TERRA-NEO

or

Is 2.44 trillion unknowns the largest finite element system that can be solved today?

U. Rüde (LSS Erlangen, ruede@cs.fau.de)

Lehrstuhl für Informatik 10 (Systemsimulation)
Universität Erlangen-Nürnberg
www10.informatik.uni-erlangen.de

Doktoranden-Seminar zum SPPEXA

Erlangen, im SS 2014
Project Consortium

- H.-P. Bunge (LMU, Geophysik)
  - Dr. M. Mohr
  - J. Weismüller
  - NN (Doktorand)
- B. Wohlmuth (TUM, Mathematik)
  - Dr. C. Waluga
  - M. Hoffmann (bis 12/13)
  - Dr. L. John (ab 4/14)
- M. Huber (seit 1/14 derzeit am LSS)
- G. Wellein (FAU, RRZE)
  - H. Stengel
- U. Rüde (FAU, Inf)
  - Dr. H. Köstler
  - Dr. B. Gmeiner (TUM bis 3/14)
Co-Design of an Exascale Earth Mantle Modeling Framework

\begin{align*}
- \nabla \cdot (2\eta \varepsilon(u)) + \nabla p &= \rho(T)g, \\
\nabla \cdot u &= 0, \\
\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla \cdot (\kappa \nabla T) &= \gamma.
\end{align*}

\begin{itemize}
  \item $u$: velocity
  \item $p$: dynamic pressure
  \item $T$: temperature
  \item $\nu$: viscosity of the material
  \item $\varepsilon(u) = \frac{1}{2}(\nabla u + (\nabla u)^T)$: strain rate tensor
  \item $\rho$: density
  \item $\kappa, \gamma, g$: thermal conductivity, heat sources, gravity vector
\end{itemize}

Stokes equation: 

\[ -\text{div}(\nabla u - \rho I) = f, \quad \text{div} u = 0 \]

FEM Discretization:

\[ a(u_l, v_l) + b(v_l, p_l) = L(v_l) \quad \forall v_l \in V_l, \]
\[ b(u_l, q_l) - c(p_l, q_l) = 0 \quad \forall q_l \in Q_l, \]

with: 
\[ a(u, v) := \int_{\Omega} \nabla u : \nabla v \, dx, \quad b(u, q) := -\int_{\Omega} \text{div} u \cdot q \, dx \]

Schur-complement formulation:

\[
\begin{bmatrix}
  A_l & B_l^T \\
  0 & C_l + B_l A_l^{-1} B_l^T
\end{bmatrix}
\begin{bmatrix}
  u_l \\
  p_l
\end{bmatrix}
= \begin{bmatrix}
  f_l \\
  B_l A_l^{-1} f_l
\end{bmatrix}
\]
Regular tetrahedral refinement

Structured refinement of tetrahedra

Use regular HHG patches for partitioning the domain (only 2D for simplification)

communication of ghost layers

Coarse grid with 132k elements, as assigned to supercomputer
Geodynamics

Starting with an icosahedron

refining

until $10^{12}$ FE are reached
## Scalability on JuQueen for Stokes

### Run-time distribution of the solving procedure

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Threads</th>
<th>Grid points</th>
<th>Resolution</th>
<th>Time: (A)</th>
<th>(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>$2.1 \cdot 10^{07}$</td>
<td>32 km</td>
<td>30 s</td>
<td>89 s</td>
</tr>
<tr>
<td>4</td>
<td>240</td>
<td>$1.6 \cdot 10^{08}$</td>
<td>16 km</td>
<td>38 s</td>
<td>114 s</td>
</tr>
<tr>
<td>30</td>
<td>1,920</td>
<td>$1.3 \cdot 10^{09}$</td>
<td>8 km</td>
<td>40 s</td>
<td>121 s</td>
</tr>
<tr>
<td>240</td>
<td>15,360</td>
<td>$1.1 \cdot 10^{10}$</td>
<td>4 km</td>
<td>44 s</td>
<td>133 s</td>
</tr>
<tr>
<td>1,920</td>
<td>122,880</td>
<td>$8.5 \cdot 10^{10}$</td>
<td>2 km</td>
<td>48 s</td>
<td>153 s</td>
</tr>
<tr>
<td>15,360</td>
<td>983,040</td>
<td>$6.9 \cdot 10^{11}$</td>
<td>1 km</td>
<td>54 s</td>
<td>170 s</td>
</tr>
</tbody>
</table>

