Acceleration of the Boundary Element Library BEM4I on the Knights Corner and Knights Landing Architectures

Michal Merta
IT4Innovations National Supercomputing Center
VŠB – Technical University of Ostrava
17. listopadu 15/2172, 708 33 Ostrava
Czech Republic
Email: michal.merta@vsb.cz

Jan Zapletal
IT4Innovations National Supercomputing Center
VŠB – Technical University of Ostrava
17. listopadu 15/2172, 708 33 Ostrava
Czech Republic
Email: jan.zapletal@vsb.cz

***Abstract*—The aim of the poster is to present the acceleration of the boundary element method (BEM) by the Intel Xeon Phi technology. The poster provides brief overview of BEM followed by the discretization approach and efficient numerical assembly of the BEM matrices. We discuss its parallelization by OpenMP in shared memory and the SIMD vectorization necessary to exploit the full potential of the Xeon and Xeon Phi architectures. We present numerical experiments performed both on the Knights Corner (KNC) coprocessor and the Knights Landing (KNL) standalone processor.**

I. Introduction

Although the popularity of the manycore and manycore-accelerated architectures has been steadily rising in the past decade, the paradigm shift from the multicore platforms puts high demands on the developers of scientific codes. To fully exploit the potential of such architectures, it is not only important to utilize all available cores, but to also pay close attention to the SIMD vectorization of the computationally intensive loops [1], [2].

BEM, as a representative of solvers for discretized partial differential equations [3], presents a suitable candidate for manycore acceleration due to the expensive evaluation of singular surface integrals carried out by four-dimensional regularized Gaussian quadrature. In the following text we briefly describe the acceleration of the BEM4I library developed at the IT4Innovations National Supercomputing Center, Czech Republic, both by the offload to the KNC coprocessor [2] and the native run on KNL. The results are compared to the assembly times on the Haswell (HSW) architecture.

II. Efficient Implementation of BEM

Here, we restrict ourselves to the efficient assembly of the single-layer operator matrix by the regularized Gaussian quadrature [3],

$$\begin{aligned} \mathsf{V}_h[i,j] &:= \int_{\tau_i}\int_{\tau_j} \frac{1}{4\pi}\frac{1}{\|\boldsymbol{x}-\boldsymbol{y}\|}\mathrm{d}\boldsymbol{s}_{\boldsymbol{y}}\,\mathrm{d}\boldsymbol{s}_{\boldsymbol{x}} \\ &\approx \sum_{m,n,o,p} w_m w_n w_o w_p f(z_m,z_n,z_o,z_p) \end{aligned} \tag{1}$$

assembled over the tuples of surface triangles τ_i, τ_j. In the classical BEM this operation reaches the quadratic complexity.

To utilize all available cores, the tuples are distributed to available OpenMP threads. In addition, a portion of the matrix is assembled on the KNC coprocessors by the offload pragmas. The four loops representing the sum from (1) are collapsed, the weights $w_\bullet$ and sampling points $z_\bullet$ are stored in the aligned structure-of-arrays format, and the kernel f is evaluated in the SIMD manner by OpenMP 4.0 pragmas. The efficient vectorized fused-multiply-add instructions can thus be employed.

The disadvantage of the offload implementation is the transfer of the matrix blocks via the PCIe bus. In the BEM4I code, the communication is hidden by computation (by the double-buffering technique). On the other, the stand-alone version of the KNL processor allows to avoid this transfer entirely as it has direct access to the main memory of the computational node.

In the following section we provide numerical experiments performed at the Salomon cluster of the IT4Innovation National Supercomputing Center equipped with dual-socket HSW (Intel Xeon E5-2680v3) nodes accelerated by two KNC (Intel Xeon Phi 7120P) cards and the Intel's Endeavor system equipped with KNL (Intel Xeon Phi 7210) nodes.

III. Numerical Experiments

The numerical experiments have been performed on a mesh consisting of 20,480 triangular surface elements.

Figure 1 depicts the acceleration achieved by offloading part of the V_h assembly to the KNC coprocessor(s) with various number of OpenMP threads. The maximum speedup achieved with one and two KNC cards relatively to the run on 24 threads on HSW reaches the reasonable values of 1.77 and 2.66, respectively.

In Figure 2 we present the OpenMP scalability on the KNL architecture compared to the runtime on the double-socket HSW. The speedup on 32, 64, and 128 threads on KNL compared to 24 threads on HSW reaches 1.27, 2.52, and 2.54,

