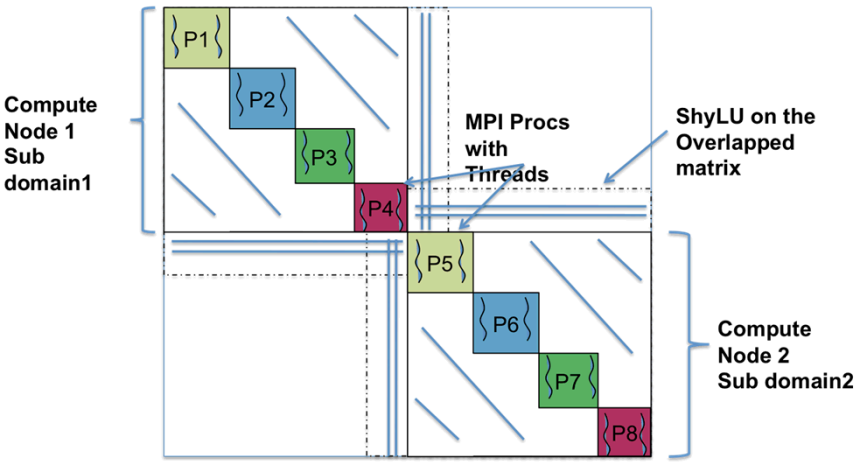
Sandia National Laboratories

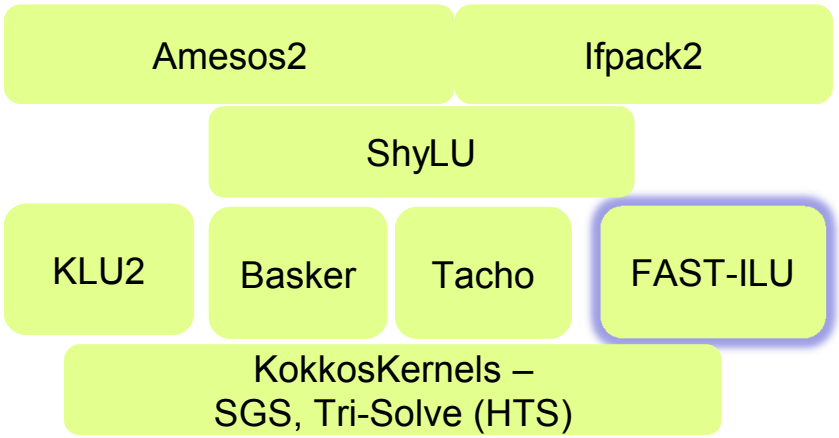
- Highly parallel architectures are disruptive
  - Require us to revisit algorithms.
  - Old ideas may come back to life?
- Coloring is powerful concept
  - Exposes fine-grain parallelism
  - Reordering slows convergence but still a win compared to sequential approach
- On-node performance critical for exascale.
  - Rewrite of solver software is needed.

Backup

Sandia National Laboratories

ShyLU and Subdomain Solvers : Overview





- MPI+X based subdomain solvers
  - Decouple the notion of one MPI rank as one subdomain: Subdomains can span multiple MPI ranks each with its own subdomain solver using X or MPI+X
  - Epetra based solver, Tpetra interface still being developed
    - Trilinos Solver Factory a big step forward to get this done (*Thanks to M. Hoemmen*)
- **Subpackages of ShyLU**: Multiple Kokkos-based options for on-node parallelism
  - **Basker :** LU or ILU (t) factorization (J. Booth)
  - **Tacho:** Incomplete Cholesky - IC (k) (*See Kyungjoo's talk*)
  - **Fast-ILU:** Fast-ILU factorization for GPUs (A. Patel)
- **KokkosKernels:** Coloring based Gauss-Seidel (M. Deveci), Triangular Solves (*See Andrew's talk for HTS*)
- Experimental code base under active development. Jointly funded by ASC, ATDM, FASTMath, LDRD.