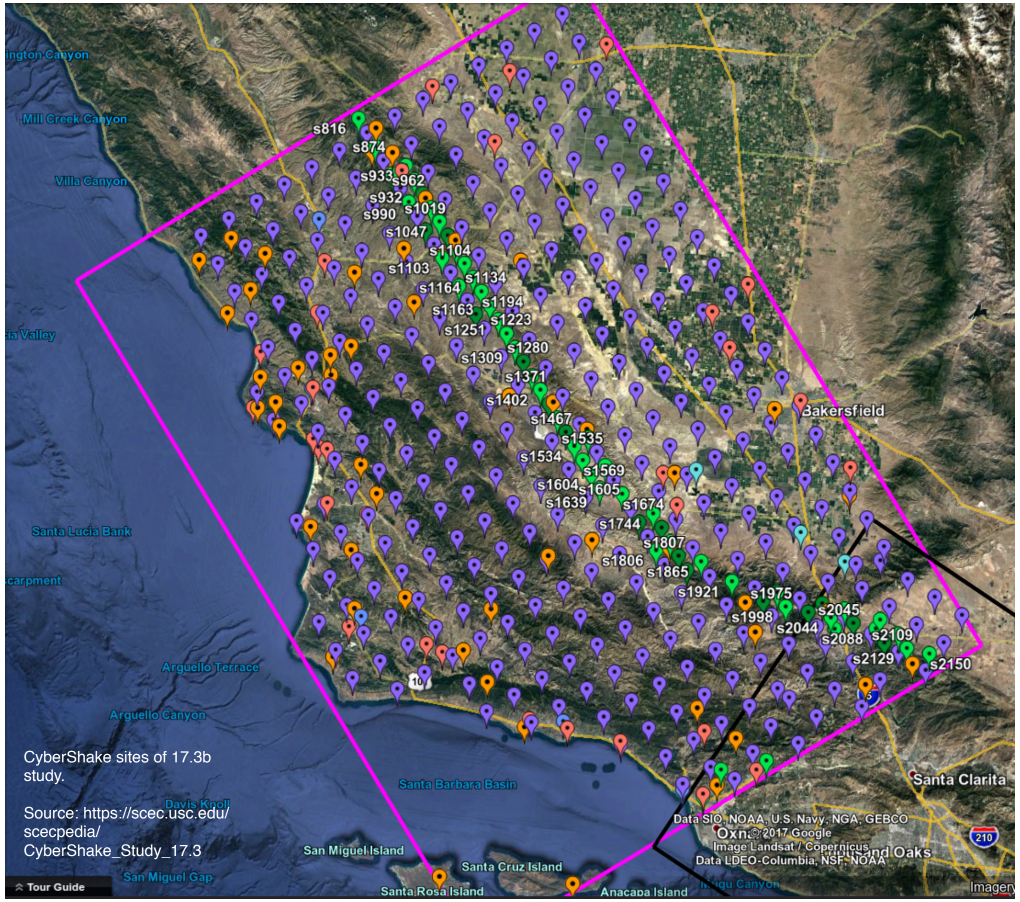
ISC High Performance
06/19/2019

Petaflop Seismic Simulations in the Public Cloud

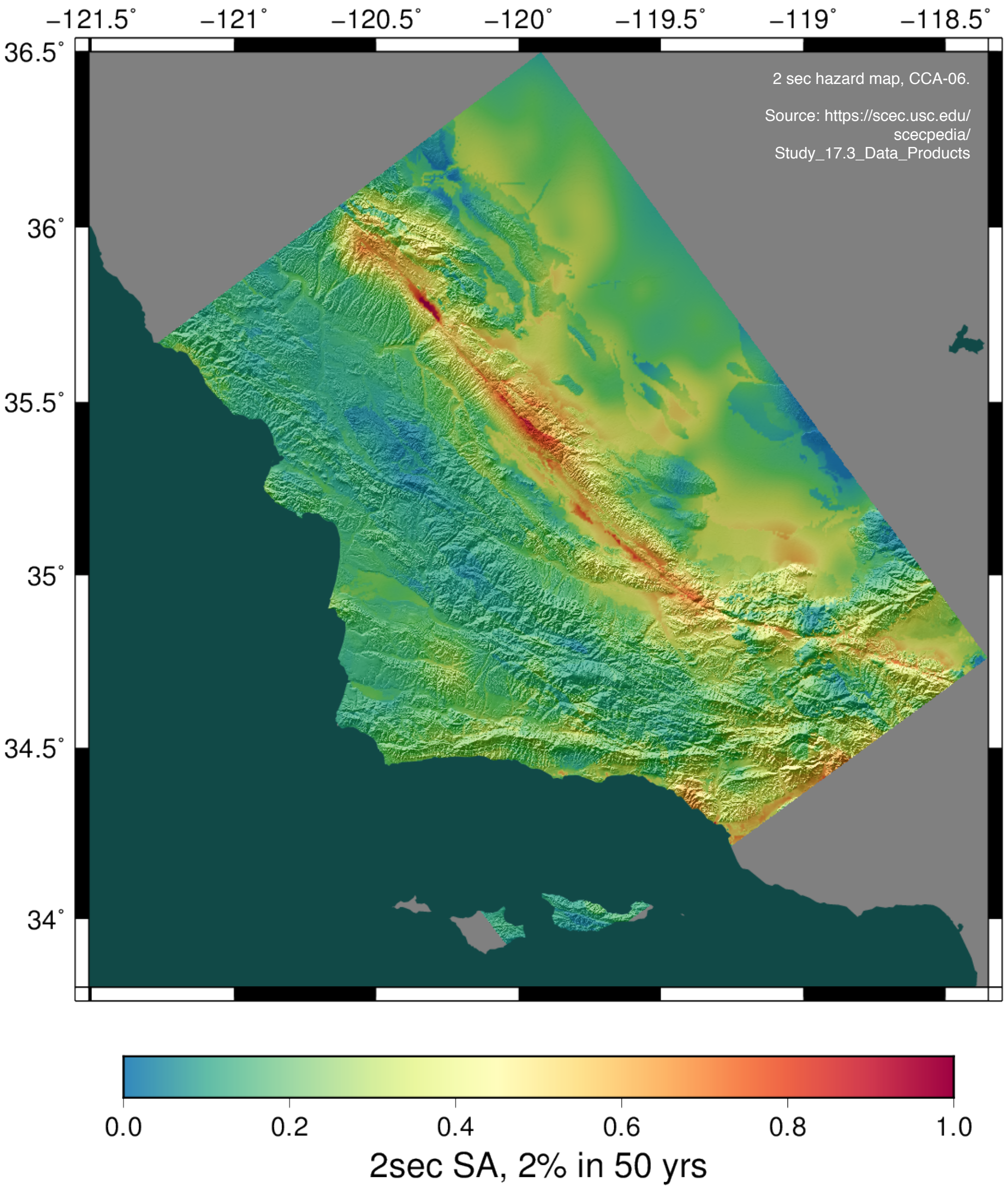
Alexander Breuer

SDSC SAN DIEGO SUPERCOMPUTER CENTER

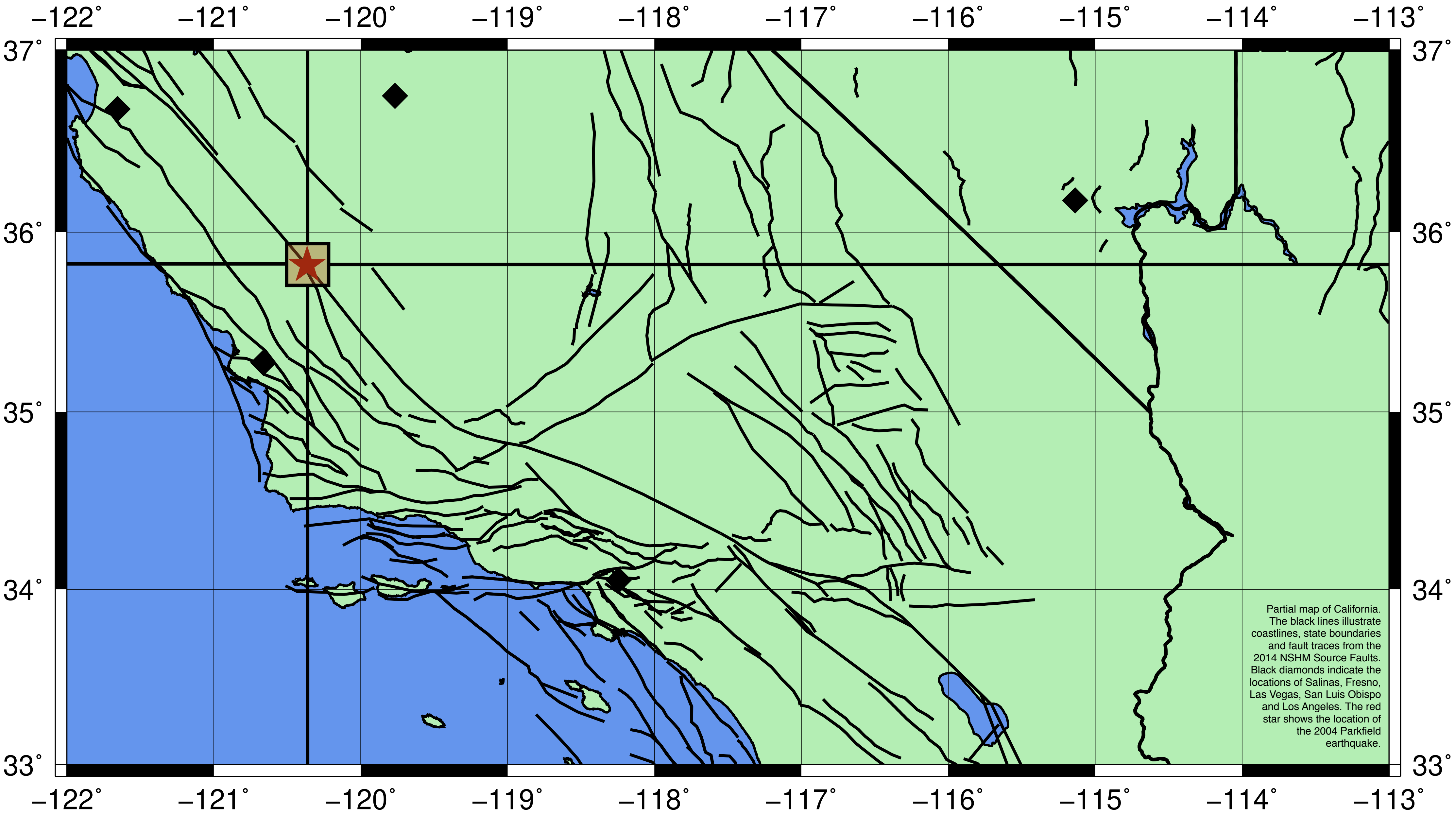
