Parallel Circuit Simulation: Challenges and Opportunities

Bruce Hendrickson

Rob Hoekstra

Sandia National Labs

Why Parallelize Circuit Simulation?

- **Solve bigger problems**
  - » Principle argument for other applications
    - – Higher resolution in solution
  - » Less compelling for some aspects of circuit modeling
    - – But allows for more integrated analysis
- **Solve coupled physics problems**
  - – E.g. radiation and thermal effects
  - – Can often be done as uncoupled computation
- **Faster turn-around time for designers**

Parallel Challenges

- **Simultaneously balance**
  - » **Problem construction (matrix fill) and**
  - » **Problem solution**

- **Parallel Direct Solvers**
  - » **Key advantage is robustness**
  - » **limited parallel and algorithmic scalability**
  - » **Load balancing**
  - » **Memory requirements**
  - » **Immature software**

Parallel Iterative Solvers for CFD

- **Parallel Iterative Solvers Work Well**
  - » Robustness is an issue, but usually OK
  - » Easier to parallelize than direct methods
  - » Excellent parallel scalability is possible
  - » Graph partitioning balances the load
  - » Effective, parallel preconditioners are available

Circuits are Different!

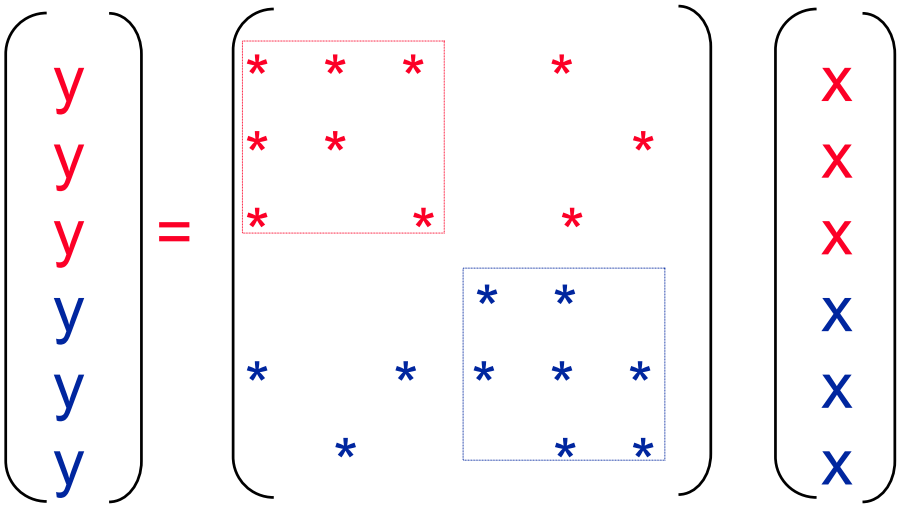
- Toto, I don't think we're in PDE-land any more!
  - » No locality coming from discretized space
  - » Very heterogeneous matrix structures
    - – Challenge for ordering and load balancing
    - – Effects both direct and iterative solvers
  - » Very poorly conditioned matrices.
  - » No underlying geometry for multigrid

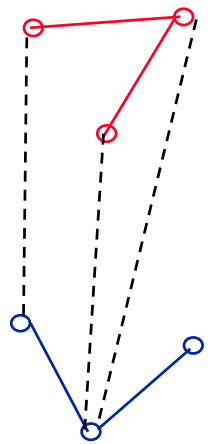
  - » Robust preconditioning is hard
  - » Load balancing is hard
    - – For both matvec and preconditioner

Parallel Iterative Solvers for Circuits

- **Parallel Iterative Solvers Sometimes Work OK**
  - » **Robustness is a huge concern**
  - » **Easier to parallelize than direct methods**
  - » **Mediocre parallel scalability**
  - » **Graph partitioning does not balance the load**
  - » **Preconditioners**
    - – **Sometimes ineffective**
    - – **Good ones can lead to significant load imbalance**

