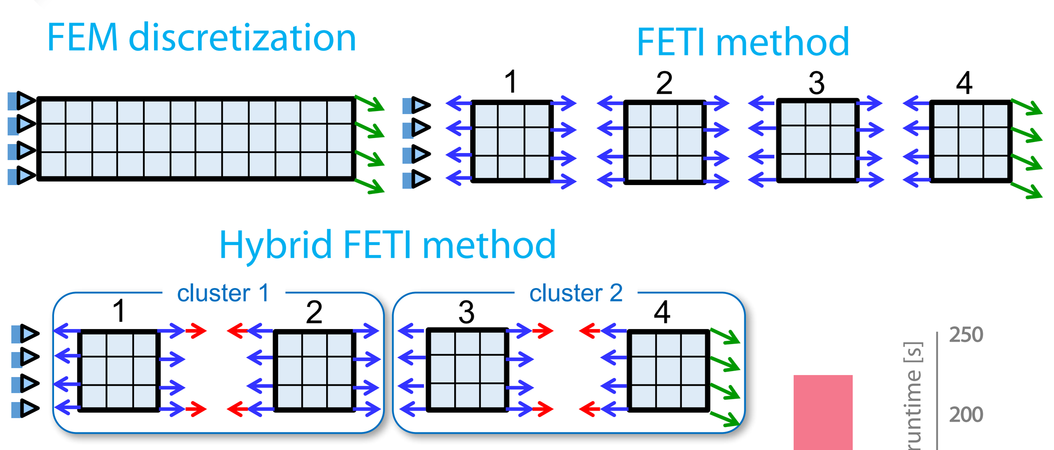
Parallel FETI Solver for Modern Architectures

