Multigrid PDE Solvers on PetaScale Systems

T. Gradl, C. Freundl, H. Köstler (LSS Erlangen)
B. Bergen (LSS Erlangen/Los Alamos)
U. Rüde (LSS Erlangen, rude@cs.fau.de)

Lehrstuhl für Informatik 10 (Systemsimulation)
Universität Erlangen-Nürnberg
www10.informatik.uni-erlangen.de
Atlanta, March 2008
Overview

- Motivation

- Applications of Large-Scale FE Simulations
  - Direct and Inverse Bio-Electric Field Problems
  - 3D image registration/image processing/analysis
  - Flow Induced Noise and Acoustics
  - Multiphysics-Problems in Nano Technology
  - Simulation of High Temperature Furnaces
  - Earth Mantle Convection

- **Scalable** Finite Element Solvers
  - **Scalable Algorithms:** Multigrid
  - **Scalable Software Architecture:** Hierarchical Hybrid Grids (HHG)
  - Performance & Scalability Results
  - Adaptive Grids in HHG

- Outlook
  - Towards Peta-Scale FE Solvers
Part I

Motivation
HHG Motivation I
Structured vs. Unstructured Grids
(on Hitachi SR 8000)

Extinct Pet Dinosaur HLRB-I:
Hitachi SR 8000
at the Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften
No. 5 in TOP-500 at time of installation in 2000

- gridlib/HHG MFlops rates for matrix-vector multiplication on one node
- compared with highly tuned JDS results for sparse matrices
- Many emerging architectures have similar properties (Cell, GPU)
Our Current Play Station: HLRB-II

- ccNuma architecture
  - 9700 CPU Cores (Itanium II)
  - 40 TBytes RAM
  - NUMAlink network
  - > 60 TFLOP/s Peak Performance
- Our testing ground for scalability experiments

HLRB-II: SGI Altix 4700
at the Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften
No. 15 in in TOP-500 of Nov. 2007
HHG Motivation II: DiMe - Project

Data Local Iterative Methods for the Efficient Solution of Partial Differential Equations

www10.informatik.uni-erlangen.de/de/Research/Projects/DiME/

- Cache-optimizations for sparse matrix codes (since 1996)
- High Performance Multigrid Solvers
- Efficient LBM codes for CFD (with free surface flow)
Part III - a

Towards Scalable FE Software

**Multigrid Algorithms**
What is Multigrid?

- Has nothing to do with „grid computing“
- A general methodology
  - multi-scale (actually it is the „original“)
  - many different applications
  - developed in the 1970s - ...
- Useful e.g. for solving elliptic PDEs
  - large sparse systems of equations
  - iterative
  - convergence rate independent of problem size
  - asymptotically optimal complexity -> algorithmic scalability!
  - can solve e.g. 2D Poisson Problem in ~ 30 operations per gridpoint
  - efficient parallelization possible
  - best (maybe the only?) basis for fully scalable FE solvers
**Multigrid: V-Cycle**

**Goal:** solve $A^h u^h = f^h$ using a hierarchy of grids

Relax on

$$A^h u^h = f^h$$

Correct

$$u^h \leftarrow u^h + e^h$$

Residual

$$r^h = f^h - A^h u^h$$

Restrict

$$r^H = I^H_h r^h$$

Interpolate

$$e^h = I^h_H e^H$$

Solve

$$A^H e^H = r^H$$

by recursion

...
Parallel High Performance FE Multigrid

- Parallelize "plain vanilla" multigrid
  - partition domain
  - parallelize all operations on all grids
  - use clever data structures

- Do not worry (so much) about coarse grids
  - idle processors?
  - short messages?
  - sequential dependency in grid hierarchy?

