Acceleration of the boundary element library BEM4I on the Knights Corner and Knights Landing architectures

IT4Innovations national supercomputing center

Michal Merta, Jan Zapletal

michal.merta@vsb.cz, jan.zapletal@vsb.cz

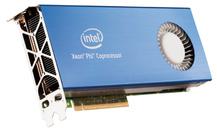
BEM4I

BEM4I [1,4,5] is a library of parallel **boundary element method** (BEM) [2] based solvers developed at IT4Innovations National Supercomputing Center. It utilizes the main aspect of BEM and reduces the problem to the boundary of a computational domain, which makes it ideal for problems on **unbounded domains** or **shape optimization** problems. The library features explicit vectorization using **SIMD** instruction set extensions, parallelization in shared and distributed memory via **OpenMP** and **MPI**, respectively. The library provides solvers for problems from simple heat transfer to time-dependent wave scattering.

Laplace equation	BEM4I	Electromagnetism
Lamé equation		Sound scattering
Wave equation		Heat transfer
Helmholtz equation		Linear elasticity

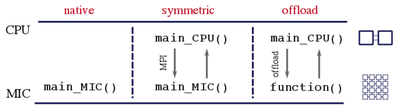
INTEL XEON PHI COPROCESSOR

The Intel Xeon Phi coprocessor is a representative of Intel's **Many Integrated Core** (MIC) architecture. The first generation (Knights Corner, KNC) features up to **61 cores** running on 1.2 GHz, four hardware threads per core, and 16 GB of RAM. The cores are equipped with 512-bit long **vector registers**.



The theoretical peak performance of the coprocessor is **1.2 Tflops** in double precision arithmetics. It runs its own Linux operating system and can be treated as a stand-alone node in a heterogeneous cluster.

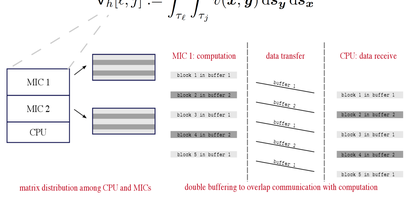
There are three main computational modes – **symmetric**, **native**, and **offload**.



OFFLOADING THE BEM COMPUTATION

Since the **assembly of the system matrix** is the most computationally demanding part of classical BEM (having quadratic complexity with respect to number of surface elements), we accelerate it by **offloading** part of the **numerical quadrature** over pairs of elements to multiple coprocessors.

$$\mathsf{V}_h[\ell,j] := \int_{\tau_\ell}\int_{\tau_j} v(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}}\,\mathrm{d}s_{\boldsymbol{x}}$$



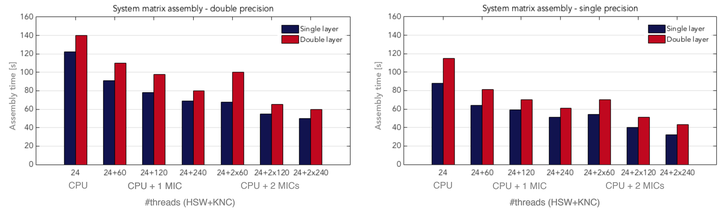
To overcome the main bottleneck (data transfer via PCIe bus) we use the technique of **double buffering**.

THE SALOMON SUPERCOMPUTER

The numerical experiments were performed using the Salomon supercomputer at the IT4Innovations National Supercomputing Center in Ostrava, Czech Republic. It consists of **1008 compute nodes**, each equipped with two Intel Xeon E5-2680v3 12-core processors and 128 GB of RAM. 432 of the compute nodes are accelerated by **two Intel Xeon Phi 7120P** coprocessors which makes it the biggest installation of this kind in Europe. The theoretical peak performance of the cluster is **2 Pflops**.

ASSEMBLY OF THE SYSTEM MATRICES FOR THE LAPLACE EQUATION

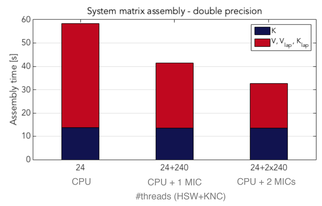
The following numerical experiments document accelerated assembly of the single layer and double layer matrix for the Laplace equation. The testing mesh consists of 81,920 surface elements [4]. The maximum **speedup** using CPU and two Xeon Phi cards with respect to CPU-only computation is **2.5**.



ASSEMBLY OF THE SYSTEM MATRICES FOR THE LAMÉ EQUATION

To accelerate the **linear elasticity** solver we offload computation of the single layer operator matrix V to the coprocessor. The double layer operator matrix K is obtained by transformations of V and appropriate matrices for the Laplace operator.

