Scalable Algebraic Multigrid on Blue Gene/L

Hans De Sterck, Jeff Butler
Department of Applied Mathematics
University of Waterloo
Ulrike Meier Yang
Center for Applied Scientific Computing
Lawrence Livermore National Lab, USA

CAIMS Annual Meeting,
Toronto, 19 June 2006
Outline

1. introduction: algebraic multigrid (AMG)
2. classical coarsening may lead to complexity growth
3. Parallel Modified Independent Set (PMIS) coarsening
4. improving interpolation
5. scaling results on Blue Gene/L
(1) Introduction: Algebraic Multigrid (AMG)

- \( \text{solve } A u = f \)
- \( A \) from 3D PDE – sparse!
- large problems \((10^9 \text{ dof})\) - parallel
- unstructured grid problems
algebraic multigrid (AMG)

- multi-level
- iterative
- algebraic: suitable for unstructured grids!
AMG building blocks

Setup Phase:

- Select coarse “grids”
- Define interpolation, \( P^{(m)}, \ m = 1,2,... \)
- Define restriction and coarse-grid operators

\[
R^{(m)} = P^{(m)T} \quad A^{(m+1)} = P^{(m)T} A^{(m)} P^{(m)}
\]

Solve Phase

Relax \( A^{(m)} u^{m} = f^{m} \)

Compute \( r^{m} = f^{m} - A^{(m)} u^{m} \)

Restrict \( r^{m+1} = R^{(m)} r^{m} \)

Solve \( A^{(m+1)} e^{m+1} = r^{m+1} \)

Correct \( u^{m} \leftarrow u^{m} + e^{m} \)

Interpolate \( e^{m} = P^{(m)} e^{m+1} \)
2D model problem: \[-u_{xx} - u_{yy} = f(x, y)\]

- high-frequency error is removed by relaxation
- low-frequency error needs to be removed by coarse-grid correction
- low-frequency error on fine grid becomes higher frequency error on coarse grid
AMG complexity - scalability

- **Operator complexity** $C_{op} = \frac{\sum_i \text{nonzeros}(A_i)}{\text{nonzeros}(A_0)}$

  e.g., 3D, ideally: $C_{op} = 1 + 1/8 + 1/64 + \ldots < 8/7$

  *measure of memory use, and work in solve phase*

- **scalable algorithm:**
  
  $\mathcal{O}(n)$ operations per V-cycle ($C_{op}$ bounded)

  AND

  number of V-cycles independent of $n$

  ($\rho_{\text{AMG}}$ independent of $n$)
O(n) scalability
AMG coarsening and interpolation

- large $a_{ij}$, ‘strong connections’ are important

- define strength matrix $S$:

$$\begin{bmatrix}
x & x & x \\
x & x & x & x \\
x & x & x \\
x & x & x & x \\
x & x & x \\
\end{bmatrix} \quad \begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 1 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{bmatrix}$$

- consider the undirected graph of $S$

- apply parallel maximal independent set algorithm to $\text{graph}(S)$ [Luby, 1986]
classical AMG coarsening (CLJP)

- *(C1)* Maximal Independent Set:
  Independent: no two C-points are connected
  Maximal: if one more C-point is added, the independence is lost

- *(C2)* All F-F connections require connections to a common C-point (for good interpolation)

- F-points have to be changed into C-points, to ensure *(C2); (C1)* is violated

more C-points, higher complexity
classical coarsening: scalability results

- example: finite difference Laplacian, parallel CLJP coarsening algorithm

- 2D (5-point): near-optimal scalability \((250^2 \text{ dof/proc})\)

<table>
<thead>
<tr>
<th>Procs</th>
<th>(C_{op})</th>
<th>(t_{tot})</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>4.48</td>
<td>2.89</td>
<td>9</td>
</tr>
<tr>
<td>64</td>
<td>4.50</td>
<td>3.85</td>
<td>9</td>
</tr>
<tr>
<td>256</td>
<td>4.50</td>
<td>5.01</td>
<td>9</td>
</tr>
</tbody>
</table>
(2) classical coarsening may lead to complexity problems