**Graph**

- **Computation**
- **Communication**

- **Tasks**: Pressure correction, Solve momentum, Multigrid, finest level, Multigrid, remaining levels, Multigrid, CG, residual, Multigrid, CG, scalar product
Parallel Efficiency of HHG

Problem Size vs Parallel Efficiency

- JUGENE
- JUQUEEN
- SuperMUC

- 1 Node Card
- 1 Midplane
- 1 Island
- Hybrid Parallel

- 262,144 cores
- 393,216 cores
- 1,310,720 cores
Textbook Multigrid Efficiency (TME)?

- TME introduced by Brandt to distinguish between „optimal“ and „fast“ algorithms
  - The cost of solution should be less than 10 times the cost of a Gauss-Seidel relaxation (or Matrix-vector multiply) of the system
- We define the
  - algorithmic TME-factor:
    • operations for solution/ operations for a single relaxation sweep
  - parallel TM-scalability factor
    • time for solution / (time for elementary relaxation × #unknowns)
- For the Stokes system: 4 equations (3x velocity, 1 pressure)
  - Idealized sweep is possible at 26/4 ≈ 7 (× 10^{12} grid points/sec)
  - this ignores cost of communication
- Full SuperMuc solves 6.9×10^{11} mesh points in 41 seconds
  - equivalent to >2.44 ×10^{12} unknowns
  - for residual reduction by 10^3
    • sufficient for FMG solution or
    • single time step
for (int i=1; i < (tsize-j-k-1); i=i+2) {
    u[mp_mr+i] = c[0] * (-
c[1] * u[mp_mr+i+1] - c[2] * u[mp_tr+i-1] -
c[9] * u[bp_mr+i+1] - c[10] * u[bp_tr+i-1] -
f[mp_mr+i] );
}

This loop should be executed on each SuperMuc core at

- 720 M updates/sec (in theory - peak performance)
- 176 M updates/sec (in practice - memory access bottleneck; RB-ordering prohibits vector loads)

Thus whole SuperMuc should perform

- 147456*176M ≈ 26T elementary updates/sec
Do we reach TME?

Our overall TM-scalability factor is

- $\frac{41 \text{ sec}}{(2.44 \times 10^{12}/26 \times 10^{12})} \approx 386$
- the solution is 386 times as expensive as an idealized single relaxation sweep

Why do we fail reach a TM-scalability factor $\approx 10$?

- have ignored (parallel) overheads, in particular the reduced efficiency of parallel MG on coarser grids
- pressure correction Schur solver is suboptimal
  - costing 7 CG iterations
  - requiring the CG preconditioner sweep, etc.
  - error reduction by $10^3$ is over-solving
- Multigrid only as preconditioner for scalar velocity systems
- Chance for further tuning of parameters, i.e. are V(3,3) cycles optimal?
TM Scalability as Guideline for Performance Engineering

- Quantify the cost of an elementary relaxation
  - through systematic micro-kernel benchmarks
  - optimize kernel performance, justify any performance degradation
  - compute idealized aggregate performance for large scale parallel system

- Evaluate (parallel) solver performance

- Quantify TM-scalability factor
  - as overall measure of efficiency
  - analyse discrepancies
  - identify possible improvements
  - improve TM-scalability factor
Thank you for your attention!

Questions?

Video generated at LSS with the massively parallel waLberla Software framework for Lattice Boltzmann based multi-physics applications.

Slides, reports, thesis, animations available for download at:

www10.informatik.uni-erlangen.de

Ulrich Rüde - Lehrstuhl für Simulation