CO-PROCESSOR ACCELERATION OF AN UNMODIFIED PARALLEL STRUCTURAL MECHANICS CODE WITH FEAST-GPU

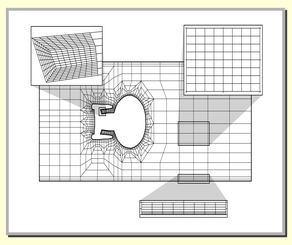
Dominik Göddeke
Hilmar Wobker, Stefan Turek
Applied Mathematics, University of Dortmund
dominik.goeddeke@math.uni-dortmund.de

Robert Strzodka
Max Planck Center, MPII
strzodka@mpi-inf.mpg.de

Jamaludin Mohd-Yusof
Patrick McCormick
Los Alamos National Laboratory
jamal|pat@lanl.gov

FEM package: FEAST

Existing MPI-based Finite Element package with more than 100,000 lines of code.



Global partition into unstructured collection of macros, refined independently via generalized tensor-product meshes. Generalized DD/MG solvers: parallel MG smoothed by patchwise MG, implicit overlap.

Exploit structured parts:
high performance, co-processor acceleration
Hide anisotropies locally:
numerical robustness, (weak) scalability

Why Graphics Processor Units (GPUs)?

Scientific computing paradigm shift: Fine-grained parallelism addresses thermal constraints, frequency and power scaling, and the memory wall.

multicore CPUs (commodity based clusters)
Cell (heterogeneity on a chip)
GPUs (> 128 parallel cores, 1000s of threads)

GPUs are very attractive for commodity based clusters: fast, readily available, excellent price/performance ratio, superior bandwidth.

Use GPUs as co-processors!

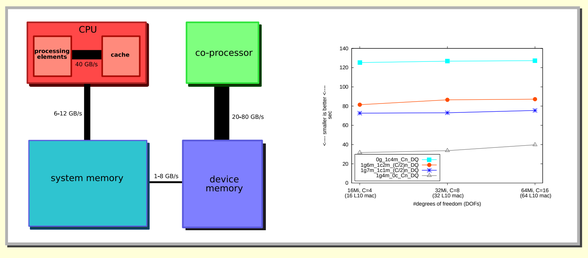
Common concerns:

Bus between host and co-processor can limit performance: Perform as much independent local work as possible between global synchronizations [1].

GPUs only support single precision: Apply a mixed precision iterative refinement scheme [2].

Importance of bandwidth

FEAST is 95% memory bound on a single node.

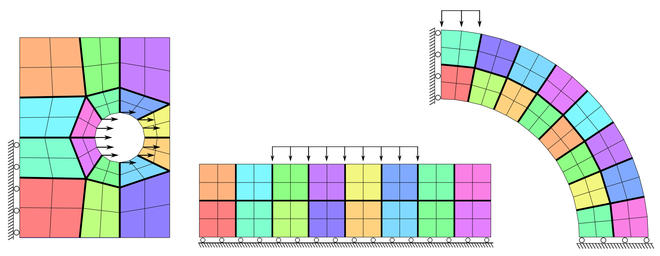


Example: DQ cluster, LANL
2x Xeon EM64T (3.4 GHz, 6.4 GB/s shared bandwidth),
1x Quadro FX4500 (strong GPUs, 512MB, 33.6 GB/s),
or 1x Quadro FX1400 (weak GPUs, 128MB, 19.2 GB/s),

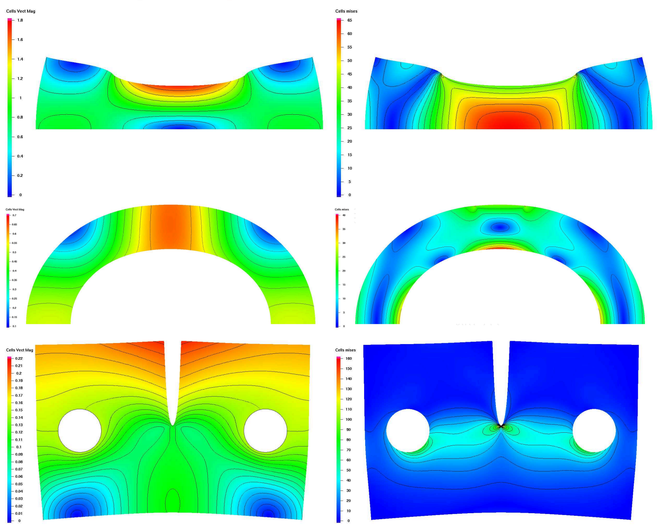
GPUs increase overall system bandwidth by 4–6x, of which 2–3x translates into application speedup.