- 3D (7-point): complexity growth

<table>
<thead>
<tr>
<th>dof</th>
<th>$C_{op}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$32^3$</td>
<td>16.17</td>
</tr>
<tr>
<td>$64^3$</td>
<td>22.51</td>
</tr>
</tbody>
</table>
(3) Parallel Modified Independent Set (PMIS) coarsening

our approach to reduce complexity:

- *do not add C points for strong F-F connections that do not have a common C point*

- *less C points, reduced complexity, but worse convergence factors expected*

- *compensate by GMRES acceleration*
parallel PMIS results: 7-point finite difference Laplacian in 3D, $40^3$ dof per proc

CLJP and PMIS-GMRES(10)

<table>
<thead>
<tr>
<th>proc</th>
<th>$C_{op}$</th>
<th>Iter</th>
<th>$t_{total}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.39</td>
<td>6</td>
<td>3.35</td>
</tr>
<tr>
<td>512</td>
<td>17.02</td>
<td>10</td>
<td>35.83</td>
</tr>
<tr>
<td>1331</td>
<td>17.19</td>
<td>10</td>
<td>46.25</td>
</tr>
<tr>
<td>1</td>
<td>2.32</td>
<td>13</td>
<td>1.28</td>
</tr>
<tr>
<td>512</td>
<td>2.37</td>
<td>25</td>
<td>12.77</td>
</tr>
<tr>
<td>1331</td>
<td>2.37</td>
<td>28</td>
<td>17.99</td>
</tr>
</tbody>
</table>
convergence problems on PMIS-coarsened grids

- PMIS coarsening works well for many problems, but requires GMRES acceleration
- for some problems, too many iterations are necessary because interpolation is not accurate enough ("not enough C-points")
- one solution: add C-points (CLJP…)
- other solution: use distance-two C-points for interpolation = long-range interpolation
  ➔ F-F interpolation
convergence problems

- 3D elliptic PDE with jumps in coefficient $a$

\[(au_x)_x + (au_y)_y + (au_z)_z = 1\]

- 1000 processors, $40^3$ dof/proc

<table>
<thead>
<tr>
<th></th>
<th>$t_{tot}$</th>
<th>$C_{op}$</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLJP</td>
<td>52.48</td>
<td>17.00</td>
<td>17</td>
</tr>
<tr>
<td>PMIS</td>
<td>211.79</td>
<td>2.40</td>
<td>686</td>
</tr>
</tbody>
</table>

- remedy: improve interpolation used with PMIS
(4) improving interpolation: F-F interpolation

- when strong F-F connection without a common C-point is detected, do not add C-point, but extend interpolation stencil to distance-two C-points
- no C-points added, but larger interpolation stencils
results using long-range F-F interpolation

- 3D elliptic PDE with jumps in coefficient $a$

\[
(au_x)_x + (au_y)_y + (au_z)_z = 1
\]

- 1 processor, AMG+GMRES, $80^3$ dof

<table>
<thead>
<tr>
<th></th>
<th>$t_{tot}$</th>
<th>$C_{op}$</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLJP</td>
<td>48.0</td>
<td>21.54</td>
<td>7</td>
</tr>
<tr>
<td>PMIS</td>
<td>94.6</td>
<td>2.46</td>
<td>188</td>
</tr>
<tr>
<td>PMIS + F-F</td>
<td>21.4</td>
<td>4.90</td>
<td>9</td>
</tr>
</tbody>
</table>
reduce complexity: FF1 Interpolation

- **Modified FF Interpolation (FF1)**

- To reduce operator complexity, only include one distance-two C-point when a strong FF connection is encountered

- Setup time, complexity are reduced
results: 7-pt Laplacian Problem

- PMIS coarsening, 1 processor, $128^3$ dof

- AMG

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>2.36</td>
<td>77</td>
<td>16.63</td>
<td>85.93</td>
<td>102.56</td>
</tr>
<tr>
<td>FF</td>
<td>4.80</td>
<td>13</td>
<td>83.81</td>
<td>22.86</td>
<td>106.67</td>
</tr>
<tr>
<td>FF1</td>
<td>3.68</td>
<td>15</td>
<td>44.22</td>
<td>22.07</td>
<td>66.29</td>
</tr>
</tbody>
</table>
scalability of PMIS-FF1