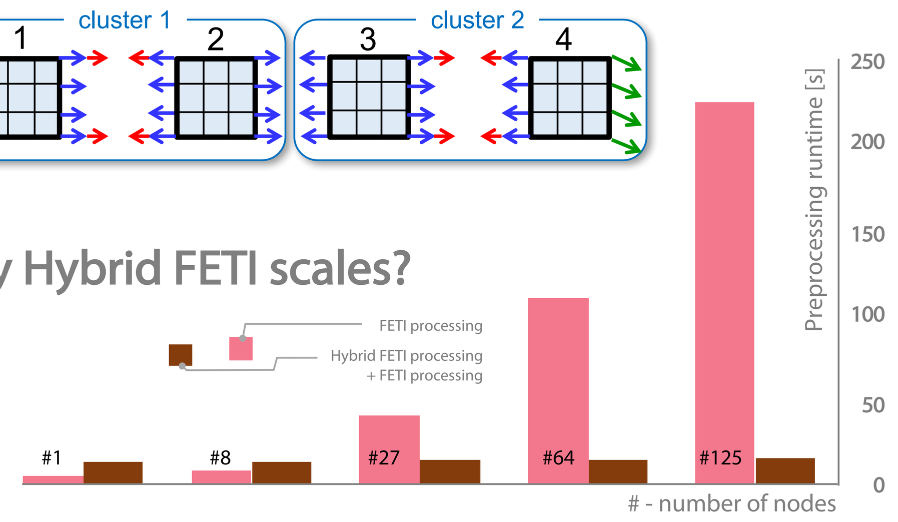
ESPRESO Solver – espreso.it4i.cz

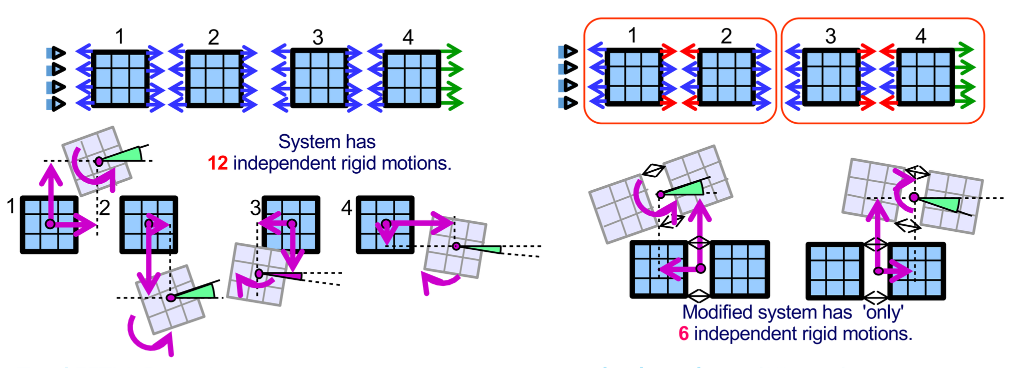
Lubomír Říha (lubomir.riha@vsb.cz), Tomáš Brzobohatý, Michal Merta, Alexandros Markopoulos, Ondřej Meca, and Tomáš Kozubek
IT4Innovations National Supercomputing Center, Ostrava, Czech Republic - http://www.it4i.cz

Hybrid Total FETI Method - Multilevel FETI



Why Hybrid FETI scales?





Total FETI (2D case)
Problem decomposed into 4 subdomains generates coarse problem matrix (GGT) with dimension:

3 *(number of SUBDOMAINS) = 12

Hybrid Total FETI (2D case)
Beam decomposed into 2 clusters (each consists of N subdomains) generates coarse problem matrix (GGT) with dimension

3 *(number of CLUSTERS) = 6

Number of clusters = number of nodes

FETI and HTFETI References:

[1] A METHOD OF FINITE-ELEMENT TEARING AND INTERCONNECTING AND ITS PARALLEL SOLUTION ALGORITHM
By: FARHAT, C; ROUX, FX; INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, Volume: 32, Issue: 6, 1991

[2] Highly scalable parallel domain decomposition methods with an application to biomechanics
By: Klawonn, Axel; Rheinbach, Oliver; ZAMM-ZEITSCHRIFT FUR ANGEWANDTE MATHEMATIK UND MECHANIK, Volume: 90, Issue: 1, 2010

[2] Total FETI domain decomposition method and its massively parallel implementation
By: Kozubek, T.; Vondrak, V.; Mensik, M.; et al.; ADVANCES IN ENGINEERING SOFTWARE Volume: 60-61, 2013

FETI Iterative Solver for Many-core Accelerators

- FETI methods relies on sparse direct solvers (mainly on forward and backward substitutions) and sparse matrix vector multiplications
- Operation on sparse matrices have low arithmetic density and "bad" memory access patterns
- Using Local Schur complements in form of dense matrices is still memory bounded, but can fully utilize fast accelerator memory
- This is the main factor that brings the speedup when compared to CPU lower main memory bandwidth

Projected Conjugate Gradient in FETI

```
r_0 := b − A x_0; u_0 := M^{-1} r_0; p_0 := u_0
for i = 0, ..., m − 1 do
    s := A p_i
    α := ⟨r_i, u_i⟩ / ⟨s, p_i⟩
    x_{i+1} := x_i + α p_i
    r_{i+1} := r_i − α s
    u_{i+1} := M^{-1} r_{i+1}
    β := ⟨r_{i+1}, u_{i+1}⟩ / ⟨r_i, u_i⟩
    p_{i+1} := u_{i+1} + β p_i
end for
```

Pre-processing – K factorization
1.) $x = B_1^T \cdot \lambda$ - **SpMV**
2.) $y = K^{-1} \cdot x$ - **solve**
3.) $\lambda = B_1 \cdot y$ - **SpMV**
4.) stencil data exchange in λ
- MPI – Send and Recv
- OpenMP – shared mem. Vec

