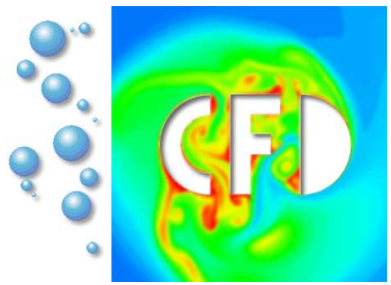
Minimally invasive integration of GPUs to improve multigrid solver performance on a cluster

Dipl.-Inform. Dominik Göddeke

dominik.goeddeke@math.uni-dortmund.de

Mathematics III: Applied Mathematics
Computer Science VII: Computer Graphics
University of Dortmund, Germany

SCSE07 – Software Issues in Computational Science and Engineering
Uppsala University, Sweden, August 15-16, 2007



Motivation

The free ride is over, scientific computing faces a paradigm shift:

physical barriers (heat, power consumption, leaking voltage)

memory wall (in particular for sparse LA problems)

applications no longer run faster automatically on newer hardware

Fine-grained parallelism addresses these problems:

multi-core commodity CPUs Cell BE processor graphics cards

HPC accelerators (e.g. ClearSpeed) reconfigurable hardware

We have to rethink the way we design algorithms for these architectures. We do not want to reimplement entire programs.

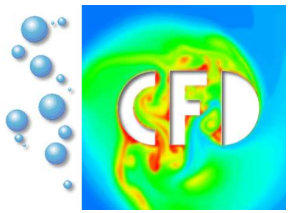
Acknowledgements

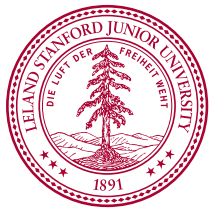
Collaborative work with:

Hilmar Wobker, Stefan Turek and the FEAST group (Dortmund)

Robert Strzodka (Stanford University, Max Planck Center)

Jamaludin Mohd-Yusof, Patrick McCormick (Los Alamos National Laboratory)