UC San Diego

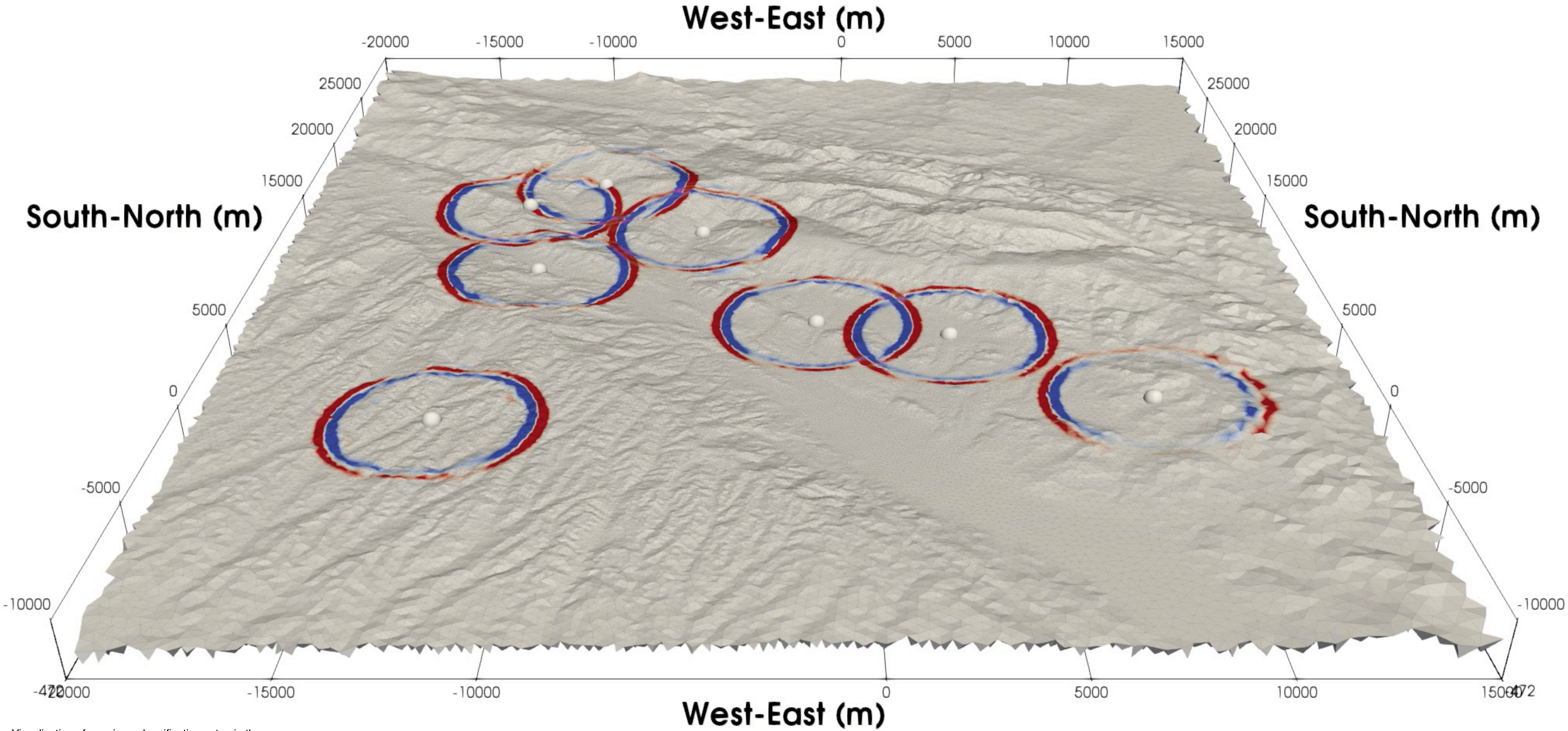


CyberShake sites of 17.3b study.
Source: https://scec.usc.edu/scecpedia/CyberShake_Study_17.3
s816
s874
s933
s962
s932
s1019
s990
s1047
s1104
s1103
s1134
s1164
s1194
s1163
s1223
s1251
s1280
s1309
s1371
s1402
s1467
s1535
s1534
s1569
s1604
s1605
s1639
s1674
s1744
s1807
s1806
s1865
s1921
s1975
s1998
s2044
s2045
s2088
s2109
s2129
s2150
Bakersfield
Santa Clarita
Mill Creek Canyon
Villa Canyon
Santa Lucia Bank
Arguello Terrace
Arguello Canyon
Santa Barbara Basin
San Miguel Island
Santa Cruz Island
Santa Rosa Island
Anacapa Island
San Miguel Gap
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
© 2017 Google
Image Landsat / Copernicus
Data LDEO-Columbia, NSF, NOAA
Tour Guide



