

# Tensor-Optimized Hardware Accelerates Fused Discontinuous Galerkin Simulations

Alexander Breuer\*, Alexander Heinecke<sup>†</sup> and Yifeng Cui\*

\*San Diego Supercomputer Center, UC San Diego, San Diego, CA, USA

<sup>†</sup>Parallel Computing Lab, Intel Corporation, Santa Clara, CA, USA

**Abstract**—In recent years the computation/memory balance of processors continuously shifted towards computation. The rise of Deep Learning, which is based on matrix multiplications, accelerated this path, especially in terms of single precision and lower precision computation. An important research question is if this development can be leveraged for traditional HPC. In this work we demonstrate that a high order discontinuous Galerkin solver for seismic wave propagation can execute in single precision without loss of modeling accuracy. Additionally, we extended its kernels to support the Intel Knights Mill CPU with 14 TFLOPS of tensor-optimized single precision performance. This allows us to exploit the hardware’s special computation capabilities, even in a regular HPC application with sparse linear algebra kernels. At the cluster-level, Knights Mill can obtain the same application performance as the latest top-bin dual socket Intel Xeon Platinum nodes. Compared to the HPC-focused Knights Landing processor, speed-ups of up to  $1.6\times$  are possible, depending on the scenario. Additionally, we are able to increase the throughput of seismic discontinuous Galerkin methods by  $4.2\times$ , when comparing our solver’s single precision and fifth order performance to the SC 2017 best-paper award winning work.

## I. INTRODUCTION AND RELATED WORK

Until two years ago, the development of processors for traditional High Performance Computing (HPC) use cases has targeted a more and more complicated design principle: “Increase the memory bandwidth at the same rate as the compute capabilities of the processor itself”. The need for a balanced platform is based on the algorithmic patterns of HPC workloads. The fast rise of deep learning changed this design view of computer architectures. Deep learning algorithms rely on tensor operations, implemented as General Matrix-Matrix Multiplications (GEMMs). More importantly, single precision (FP32) GEMM, half precision (FP16), or even quantized integers are used for deep learning. Therefore, several vendors invested in developing processors with focus on lowered precision, while not increasing memory bandwidth.

We examine how the seismic solver EDGE [1] can benefit from less-balanced, tensor-optimized hardware. In addition, we compare our obtained performance to regular general purpose processors with similar machine balances but large caches and wide execution engines. Optimal utilization of tensor-optimized hardware by EDGE required several steps, which also helps with best performance on general purpose CPUs. As illustrated in Fig. 1, our poster’s contribution is two-fold: a) applying, analyzing and verifying algorithmic advances to EDGE for general purpose CPUs, b) discussion

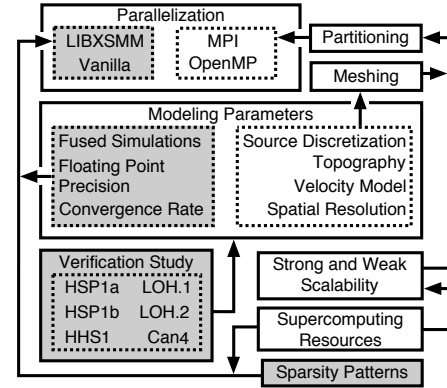


Fig. 1. Illustration of EDGE’s modeling and simulation pipeline. The parallelization is influenced by a multitude of factors. Our areas of contribution are highlighted in gray.

and evaluation of optimizations needed for tensor-optimized hardware. Our poster reports on the following contributions:

- Extension of previous fused DG-FEM simulation technology to arbitrary orders of convergence and high-performance x86 CPU architectures.
- Analysis of FP32 accuracy and performance in an extensive verification study for seismic wave propagation. The analysis of the LOH.1 benchmark is part this work, including an electronic supplement and references to further benchmark results (see App. A).
- Revised Just-In-Time (JIT) assembly kernel generation of core computational routines in the LIBXSMM library, targeting the various processors of our study. Tensor-optimized hardware, which is very specific to dense GEMM, allows us to outperform the SC 2017 best-paper award winning work [2] by  $4.2\times$ .