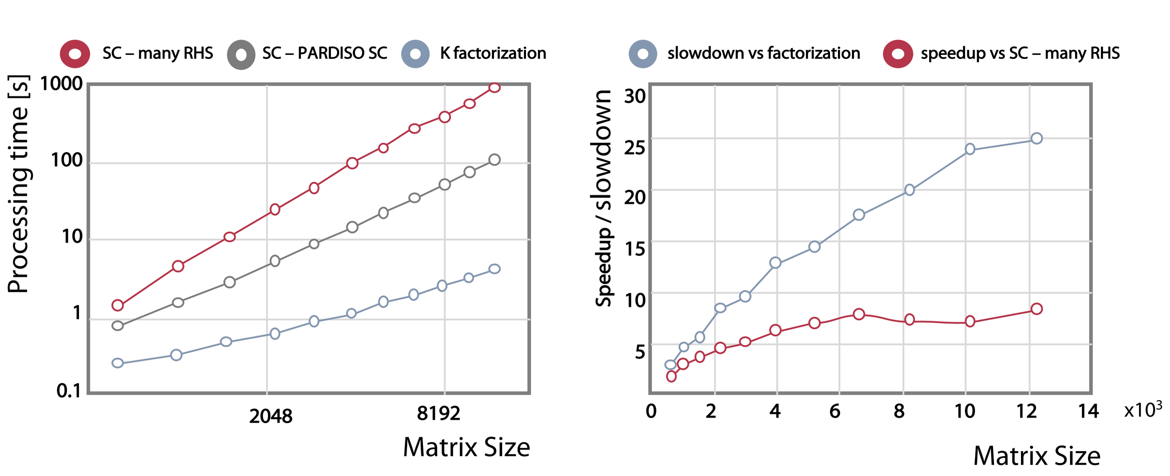
Pre-processing - $S_c = B_1 K^{-1} B_1^T$
1.) - **nop**
2.) $\lambda = S_c \cdot \lambda$ - **DGEMV, DSYMV**
3.) - **nop**
4.) stencil data exchange in λ
- MPI – Send and Recv
- OpenMP – shared mem. Vec

Pre-processing - $S_c = B_1 K^{-1} B_1^T$ → GPU/MIC
1.) λ → GPU/MIC - **PCIe transfer from CPU**
2.) $\lambda = S_c \cdot \lambda$ - **DGEMV, DSYMV on GPU/MIC**
3.) λ ← GPU/MIC - **PCIe transfer to CPU**
4.) stencil data exchange in λ
- MPI – Send and Recv
- OpenMP – shared mem. vec