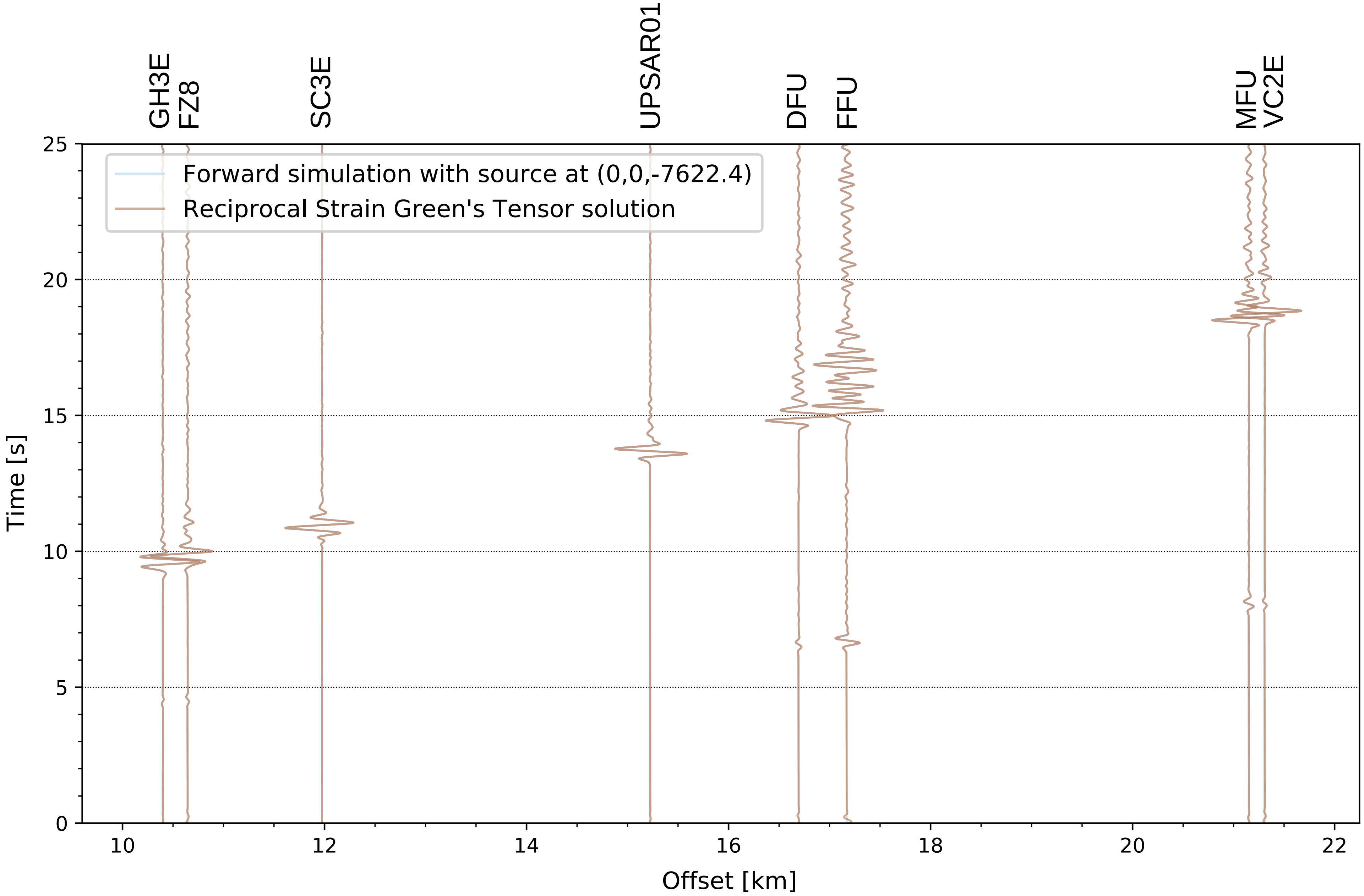
2 sec hazard map, CCA-06.
Source: https://scec.usc.edu/scecpedia/Study_17.3_Data_Products
−121.5˚
−121˚
−120.5˚
−120˚
−119.5˚
−119˚
−118.5˚
36.5˚
36˚
35.5˚
35˚
34.5˚
34˚
0.0
0.2
0.4
0.6
0.8
1.0
2sec SA, 2% in 50 yrs


–122°
–121°
–120°
–119°
–118°
–117°
–116°
–115°
–114°
–113°
37°
36°
35°
34°
33°
Partial map of California. The black lines illustrate coastlines, state boundaries and fault traces from the 2014 NSHM Source Faults. Black diamonds indicate the locations of Salinas, Fresno, Las Vegas, San Luis Obispo and Los Angeles. The red star shows the location of the 2004 Parkfield earthquake.



Visualization of a reciprocal verification setup in the Parkfield region of the San Andreas Fault. Shown are the South-North particle velocities for eight fused point forces at respective receiver locations.