- Why we do not use Domain Decomposition
  - DD without coarse grid does not scale (algorithmically) and is inefficient for large problems/many processors
  - DD with coarse grids is still less efficient than multigrid and is as difficult to parallelize
Multigrid Details

- Finite Element induced
  - interpolation
  - restriction
  - Galerkin coarsening
- V(2,2) cycle, sometimes V(3,3)
- line-wise Gauss-Seidel smoother
  - lines in red-black ordering
  - neglecting a few dependencies across processor boundaries (Jacobi-like points)
- mostly standard
Part III - b

Towards Scalable FE Software

Hierarchical Hybrid Grids
Hierarchical Hybrid Grids (HHG)

- Unstructured input grid
  - resolves geometry of problem domain
- Patch-wise regular refinement
  - generates nested grid hierarchies naturally suitable for geometric multigrid algorithms
- New:
  - Modify storage formats and operations on the grid to exploit the regular substructures
- Does an unstructured grid with 1,000,000,000,000 elements make sense?

HHG Goal: Ultimate Parallel FE Performance!
HHG refinement example

Input Grid
HHG Refinement example

Refinement Level one
HHG Refinement example

Refinement Level Two
HHG Refinement example

Structured Interior
HHG Refinement example

Structured Interior
HHG Refinement example

Edge Interior
HHG Refinement example

Edge Interior
Hierarchical hybrid grids (HHG)
- are not only another block structured grid
HHG are more flexible (unstructured, hybrid input grids)
- are not only another unstructured geometric multigrid package
HHG achieve better performance
- unstructured treatment of regular regions does not improve performance
HHG for Parallelization

Use regular HHG patches for partitioning the domain
for each vertex do
    apply operation to vertex
end for

update vertex primary dependencies
for each edge do
    copy from vertex interior
    apply operation to edge
    copy to vertex halo
end for

update edge primary dependencies
for each element do
    copy from edge/vertex interiors
    apply operation to element
    copy to edge/vertex halos
end for

update secondary dependencies
HHG for Parallelization

SEND BUFFER

RECEIVE BUFFER

LOCAL BUFFER

Partition 0

MPI

RECEIVE BUFFER

SEND BUFFER

LOCAL BUFFER UNUSED

Partition 1

Cut Face

$N_i$

$N_j$

$N_l$
Part II - c

Towards Scalable FE Software

Performance Results
### Performance of 3D-MG-Smoother for 7-pt stencil in Mflops on Itanium 1.4 GHz

<table>
<thead>
<tr>
<th>grid size</th>
<th>$17^3$</th>
<th>$33^3$</th>
<th>$65^3$</th>
<th>$129^3$</th>
<th>$257^3$</th>
<th>$513^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>1072</td>
<td>1344</td>
<td>715</td>
<td>677</td>
<td>490</td>
<td>579</td>
</tr>
<tr>
<td>no blocking</td>
<td>2445</td>
<td>1417</td>
<td>995</td>
<td>1065</td>
<td>849</td>
<td>819</td>
</tr>
<tr>
<td>2x blocking</td>
<td>2400</td>
<td>1913</td>
<td>1312</td>
<td>1319</td>
<td>1284</td>
<td>1282</td>
</tr>
<tr>
<td>3x blocking</td>
<td>2420</td>
<td>2389</td>
<td>2167</td>
<td>2140</td>
<td>2134</td>
<td>2049</td>
</tr>
</tbody>
</table>

# Node Performance is Difficult! (B. Gropp)

DiMe project: Cache-aware Multigrid (1996 - 2008)

Node Performance is Possible!

- Performance of 3D-MG-Smoother for 7-pt stencil in Mflops on Itanium 1.4 GHz
- Array Padding
- Temporal blocking - in EPIC assembly language
- Software pipelining in the extreme (M. Stürmer - J. Treibig)
Multigrid Convergence Rates