90 – 95% of runtime spent in Ap_i

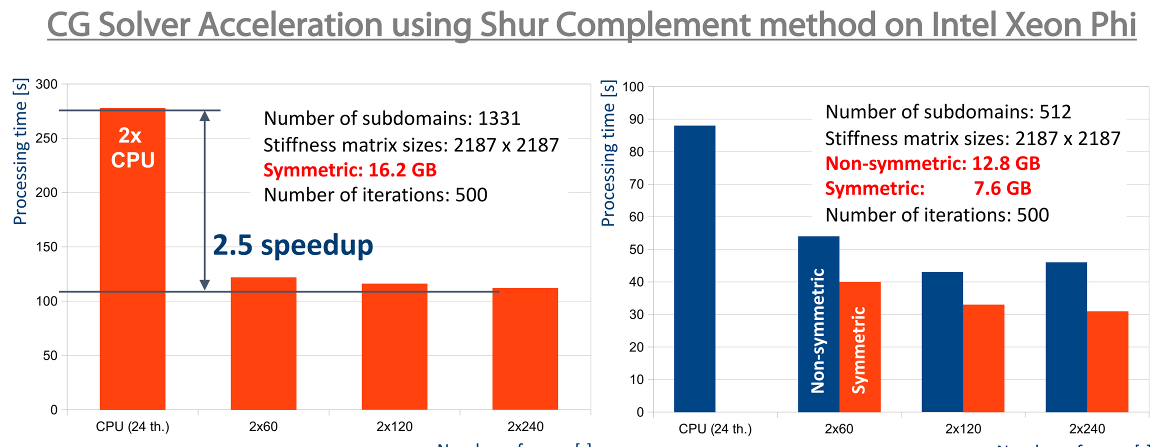
Local Schur Complement Method for FETI

Local Schur Complement Calculation using PARDISO



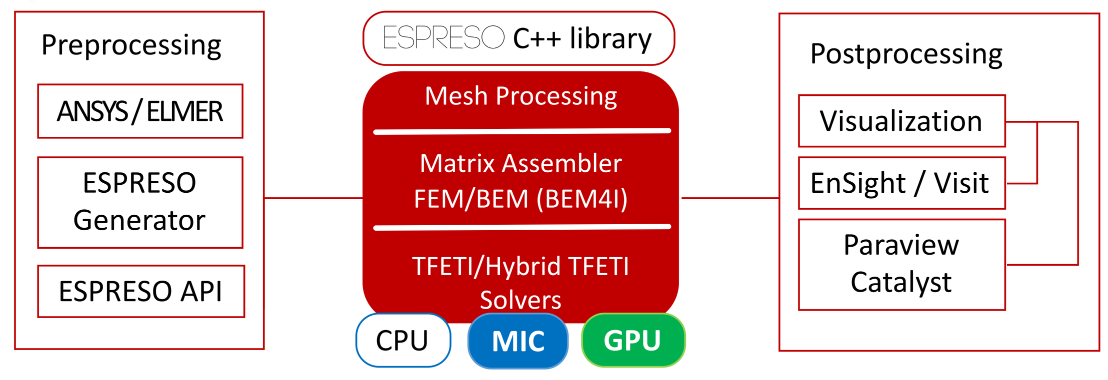
- Schur complement calculation is the main bottleneck of the method
- PARDISO solvers contain efficient algorithms for SC calculation (PARDISO SC and PARDISO MKL exhibit almost identical performance)

CG Solver Acceleration using Shur Complement method on Intel Xeon Phi



Two Intel Xeon Phi 7120P are 2.5 times faster than two 12 core Xeon E5-2680v3 (Symmetric format for SC further improves performance).

ESPRESO Library



Key Features of the ESPRESO Library

- support FEM and BEM (uses BEM4I library) discretization for: **Advection-diffusion, Stokes flow and Structural mechanics**
- Support for Ansys and OpenFOAM database file format
- Multiblock benchmark generator for large scalability tests
- C API allows ESPRESO to be used as solver library - tested with CSC ELMER
- Postprocessing and Vizualization is based on VTK library and Paraview (support for realtime vizualization using Paraview Catalyst)

Massively Parallel Solver

- based on highly scalable Hybrid Total FETI Method – scales to ~18 000 compute nodes
- Support for symmetric (CG with full orthogonalization) and nonsymmetrical systems (BiCGStab, GMRES)
- supports modern many core architectures – GPGPU and Intel Xeon Phi
- contains pipelined Conjugated Gradient Solver – communication hiding
- supports hybrid parallelization in form of MPI and Cilk++

Key Research Funding Projects

- **Intel Xeon Phi acceleration developed under Intel PCC at IT4Innovations**
- **Hybrid FETI implementation developed under EXA2CT FP7 project**
- **Scalability tests on Titan machine and GPU acceleration developed under the ORNL Director Discretion project for 2.7 million core hours**

Parallel Computing Center

IT4Innovations Salomon
2,016 Intel Xeon E5-2680v3, 2.5GHz, 12cores
864 Intel Xeon Phi 7120P, 61cores, 16GB RAM

Experience with Knights Landing (KNL)

Intel MKL PARDISO for FETI Solvers

Cholesky decomposition and solve using Parallel Studio 2017 beta

	MCDRAM		DDR	
# threads	Factorization [s]	Solve (100x) [s]	Factorization [s]	Solve (100x) [s]
64	5.5	10.8	**5.8**	**11.6**
128	6.4	8.5	7.5	11.9
256	7.2	5.1	10.9	13.0
2x Haswell	--	--	3.0	7.3