Graph Partitioning & Symmetric Matrix-Vector Multiplication



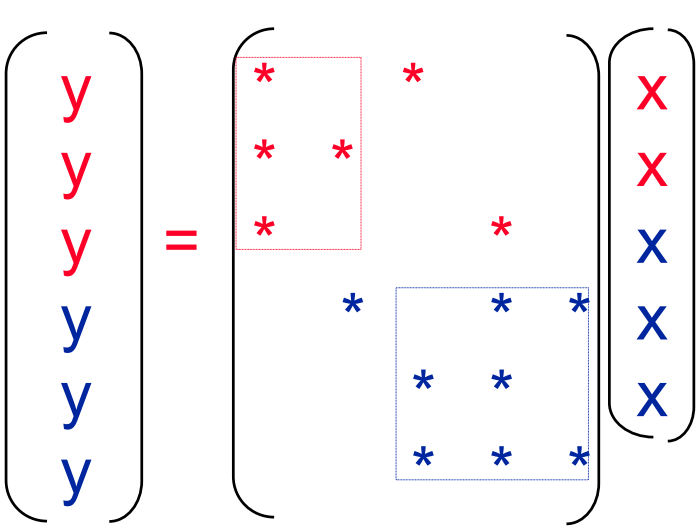


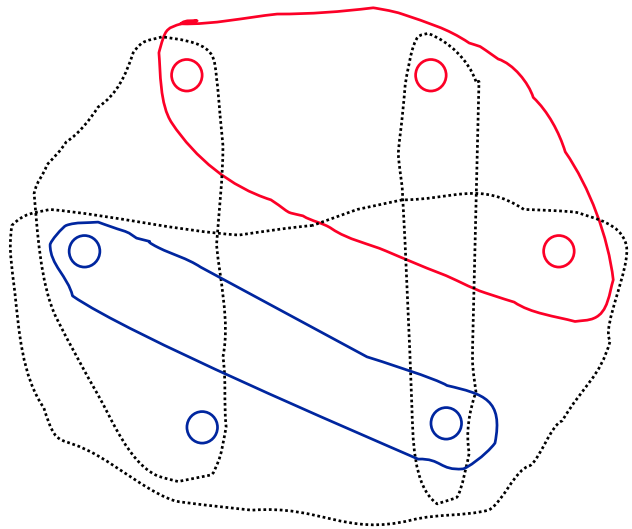
- **Processor owns rows of x, y and matrix**
  - » **Can compute contributions from diagonal blocks**
  - » **Communicate x values for off-diagonal blocks**
- **Twice Edge-cuts = 6, but communication = 5!**

Why Does Graph Partitioning Work?

- **Vast majority of applications are computational meshes.**
  - » **Matrices are generally structurally symmetric**
  - » **Geometric properties ensure that good partitions exist.**
    - – **Communication/Computation = $n^{1/2}$ in 2D, $n^{2/3}$ in 3D.**
    - – **Runtime is dominated by computation.**
  - » **Vertices have bounded numbers of neighbors.**
    - – **Error in edge cut metric is bounded.**
  - » **Homogeneity ensures all processors have similar subdomains.**
    - – **No processor has dramatically more communication.**
- **None of these properties holds true for circuits!**

Alternative Model: Hypergraphs





- Row = vertex
- Column = hyperedge (all nonzeros in column)
- Can encode non-symmetric matrices
  - » (Çatalyürek/Aykanat'96)

Hypergraph Model, Continued

- Vertex is computation (weighted by cost)
- Hyperedge connects producer & consumers of datum (weighted by #bits in datum)
- Cut hyperedge leads to communication.
- Total weight of cut hyperedges = volume of communication!
- Partition to minimize cut hyperedges.

Does it Matter?

- Big win for unstructured problems!
- On set of linear programming problems…
  - » Communication volume reduced by > 30%
  - » (Çatalyürek/Aykanat'96)
- Only moderate impact on easier problems
  - » E.g. On problems from meshes only 5-10% better.

Hypergraph Partitioning Tools

- Developed for/by circuit placement folks
  - » Often proprietary
  - » hMETIS is gold standard – no source code
- No parallel code
  - » We are adding capability to Zoltan
    - – Sandia's dynamic load balancing tool

Load Balancing for Complex Preconditioners

- **Simple preconditioners balance easily**
  - » **E.g Jacobi (diagonal scaling)**
- **But complex preconditioners don't**
  - » **Overlapped Schwarz domain decomposition**
    - – **Overlapped domains may differ in size**
      - **Particularly for circuits!**
  - » **Domain-by-domain incomplete factorizations**
    - – **Different domains may have differing fill**
      - **Particularly for circuits!**

Improving Imbalanced Preconditioners

- Key source of poor scalability for circuits
- One idea:
  - Detect imbalance
  - Migrate elements off of overloaded processors
  - Iterate as needed
  - (For first effort, see Pinar/H. 2001)
- New ideas are needed

More Information

- **Contact info:**
  - bah@cs.sandia.gov
  - http://www.cs.sandia.gov/~bahendr
  - http://www.cs.sandia.gov/Zoltan

- **Collaborators:**
  - Karen Devine Tammy Kolda Ali Pinar

- **Work supported by DOE's MICS program**

- Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the US DOE under contract DE-AC-94AL85000