PetaScale Finite Element Methods for Hierarchical Hybrid Grids using Hybrid Parallelization

16th Copper Mountain Conference on Multigrid Methods, 2013

Björn Gmeiner, Ulrich Rüde

March, 2013
Contents

Hierarchical Hybrid Grids

Comparison on Different HPC-Clusters

Stokes Solver within HHG
Hierarchical Hybrid Grids
HHG - Combining finite element and multigrid methods

FE mesh may be unstructured. What nodes to remove for coarsening? Not straightforward! Why not start from the coarse grid?
Properties of the HHG approach

Advantages

• Multigrid is straightforward
• Very memory efficient
• Massive performance benefits on current computer architectures
• Subserves parallelization

⇒ $3 \cdot 10^{12}$ unknowns are possible

Limitation

• Coarse input grid needed
• Adaptivity
Performance Comparison on Different HPC-Clusters
Benchmark Problem and Discretization

Scalar elliptic equation:

\[
\int_{\Omega} \epsilon(x) \nabla u(x) \cdot \nabla v(x) \, dx = \int_{\Omega} f(x) v(x) \, dx \tag{1}
\]

Finite elements settings:

- linear (tetrahedral) finite elements
- constant stencil for constant coefficients \( \epsilon(x) \)
- on-the-fly assembly for variable coefficients inside the macro-elements, storage of the stencils on the interfaces
Multigrid Setup

Multigrid sub-algorithms and parameters:

- three Gauss-Seidel for pre- and post-smoothing steps
- linear interpolation
- six multigrid levels
- parallel conjugate gradient algorithm on the coarsest grid
- direct coarse grid approximation with coefficients averaging
### Benchmark-Machine Overview

<table>
<thead>
<tr>
<th></th>
<th>JUGENE</th>
<th>JUQUEEN</th>
<th>SuperMUC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
<td>IBM BlueGene/P</td>
<td>IBM BlueGene/Q</td>
<td>IBM System x iDataPlex</td>
</tr>
<tr>
<td><strong>Processor</strong></td>
<td>IBM PowerPC 450</td>
<td>IBM PowerPC A2</td>
<td>Intel Xeon E5-2680 8C</td>
</tr>
<tr>
<td><strong>Clock Frequency</strong></td>
<td>0.85 GHz</td>
<td>1.6 GHz</td>
<td>2.8 GHz</td>
</tr>
<tr>
<td><strong>Number of Nodes</strong></td>
<td>73 728</td>
<td>28 672</td>
<td>9 216</td>
</tr>
<tr>
<td><strong>HW Threads per Core</strong></td>
<td>4</td>
<td>16x4</td>
<td>16x2</td>
</tr>
<tr>
<td><strong>Memory per HW Thread</strong></td>
<td>0.5 GB</td>
<td>0.25 GB</td>
<td>1 GB</td>
</tr>
<tr>
<td><strong>Network Topology</strong></td>
<td>3D Torus</td>
<td>5D Torus</td>
<td>Tree</td>
</tr>
<tr>
<td><strong>GFlops per Watt</strong></td>
<td>0.44</td>
<td>2.54</td>
<td>0.94</td>
</tr>
</tbody>
</table>

**Table**: System overview of the supercomputers JUGENE, JUQUEEN, and SuperMUC
Performance comparison at $\approx 0.8$ PFlop peak

<table>
<thead>
<tr>
<th></th>
<th>JUGENE</th>
<th>JUQUEEN</th>
<th>SuperMUC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single Node</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak FLOP/s (const. coeff. $\epsilon$)</td>
<td>6%</td>
<td>7%</td>
<td>12%</td>
</tr>
<tr>
<td>Peak FLOP/s (var. coeff. $\epsilon$)</td>
<td>9%</td>
<td>10%</td>
<td>13%</td>
</tr>
<tr>
<td>Peak bandwidth eq. (const. coeff. $\epsilon$)</td>
<td>11%</td>
<td>53%</td>
<td>60%</td>
</tr>
</tbody>
</table>

| **Parallel Efficiencies** (at $\approx 0.8$ PFlop peak) |        |         |          |
| Scaling (const. coeff. $\epsilon$) | 65%     | 64%     | 72%      |
| Scaling (var. coeff. $\epsilon$)  | 94%     | 93%     | 96%      |
| Number of processes | 262 144 | 262 144 | 32 768   |

| **Energy Efficiency** |        |         |          |
| Energy improvement compared to JUGENE (const. coeff. $\epsilon$) | 1       | 6.6     | 4.7      |
| Energy improvement compared to JUGENE (var. coeff. $\epsilon$) | 1       | 6.4     | 3.2      |

Table: Single node, parallel performance and power consumptions on the different clusters.
Parallel Efficiencies of HHG on Different Clusters

- **Problem Size**
- **Parallel Efficiency**

- **Clusters**:
  - JUGENE
  - JUQUEEN
  - SuperMUC

- **Problem Sizes**:
  - 1 Node Card
  - 1 Midplane
  - 1 Island
  - Hybrid Parallel

- **Core Counts**:
  - 262,144 cores
  - 393,216 cores
  - 131,072 cores
  - 262,144 cores

Department for Computer Science 10 (System Simulation)
Stokes Solver within HHG
Boussinesq Model for Mantle Convection Problems

\[-\nabla \cdot (2\mu\varepsilon(u)) + \nabla p = RaTe_r,\]
\[\nabla \cdot u = 0,\]
\[\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla \cdot (Pe^{-1} \nabla T) = \gamma.\]