Test configurations

Coarse grids, partitions, boundary conditions

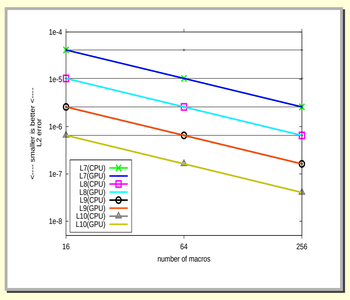


Computed displacements and van Mises stresses:



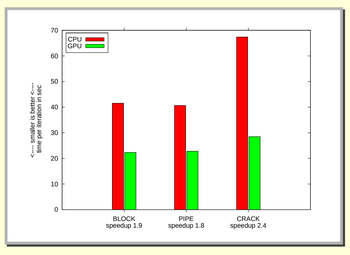
Results

Identical L_2 errors despite single precision smoothing:



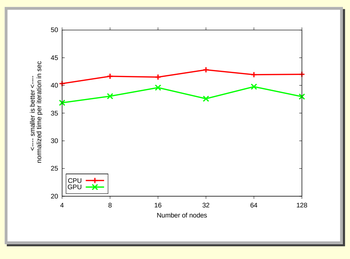
Vertical: L_2 error reduction by a factor of 4 (h^2) per refinement level.
Horizontal: Independence of macro distribution and refinement level: 16 macros on $L9$ are equivalent to 64 macros on $L8$.
Diagonal: Quadrupling the number of macros on the same level gives same error reduction as refining.

Speedup of 1.6x–2.4x for 2x Xeon vs. 1x Quadro 4500:



16 nodes, Infiniband, strong GPUs,
128 Mi DOFs per configuration

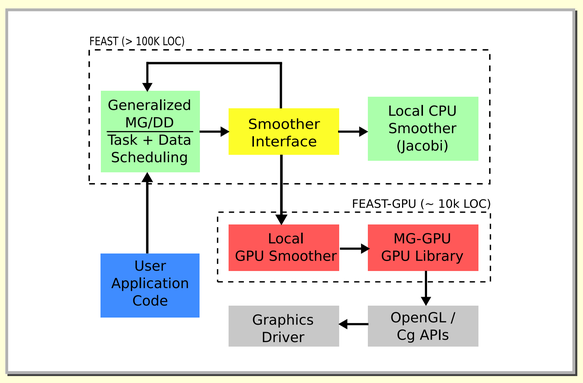
Good weak scalability for 2x Xeon vs. 1x Quadro 1400:



4–128 nodes with Infiniband, weak GPUs,
8 Mi DOFs per node (1 Bi max)

Additional benefits:
Additional benefits in metrics as power/performance, price/performance, performance/dollar etc.

Co-processor integration: FEAST-GPU



Hardware acceleration integrated on the node level: GPUs accelerate local smoothing for the global, parallel solver. Each type of hardware performs the tasks it is best suited for: The GPU executes the MG smoother, the CPU compensates for low precision and is responsible for communication and synchronization of the parallel solvers.

All local smoothers implement the same interface, enabling hardware acceleration through simple changes in configuration files.

This decoupled approach results in a sufficient amount of local work between communication, both between CPU and GPU in a node and between nodes using MPI.

Less than 1000 lines of new kernel code.
No changes to applications on top of FEAST.

More details: *Using GPUs to improve multigrid solver performance on a cluster* [1].

CSM application: FEAST-SOLID

Fundamental model problem: elastic, compressible material, exposed to small deformations in a static loading process.

Use linearized strain tensor and kinematic relation; apply Hooke's law for isotropic elastic material: Lamé equation for the unknown displacements $\boldsymbol{u}$:

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

FEM discretization and 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 the scalar blocks $\mathbf{K}_{11}$ and $\mathbf{K}_{22}$ with FEAST solvers.

References

[1] Dominik Göddeke, Robert Strzodka, Jamal Mohd-Yusof, Patrick McCormick, Hilmar Wobker, Christian Becker, and Stefan Turek. Using GPUs to improve multigrid solver performance on a cluster. *accepted for publication in the International Journal of Computational Science and Engineering*, 2008.

[2] Dominik Göddeke, Robert Strzodka, and Stefan Turek. Performance and accuracy of hardware-oriented native-, emulated- and mixed-precision solvers in FEM simulations. *International Journal of Parallel, Emergent and Distributed Systems*, 22(4):221–256, August 2007.