## II. FUSED SEISMIC SIMULATIONS

EDGE solves the three-dimensional isotropic elastic wave equations in velocity-stress formulation. These are given as a system of linear hyperbolic partial differential equations:

$$q_t + Aq_x + Bq_y + Cq_z = S. \quad (1)$$

$t$  is time and  $\vec{x} = (x, y, z)$  the vector of Cartesian coordinates. Subscripts in Eq. 1 denote partial derivatives.  $q(\vec{x}, t) = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{xz}, \sigma_{yz}, u, v, w)$  is the space-time-dependent vector of quantities, containing the three nor-

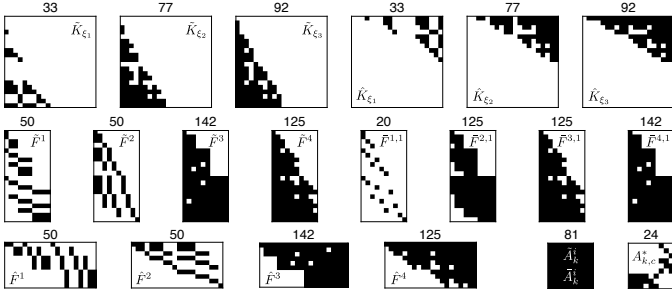


Fig. 2. Sparsity patterns of the ADER-DG matrices for a fourth order method. On top of each matrix, the number of non-zero entries is given.

mal stresses  $\sigma_{xx}$ ,  $\sigma_{yy}$  and  $\sigma_{zz}$ , the three shear stresses  $\sigma_{xy}$ ,  $\sigma_{xz}$ , and  $\sigma_{yz}$ , and the three particle velocities  $u$ ,  $v$  and  $w$ .  $A(\vec{x})$ ,  $B(\vec{x})$ ,  $C(\vec{x})$  are the three space-dependent Jacobians, discretizing the seismic velocity model. Application of the ADER-DG machinery leads to a scheme, composed of sparse matrix-matrix operations [1]. These matrices are repeatedly applied to the degrees of freedom to advance the simulation in time and illustrated for a fourth order method in Fig. 2.

EDGE heavily exploits the fact that many of the grand challenges in earthquake system science require large ensembles of geometrically similar forward simulations. A traditional solver  $s$  operates on fixed input  $i$  to obtain observations  $o = s(i)$ . Now, if we require observations for  $n$  different inputs  $I_n = (i_1, i_2, \dots, i_n)$ , e.g., different seismic sources, we would execute the solver  $n$  times to obtain the set of observations  $O_n = (o_1, o_2, \dots, o_n) = (s(i_1), s(i_2), \dots, s(i_n))$ . In contrast, EDGE's solver  $S_m$  operates in parallel on  $m \leq n$  different inputs  $I_m = (i_1, i_2, \dots, i_m)$  to obtain a set of observations in a single execution:  $O_m = (o_1, o_2, \dots, o_m) = S_m(I_m)$ . The advantages of this basic paradigm range from higher data-reuse through shared data structures, e.g., the velocity model, towards better parallelization opportunities at all levels.

*Single vs. Double Precision Floating Point Arithmetic:* In the past, seismic ADER-DG simulations have been carried out in double precision arithmetic [2]–[6]. The reasons are: a) extensive verification was only performed for double precision, and b) potential speedups are annihilated through low SIMD parallelism of non-fused ADER-DG kernels. As part of a larger parameter study, c.f. App. A, we present verification results using the Layer Over Halfspace benchmark. We executed the benchmark for convergence rates  $\mathcal{O} \in \{2, \dots, 6\}$  in single and double precision arithmetic. We observe a good fit of EDGE's solutions with the reference, while the 32-bit synthetics are visually indistinguishable from the 64-bit solution (see App. A). In conjunction with our extensive multi-parameter study and other verification settings [7], we conclude that single precision arithmetic is sufficient for EDGE's wave propagation solver. This observation agrees with other seismic solvers, e.g., [8] or [9].

### III. KERNEL OPTIMIZATIONS

For the presented work we extended LIBXSMM heavily. This extension is many-fold: a) small dense regular GEMM

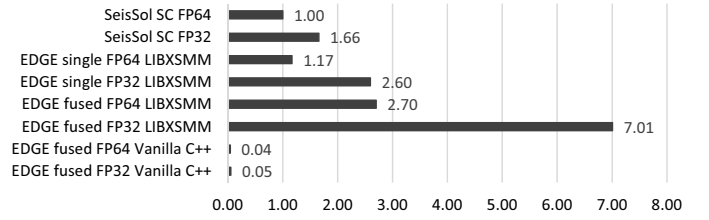


Fig. 3. Speed-up over the SC 2017 best paper award-winning work [2] on the KNM processor using fifth order and global time stepping.

with QFMA (Knight Mill) support, b) small dense fused GEMM with QFMA (Knights Mill) support (for fused simulation with low sparsity, e.g. the flux solver), c) sparse fused GEMM with QFMA (Knights Mill) support, d) small dense fused GEMM for regular AVX2 and AVX512 architectures e) sparse fused GEMM for regular AVX2 and AVX512 architectures. Our optimizations hardware the sparsity pattern of the sparse operator matrices to maximize execution of non-zero FLOPs. That means, runtime-compiling an assembly kernel into the instruction stream that exactly matches the sparse matrix operator. We optimized for low overheads stemming from processing the sparse matrix structure and selecting the corresponding DOF-matrix entries at runtime. A pseudo-code example for c) is given in App. A.