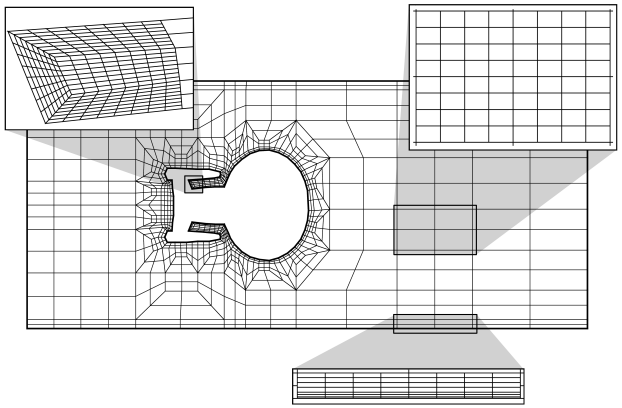


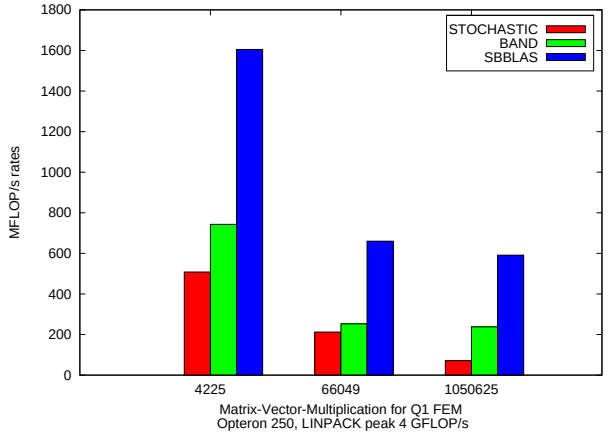


Overview

Overview

Good performance for (serial) FEM



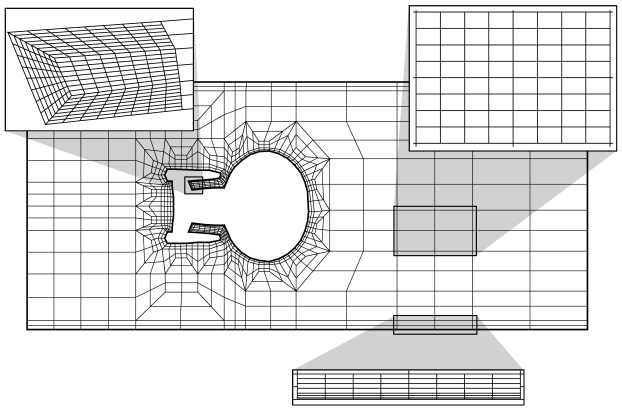


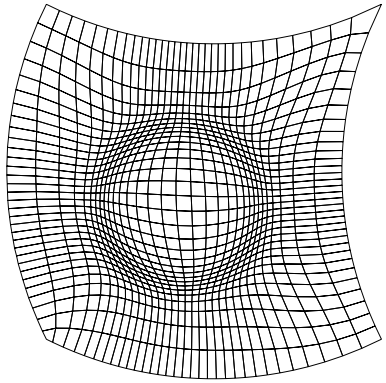
Fully adaptive grids, stochastic numbering: maximum flexibility, unstructured sparse matrices, huge performance drop for large problems

Generalised tensorproduct grids: banded matrices, better sustained performance

Exploit structured parts of the discretisation: high performance, tuned local solvers, (co-processor acceleration)

Good performance for (parallel) FEM





FEAST, MPI-based Finite Element package for PDEs

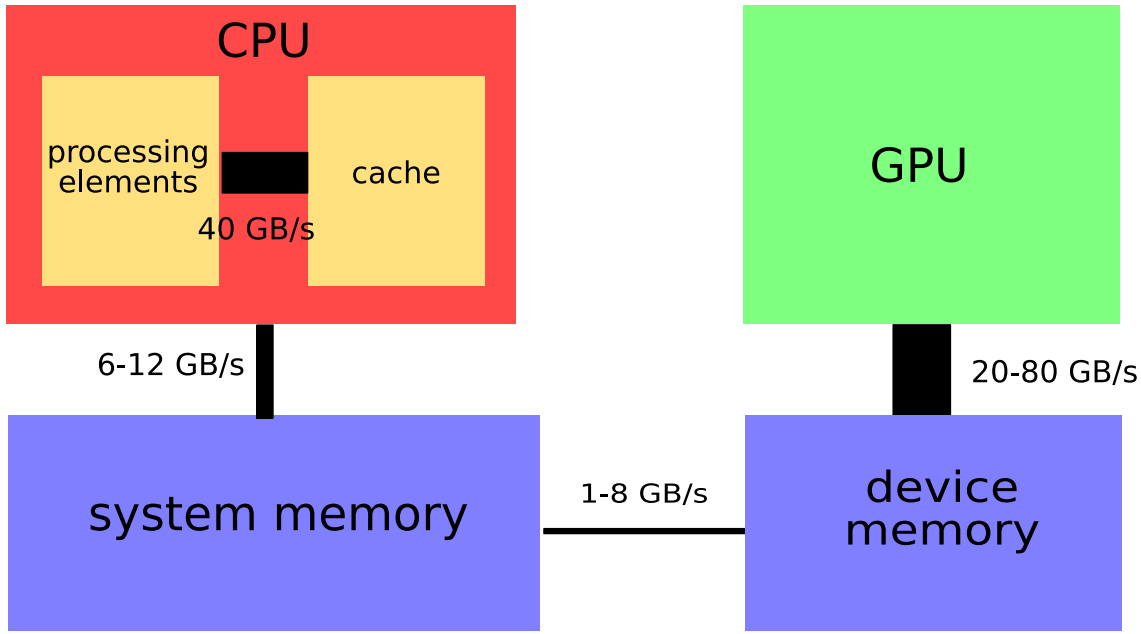
Global discretisation with unstructured collection of tensorproduct macros

Generalised DD/MG approach, parallel recursive MG smoothers with only implicit overlap instead of (standard) data-parallel multigrid.

Hide anisotropies locally via recursive global/local MG:
numerical robustness, decoupled local work, scalability

Bandwidth, bandwidth, bandwidth

FEM codes are 95% memory bound,
bandwidth is the crucial factor for performance.



GPUs offer superior bandwidth, are readily available, fast, cheap, ..., in short, seem like an ideal candidate to improve commodity based clusters.

Goals

Goal of this project: Include GPUs into an existing FE package...