Helps understanding geophysical phenomena, like
- Continental drift
- Plate tectonics
- Earthquakes
Stokes System

\[ \nabla \cdot (2\mu \nabla u) - \nabla p = RaTe_r, \quad (2) \]

\[ \nabla \cdot (\rho u) = 0. \quad (3) \]

Decoupling with a pressure correction approach using the Schur complement leads to

\[
\begin{pmatrix}
A & B^T \\
0 & -BA^{-1}B^T
\end{pmatrix}
\begin{pmatrix}
U \\
P
\end{pmatrix} =
\begin{pmatrix}
F \\
0
\end{pmatrix}.
\]

with the matrices

\[ A_{ij} = (\nabla \phi_i^u, 2\mu \nabla \phi_j^u) \quad (4) \]

\[ B_{ij} = -(\phi_i^q, \nabla \cdot \phi_j^u) \quad (5) \]

\[ F_i = (\phi_i^u, RaTe_r). \quad (6) \]
Spherical refinement of an icosahedron (0)
Discretization with prismatic elements
Spherical refinement of an icosahedron (0)
Spherical refinement of an icosahedron (1)
Spherical refinement of an icosahedron (2)
Spherical refinement of an icosahedron (3)
Spherical refinement of an icosahedron (4)
Icosahedral Finite Element Mesh
Benchmark Problem within HHG
Solution of the Stationary Stokes Equation on the Mantle Geometry

<table>
<thead>
<tr>
<th>Compute nodes</th>
<th>Threads</th>
<th>Grid points</th>
<th>Resolution</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>$2.1 \cdot 10^7$</td>
<td>32 km</td>
<td>73 s</td>
</tr>
<tr>
<td>4</td>
<td>240</td>
<td>$1.6 \cdot 10^8$</td>
<td>16 km</td>
<td>95 s</td>
</tr>
<tr>
<td>30</td>
<td>1920</td>
<td>$1.3 \cdot 10^9$</td>
<td>8 km</td>
<td>98 s</td>
</tr>
<tr>
<td>240</td>
<td>15260</td>
<td>$1.1 \cdot 10^{10}$</td>
<td>4 km</td>
<td>111 s</td>
</tr>
<tr>
<td>1920</td>
<td>122880</td>
<td>$8.5 \cdot 10^{10}$</td>
<td>2 km</td>
<td>125 s</td>
</tr>
<tr>
<td>15360</td>
<td>983040</td>
<td>$6.9 \cdot 10^{11}$</td>
<td>1 km</td>
<td>139 s</td>
</tr>
</tbody>
</table>
Run-time Distribution of the Solving Procedure measured with Scalasca

- Pressure correction: 80.6 s
- Multigrid: 61.8 s
- Multigrid, without finest level: 28.1 s
- Multigrid, CG: 21.5 s
- Multigrid, CG, scalar product: 13.1 s
- Multigrid, CG, scalar product: 5.7 s
- Communication: 7.9 s
- Communication: 0.5 s
- Communication: 4.7 s

Department for Computer Science 10 (System Simulation)
Temperatur Field of a First Convection-Diffusion Simulation within HHG
Conclusions and Outlook

- Comparison of a geometric multigrid on three different large-scale clusters
- First extensions of HHG to solve the Stokes system
- Utilize HHG as a test-bed for a new Earth mantle convection code (TERRA-NEO)
<table>
<thead>
<tr>
<th>Number of Threads</th>
<th>Number of Unknowns</th>
<th>Time per V-cycle</th>
<th>Number of Threads</th>
<th>Number of Unknowns</th>
<th>Time per V-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>$1.33 \cdot 10^8$</td>
<td>2.34 s</td>
<td>16384</td>
<td>$3.43 \cdot 10^{10}$</td>
<td>3.15 s</td>
</tr>
<tr>
<td>128</td>
<td>$2.67 \cdot 10^8$</td>
<td>2.41 s</td>
<td>32768</td>
<td>$6.87 \cdot 10^{10}$</td>
<td>3.28 s</td>
</tr>
<tr>
<td>256</td>
<td>$5.35 \cdot 10^8$</td>
<td>2.80 s</td>
<td>65536</td>
<td>$1.37 \cdot 10^{11}$</td>
<td>3.39 s</td>
</tr>
<tr>
<td>512</td>
<td>$1.07 \cdot 10^9$</td>
<td>2.82 s</td>
<td>131072</td>
<td>$2.75 \cdot 10^{11}$</td>
<td>3.56 s</td>
</tr>
<tr>
<td>1024</td>
<td>$2.14 \cdot 10^9$</td>
<td>2.82 s</td>
<td>262144</td>
<td>$5.50 \cdot 10^{11}$</td>
<td>3.68 s</td>
</tr>
<tr>
<td>2048</td>
<td>$4.29 \cdot 10^9$</td>
<td>2.84 s</td>
<td>524288</td>
<td>$1.10 \cdot 10^{12}$</td>
<td>3.76 s</td>
</tr>
<tr>
<td>4096</td>
<td>$8.58 \cdot 10^9$</td>
<td>2.96 s</td>
<td>1048576</td>
<td>$2.20 \cdot 10^{12}$</td>
<td>4.07 s</td>
</tr>
<tr>
<td>8192</td>
<td>$1.72 \cdot 10^{10}$</td>
<td>3.09 s</td>
<td>1572864</td>
<td>$3.29 \cdot 10^{12}$</td>
<td>4.03 s</td>
</tr>
</tbody>
</table>

**Table:** Weak scaling experiment on JUQUEEN solving a problem on the full machine with up to $3.29 \cdot 10^{12}$ unknowns.