SDSC
SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego

Tensor-Optimized Hardware Accelerates Fused Discontinuous Galerkin Simulations

Alexander Breuer, Alexander Heinecke (Intel), Yifeng Cui

The International Conference for High Performance Computing, Networking, Storage, and Analysis

Dallas, Texas
November 11-16, 2018

SC18 Dallas, TX | hpc inspires.

Extreme-scale Discontinuous Galerkin Environment (EDGE)

- ★ Focus: Problem settings with high geometric complexity, e.g., mountain topography
- ★ Unique support for fused simulations exploiting inter-simulation parallelism
- ★ Rapid prototyping through support for: Line elements, quads, triangles, hexes, tets
- ★ Parallelization: Assembly kernels for AVX, AVX2, AVX512 and AVX512_4FMA extensions, utilizing all x86 CPUs of the last five years optimally; OpenMP+MPI (custom and overlapping)
- ★ World record seismic wave propagation performance: 10.4 DP-PFLOPS on Cori II [1]
- ★ Continuity: Continuous Integration (sanity checks), Continuous Delivery (automated convergence + benchmarks runs), code coverage, license checks, container bootstrap
- ★ License: BSD 3-Clause (software), CC0 for supporting files, e.g., user guide

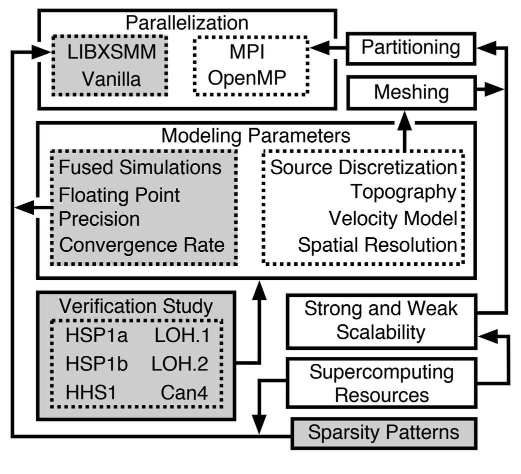
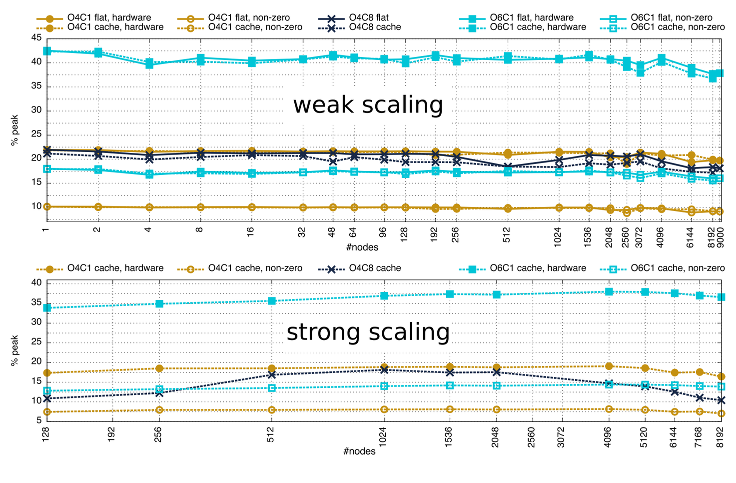
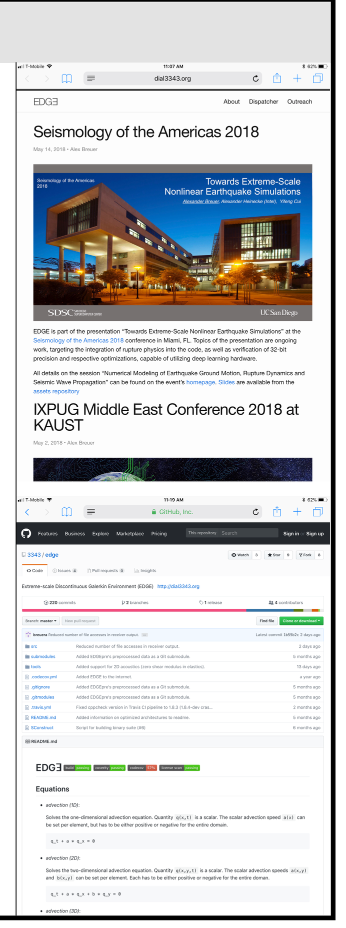