Time (s): 3.25

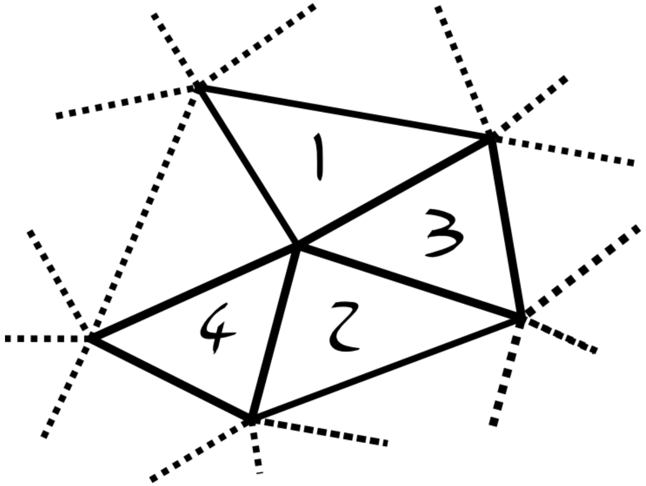


Comparison of post-processed point force simulations with a double-couple reference. Shown are the seismograms of the particle velocity in South-North direction for eight stations at the surface. The x-axis reflects hypocentral distance. The convolved SGTs are largely indistinguishable from the reference. At the very beginning of each seismogram, a small and expected offset is visible, since we processed the raw signals without tapering. [ISC19]

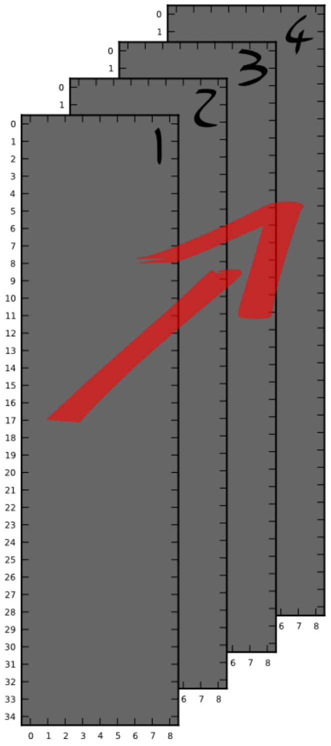
$$q_t + A^{x_1} q_{x_1} + A^{x_2} q_{x_2} + A^{x_3} q_{x_3} = 0$$

$$q(\vec{x},t) = \begin{pmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{33} \\ \sigma^{12} \\ \sigma^{23} \\ \sigma^{13} \\ u_1 \\ u_2 \\ u_3 \end{pmatrix} \qquad A^{x_1}(\vec{x}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -\lambda - 2\,\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu \\ -\rho^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\rho^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\rho^{-1} & 0 & 0 & 0 \end{pmatrix}$$