The numerical experiments were performed on the mesh consisting of 20,480 surface elements. The maximum **speed-up** obtained using two coprocessor cards is **1.78**.

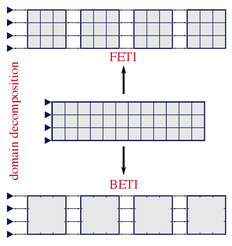


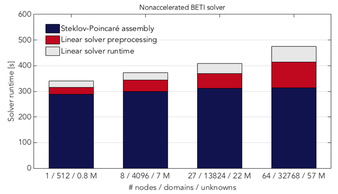
TOWARDS EXASCALE – ACCELERATED BETI DDM

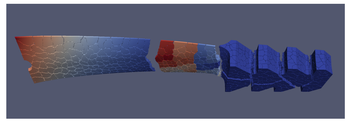
In cooperation with the **ESPRESO** (ExaScale PaRallel FETI Solver) library [3] BEM4I enables parallel solution of large scale **linear elasticity** problems using the **BETI** (Boundary Element Tearing and Interconnecting) **domain decomposition method**. BEM4I provides ESPRESO with the Steklov-Poincaré operator for each subdomain which serves as a local Dirichlet-to-Neumann map.

$$\mathsf{S}_h = (1/2\mathsf{M}_h + \mathsf{K}_h^\top)\mathsf{V}_h^{-1}(1/2\mathsf{M}_h + \mathsf{K}_h)$$

The assembly of the Steklov-Poincaré operator is the most computationally demanding part of the BETI method and its acceleration by the Intel Xeon Phi coprocessors is currently under development.



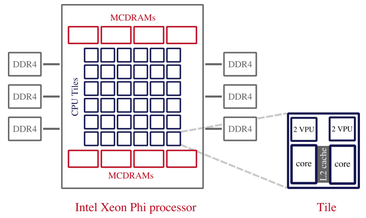




INTEL XEON PHI PROCESSOR

The Intel Xeon Phi processor (code name Knights Landing, KNL) represents the latest generation of the MIC architecture. With up to **72 cores**, **384 GB** of DDR4 memory, and **16 GB of fast MCDRAM** it can provide more than **3 Tflops** of double precision computing power. Contrary to Knights Corner it is available in the form of a stand-alone bootable processor.

The Atom-Silvermont based cores are equipped with **four hardware threads** and two vector units supporting **AVX-512** SIMD instruction set extension. This enables concurrent operations on up to 8 double or 16 single precision operands.



There are three memory modes available at the coprocessor. In the **flat mode** the the MCDRAM extends the DDR4 RAM with additional 16 GB. In **cache mode** MCDRAM serves as a fast data cache. The **hybrid mode** combines both approaches.

Knights Corner		Knights Landing
57 - 61	# cores	64 - 72
in-order, based on P54C	architecture	out-of-order, based on Airmont
IMCI	instruction set	AVX-512
22 nm	manufacturing	14 nm
16 GB GDDR5 on-card	max. RAM	16 GB MCDRAM/384 GB DDR4
350 GB/s	RAM bandwidth	400 GB/s/90 GB/s
1.2 TFlops	Rpeak (DP)	3 TFlops
up to 300 W	TDP	up to 260 W
PCIe card	availability	standalone processor, PCIe card

CODE OPTIMIZATION

Optimization, **vectorization**, and OpenMP **parallelization** are still necessary to leverage the full potential of the Xeon Phi processor. However, it is no longer necessary to transfer data via the PCI Express bus and the original Xeon-optimized code can be easily ported to Xeon Phi. The code optimization includes

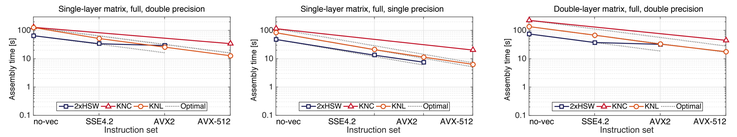
- converting arrays of structures to **structures of arrays**
- **data alignment**
- assisting with vectorization using `#pragma omp simd`
- providing **SIMD-enabled functions**
- manual **loop unrolling** and **collapsing**
- using tools like Intel Advisor to identify bottlenecks

EXPERIMENTAL CLUSTER

The following experiments were performed using **Intel Xeon Phi 7210** processors installed at Intel's Endeavor cluster. The processors feature 64 cores (256 threads) running at 1.3 GHz, 16 GB of MCDRAM, and 96 GB of DDR4 RAM. The flat memory mode with quadrant clustering is used. Computational times are compared to the Xeon processors and KNCs (in native mode) on the Salomon cluster.