Schur complement and DSYMV using Parallel Studio 2017 beta

	MCDRAM		DDR	
# threads	Decomposition	Apply (100x)	Decomposition	Apply (100x)
64	24.5	3.8	**43**	12
128	19.1	3.3	57	12
256	44.4	3.1	72	15
2x Haswell	--	--	13.4	7.7

Schur Complement Processing on KNL

Hardware: early access to Intel Xeon Phi 7210 (64 cores at 1.3 GHz) at Intel's Endeavor cluster under umbrella of Intel PCC at IT4Innovations

Schur complement calculation using MKL PARDISO
- In the best case decomposition scales up to 128 threads (MCDRAM)
- 2.3x better performance for MCDRAM when compared to DDR (19.1s/43s)
- 2x Haswell CPUs of IT4I Salomon is 1.4 times faster than KNL (19.1s/13.4s)

Apply – Symmetric Matrix Vector Multiplication (MKL)
- in DDR very bad scalability - from 8 to 32 threads 1.2x speedup
- in MCDRAM scales up to 256 threads up to 4x speed compared to DDR
- **1x KNL is 2.5x times faster than 2x Haswell of IT4I Salomon (3.1s/7.7s)**
- **1x KNL is 2.0x times faster than 1x Xeon Phi 7120p**

Notes:
- Decomposition on KNL is two times faster than on KNC

MCDRAM significantly boosts the performance of the Schur complement method in ESPRESO solver.

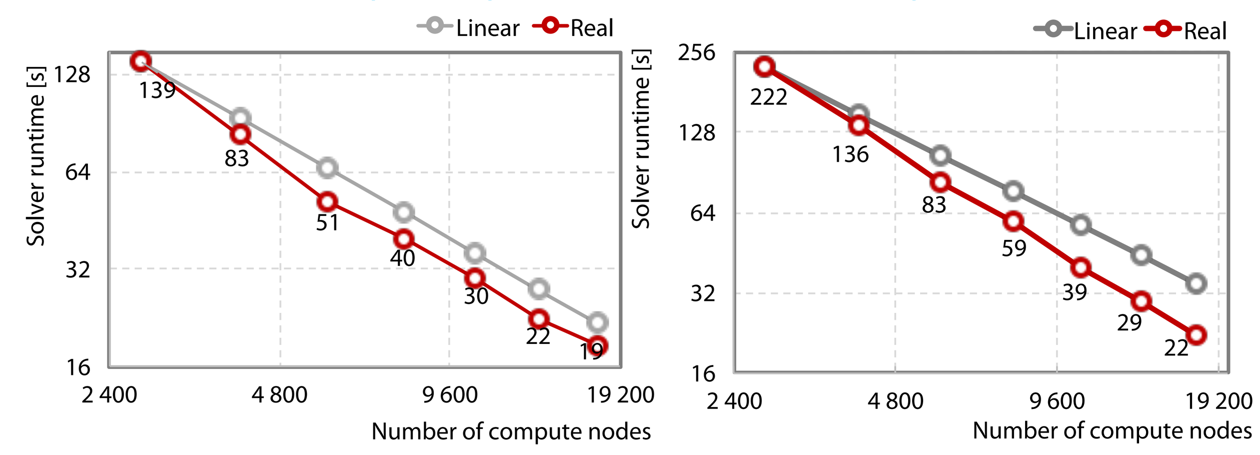
One KNL compute node can deliver similar performance to two Intel Xeon Phi 7120P accelerators for the critical part of the FETI based solvers if essential data is stored in the MCDRAM.