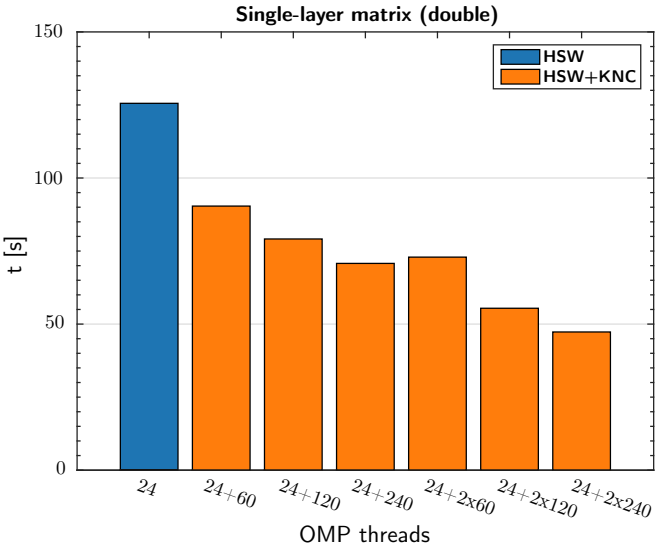

Fig. 1. Acceleration by offload to KNC.

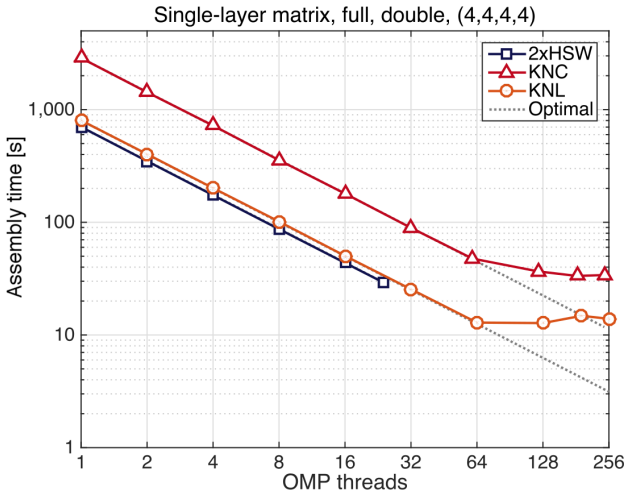

Fig. 2. OpenMP scalability of native KNC assembly.

respectively. Due to the more modern architecture of KNL cores, the hyperthreading is not as significant as in the case of KNC.

Finally, Figure 3 proves the critical need for proper vectorization. On KNL with 128 threads, the speedup due to AVX-512 instructions compared to no vectorization reaches 9.9 (even though the registers accommodate eight double precision operands). The scalability on HSW is not optimal due to high pressure on the SIMD registers, which is overcome in KNL by adding further registers.

IV. Conclusion

We presented the acceleration techniques used in the BEM4I library by means of OpenMP parallelization and SIMD vectorization. The first approach included the offload of the computationally intensive kernels to the KNC coprocessors, the second one utilizes the standalone processing unit of

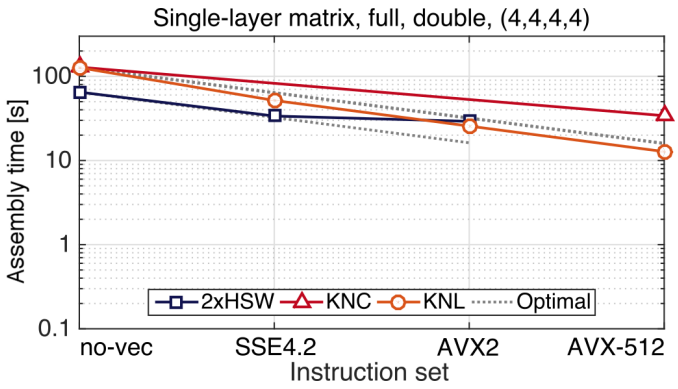

Fig. 3. SIMD scalability of native KNC assembly.

the KNL architecture. Both approaches have been validated by numerical experiments showing reasonable performance and speedup with respect to the code running solely on the HSW system. In addition, the poster also presents the possible interconnection with the massively parallel Espresso domain decomposition library [4] for solving large scale engineering problems (billions of volume degrees of freedom).

Acknowledgment

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", from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center LM2015070", and by VŠB-Technical University of Ostrava within the grant SP2016/113. The authors thank Intel for providing an early access to the KNL architecture on the Endeavor system, where part of the numerical experiments have been performed.

References

[1] M. Merta and J. Zapletal, "Acceleration of boundary element method by explicit vectorization," *Advances in Engineering Software*, vol. 86, pp. 70–79, 2015.

[2] M. Merta, J. Zapletal, and J. Jaros, "Many core acceleration of the boundary element method," in *High Performance Computing in Science and Engineering: Second International Conference, HPCSE 2015, Soláň, Czech Republic, May 25-28, 2015, Revised Selected Papers*, T. Kozubek, R. Blaheta, J. Šístek, M. Rozložník, and M. Čermák, Eds. Springer International Publishing, 2016, pp. 116–125.

[3] S. Sauter and C. Schwab, *Boundary Element Methods*, ser. Springer Series in Computational Mathematics. Springer, 2010.

[4] L. Říha, T. Brzobohatý, and A. Markopoulos, "Hybrid parallelization of the total feti solver," *Advances in Engineering Software*, p. Article in Press, 2016.