...without changes to application code built on top of the package,

...without fundamental refactoring of the package,

...without sacrificing either functionality or accuracy,

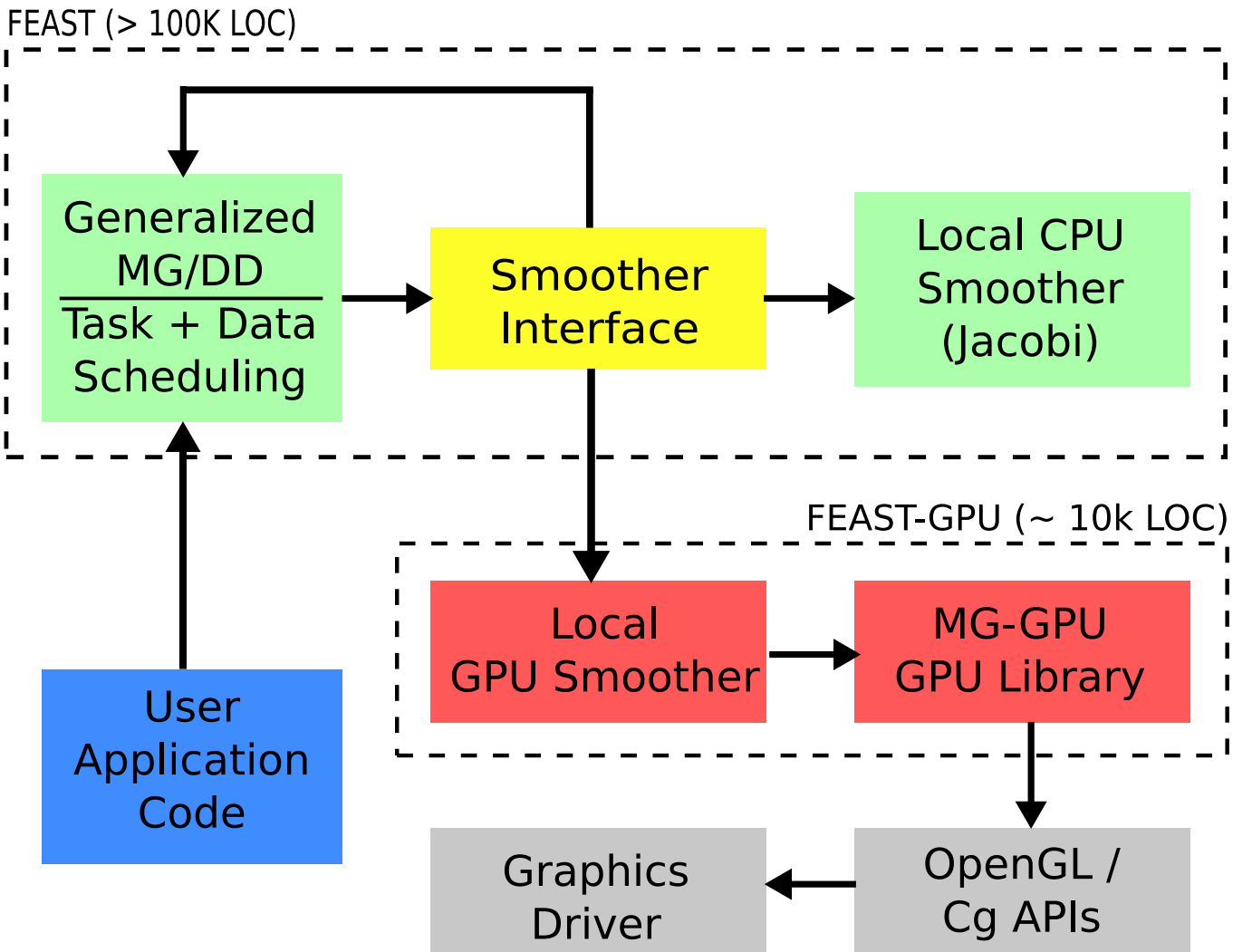
...but with noteworthy speedups,

...a reasonable amount of generality wrt. other co-processors,

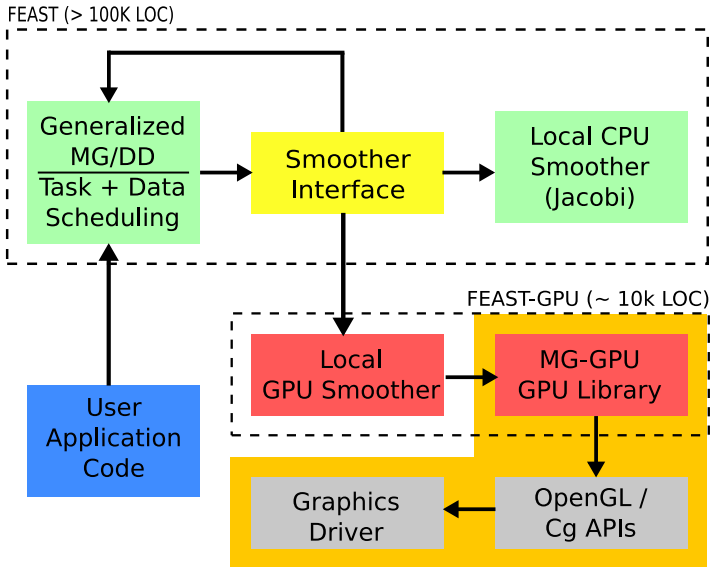
...and additional benefits in terms of space/power/etc.

Overview

Integration overview



GPU library



GPU library:

offers a lean and efficient interface to the outer package,

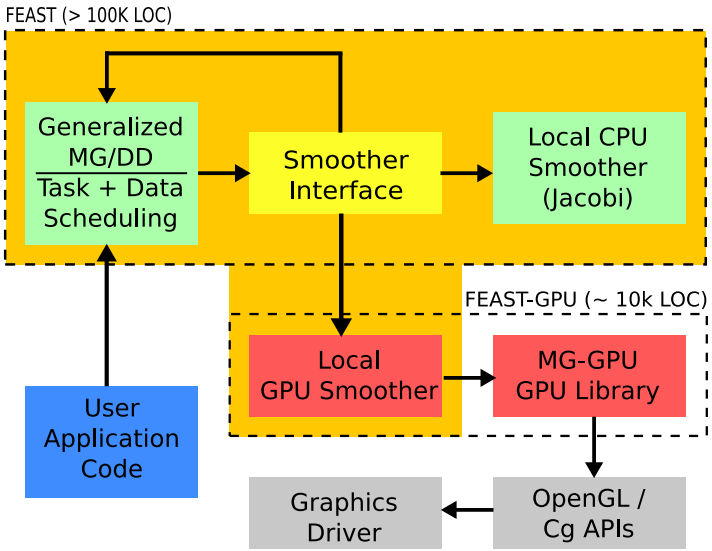
hides all hardware-dependent details,

provides tuned hardware adaptions of multigrid solvers.

