Massively Parallel Multgrid for Finite Elements

B. Bergen (LSS Erlangen/ now Los Alamos)

T. Gradl, Chr. Freundl (LSS Erlangen)

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

Lehrstuhl für Informatik 10 (Systemsimulation)

Universität Erlangen-Nürnberg

www10.informatik.uni-erlangen.de

Copper Mountain Multigrid Method

March 2007

Overview

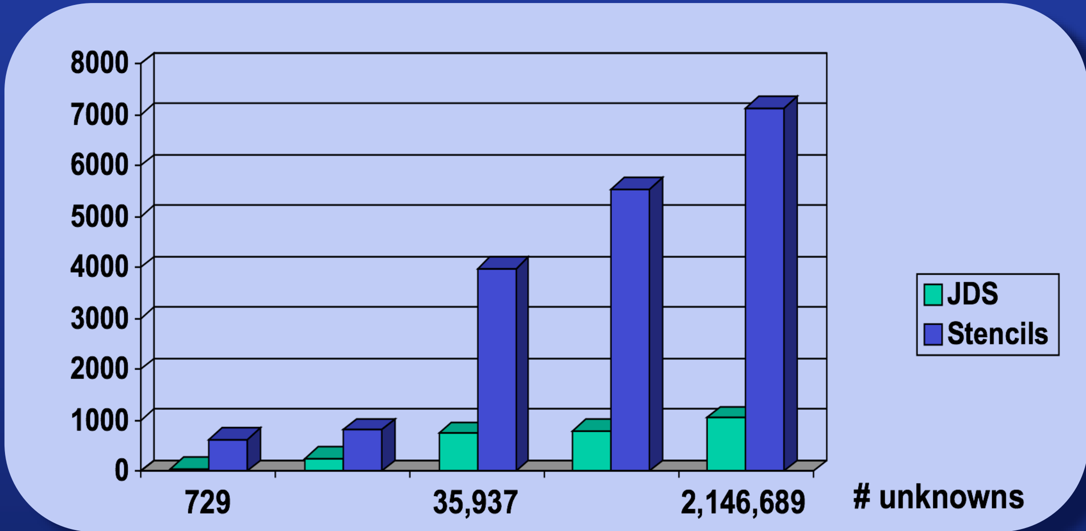
- Motivation
  - Supercomputers
  - Example Applications of Large-Scale FE Simulation
- Scalable Finite Element Solvers
  - Scalable Algorithms: Multigrid
  - Scalable Architectures:
    - **H**ierarchical **H**ybrid **G**rids (HHG)
    - **Par**allel **Ex**pression Templates for **PDE** (ParExPDE)
  - Results
- Outlook
  - Towards Peta-Scale FE Solvers

Part I

Motivation

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

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

Our New Play Station: HLRB-II

HLRB-II: SGI Altix 4700
at the Leibniz-Rechenzentrum der
Bayerischen Akademie der Wissenschaften
No. 18 in in TOP-500 of Nov. 2006

- Shared memory architecture, currently being upgarded from 4096 CPUs to
  - 9700 CPU Cores (Itanium II)
  - 40 TBytes RAM
  - NUMAlink network
  - > 60 TFLOP/s Peak Performance
- Our testing ground for scalability experiments

Part II-a

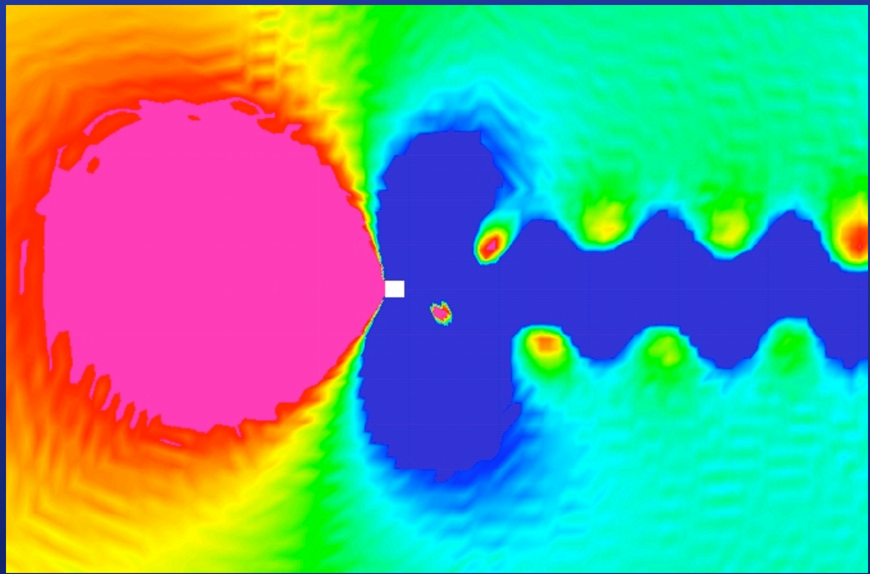
Towards Scalable FE Software

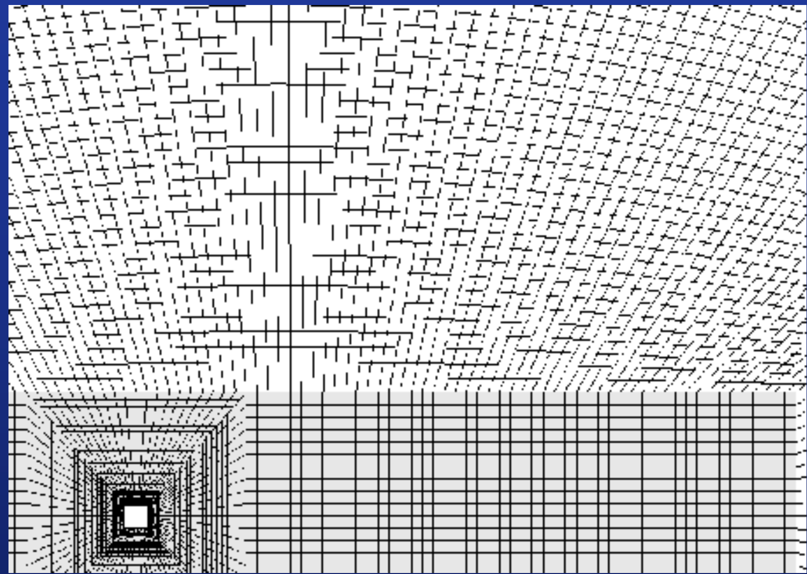
Scalable Algorithms:
Multigrid

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Example: Flow Induced Noise

Flow around a small square obstacle





Relatively simple geometries, but fine resolution for resolving physics

Images by M. Escobar (Dept. of Sensor Technology + LSTM)

Multigrid: V-Cycle

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

Relax on $A^h u^h = f^h$

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

Restrict $r^H = I_h^H r^h$

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

by recursion

... ...