*Improvement Over State-Of-The-Art:* Fig. 3 depicts the speedups over the SC 2017 best paper award-winning work [2] on the Knights Mill processor using order  $\mathcal{O} = 5$ <sup>1</sup>. This is done by using a single node setup of the LOH.1 benchmark with 350,264 elements. Due to our JIT'ed assembly kernels we can outperform the SC 2017 work. This even is true without leveraging fused simulations or single precision arithmetic. The combined speedup of EDGE over [2] is 7.0 $\times$ , when exploiting FP32 and fused simulations as modeling parameters (see Fig. 1). When also running SeisSol in single precision, EDGE's fused simulation throughput is still 4.2 $\times$  higher. Since [2] only uses LIBXSMM in a static way, but not at runtime, SeisSol cannot make use of QFMA in case of the FP32 execution. Fig. 3 also highlights that our fused assembly kernels utilize the hardware much more efficiently than single runs or [2] because all multiplications with padded zeros disappear and perfect vector code is executed. Additionally, we can see why runtime code generation tailored to the sparse matrix operators is essential. The performance delivered by running EDGE using the Intel compiler with vectorization pragmas (denoted by C++) is extremely poor, although it executes fused simulations. Our runtime assembly kernel generators, that have been added to LIBXSMM, are able to outperform the compiler by 140 $\times$  (FP32) and 65 $\times$  (FP64) as the compiler fails to vectorize the dense C++ vanilla kernels, but also misses essential prefetching [10] and blocking strategies for ADER-DG.

<sup>1</sup>We compiled SeisSol according to [2]'s artifact description and confirmed that our binary reproduces the reported performance on the same hardware.

## REFERENCES

- [1] A. Breuer, A. Heinecke, and Y. Cui, "Edge: Extreme scale fused seismic simulations with the discontinuous galerkin method," in *International Supercomputing Conference*. Springer, 2017, pp. 41–60.
- [2] C. Uphoff, S. Rettenberger, M. Bader, E. H. Madden, T. Ulrich, S. Wollherr, and A.-A. Gabriel, "Extreme scale multi-physics simulations of the tsunamigenic 2004 sumatra megathrust earthquake," 2017.
- [3] A. Breuer, A. Heinecke, and M. Bader, "Petascale local time stepping for the ader-dg finite element method," in *Parallel and Distributed Processing Symposium, 2016 IEEE International*, 2016.
- [4] A. Heinecke, A. Breuer, M. Bader, and P. Dubey, "High order seismic simulations on the intel xeon phi processor (knights landing)," in *International Conference on High Performance Computing*, 2016.
- [5] A. Heinecke, A. Breuer, S. Rettenberger, M. Bader, A.-A. Gabriel, C. Pelties, A. Bode, W. Barth, X.-K. Liao, K. Vaidyanathan *et al.*, "Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014.
- [6] A. Breuer, A. Heinecke, S. Rettenberger, M. Bader, A. Gabriel, and C. Pelties, "Sustained petascale performance of seismic simulations with SeisSol on SuperMUC," in *International Supercomputing Conference*, 2014.
- [7] Alexander Breuer, Alexander Heinecke, Yifeng Cui, "EDGE: Benchmarking the Seismic Wave Propagation Solver," in *Proceedings of the 11th National Conference in Earthquake Engineering, Earthquake Engineering Research Institute, Los Angeles, CA. 2018 (to appear)*.
- [8] H. Fu, C. He, B. Chen, Z. Yin, Z. Zhang, W. Zhang, T. Zhang, W. Xue, W. Liu, W. Yin *et al.*, "18.9-pflops nonlinear earthquake simulation on sunway taihulight: enabling depiction of 18-hz and 8-meter scenarios," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2017, p. 2.
- [9] M. Rietmann, P. Messmer, T. Nissen-Meyer, D. Peter, P. Basini, D. Komatitsch, O. Schenk, J. Tromp, L. Boschi, and D. Giardini, "Forward and adjoint simulations of seismic wave propagation on emerging large-scale gpu architectures," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society Press, 2012, p. 38.
- [10] "Ch. 21," in *Intel Xeon Phi Processor High Performance Programming Knights Landing Edition*, J. Reinders, J. Jeffers, and A. Sodani, Eds., 2016.