Advantages: Interface is independent of the actual hardware underneath. Backends can be implemented, tested and tuned independently and standalone.

Main challenge: Efficient data transfers interleaved with format conversions to reduce the impact of the PCIe bottleneck.

GPU smoother



Hardware acceleration:

Integrated on the node level, as a new type of **local smoother**,

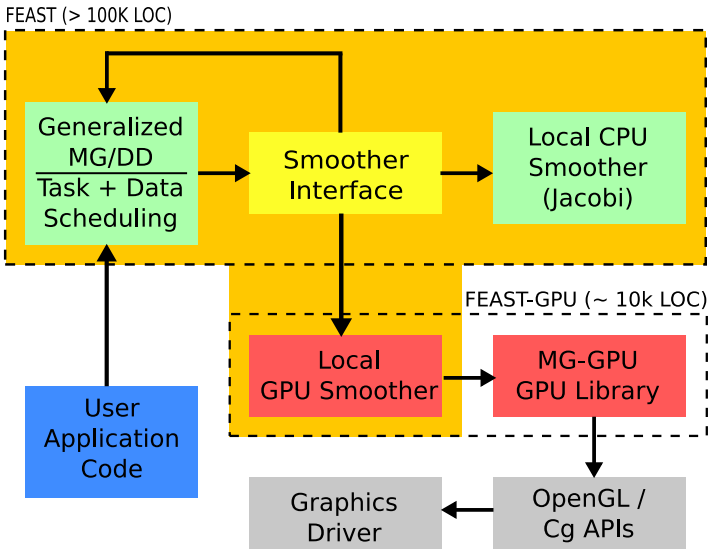
by implementing the existing local interfaces.

Global communication framework remains unchanged.

Decoupled approach: Sufficient amount of local work between MPI communication makes hardware acceleration feasible despite the transfer bottleneck.