Interpolate $e^h = I_H^h e^H$

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

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 procssors
  - DD with coarse grids is still less efficient than multigrid and is as difficult to parallelize

Part II - b

Towards Scalable FE Software

Scalable Architecture

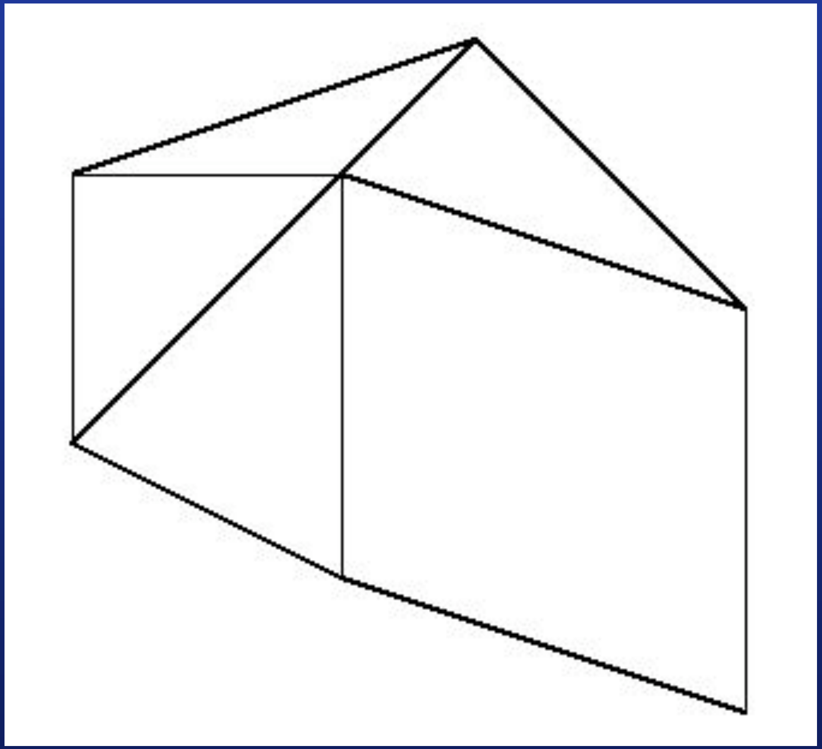
Hierarchical Hybrid Grids and Parallel Expression Templates for PDEs

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 100 000 000 000 elements make sense?

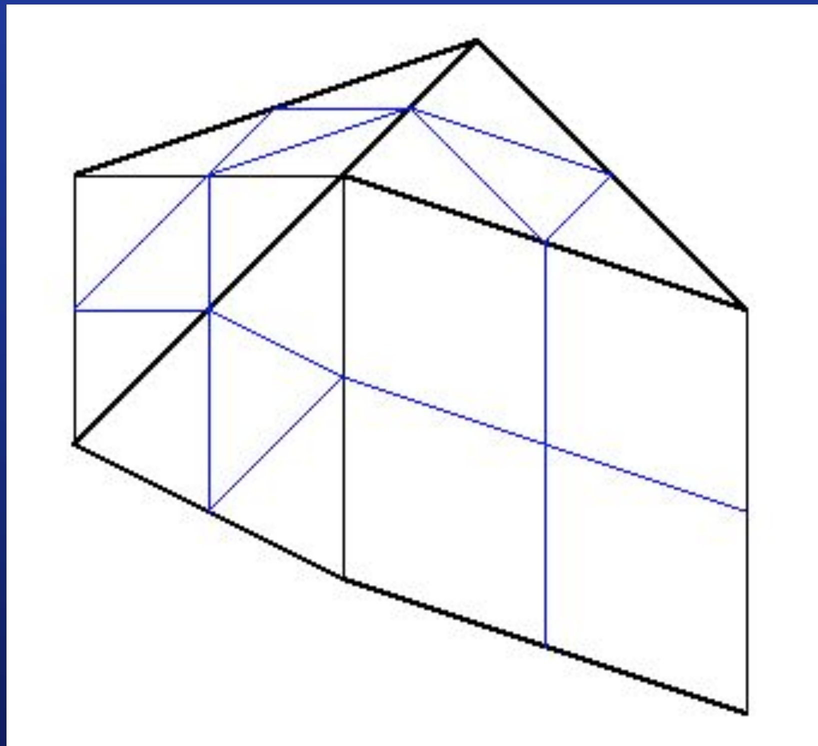
HHG - Ultimate Parallel FE Performance!