Illustration of EDGE's modeling and simulation pipeline. Science- and HPC-driven characteristics are tighly coupled, requiring a comprehensive approach. The parallelization (SIMD, shared memory, distributed memory) is influenced by a multitude of factors. The presented verification study is key in determining modeling parameters. Together with other parts of the pipeline, highlighted in gray, the study's results drive EDGE's single-core parallelization.



Top: Weak and strong scaling study on Cori-II (KNL). Shown are hardware and non-zero hardware peak efficiencies of all configurations in cache and flat mode. O denotes the order and C the number of fused simulations. The weak scaling uses a moderate total of 276,480 tetrahedrons per node. The mesh of the strong scaling contains 340,727,199 tetrahedrons, discretizing the LOH.1 benchmark (see below). [1]

Right: EDGE's homepage http://dial3343.org and software repository https://github.com/3343/edge.



Fused Simulations Exploit Inter-Simulation Parallelism

"Why is this a good idea?"

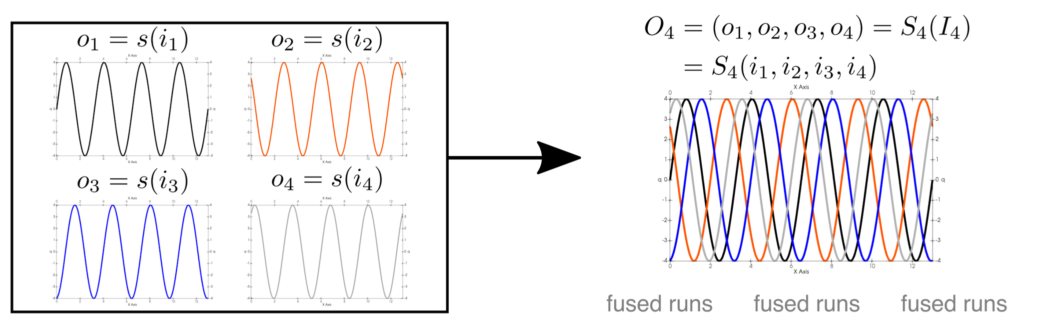
- ★ Idea: Exploit input parallelism by fusing multiple, similar simulations in a single execution of the solver
- ★ EDGE supports this idea at all levels of parallelism, starting at a single vector op
- ★ Fusing multiples of the vector-width (KNL, SKX, KNM): 16 simulations in single precision allow for perfect vectorization without zero ops
- ★ Fusion of multiples of 64 bytes (16 simulations) leads to alignment to cache-lines without artificial zero-padding
- ★ Read-only data structures are shared among all fused simulations

"Similar simulations?"

- ★ EDGE imposes restrictions on fused seismic simulations:
  1) Identical mesh for all fused simulations
  2) Identical simulation parameters: start and end time, convergence rate, "frequency" of wave field output, "frequency" and location of seismic receivers
  3) Identical material parameters (velocity model)
  4) "Sources" mostly arbitrary: Arbitrary initial DOFs, kinematic sources: arbitrary location and moment rates, spontaneous rupture: identical friction law, other initial parameters arbitrary

$$q(x,t)_t + v \cdot q(x,t)_x = 0,\ v \in \mathbb{R}$$

Illustration of EDGE's fused approach (4 simulations), applied to the advection equation with sinusoidal initial values. While a traditional solver handles different initial values in multiple executions, EDGE exploits the input-parallelism and computes the 4 simulations in parallel.