Effort: Less than 300 lines of wrapper code.

Tradeoffs in parallelism



Coarse-grained parallelism:

Heterogeneous task scheduling based on problem size, expected run-time, smoother capabilities.

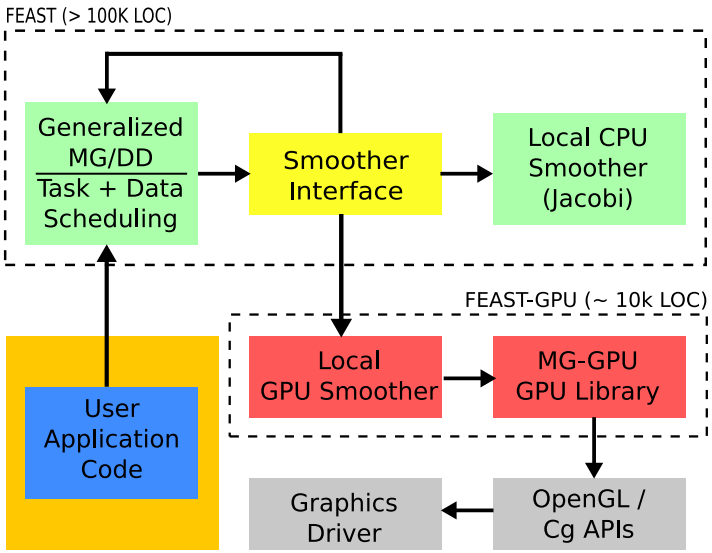
Careful assignment of MPI jobs to existing resources.

Approx. 700 lines of code.

Fine-grained parallism: Outer multigrid generates small sub-problems on lower levels, transfer overhead prevents hardware acceleration: Provide CPU fallback solver.

Rule of thumb: small tasks to the CPU, large tasks to the GPU

Applications



Application perspective:

No changes to application code.

Acceleration enabled by simple change in parameter files.

If hardware acceleration is unavailable, FEAST solvers provide software fallback.

```
SOLVER: BICG,256,REL:1.0E-6
  PREC: BGS,1.0
    SOLVER: MG,1,REL:1.0E-99,V,1,1,GLOBAL
      SMOOTHER: 0.7
        SOLVER: MG,1,REL:1.0E-99,F,4,4,MACRO
          SMOOTHER: 0.7
              LSMOOTHER: JACOBI
           COARSE: CG,4096,REL:1.0E-6,MACRO
            PREC: 1.0
              LSMOOTHER: JACOBI
       COARSE: LU_DECOMP,GLOBAL
```

```
SOLVER: BICG,256,REL:1.0E-6
  PREC: BGS,1.0
    SOLVER: MG,1,REL:1.0E-99,V,1,1,GLOBAL
      SMOOTHER: 0.7

        LSMOOTHER: MG-GPU,1,REL:1E-99,F,4,4,JACOBI,0.7,
           COARSE: CG,4096,REL:1.0E-6,JACOBI,
           CPU: 4,CG,4096,REL:1.0E-6,JACOBI

       COARSE: LU_DECOMP,GLOBAL
```

Overview

Application: Structural Mechanics

Fundamental model problem: elastic, compressible material (e.g. steel) exposed to small deformations in a static loading process.

Linearized strain tensor and kinematic relation, Hooke's material law:
Lamé equation

$$\begin{aligned} -2\mu \operatorname{div} \boldsymbol{\varepsilon}(\mathbf{u}) - \lambda \operatorname{grad} \operatorname{div} \mathbf{u} &= \mathbf{f}, & \mathbf{x} &\in \Omega \\ \mathbf{u} &= \mathbf{g}, & \mathbf{x} &\in \Gamma_D \\ \boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} &= \mathbf{t}, & \mathbf{x} &\in \Gamma_N \end{aligned}$$