HHG refinement example



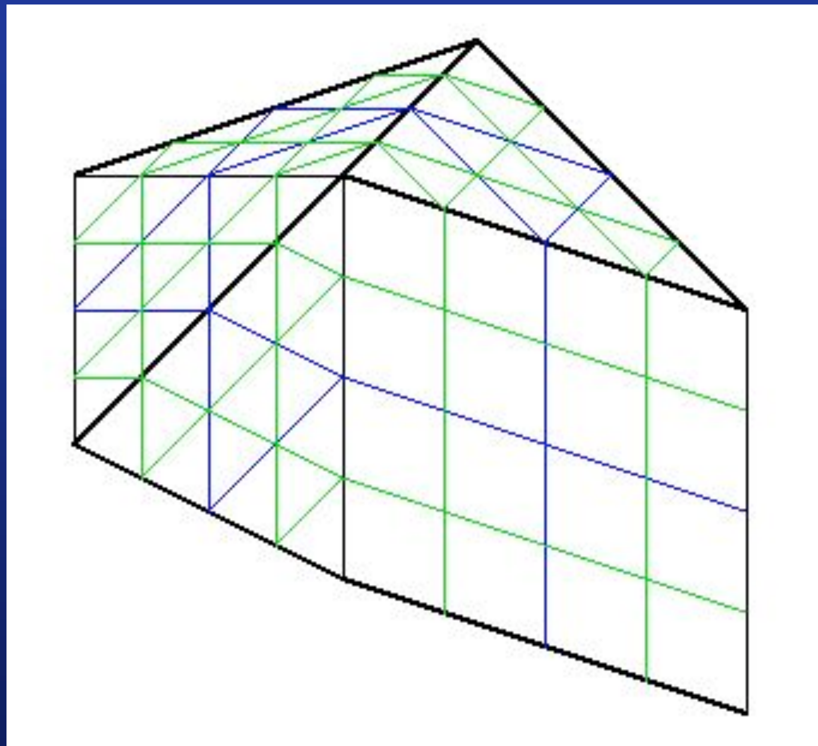
Input Grid

HHG Refinement example



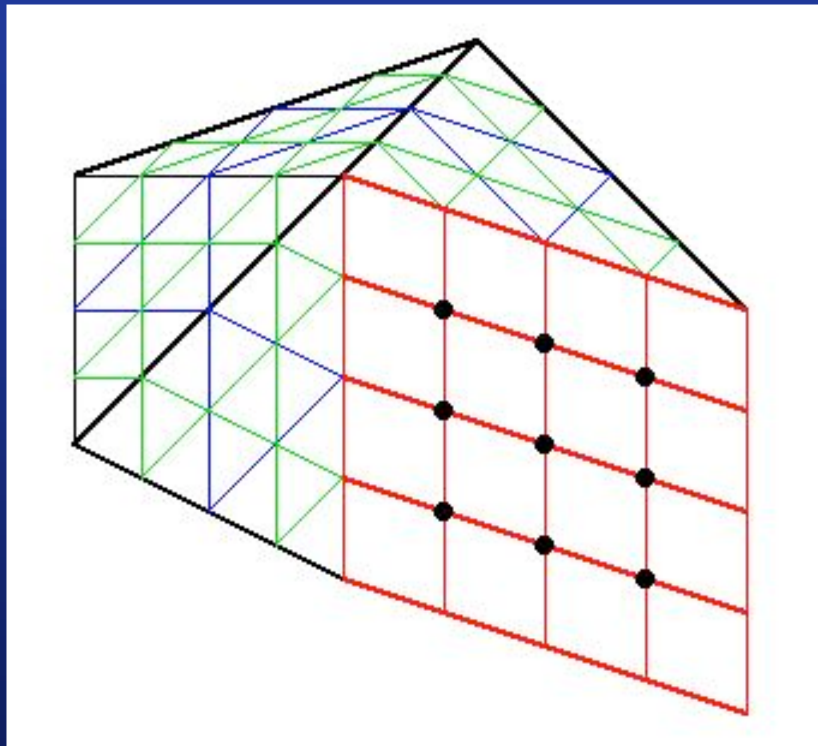
Refinement Level one

HHG Refinement example



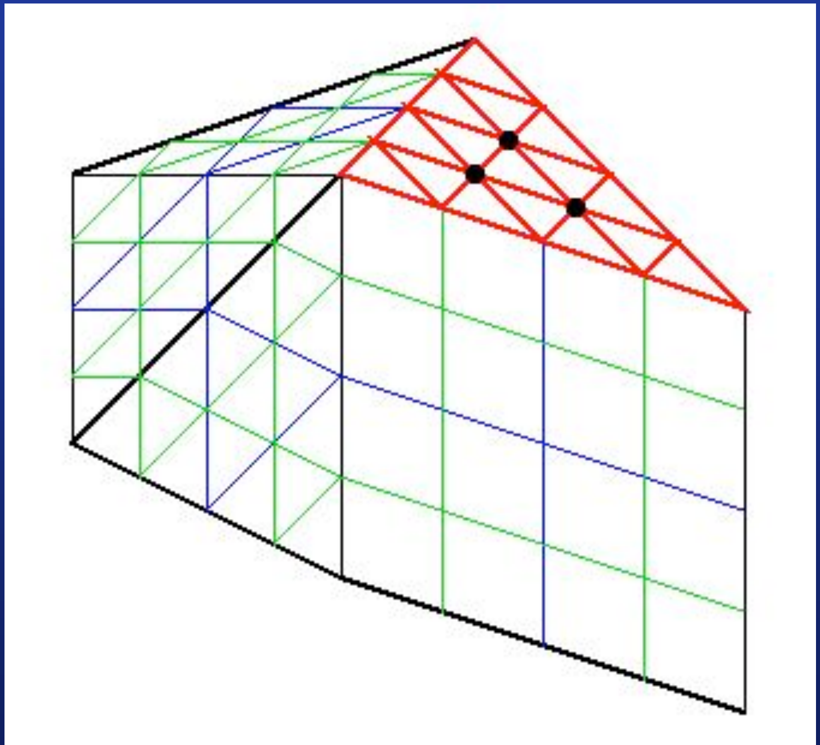
Refinement Level Two

HHG Refinement example



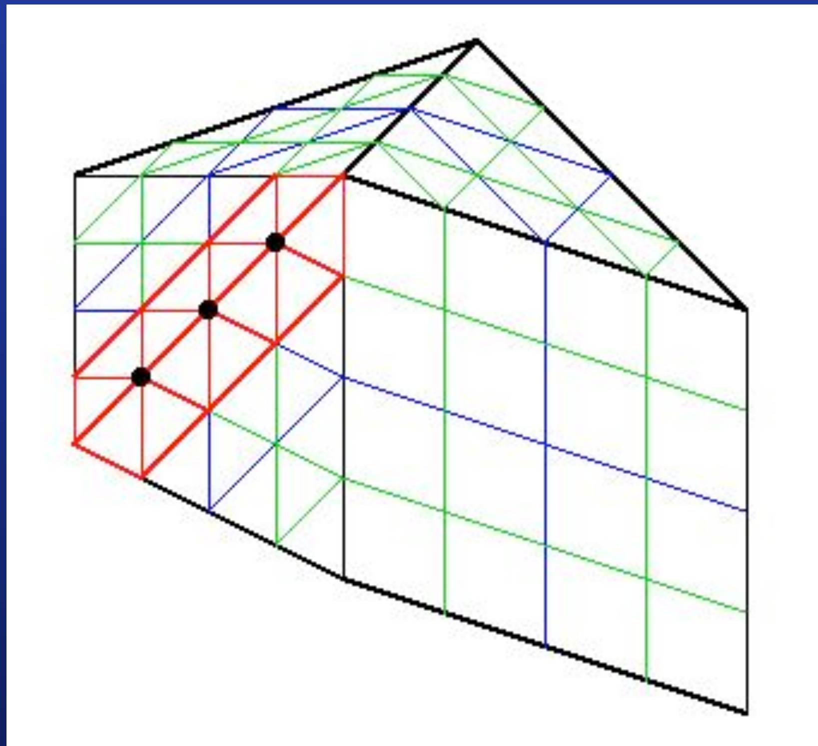
Structured Interior

HHG Refinement example



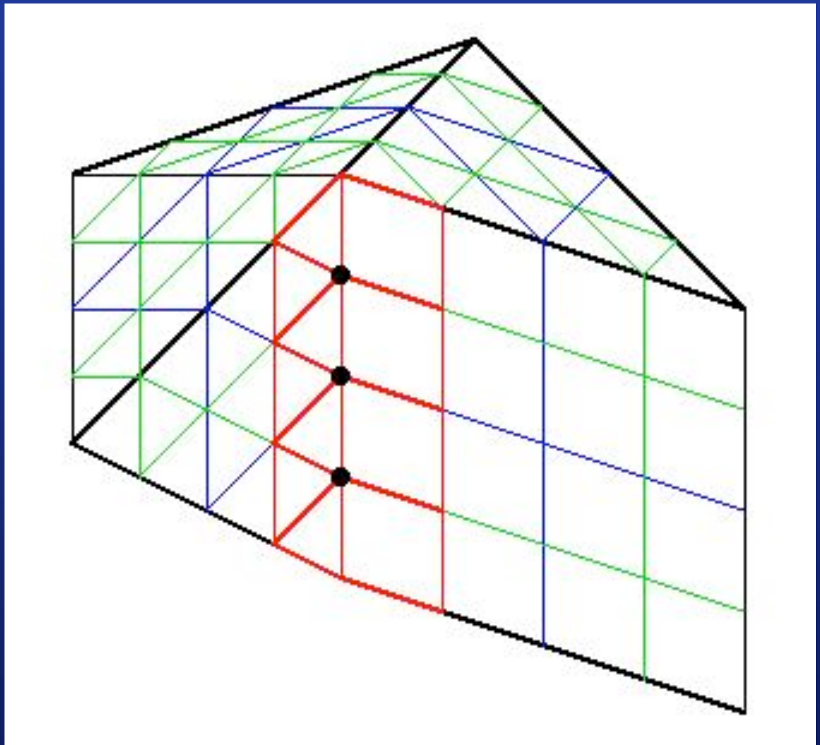
Structured Interior

HHG Refinement example



Edge Interior

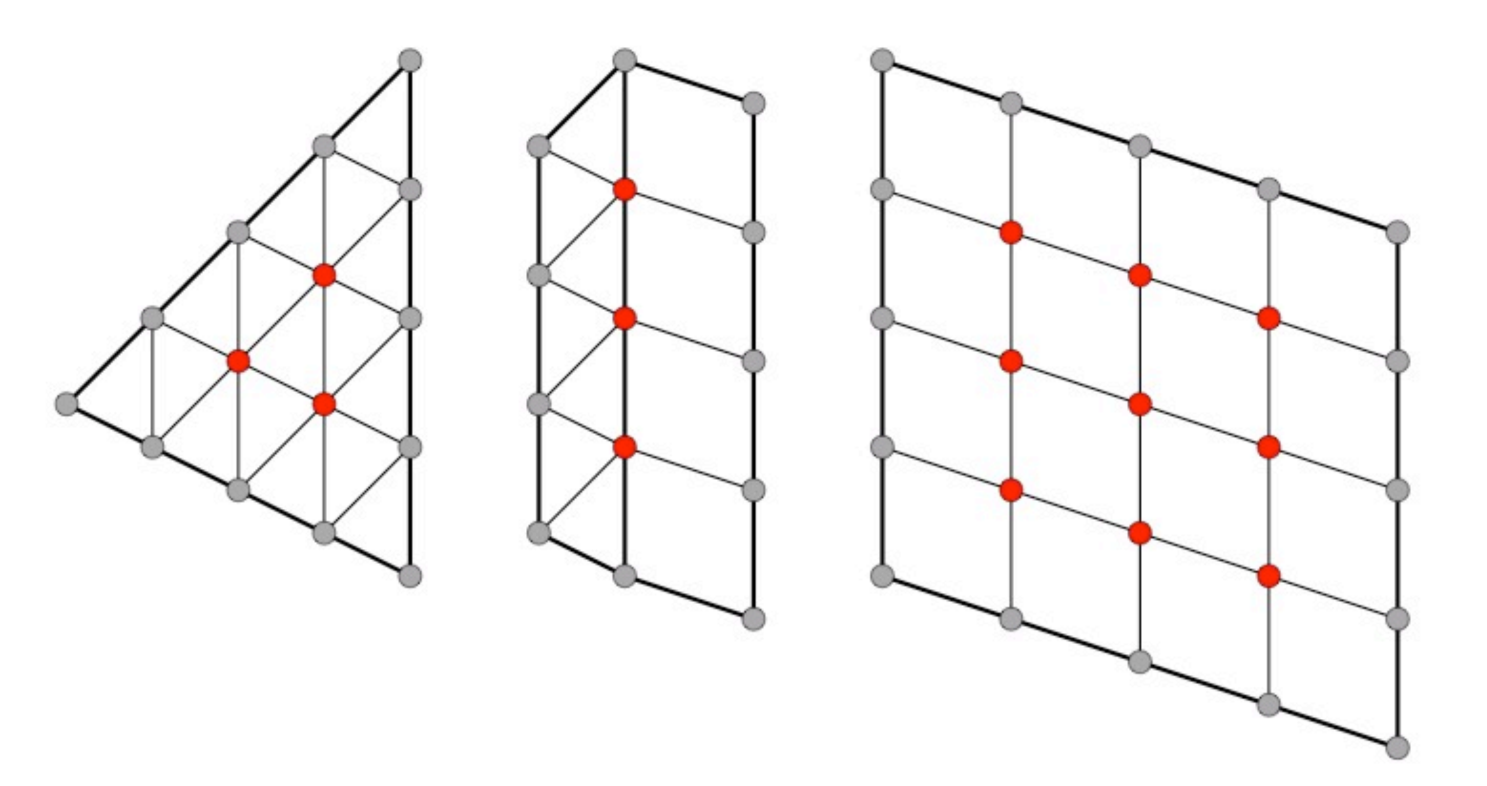
HHG Refinement example



Edge Interior

HHG for Parallelization

- Use regular HHG patches for partitioning the domain



Friedrich-Alexander-Universität Erlangen-Nürnberg

HHG Parallel Update Algorithm