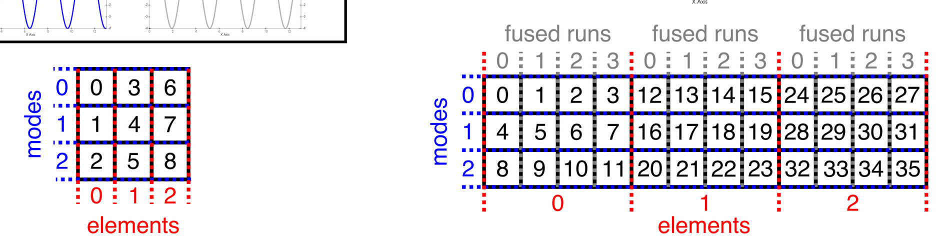
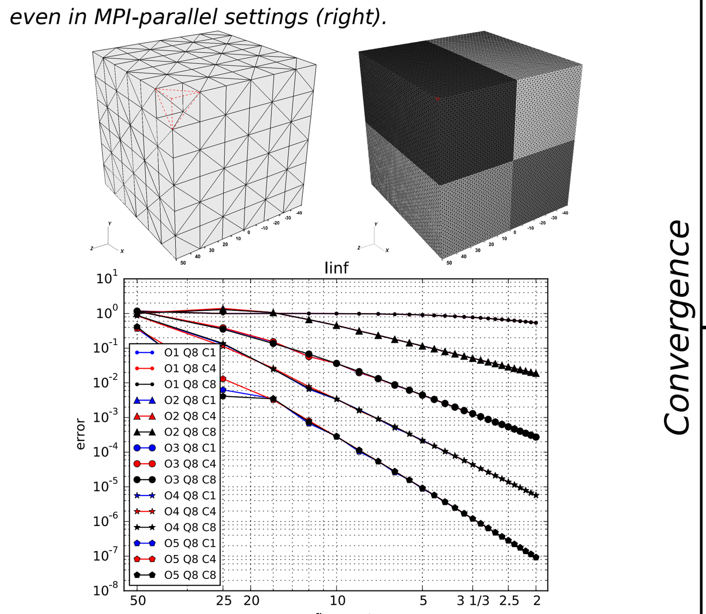


Illustration of EDGE's memory layout for a third order (P2 elements) ADER-DG solver for the advection equation. In the case of the traditional, non-fused approach, the DG-modes are the fastest dimension in memory, followed by the elements. In case of fused simulations, the fused runs are the fastest dimension, followed by the DG-modes and the elements.

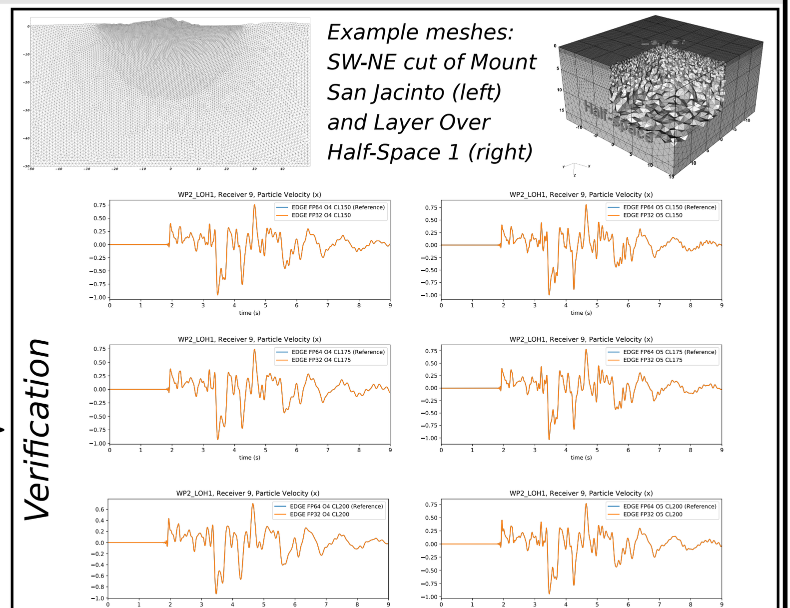
Contribution #1: Verification of 32bit Precision - Beyond Artificial Convergence Benchmarks