![Graph showing convergence rates over cycle number](image-url)
Scaleup HHG and ParExPDE
<table>
<thead>
<tr>
<th>#Proc</th>
<th>#unkn. x 10^6</th>
<th>Ph.1: sec</th>
<th>Ph. 2: sec</th>
<th>Time to sol.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>134.2</td>
<td>3.16</td>
<td>6.38*</td>
<td>37.9</td>
</tr>
<tr>
<td>8</td>
<td>268.4</td>
<td>3.27</td>
<td>6.67*</td>
<td>39.3</td>
</tr>
<tr>
<td>16</td>
<td>536.9</td>
<td>3.35</td>
<td>6.75*</td>
<td>40.3</td>
</tr>
<tr>
<td>32</td>
<td>1,073.7</td>
<td>3.38</td>
<td>6.80*</td>
<td>40.6</td>
</tr>
<tr>
<td>64</td>
<td>2,147.5</td>
<td>3.53</td>
<td>4.92</td>
<td>42.3</td>
</tr>
<tr>
<td>128</td>
<td>4,295.0</td>
<td>3.60</td>
<td>7.06*</td>
<td>43.2</td>
</tr>
<tr>
<td>252</td>
<td>8,455.7</td>
<td>3.87</td>
<td>7.39*</td>
<td>46.4</td>
</tr>
<tr>
<td>504</td>
<td>16,911.4</td>
<td>3.96</td>
<td>5.44</td>
<td>47.6</td>
</tr>
<tr>
<td>2040</td>
<td>68,451.0</td>
<td>4.92</td>
<td>5.60</td>
<td>59.0</td>
</tr>
<tr>
<td>3825</td>
<td>128,345.7</td>
<td>6.90</td>
<td></td>
<td>82.8</td>
</tr>
<tr>
<td>4080</td>
<td>136,902.0</td>
<td></td>
<td>5.68</td>
<td></td>
</tr>
<tr>
<td>6102</td>
<td>205,353.1</td>
<td></td>
<td>6.33</td>
<td></td>
</tr>
<tr>
<td>8152</td>
<td>273,535.7</td>
<td></td>
<td>7.43*</td>
<td></td>
</tr>
<tr>
<td>9170</td>
<td>307,694.1</td>
<td></td>
<td>7.75*</td>
<td></td>
</tr>
</tbody>
</table>

Parallel scalability of scalar elliptic problem discretized by tetrahedral finite elements.

Times for 12 V(2,2) cycles on SGI Altix: Itanium-2 1.6 GHz.

Largest problem solved to date: 3.07 x 10^{11} DOFS on 9170 Procs: 7.8 s per V(2,2) cycle

B. Bergen, F. Hülsemann, U. Rüde, G. Wellein: ISC Award 2006, also:
„Is 1.7\times10^{10} unknowns the largest finite element system that can be solved today?“, SuperComputing, Nov‘ 2005.
What can we do with \(10^{10} - 10^{12}\) unknowns?

- resolve a moderate sized concert hall of \(10^4\) \(m^3\) volume with resolution smaller than the wavelength of the audible spectrum (<1 cm)
- compute long range interactions (gravitational/Coulomb potential) for systems with trillions of stars/atoms
- resolve the whole volume of earth’s atmosphere with \(~100\)m resolution (in 3D)
Part II - d

Towards Scalable FE Software

Adaptivity
Two approaches

- Red-green → conforming grids

- Hanging nodes → non-conforming grids
Refinement with hanging nodes in HHG

$Au = f$, $r = f - Au$, compactly supported basis functions

Uniform refinement: only one boundary layer

Adaptive refinement: two interface layers

1. Smooth $u$

2. Compute & restrict $r$

See: UR. Fully Adaptive Multigrid Methods, SINUM, 1993
Part IV

Outlook
Conclusions

Supercomputer Performance is Easy!

- If parallel efficiency is bad, choose a slower serial algorithm
  - it is probably easier to parallelize
  - and will make your speedups look much more impressive

- Introduce the “CrunchMe” variable for getting high Flops rates
  - advanced method: disguise CrunchMe by using an inefficient (but compute-intensive) algorithm from the start

- Introduce the “HitMe” variable to get good cache hit rates
  - advanced version: disguise HitMe by within “clever data structures” that introduce a lot of overhead

- Never cite “time-to-solution”
  - who cares whether you solve a real life problem anyway
  - it is the MachoFlops that interest the people who pay for your research

- Never waste your time by trying to use a complicated algorithm in parallel (such as multigrid)
  - the more primitive the algorithm the easier to maximize your MachoFlops.
Talk is Over

Please wake up!