Preconditioning Linear Systems Arising from Graph Laplacians of Complex Networks

Kevin Deweese¹  Erik Boman²

¹Department of Computer Science
University of California, Santa Barbara

²Scalable Algorithms Department
Sandia National Laboratories

Copper Mountain Conference, 2014
Our focus: Solve the system of equations $Lx = b$ where $L$ is a combinatorial Laplacian matrix.
Best Techniques

Direct Solver: Sparse Cholesky
Iterative Solver: Conjugate Gradient
Use a preconditioner to accelerate convergence. $B^{-1}Ax = B^{-1}b$
A key challenge is to find a good preconditioner.
Traditional preconditioners include Jacobi, Incomplete Cholesky, and Multigrid methods.
Are different techniques needed for network science problems?
Combinatorial Preconditioning

- Solving linear equations with SDD matrices by constructing good preconditioners (Vaidya, 1991)
- Support theory for preconditioning (Boman and Hendrickson, 2003)
- Vaidya’s preconditioners: Implementation and experimental Study (Chen and Toledo, 2003)
- Nearly linear time algorithms for preconditioning and solving SDD linear systems (Spielman and Teng, 2006)
- A nearly $m \log n$ time solver for SDD linear systems (Koutis, Miller, and Peng, 2011)
- mostly theory, few experiments
Support Graphs

Use a graph approximation as a preconditioner.
- Complete factorization of an incomplete matrix
Specifically we use max-weight spanning trees.
Experimental Setup

- We use the Trilinos software framework (Heroux et al. 2005) for our experiments.
- We implemented MST as an Ifpack2 preconditioner.
- We use
  - Ifpack2’s Jacobi and RILUK preconditioners.
  - Tpetra for matrix/vector operations.
  - Belos for iterative methods (PCG).
  - Zoltan for graph partitioning.
Flickr graph, 800k Rows (Vertices), 13M NNZ (2xEdges)

<table>
<thead>
<tr>
<th>Preconditioner</th>
<th>Iterations</th>
<th>Solve Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>2689</td>
<td>266.3</td>
</tr>
<tr>
<td>Jacobi</td>
<td>76</td>
<td>7.841</td>
</tr>
<tr>
<td>ILUT</td>
<td>30</td>
<td>8.331</td>
</tr>
<tr>
<td>MST</td>
<td>42</td>
<td>6.247</td>
</tr>
</tbody>
</table>
Parallel Setup

- We use the Additive Schwarz domain decomposition scheme to split the problem over subdomains and do inexact subdomain solves.
  - No overlap (Block Jacobi)
- We use Zoltan’s interface to the Scotch graph partitioning library to
  - Load balance.
  - Reduce communication cost.
  - Improve subdomain preconditioner quality.
as-Skitter, 1.7M Rows (Vertices), 22M NNZ (2xEdges), 24 cores

<table>
<thead>
<tr>
<th>Preconditioner</th>
<th>Iterations No Partitioning</th>
<th>Iterations Scotch</th>
<th>Solve Time (s) No Partitioning</th>
<th>Solve Time (s) Scotch</th>
</tr>
</thead>
<tbody>
<tr>
<td>MST</td>
<td>128</td>
<td>103</td>
<td>9.142</td>
<td>5.51</td>
</tr>
<tr>
<td>Jacobi</td>
<td>130</td>
<td>116</td>
<td>7.396</td>
<td>3.645</td>
</tr>
<tr>
<td>RILUK</td>
<td>124</td>
<td>72</td>
<td>13.31</td>
<td>4.497</td>
</tr>
</tbody>
</table>
Experiments

- **Strong Scaling** (24, 48, 96 cores of hopper)
  - as-Skitter (11M edges)
  - as-Skitter with random edge weights (0-100)

- **Weak Scaling** (12, 24, 48 cores of hopper)
  - Used the BTER graph generator (Kolda et al. 2014) to generate graphs of 250k, 500k, and 1M nodes.
  - BTER (average degree 20, global clustering coefficient .3)
  - BTER with random edge weights (0-100)
  - 2.6M, 5M, 10M edges respectively
as-Skitter, 1.7M Rows (Vertices), 22M NNZ (2xEdges)

Strong Scaling

Iterations

Solve Time (s)

Processors

Jacobi

RILUK

MST

Kevin Deweese, Erik Boman (UCSB,SNL)

Preconditioning Graph Laplacians
Strong Scaling

as-Skitter with edge weights,
1.7M Rows (Vertices), 22M NNZ (2xEdges)

![Graph showing iterations vs. processors](image1)

![Graph showing solve time vs. processors](image2)

Kevin Deweese, Erik Boman (UCSB,SNL)
Preconditioning Graph Laplacians
CM 2014 13 / 17
Weak Scaling

BTER

Processors

Iterations

10^{1.2} 10^{1.4} 10^{1.6}

10^{1.6}

10^{1.4}

10^{1.2}

Iter
dations

Solve Time (s)

10^{-0.4}

10^{-0.2}

10^{0}

10^{1.2} 10^{1.4} 10^{1.6}

Processors

Jacobi

RILUK

MST

Kevin Deweese, Erik Boman (UCSB,SNL)

Preconditioning Graph Laplacians

CM 2014
Weak Scaling

BTER with edge weights

- Iterations vs. Processors
- Solve Time (s) vs. Processors

- Jacobi
- RILUK
- MST

Kevin Deweese, Erik Boman (UCSB, SNL)
Support graph preconditioners with domain decomposition scale well and are a promising alternative for solving linear problems of Laplacians.

Other support graphs should be examined. We plan to target low-stretch trees next.

Edge weights matter. Can we generate graphs with more realistic edge weights?
Preconditioning Graph Laplacians