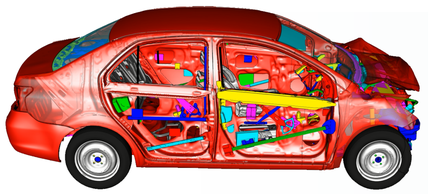
Optimization and Highly Parallel Implementation of Domain Decomposition Based Algorithms

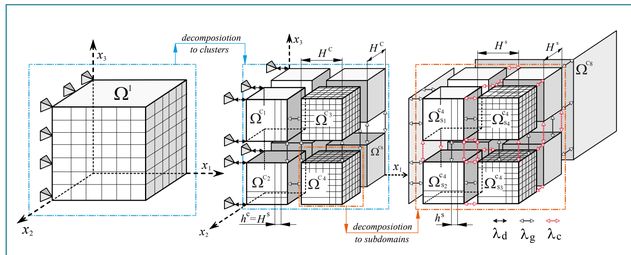


industry.it4i.cz
www.it4i.cz
am.vsb.cz

IT4Innovations national supercomputing center

FETI and HFETI Solvers

- C++ implementation based on Intel MKL sparse and dense BLAS routines and MKL version of PARDISO sparse direct solver
- Parallelization tools and strategies
  - hybrid parallelization for multisocket, multicore comp. nodes - enables over-subscription of cores with multiple subdoma ins
  - distributed memory parallelization - use of MPI 3.0 non-blocking collective operations for global reductions - Intel MPI 5.0
  - shared memory parallelization using Intel Cilk+ - enables parallel reduction using custom reduce operations on C++ classes
  - ready to use MIC accelerators

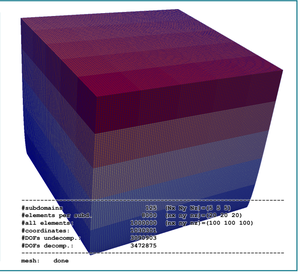


3D CUBE - HFETI elasticity benchmark

- FETI - decomposition into subdomains (Hc)
- HFETI - decomposition into clusters (Hc) and subdomains (Hs)

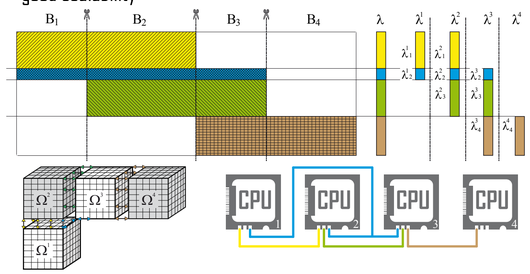
3D CUBE - Large Scale FETI Benchmark

- highly parallel generator scales up to thousands of subdomains
- large subdomains from 120,000 to 160,000 DOFs
- sparse direct solver uses most of the memory and CPU time (99%)
- no preconditioner



Reduction of global communication for (H)FETI

- each MPI process iterates only over lambdas associated with given subdomain
  - FETI – λ required for multiplication with the restriction of B matrix to given subdomain
  - HFETI – λ required for multiplication with the restriction of Bl matrix to given subdomain
- global update of vector λ becomes only nearest the neighbor type of communication with good scalability



Hiding latencies in Krylov Solvers

Preconditioned pipelined Conjuagent Gradient (CG) algorithm

1: $r_0 := b - Ax_0$; $u_0 := M^{-1}r_0$; $w_0 := Au_0$
2: **for** $i = 0,\ldots..$ **do**
3: $\gamma_i := \langle r_i, u_i \rangle$ — Dot-product - global communication - scales as log(P)
4: $\delta := \langle w_i, u_i \rangle$
5: $m_i := M^{-1}w_i$ — Sparse Matrix-Vector product - nearest neighbor comm., good scaling
6: $n_i := Am_i$
7: **if** $i > 0$ **then**
8: $\beta_i := \gamma_i/\gamma_{i-1}$; $\alpha_i := \gamma_i/(\delta - \beta_i\gamma_i/\alpha_{i-1})$
9: **else**
10: $\beta_i := 0$; $\alpha_i := \gamma_i/\delta$
11: **end if**
12: $z_i := n_i + \beta_i z_{i-1}$
13: $q_i := m_i + \beta_i q_{i-1}$
14: $s_i := w_i + \beta_i s_{i-1}$
15: $p_i := u_i + \beta_i p_{i-1}$
16: $x_{i+1} := x_i + \alpha_i p_i$

Scalar vector multiplication, vector-vector addition no communication

P. Ghysels, T. Ashby, K. Meerbergen and W. Vanroose
Hiding global communication latency in the GMRES algorithm on massively parallel machines.
SIAM J. Scientific Computing, 35(1):C48C71, (2013).