- ★ Benchmarking is key to assess the accuracy of seismic wave propagation solvers
- ★ EDGE has a multitude of modeling parameters:
  1) Fused vs. non-fused simulations
  2) Single vs. double precision
  3) Convergence rate in space and time
  4) Feature- and velocity-aware mesh refinement
  5) Source discretization
  6) Topography: Flat vs. DEM-derived
  7) Velocity model: Layered vs. data-input
- ★ Choosing the right modeling parameters is crucial for best time-to-solution
- ★ Our verification study considers the entire modeling and simulation pipeline and covers essential modeling decisions (1-3 above) for best practices

Two regular tetrahedral meshes. By imposing periodic boundary conditions, the plane wave initial solution is reproduced after diagonal propagation through the domain, even in MPI-parallel settings (right).



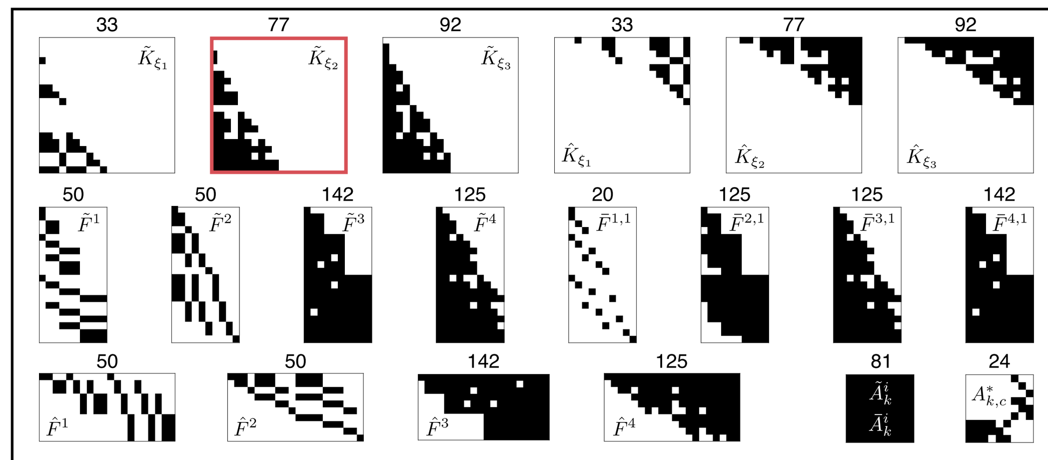
Convergence of EDGE for regular, tetrahedral meshes in the Linf-norm for the elastic wave equations in velocity-stress form. Shown are orders O1-O5 for quantity v (Q8) when utilizing EDGE's fusion capabilities with shifted initial conditions. [1]



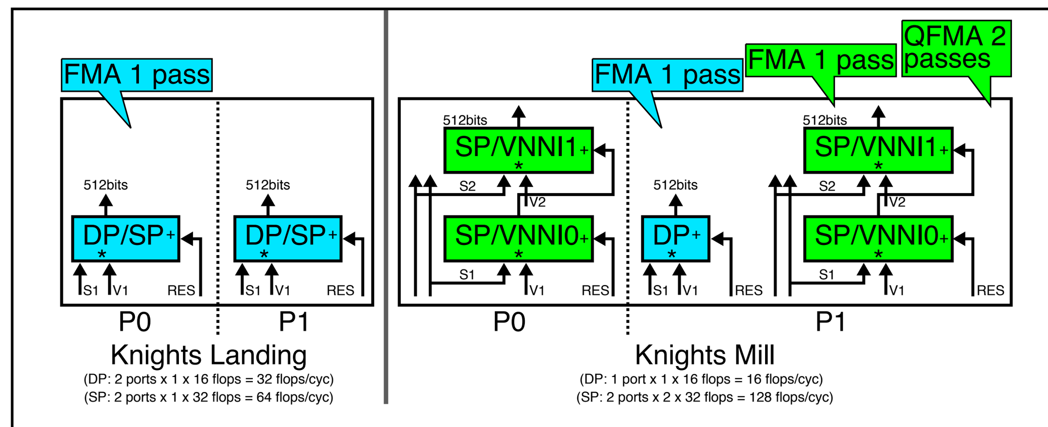
Synthetic seismograms for the ninth receiver and quantity u of the LOH.1 wave propagation benchmark. Shown is a comparison of single and double precision arithmetic in EDGE. The left synthetics show fourth order runs, the ones on the right fifth order runs. The two plots on top used a specified characteristic length of 150m, the middle plots 175m, and the bottom plots 200m. While higher resolutions and orders increase the accuracy, 32bit and 64bit precision are visually indistinguishable.