This reflects the 2.5x speedup over two 12core Intel Xeon E5-2680v3@2.5GHz CPUs

Strong Scalability Evaluation of the HTFETI Method

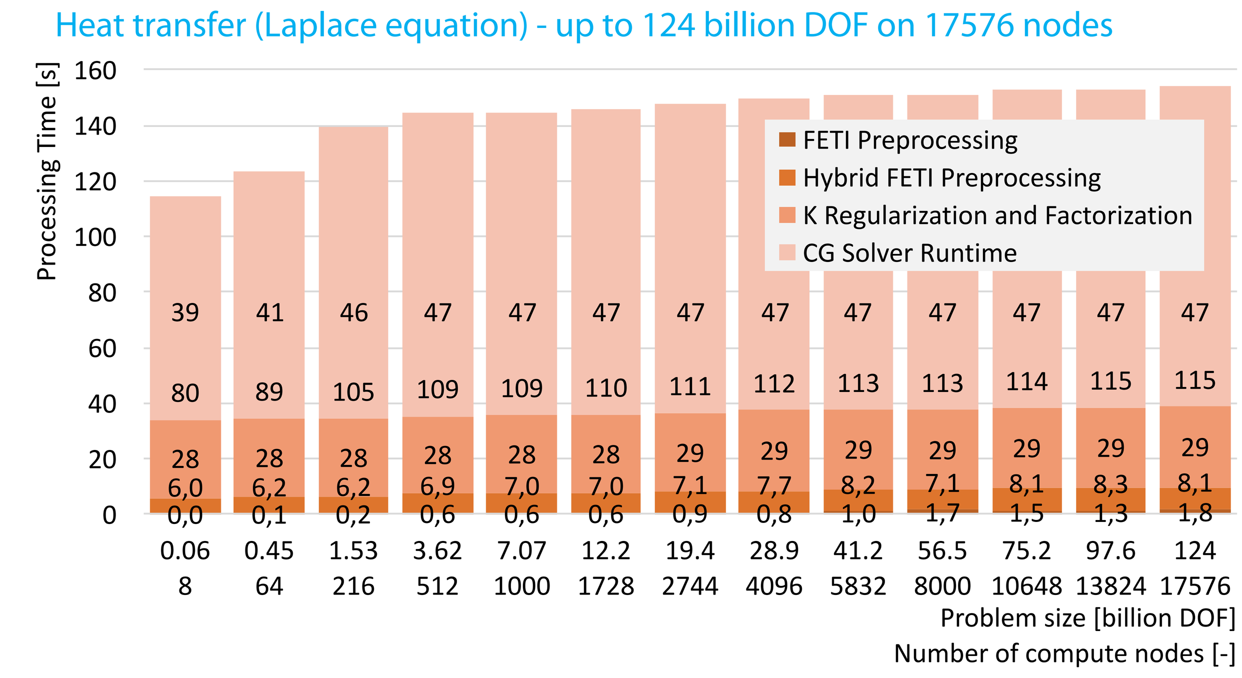
Test performed on ORNL Titan (18,688 compute nodes)

Structural mechanics - 11 billion DOF on up to 17 576 nodes (281 216 cores) and Heat transfer (Laplace equation) - 20 billion DOF on up to 17 576 nodes