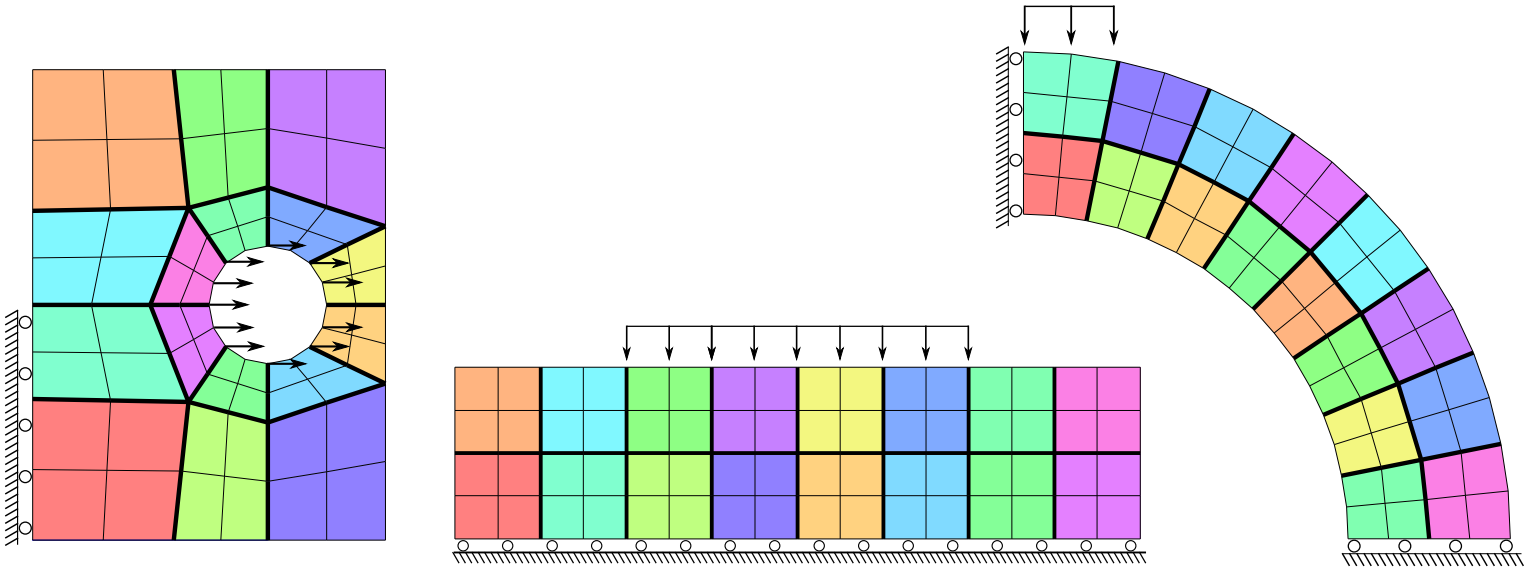
FEM discretisation, separate displacement ordering:

$$\begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}$$

Solver: Global Krylov subspace method, block-preconditioned (Gauss-Seidel) by treating $\mathbf{K}_{11}$ and $\mathbf{K}_{22}$ with FEAST solvers.

Test cases

Coarse grids, partitions, boundary conditions:



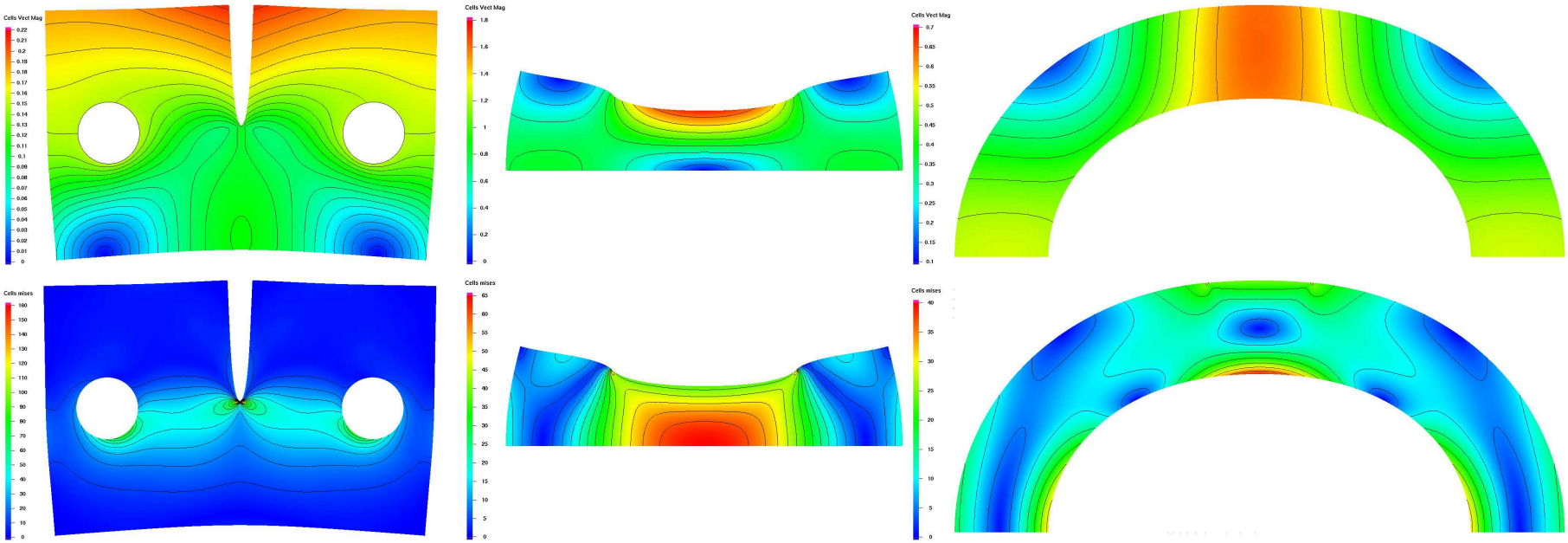
CRACK simulated test environment for assessing material properties