Contribution #2: Optimizing Fused Sparse Matrix-Matrix Kernels for Single Precision and All Orders

- ★ Hardware architectures are moving to an era of free computations and expensive data movement
- ★ Deep Learning is the new number one driver for hardware developments of all major vendors
- ★ Deep Learning hardware increases computation-intensive, reduced precision dense linear algebra capabilities, while sacrificing double precision performance
- ★ Verification study shows: Single precision arithmetic is sufficient for seismic wave propagation using ADER-DG
- ★ Extended previous fused DG-FEM efforts [1] to arbitrary orders of convergence and all recent x86 CPU architectures
- ★ Revised Just-In-Time (JIT) assembly kernel generation in the LIBXSMM library [2], targeting all recent x86 processors. Includes exploitation of Knights Mill's QFMA instruction in the code generation
- ★ Fused simulation technology is critical to utilize single precision performance: EDGE increases throughput by 4.2x over SeisSol [3] (16 Knights Mill processors, GTS, 16 fused FP32 simulations, fifth order)

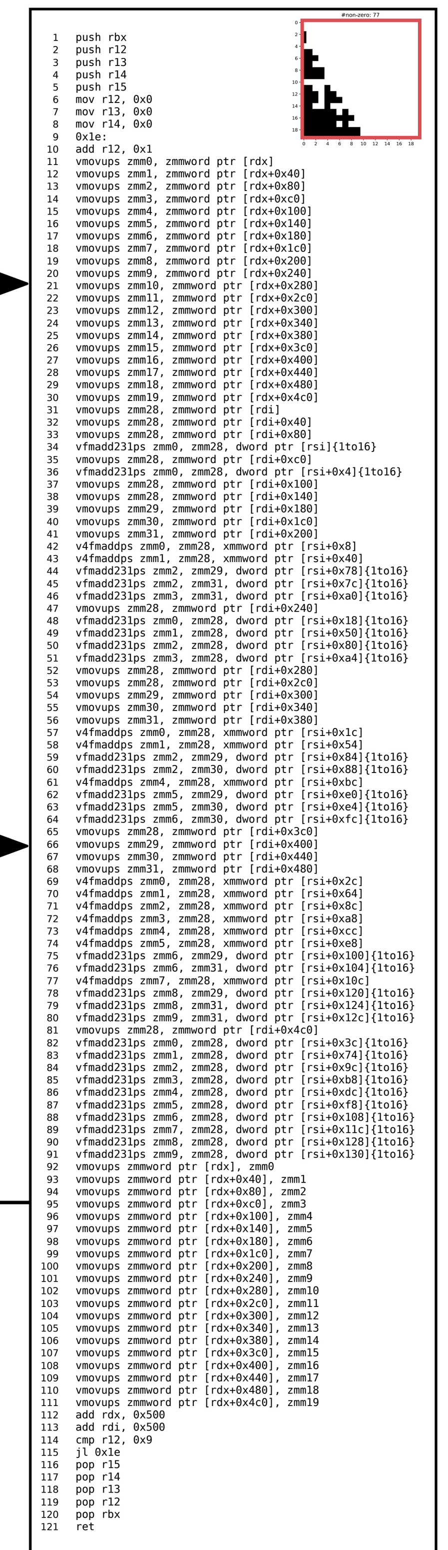


Sparsity patterns of the ADER-DG matrices for a 4th order method. On top of each matrix, the number of non-zero entries is given. The LIBXSMM generated sparse matrix-tensor assembly kernel for the red stiffness matrix and the Knight Mill architecture is shown on the right.