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

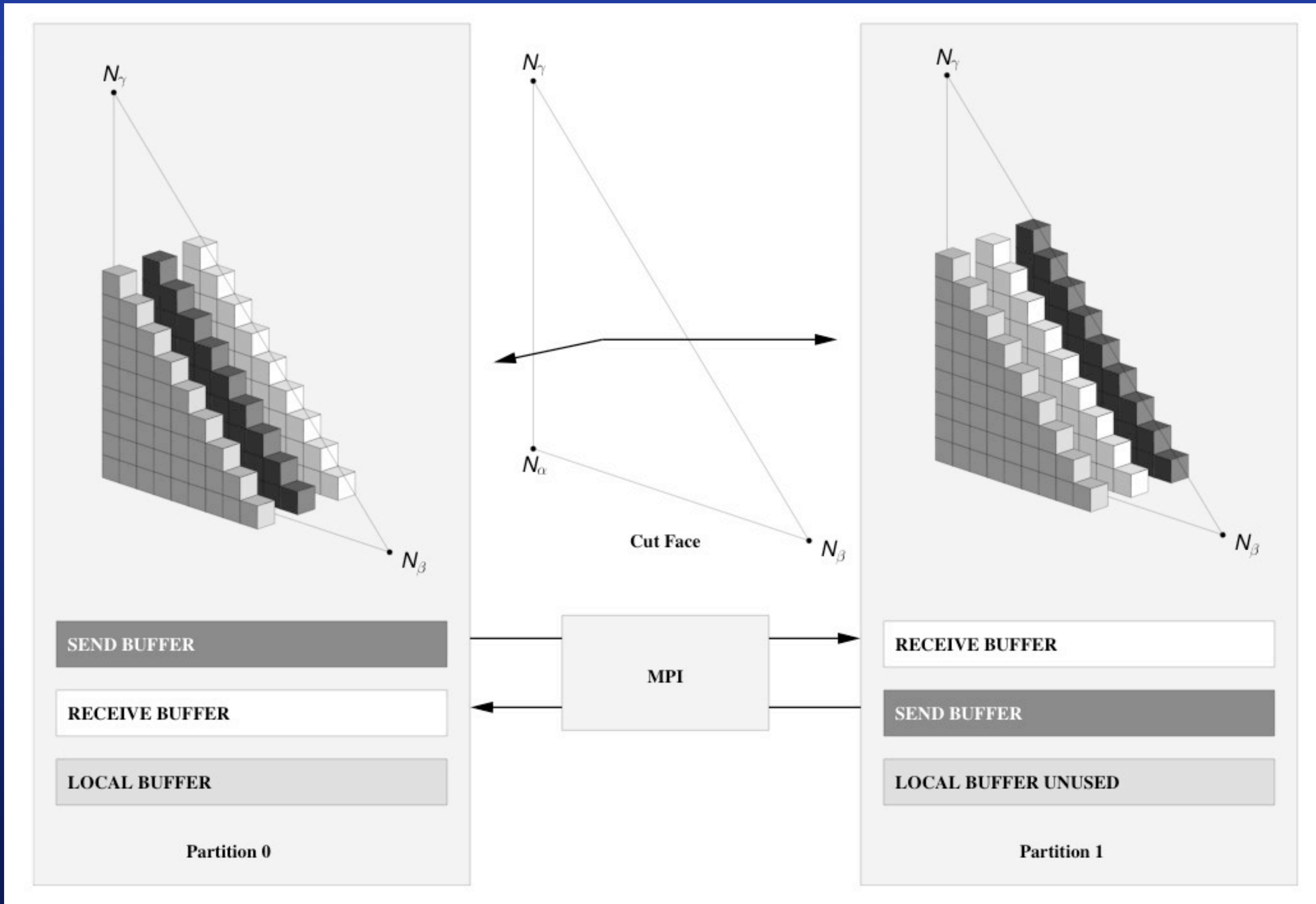
    apply operation to element

    copy to edge/vertex halos

end for

update secondary dependencies

HHG for Parallelization



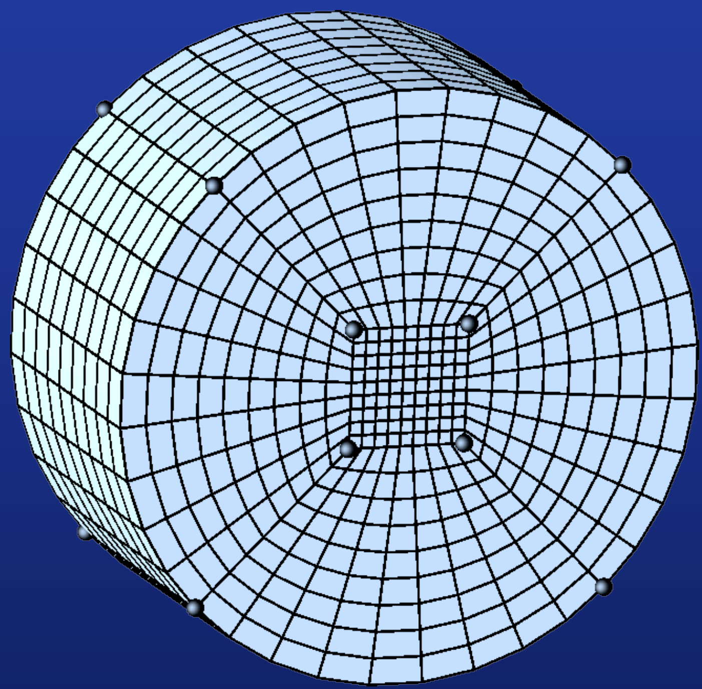
Parallel HHG - Framework Design Goals

To achieve good parallel scalability:

- Minimize latency by reducing the number of messages that must be sent
- Optimize for high bandwidth interconnects ⇒ large messages
- Avoid local copying into MPI buffers

ParExPDE

- A library for the user friendly, rapid development of numerical PDE solvers on parallel (super-)computers
- Provides a high level and intuitive user interface without compromising on efficiency