BLOCK standard benchmark configuration in CSM

PIPE circular cross-section of a pipe clamped in a bench vise

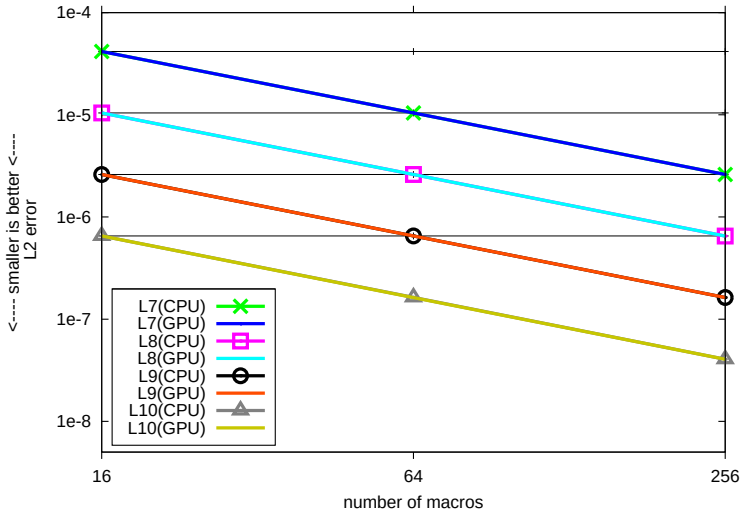
Test cases

Computed displacements and van Mises stresses:



Accuracy

Test procedure:
Analytically prescribed displacements, L_2 error for various refinement levels L. Each macro has $(2^L + 1)^2$ DOFs.



Vertical: Error reduction by a factor of 4 (h^2) when increasing L.
Horizontal: Independence of macro distribution and refinement level: 16 macros on $L9$ are equivalent to 64 macros on $L8$.
Diagonal: Quadrupling the number of macros on the same level gives same error reduction as refining.

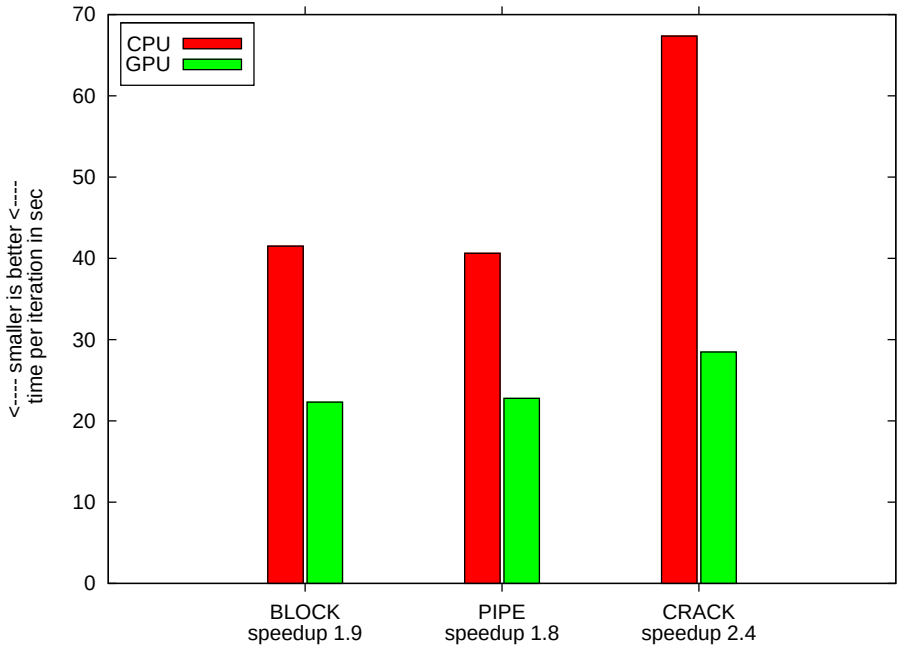
Identical convergence and computed solutions with and without hardware acceleration despite restriction to single precision for smoothing