Intercluster Processing

Cluster size: ~120 000 DOFs; optimal decomposition into 27 subdomains

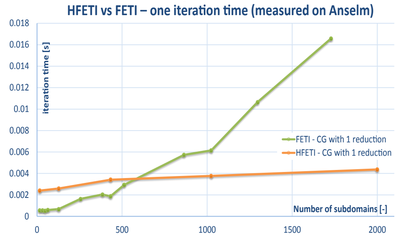
domain size	subdomains	size	avg	sum	prec	iter
4	343	128625	0.123941	5.949182	75.503219	48
5	216	139968	0.072914	3.718598	37.732576	51
6	125	128625	0.040325	2.217873	16.381135	55
8	64	139968	0.030154	1.718762	10.04497	57
11	27	139968	0.034425	1.893399	7.490033	55
17	8	139968	0.064372	3.347334	7.876874	52

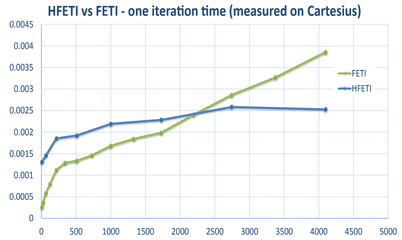
Cluster size: ~330 000 DOFs; optimal decomposition into 27 subdomains

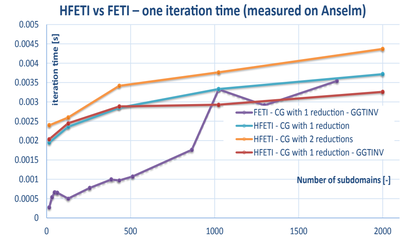
domain size	subdomains	size	avg	sum	prec	iter
5	512	331776	0.309891	17.973677	338.035839	58
6	343	352947	0.198848	12.328554	180.041968	62
7	216	331776	0.118965	7.613787	90.402162	64
11	64	331776	0.080177	5.13133	27.456335	64
15	27	331776	0.095755	6.03256	19.761467	63
23	8	331776	0.156146	8.431877	29.455252	54

Note: domains size 'N' is the number of elements per edge of the cube – real size is equal to $3*(N+1)^3$
Note: avg – average iteration time [s]; sum – total time of all iterations = solution time; prec – cluster preprocessing time; iter – number of iterations;

FETI and HFETI per iteration time



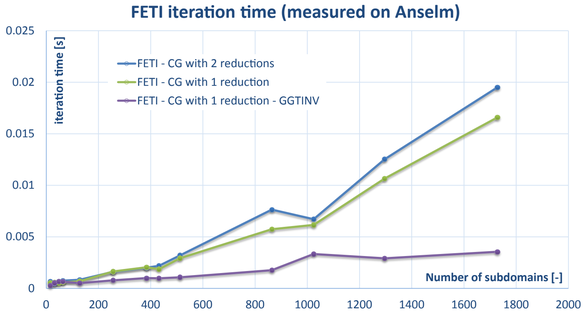




Weak scaling – domain size is fixed; cluster size is fixed; number of domains goes from 1 to 2000
- FETI a HFETI uses CG with 1 reduction and coarse problem is solved using distributed inverse matrix of GGT
- Cartesius: domain size $5^3 = 3*(5+1)^3 = 648$; cluster size is 8 domains (3 clusters per node; 1 subdom. per core)
- Anselm: domain size $5^3 = 3*(5+1)^3 = 648$; cluster size: 16 domains (1 cluster per node; 1 subdomain per core)