VECTORIZATION TESTS

To demonstrate the utilization of the SIMD instruction set extensions, we test the assembly of the system matrices V and K for the Laplace equation using SSE4.2, AVX2, AXV-512, IMCI, and no vectorization. Two SIMD units per KNL core reduce the vector register pressure, resulting in better SIMD efficiency than using KNC or Haswell.

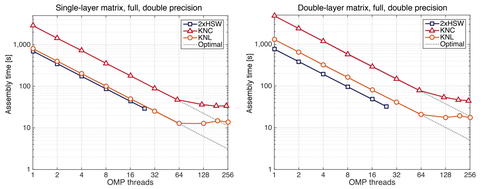


		no-vec	SSE4.2	AVX2	AVX-512	IMCI
Intel Xeon Phi 7210, 128 threads	double	1.00	2.45	4.95	9.94	-
	single	1.00	3.95	7.23	13.05	-
Intel Xeon Phi 7120P, 244 threads	double	1.00	-	-	-	3.77
	single	1.00	-	-	-	5.67
2xIntel Xeon E2680v3, 24 threads	double	1.00	1.92	2.23	-	-
	single	1.00	3.59	6.35	-	-

Speedup (assembly of the matrix V)

OPENMP SCALABILITY

Scalability of the assembly of the full system matrices V, K for the Laplace equation was tested on a mesh consisting of 20,480 surface elements.



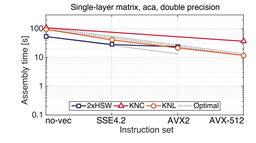
threads	64	128	192	256
V	62	63	54	58
K	61	73	64	65

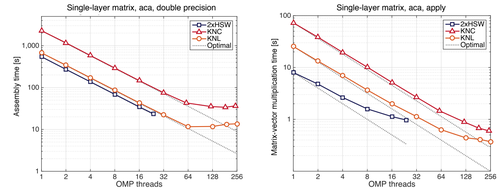
Speedup on KNL

Thanks to the out-of-order architecture of the KNL chips utilization of multiple hardware threads per core is not as important as in the case of KNC architecture. However, they still deliver additional speedup.

ADAPTIVE CROSS APPROXIMATION

The classical BEM produces dense system matrices with quadratic memory and computational complexity. To reduce this complexity we implement the **adaptive cross approximation method** based on the hierarchical mesh decomposition and **low rank approximation** of the blocks of the system matrices. Following tests were performed using mesh consisting of 81,920 surface elements. The right-hand side graph depicts the scalability of the system matrix assembly using different SIMD instruction set extensions.





The parallelization of ACA in BEM4I is done by distributing the assembly of individual blocks among available OpenMP threads. The matrix-vector multiplication is parallelized in a similar fashion. The left-hand side graphs describe the scalability of these methods.

CONCLUSION

The Intel Xeon Phi processor (KNL) provides a significant performance gain compared to the KNC generation of Intel's Many Integrated Core architecture. Since it is no longer necessary to transfer data via PCIe bus, porting and optimization of current codes is much easier. In most of the tests we were able to obtain comparable or better results than using the Intel Xeon processors. See the right-hand side for **comparison of best system matrix assembly times among platforms**.

		KNL vs. HSW	KNL vs. KNC
Full	V	2.29	2.62
	K	1.82	2.53
ACA	V	2.64	2.03
	K	1.40	2.91

Speedup

REFERENCES

[1] Merta, M., Zapletal, J. *BEM4I*. Retrieved from http://bem4i.it4i.cz/

[2] Steinbach, O. *Numerical Approximation Methods for Elliptic Boundary Value Problems*. Springer 2008.

[3] Riha, L. et al. *ESPRESO – ExaScale PaRallel FETI SOlver*. Retrieved from http://espreso.it4i.cz/

[4] Merta, M., Zapletal, J., Jaros, J. *Many Core Acceleration of Boundary Element Method*. LNCS 9611, 2016.

[5] Merta, M., Zapletal, J. *Acceleration of Boundary Element Method by Explicit Vectorization*. Adv Eng Softw 86, 2015.

ACKNOWLEDGEMENT

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project „IT4Innovations excellence in science - LQ1602“ and from the Large Infrastructures for Research, Experimental Development and Innovations project „IT4Innovations National Supercomputing Center – LM2015070“

The access to the Endeavor cluster was provided by the Intel Corporation within the project Intel Parallel Computing Centre at IT4Innovations National Supercomputing Center. The authors were also supported by VŠB-Technical University of Ostrava by the grant SGS SP2016/113.