- Regularly refined hexahedral grids
- Support for deformed edges

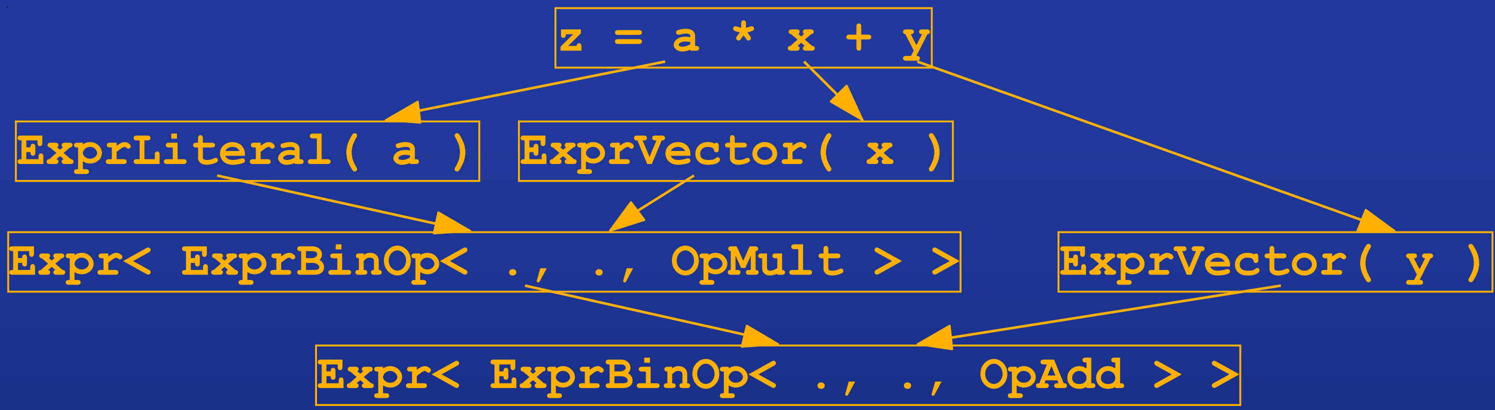


C++ Expression Templates

- Encapsulation of arithmetic expressions in a tree-like structure by using C++ template constructs
- Evaluation of expression happens at compile time
- Avoidance of unnecessary copying and generation of temporary objects

Elegance & Performance

Introduction to Expression Templates



- Evaluation of an expression:

```
template <class T>
void Vector::operator=(Expr<T>& expr) {
  for (int i=0; i<_size; i++)
    _values[i] = expr.valueAt(i);
}
```

Part II - c

Towards Scalable FE Software

Performance Results

Node Performance is Difficult! (B. Gropp)