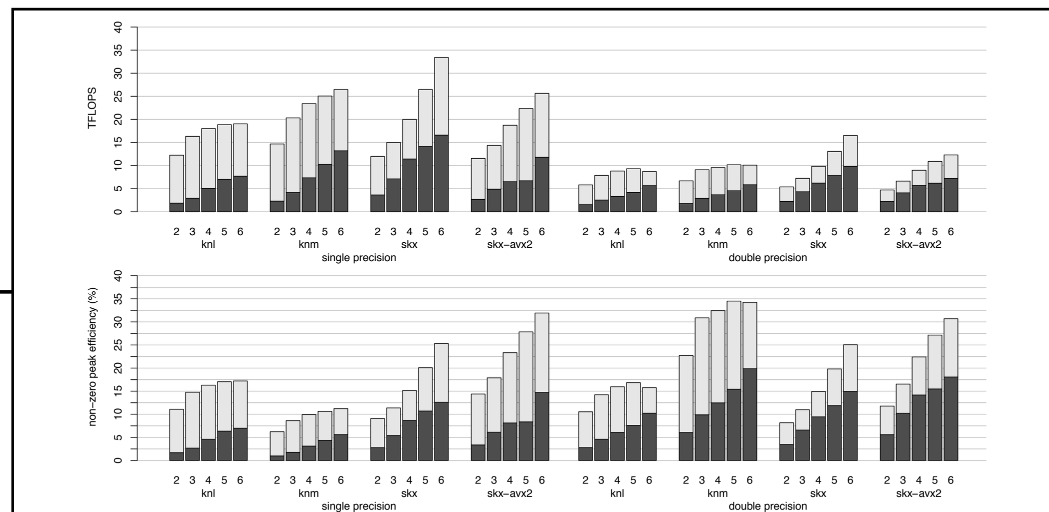
Knights Landing vs. Knights Mill VPU: a symmetric, single-pumped combo VPU is replaced by an asymmetric (single precision biased) VPU, which is double-pumped for high efficiencies on the two-issue wide Xeon Phi frontend. A LIBXSMM-generated kernel, using Knights Mill's QFMA instruction is shown on the right.

LIBXSMM



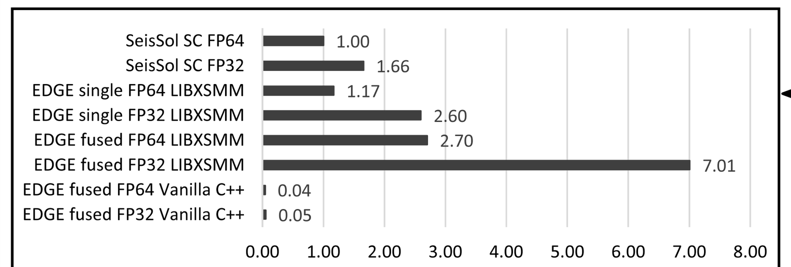
LIBXSMM-generated assembly sparse matrix-tensor kernel for the shown stiffness matrix, 16 fused simulations, Knights Mill, and single precision arithmetic. The kernel reaches over 35% of non-zero peak utilization, when running out of a hot cache.

MPI + OMP



EDGE's full-application performance on 16 nodes, when running the LOH.1 benchmark (see verification). All results are reported in terms of non-zero operations. Dark gray bars show single seismic forward simulations. Light gray bars show the performance of fused simulations. Eight simulations are fused in FP64 and sixteen in FP32 arithmetic. Dimensions: a) convergence rate, b) computer architecture: Xeon Phi 7250 (knl), Xeon Phi 7295 (knm), and 2× Intel Scalable Xeon 8180 using AVX512 (skx) and AVX2 (skx-avx2), c) floating point precision.