7-pt Laplacian, PMIS Coarsening: AMG

# iterations

n

CAIMS 2006
results: 3D elliptic PDE with jumps

\[(au_x)_x + (au_y)_y + (au_z)_z = 1\]

- AMG, 1 processor, 120^3 dof

<table>
<thead>
<tr>
<th></th>
<th>C_{op}</th>
<th># iterations</th>
<th>t_{setup} (s)</th>
<th>t_{solve} (s)</th>
<th>t_{total} (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>2.44</td>
<td>&gt;&gt; 200</td>
<td>Slow to \textit{converge}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF</td>
<td>4.94</td>
<td>14</td>
<td>62.95</td>
<td>20.54</td>
<td>83.49</td>
</tr>
<tr>
<td>FF1</td>
<td>3.84</td>
<td>18</td>
<td>35.36</td>
<td>22.47</td>
<td>57.83</td>
</tr>
</tbody>
</table>
Scaling results on Blue Gene/L

### Top 500 Supercomputer list (November 2005)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>System</th>
<th>Processors</th>
<th>Rmax (Gflop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Livermore Lab, US</td>
<td>IBM Blue Gene/L</td>
<td>131,072</td>
<td>280,600</td>
</tr>
<tr>
<td>2</td>
<td>Thomas J. Watson, US</td>
<td>IBM Blue Gene</td>
<td>40,960</td>
<td>91,290</td>
</tr>
<tr>
<td>3</td>
<td>Livermore Lab, US</td>
<td>IBM pSeries</td>
<td>10,240</td>
<td>63,390</td>
</tr>
<tr>
<td>4</td>
<td>NASA Ames, US</td>
<td>SGI Altix</td>
<td>10,160</td>
<td>51,870</td>
</tr>
<tr>
<td>5</td>
<td>Sandia Lab, US</td>
<td>Dell PowerEdge</td>
<td>8,000</td>
<td>38,270</td>
</tr>
<tr>
<td>6</td>
<td>Sandia Lab, US</td>
<td>Cray XT3</td>
<td>10,880</td>
<td>36,190</td>
</tr>
<tr>
<td>7</td>
<td>Earth Simulator Center, Japan</td>
<td>Earth Simulator NEC</td>
<td>5,120</td>
<td>35,860</td>
</tr>
<tr>
<td>8</td>
<td>Barcelona Supercomputer Center, Spain</td>
<td>IBM JS20 Cluster</td>
<td>4,800</td>
<td>27,910</td>
</tr>
<tr>
<td>9</td>
<td>University Groningen, Netherlands</td>
<td>IBM Blue Gene</td>
<td>12,288</td>
<td>27,450</td>
</tr>
<tr>
<td>10</td>
<td>Oak Ridge Lab, US</td>
<td>Cray XT3</td>
<td>5,200</td>
<td>20,527</td>
</tr>
</tbody>
</table>
LLNL Blue Gene/L

- dual-processor nodes optimized for data access
- each node: one processor for simulation, one for communication; only 256MB ram per processor
- lightweight, single-process linux kernel
LLNL Blue Gene/L results

7-pt Laplacian, total execution time, AMG-CG

- PMIS, virt.
- Fal, virt.

Times in secs vs. no. of procs
LLNL Blue Gene/L results on full machine

7-pt Laplacian, total execution time, AMG-CG, total problem size ~2 billion
PMIS: select 1

- select C-pts with maximal measure locally
- make neighbour F-pts
- remove neighbour edges
PMIS: remove and update 1

- select C-pts with maximal measure locally
- make neighbours F-pts
- remove neighbour edges
PMIS: select 2

- select C-pts with maximal measure locally
- make neighbours F-pts
- remove neighbour edges
PMIS: remove and update 2

- select C-pts with maximal measure locally
- make neighbours F-pts
- remove neighbour edges
PMIS: final grid

- select C-pts with maximal measure locally
- make neighbour F-pts
- remove neighbour edges
- parallel algorithm
LLNL Blue Gene/L results

7-pt Laplacian, total execution time, AMG-CG

- Falgout
- PMIS
- Falg., virt.
- PMIS, virt.