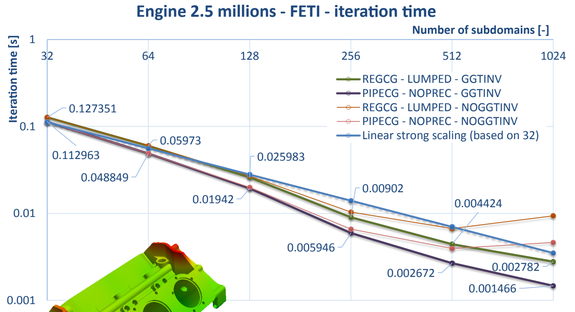
Weak scaling – domain size is fixed; cluster size is fixed; number of domains goes from 1 to 2000
- domain size $5^3 = 3*(5+1)^3 = 648$; cluster size 16 (1 cluster per node; 1 subdomain per core)
- efficient communication algorithms helps both FETI and HFETI
- **CG with 1 reduction + coarse problem solved using distributed inverse matrix (GGTINV)**

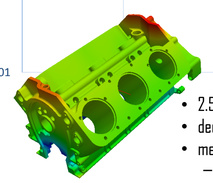
Communication layer performance evaluation on FETI



- **CG algorithm with 2 reductions** – is a general version of the CG algorithm
- **CG algorithm with 1 reduction** – is based on Preconditioned Pipelined CG algorithm, where projector is used in place of preconditioner (this algorithm is ready to use non-blocking global reduction for further performance improvements (comes in Intel MPI 5.0))
- **GGTINV** – parallelizes the solve of coarse problem plus merges two Gather and Scatter global operations into single AllGather
- $5^3 = 3*(5+1)^3 = 648$ DOFs – small subdomain size is chosen to identify all communication bottlenecks of the solver

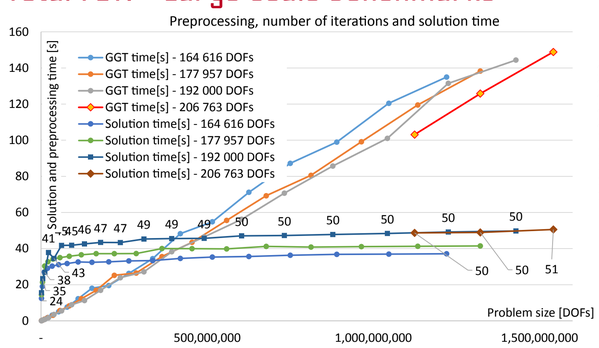
Superlinear Strong Scaling of FETI

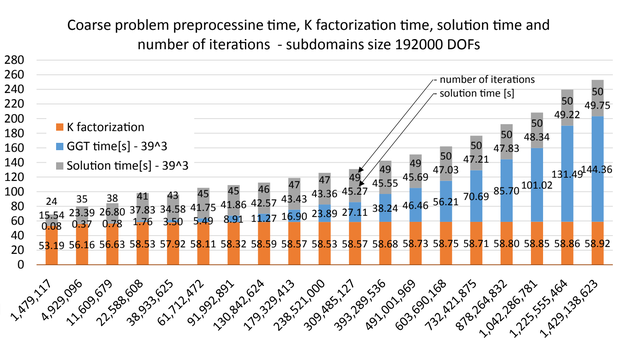




- 2.5 millions of DOFs using FETI solver
- decomposed into: 34, 64, 128, 256, 512 and 1024 subdomains
- measured on Anselm supercomputer
  - 128 nodes with 16 CPU cores (2x8)8 subdomains per node

Total FETI - Large scale benchmarks





Benchmark Systems

IT4Innovations – www.it4i.cz – Anselm – up to ~3300 cores
- non-blocking cluster of 209 nodes each with:
  - 2x 8-core Intel Sandy Bridge E5-2665, 2.4GHz and 64GB of RAM
- InfiniBand QDR network - 40 Gbit/s inter-node bandwidth

SurfSara.nl – Cartesius – up to ~8600 cores
- non-blocking island of 360 nodes each with:
  - 2× 12-core Intel Xeon E5-2695 v2 (Ivy Bridge), 2.4 GHz and 64 GB RAM
- InfiniBand FDR network - 56 Gbit/s inter-node bandwidth

https://projects.imec.be/exa2ct/

http://www.prace-ri.eu/