Weak Scalability Evaluation of the HTFETI Method

Heat transfer (Laplace equation) - up to 124 billion DOF on 17576 nodes

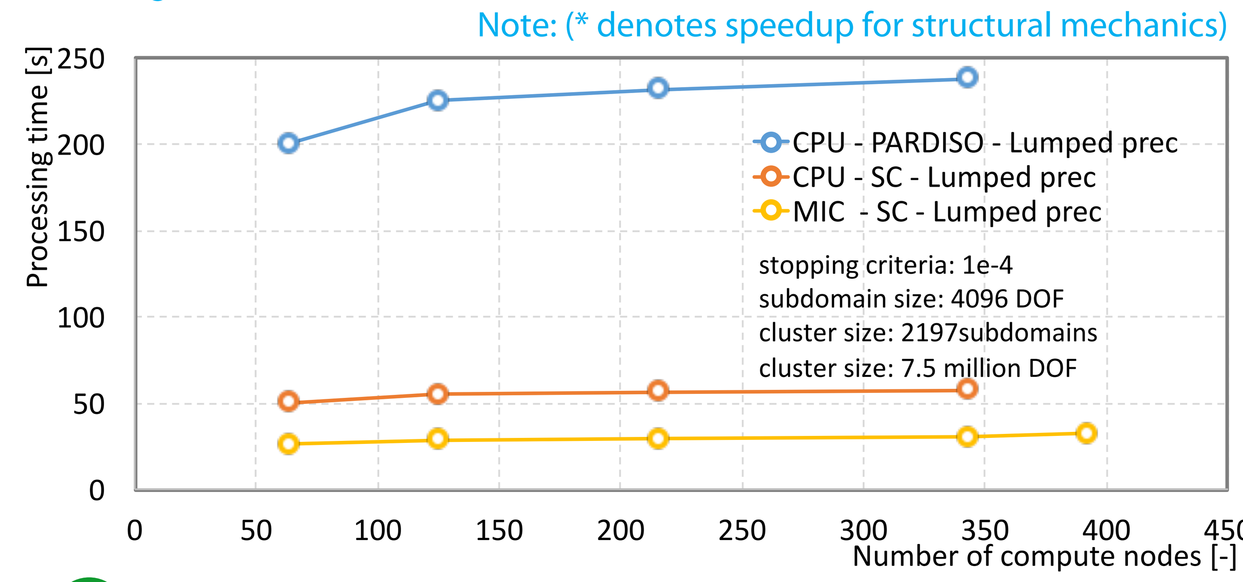


Intel Xeon Phi Acceleration of the Iterative Solver

Tests performed on IT4Innovations Salomon Supercomputer
Heat Transfer - CG Solver Runtime w. Lumped Preconditioner solving 7.5 to 2912 million DOF

speedup
7.8 2.7*
1.9 2.2*

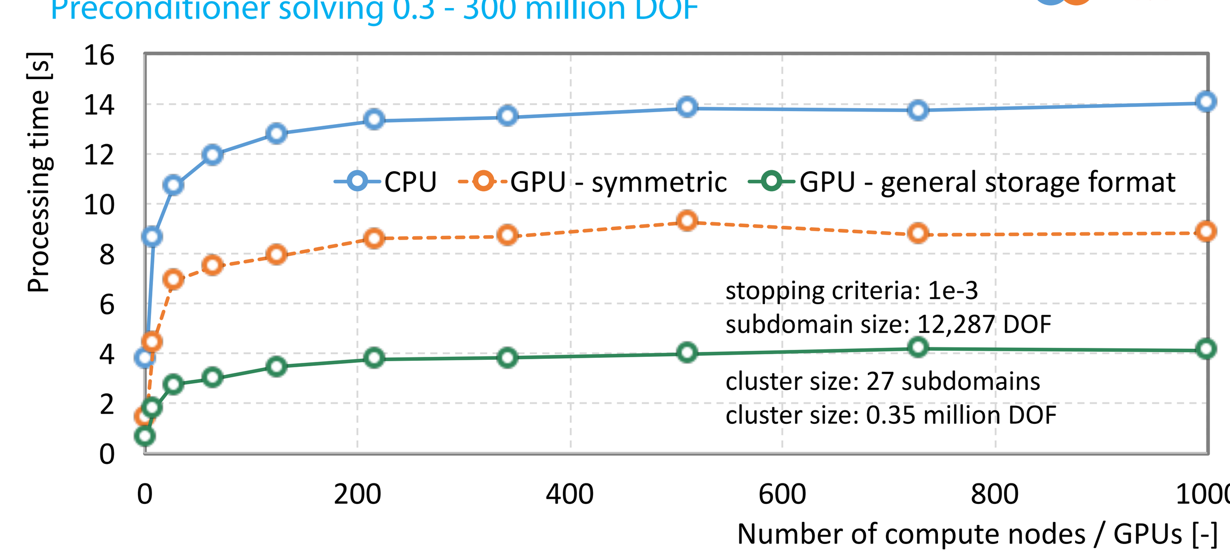
Note: (* denotes speedup for structural mechanics)



GPGPU Acceleration of the Iterative Solver

Test performed on ORNL Titan
Structural Mechanics - CG Solver Runtime with Lumped Preconditioner solving 0.3 - 300 million DOF