Speedup



Speed-up over the SC 2017 best paper award-winning solver SeisSol [3] on the Knights Mill processor using fifth order order of convergence. Setup is a single node variant of the LOH.1 benchmark with 350,264 elements and global time stepping. A speedup of up to 4.2x is obtained through the utilization of QFMA in FP32 execution and the exploitation of sparsity patterns in fused simulation technology.

General Applicability and Availability

- ★ Verification efforts transfer directly to DG-FEM solvers, relying on unstructured tetrahedral meshes, e.g., SeisSol or DGCrack
- ★ LIBXSMM optimizations transfer easily to community codes, using the library, e.g., CP2K, NekBox, Nek5000 or SeisSol
- ★ Parallelization through fused simulations / multiple right hand sides is a novel technique for improved efficiency. Approach is applicable to other software, facing challenging efficient SIMD utilization due to small matrix kernels, e.g., SpecFEM3D or Nek5000
- ★ Commercial libraries, e.g., Intel's MKL, recently added compact-vectorized routines, running batched BLAS routines. The underlying idea is similar, but would annihilate the presented speedups due to required element matrix duplication and zero padding. Suggests adoption on the library level
- ★ Artifact appendix of this poster contains details on the availability of all presented software (BSD-3) and data (CC0):
- ★ Presented LOH.1 verification runs are available from http://doi.org/10.17605/OSF.IO/H9G5N and shared with other modelers through http://sismowine.org/
- ★ Additional verification settings in 2D (Garvin's problem, Gaussian Hill, Gaussian Hill-Canyon and Mount San Jacinto topography) and in 3D (HHS1, HSP1a, HSP1b, LOH.1, LOH.2, Can.4) are available from http://opt.dial3343.org
- ★ All presented work on the solver EDGE is available from https://github.com/3343/edge
- ★ All presented work on the LIBXSMM library is available from https://github.com/hfp/libxsmm

References and Support

[1] EDGE: Extreme Scale Fused Seismic Simulations with the Discontinuous Galerkin Method - A. Breuer, A. Heinecke, Y. Cui. In High Performance Computing: 32nd International Conference, ISC High Performance 2017, Frankfurt, Germany, June 18-22, 2017, Proceedings

[2] LIBXSMM: accelerating small matrix multiplications by runtime code generation. - A. Heinecke, G. Henry, M. Hutchinson, M., H. Pabst. In SC '16 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis Article No. 84.

[3] Extreme scale multi-physics simulation of the tsunamigenic 2004 Sumatra megthrust earthquake - C. Uphoff, S. Rettenberger, M. Bader, E. H. Madden, T. Ulrich, S. Wollherr, A.-A. Gabriel. In SC'17 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis Article No. 21

This work was supported by the Southern California Earthquake Center (**SCEC**) through award #16247. This work is supported by SCEC through award #18211.

This research used resources of the National Energy Research Scientific Computing Center (**NERSC**), a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

This work used the Extreme Science and Engineering Discovery Environment (**XSEDE**), which is supported by National Science Foundation grant number ACI-1053575.

EDGE heavily relies on contributions of many authors to open-source software. This software includes, but is not limited to: ASan (debugger), Catch (unit tests), CGAL (surface meshing), Clang (compiler), Cppcheck (static code analysis), Easylogging++ (logging), ExprTk (expression parsing), GCC (compiler), Git (versioning), Git LFS (versioning), gitbook (documentation), Gmsh (volume meshing), GMT (DEM pre-processing), GoCD (continuous delivery), HDF5 (I/O), jekyll (homepage), LIBXSMM (matrix kernels), METIS (partitioning), MOAB (mesh interface), NetCDF (I/O), ParaView (visualization), Proj.4 (map projections), pugixml (XML interface), SAGA-Python (automated remote job submission), SCons (build tool), TF-MISFIT GOF CRITERIA (signal analysis), UCVMC (velocity model), Valgrind (memory debugging), Visit (visualization).

http://dial3343.org