Speedup

Hardware configuration:
16 nodes, Infiniband,
128 Mi DOFs total

2x Intel EM64T per node, 3.4 GHz, 6.4 GB/s shared bandwidth

1x NVIDIA Quadro FX4500 per node, 33.6GB/s bandwidth, 512 MB, PCIe



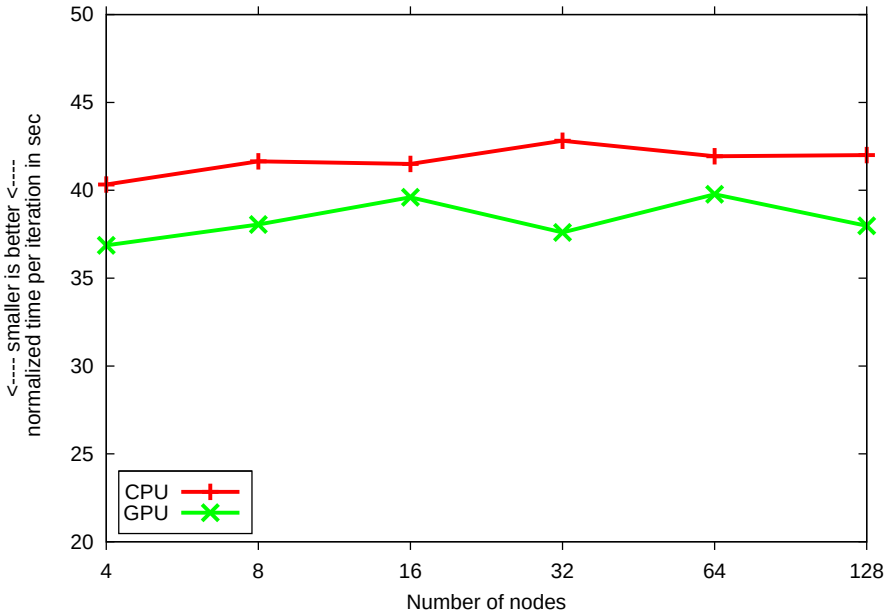
Speedup of 1.8 – 2.4 for BLOCK, PIPE and CRACK

Scalability

Hardware configuration:
4–128 nodes, Infiniband,
8 Mi DOFs per node

2x Intel EM64T per node, 3.4 GHz, 6.4 GB/s shared bandwidth

1x NVIDIA Quadro FX1400 per node, 19.2GB/s bandwidth, 128 MB, PCIe



Good weak scalability for BLOCK configuration

Overview

Summary

Goal of this project: Include GPUs into an existing FE package...

...without changes to application code built on top of the package,
ok

...without fundamental refactoring of the package,
only 1% of the code base has been touched

...without sacrificing either functionality or accuracy,
identical L_2 errors and convergence behaviour

...but with noteworthy speedups,
up to 2.4x for a reasonably challenging application

...a reasonable amount of generality wrt. other co-processors,
in progress

...and additional benefits in terms of space/power/etc.
promising preliminary results

Further details

Dominik Göddeke
`http://www.mathematik.uni-dortmund.de/~goeddeke`

Hilmar Wobker, Stefan Turek and the FEAST group (Dortmund)

Robert Strzodka (Stanford University, Max Planck Center)

Jamaludin Mohd-Yusof, Patrick McCormick (Los Alamos National Laboratory)