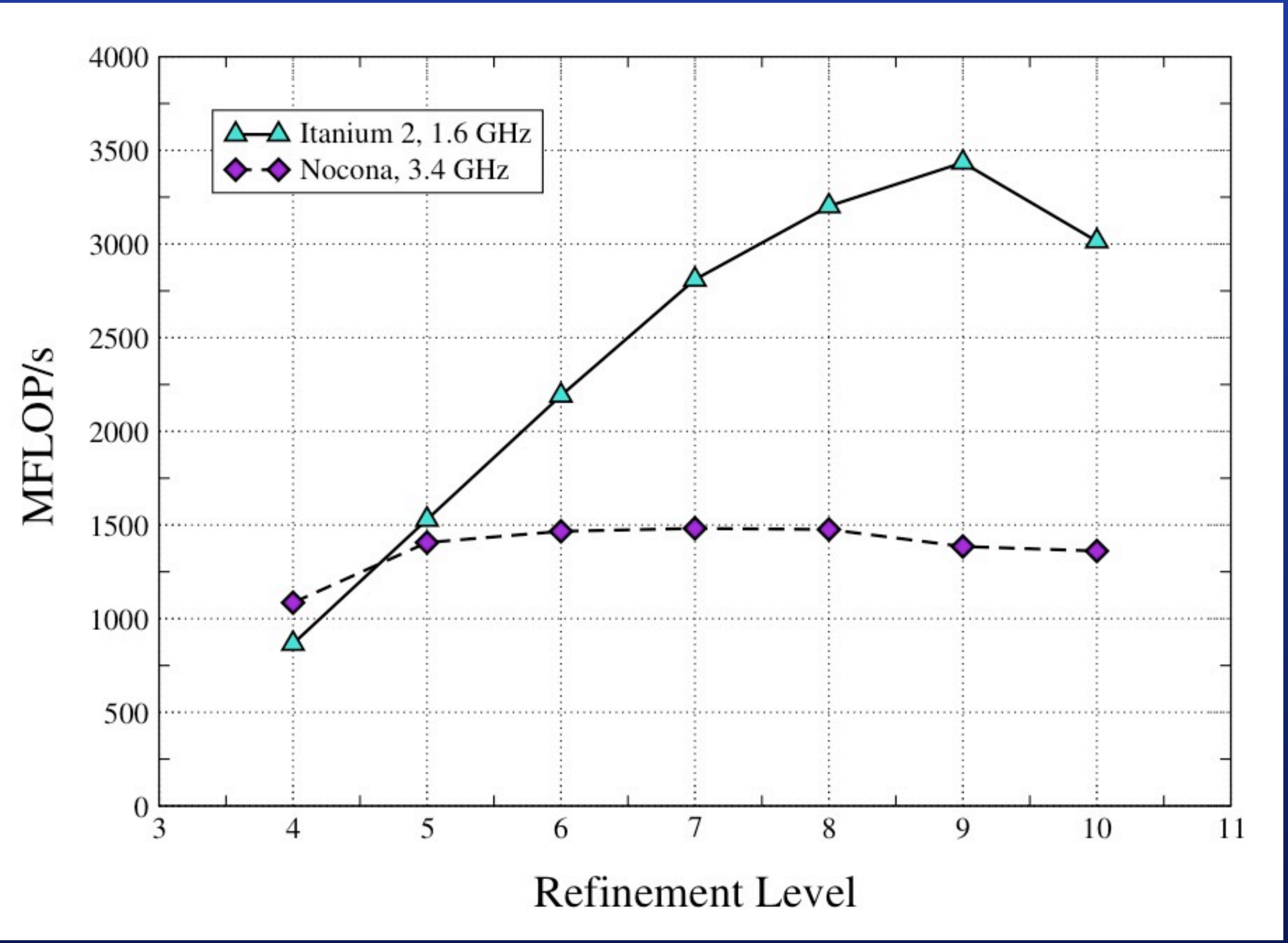
DiMe project: Cache-aware Multigrid (1996- ...)

grid size	17^3	33^3	65^3	129^3	257^3	513^3
standard	1072	1344	715	677	490	579
no blocking	2445	1417	995	1065	849	819
2x blocking	2400	1913	1312	1319	1284	1282
3x blocking	2420	2389	2167	2140	2134	2049

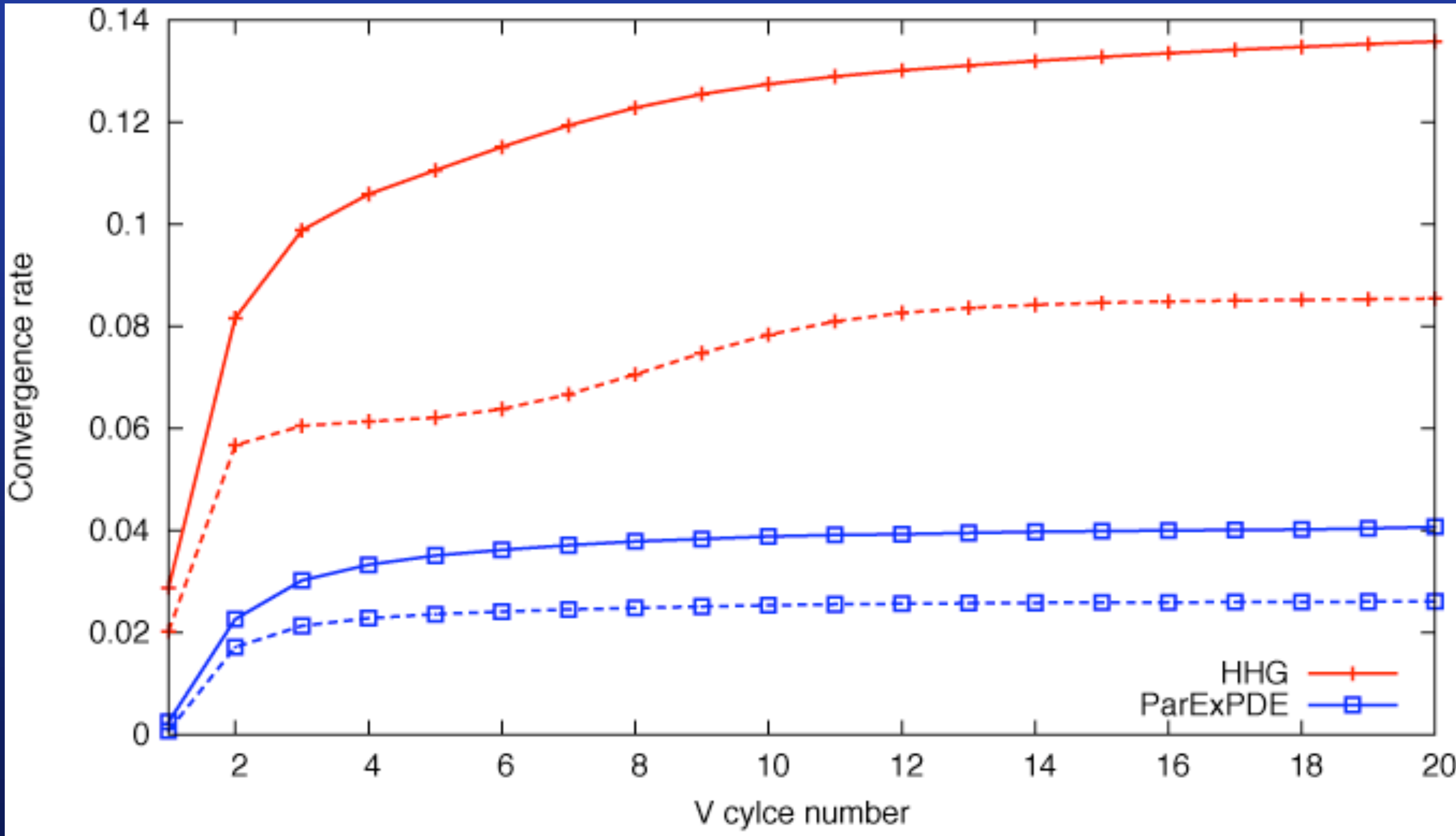
- Performance of 3D-MG-Smoother for 7-pt stencil in Mflops on Itanium 1.4 GHz
- Array Padding
- Temporal blocking - in EPIC assembly language
- Software pipelining in the extreme (M. Stürmer - J. Treibig)

Node Performance is Possible!