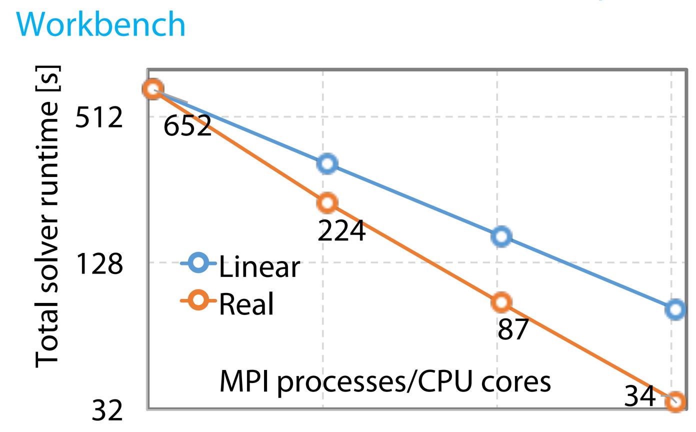
speedup
3.4
1.4

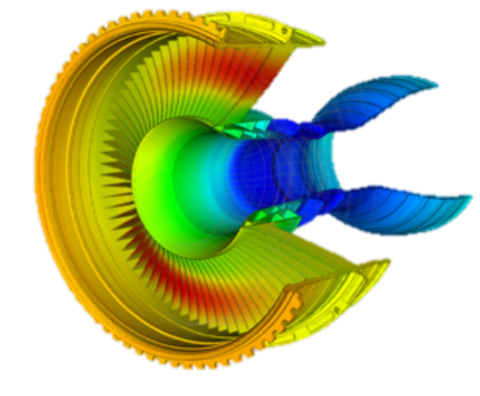


Note: AMD Opteron 6274 16-core CPU with Tesla K20X GPU w. 6 GB RAM + general storage of SC -> 0.35 mil DOF per GPU ~10x smaller problem than 2x Intel Xeon Phi 7120 w 2x16GB RAM and sym. SC

Real World Problems

Tests performed on IT4Innovations Salomon Supercomputer
Structural Mechanics - 300 million DOF problem generated from ANSYS Workbench





Hybrid Total FETI with Dirichlet preconditioner				
Number of compute nodes [-]	47	93	186	372
Number of clusters [-]	1024	2048	4096	8192
Number of subdomains per cluster [-]	128	128	32	32
Total number of subdomains [-]	131072	262144	131072	262144
Average subdomain size [DOF]	2289	1144	2289	1144
Number of iterations [-]	1104	823	522	365
Total solution time with preprocessing [s]	**652**	**224**	**87**	**34**

References:

[1] L. Riha, T. Brzobohaty, A. Markopoulos, O. Meca, T. Kozubek, Massively Parallel Hybrid Total FETI (HTFETI) Solver, in: Platform 270 for Advanced Scientific Computing Conference, PASC, ACM, 2016. doi:DOI:http://dx.doi.org/10.1145/2929908.2929909
[2] L. Riha, T. Brzobohaty, A. Markopoulos, O. Meca, T. Kozubek, O. Schenk, W. Vanroose, Efficient Implementation of Total FETI Solver for Graphic Processing Units using Schur Complement, in: HPCSE 2015, LNCS 9611, 2016. doi:DOI:10.1007/978-3-319-40361-86. 290
[3] L. Riha, T. Brzobohaty, A. Markopoulos, Hybrid parallelization of the Total FETI Solver, Advances in Engineering Software (2016) – doi:http://dx.doi.org/10.1016/j.advengsoft.2016.04.004. http://www.sciencedirect.com/science/article/pii/ S0965997816300783
[4] L. Říha, T. Brzobohatý, A. Markopoulos, M. Jarošová, T. Kozubek, D. Horák, V. Hapla, "Implementation of the Efficient Communication Layer for the Highly Parallel Total FETI and Hybrid Total FETI Solvers", Parallel Computing, DOI 10.1016/j.parco.2016.05.002

IT4Innovations national supercomputing center