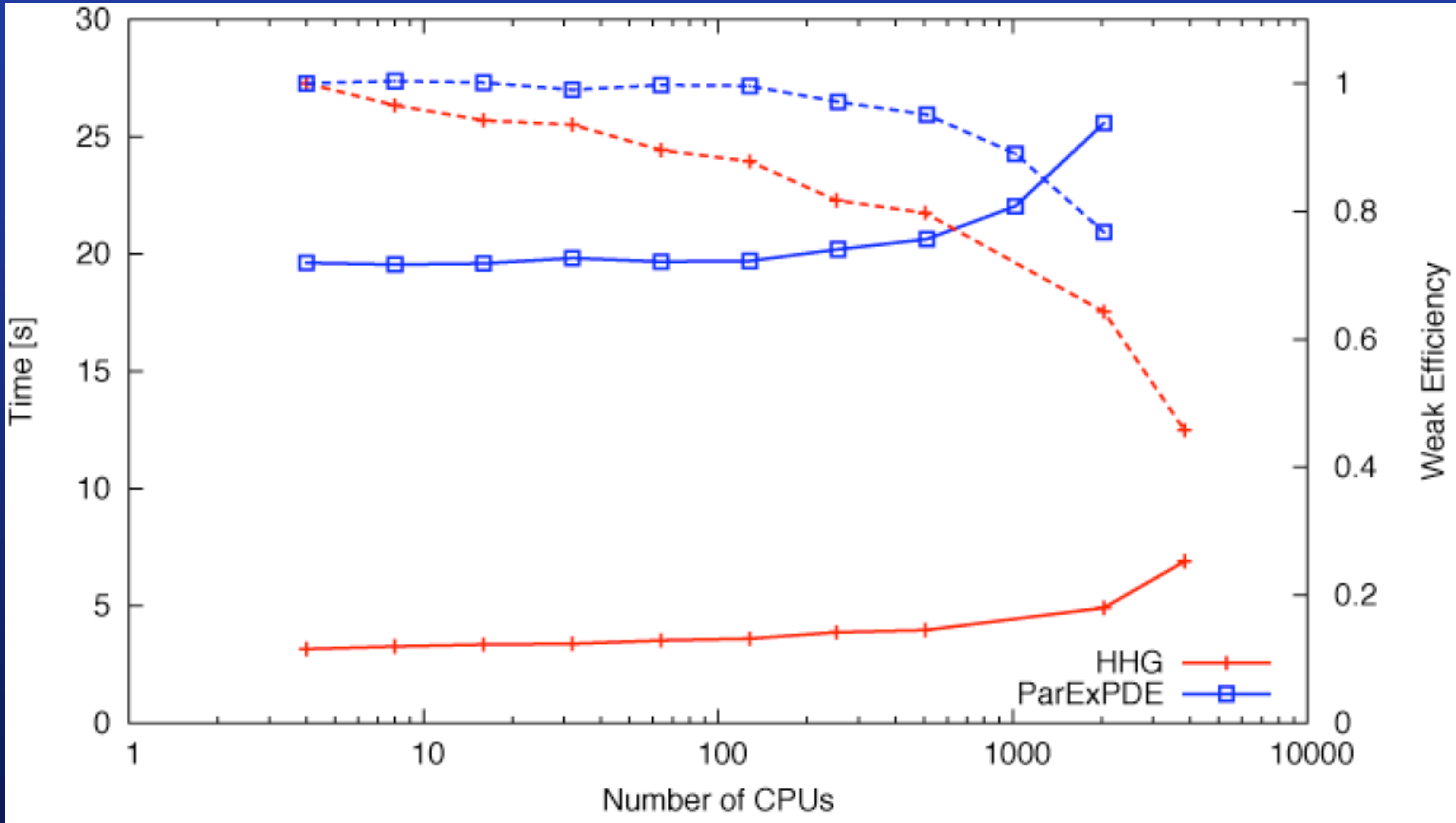
Single Processor HHG Performance on Itanium for Relaxation of a Tetrahedral Finite Element Mesh



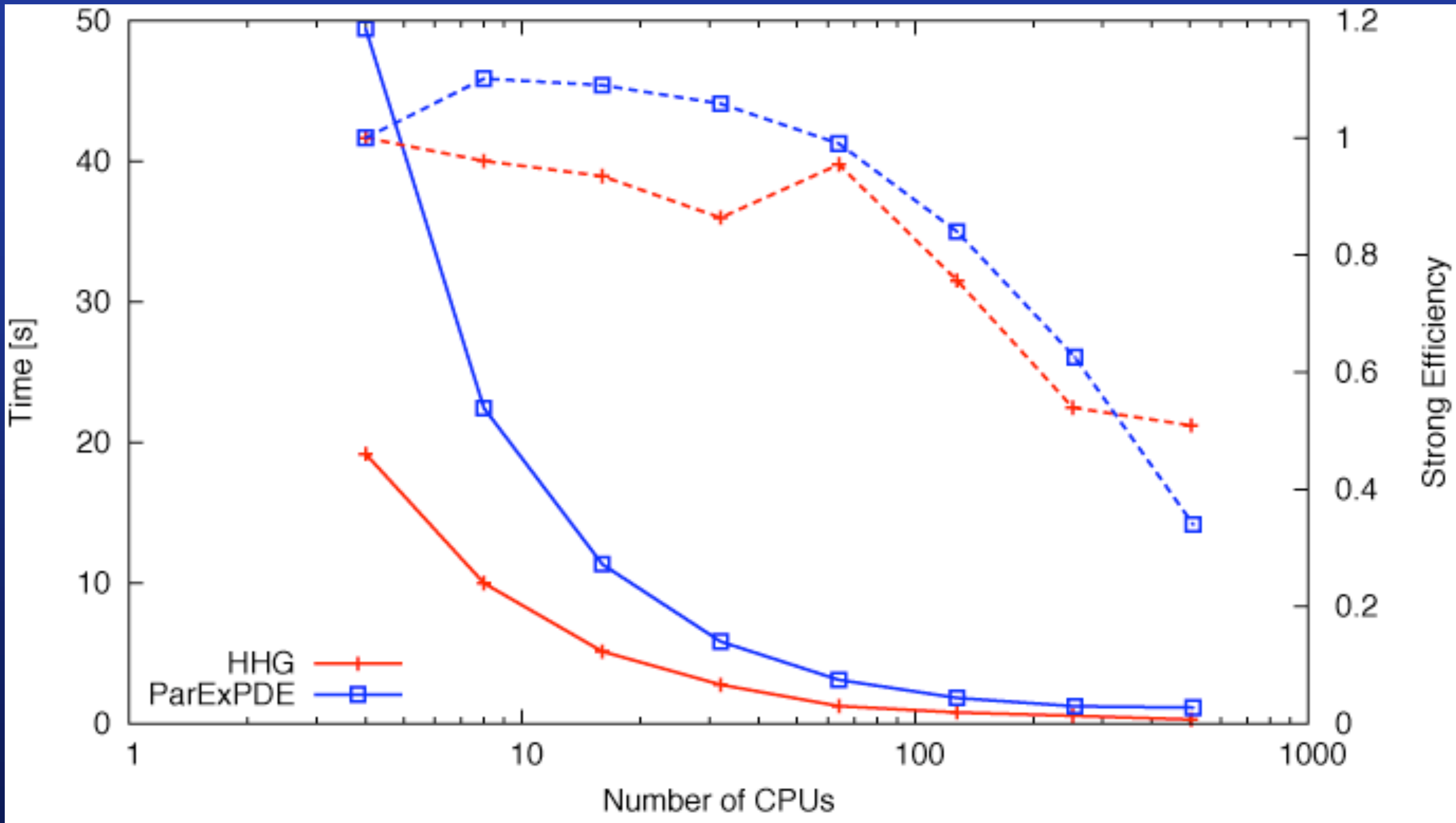
Multigrid Convergence Rates



Scaleup HHG and ParExPDE



Speedup HHG and ParExPDE



HHG: Parallel Scalability

#Procs	#DOFs x 10^6	#Els x 10^6	#Input Els	GFLOP/s	Time [s]
64	2,144	12,884	6144	100/75	68
128	4,288	25,769	12288	200/147	69
256	8,577	51,539	24576	409/270	76
512	17,167	103,079	49152	762/545	75
1024	17,167	103,079	49152	1,456/964	43

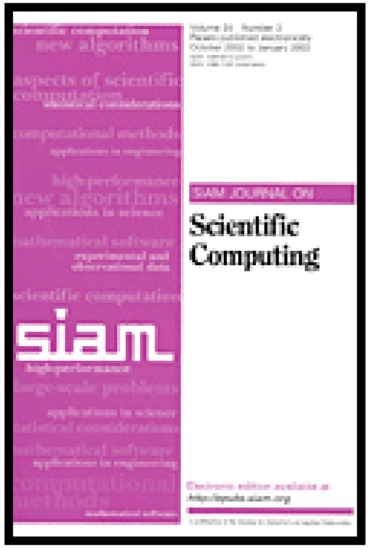
Parallel scalability of Poisson problem
discretized by tetrahedral finite elements.
Times for six V(3,3) cycles on SGI Altix (Itanium-2 1.6 GHz).

Largest problem solved to date:

1.28 x 10^{11} DOFs (45900 Input Els) on 3825 Procs: 7s per V(2,2) cycle

What can we do with $\sim 10^{10}$ - 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 intractions (gravitational/Coulomb potential) for systems with trillions of stars/atoms
- resolve the whole volume of earth‘s atmosphere with ~100m resolution (in 3D)



SIAM J. Scientific Computing

Special Issue on CS&E

http://www.siam.org/journals/sisc.php

Guest Editors-in-Chief:

Chris Johnson

David Keyes

Ulrich Ruede

Call for papers:

- Modeling techniques
- Simulation techniques
- Analysis techniques
- Tools for realistic problems

Deadline:

April 30, 2007

Guest Editorial Board: Gyan Bhanot, IBM Watson Research Center, Yorktown Hights; Rupak Biswas, NASA Ames Research Center, Moffett Field, CA; Edmond Chow, D. E. Shaw Research, LLC; Phil Colella, Lawrence Berkeley National Laboratory; Yuefan Deng, Stony Brook University, Stony Brook, NY Brookhaven National Lab; Lori Freitag Diachin, Lawrence Livermore National Laboratory; Omar Ghattas, The University of Texas at Austin, Laurence Halpern, Univ. Paris XIII; Robert Harrison, Oak Ridge National Laboratory; Bruce Hendrickson, Sandia National Laboratories; Kirk Jordan, IBM, Cambridge MA; Tammy Kolda, Sandia National Laboratories; Louis Komzsik, UGS Corp., Cypress, CA, USA; Ulrich Langer, Johann Radon Institute for Computational and Applied Mathematics (RICAM) Linz; Hans Petter Langtangen, Simula Research Laboratory, Oslo; Steven Lee, Lawrence Livermore National Laboratory; Kengo Nakajima, University of Tokyo; Aiichiro Nakano, University of Southern California, Los Angeles; Esmong G. Ng, Lawrence Berkeley National Laboratory; Stefan Turek, University of Dortmund, Germany; Andy Wathen, Oxford University; Margaret Wright, New York University; David P. Young, Boeing.

siam® Society for Industrial and Applied Mathematics

Part IV

Conclusions

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 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.

References

- B. Bergen, F. Hülsemann, U. Ruede: *Is 1.7×10^{10} unknowns the largest finite element system that can be solved today?* SuperComputing, Nov‘ 2005.
(Answer: no, it‘s 1.3×10^{11} since Dec. 2006)
- *ISC Award:* 2006 for Application Scalability.
- B. Bergen, T. Gradl, F. Hülsemann, U. Ruede: *A massively parallel multigrid method for finite elements* Computing in Science & Engineering, Vol. 8, No. 6

Acknowledgements

- Collaborators
  - In Erlangen: WTM, LSE, LSTM, LGDV, RRZE, Neurozentrum, Radiologie, etc.
  - Especially for foams: C. Körner (WTM)
  - International: Utah, Technion, Constanta, Ghent, Boulder, München, Zürich, ...
- Dissertationen Projects
  - U. Fabricius (AMG-Verfahren and SW-Engineering for parallelization)
  - C. Freundl (Parelle Expression Templates for PDE-solver)
  - J. Härtlein (Expression Templates for FE-Applications)
  - N. Thürey (LBM, free surfaces)
  - T. Pohl (Parallel LBM)
  - ... and 6 more
- 19 Diplom- /Master- Thesis
- Studien- /Bachelor- Thesis
  - Especially for Performance-Analysis/ Optimization for LBM
    - J. Wilke, K. Iglberger, S. Donath
  - ... and 23 more
- KONWIHR, DFG, NATO, BMBF
- Elitenetzwerk Bayern
  - *Bavarian Graduate School in Computational Engineering* (with TUM, since 2004)
  - Special International PhD program: *Identifikation, Optimierung und Steuerung für technische Anwendungen* (with Bayreuth and Würzburg) since Jan. 2006.

Talk is Over

Please wake up!