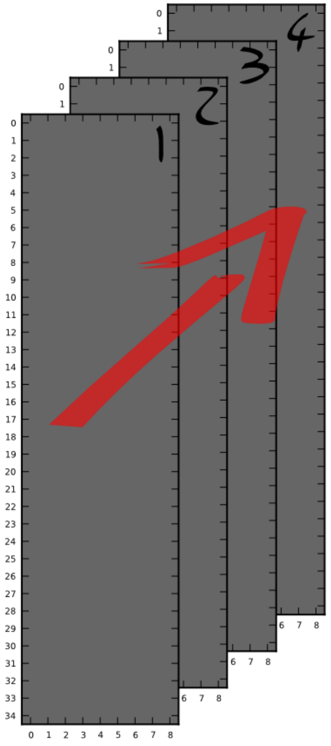
Local



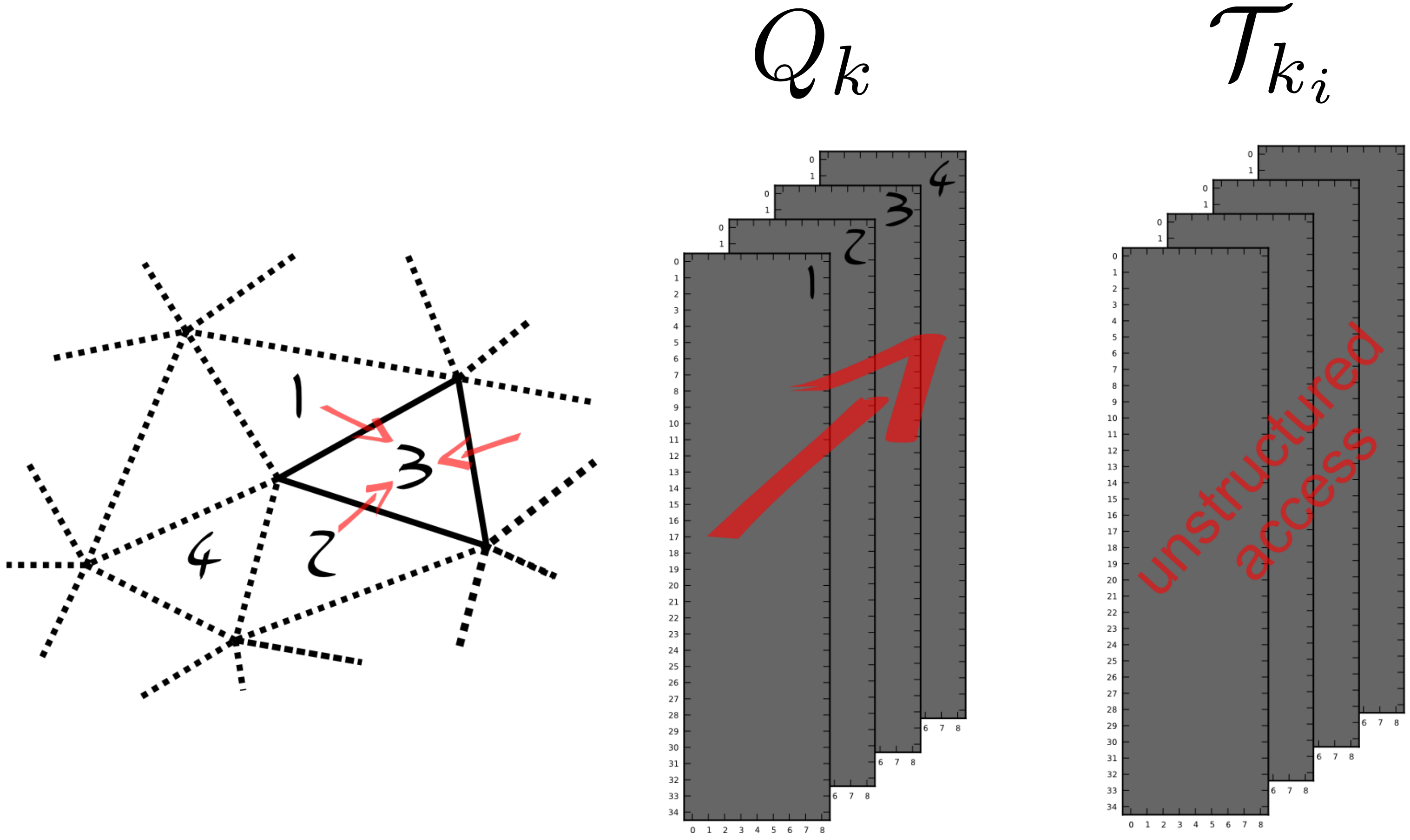
Q_k



$\mathcal{T}_k$

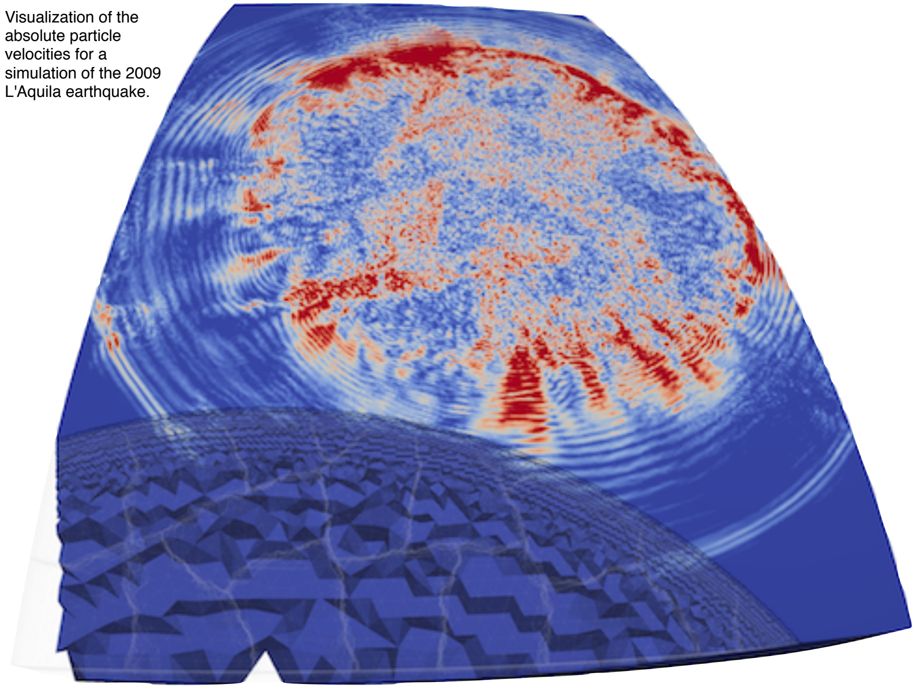


Neighboring

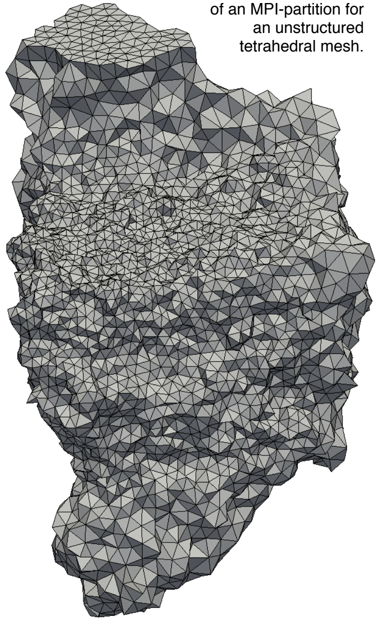


Solver

- Discontinuous Galerkin Finite Element Method (DG-FEM), ADER in time
- Full elastic wave equations in 3D and complex heterogeneous media
- Unstructured, conforming tetrahedral meshes
- Small sparse matrix operators in inner loops
- Compute bound (high orders)



Visualization of the absolute particle velocities for a simulation of the 2009 L'Aquila earthquake.



Exemplary illustration of an MPI-partition for an unstructured tetrahedral mesh.

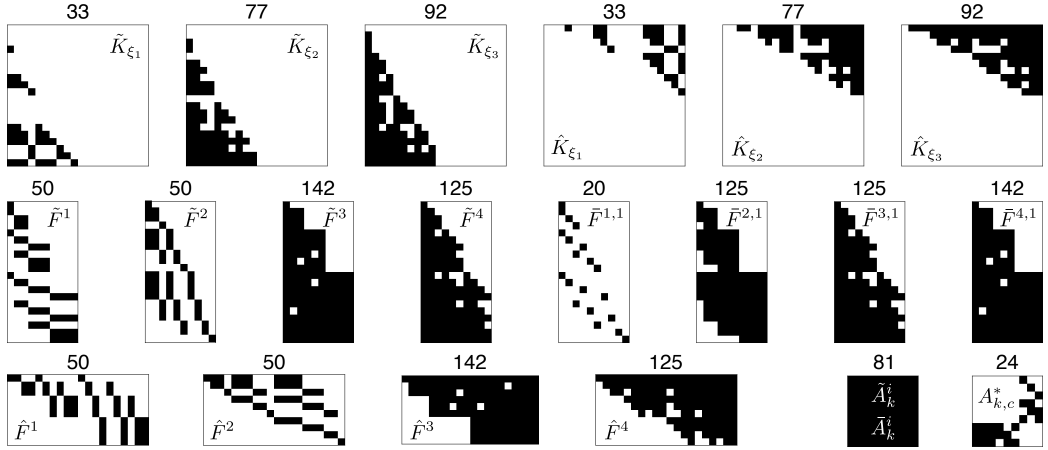


Illustration of all involves sparse matrix patterns for a fourth order ADER-DG discretization in EDGE. The numbers on top give the non-zero entries in the sparse matrices. [Parco18]

Weak Scaling Runs

Year	System	Architecture	Nodes	Cores	Order	Precision	HW-PFLOPS	NZ-PFLOPS	NZ-%Peak
2014	SuperMUC	SNB	9216	147456	6	FP64	1.6	0.9	26.6
2014	Stampede	SNB+KNC	6144	473088	6	FP64	2.3	1.0	11.8
2014	Tianhe 2	IVB+KNC	8192	1597440	6	FP64	8.6	3.8	13.5
2015	SuperMUC 2	HSW	3072	86016	6	FP64	2.0	1.0	27.6
2016	Theta	KNL	3072	196608	4	FP64	1.8	1.8	21.5
2016	Cori 2	KNL	9000	612000	4	FP64	5.0	5.0	18.1
2018	AWS EC2	SKX	768	27648	5	FP32	1.1	1.1	21.2

A collection of weak scaling runs for elastic wave propagation with ADER-DG. The runs had similar but not identical configurations. Details are available from the given sources.

Explanation of the columns:

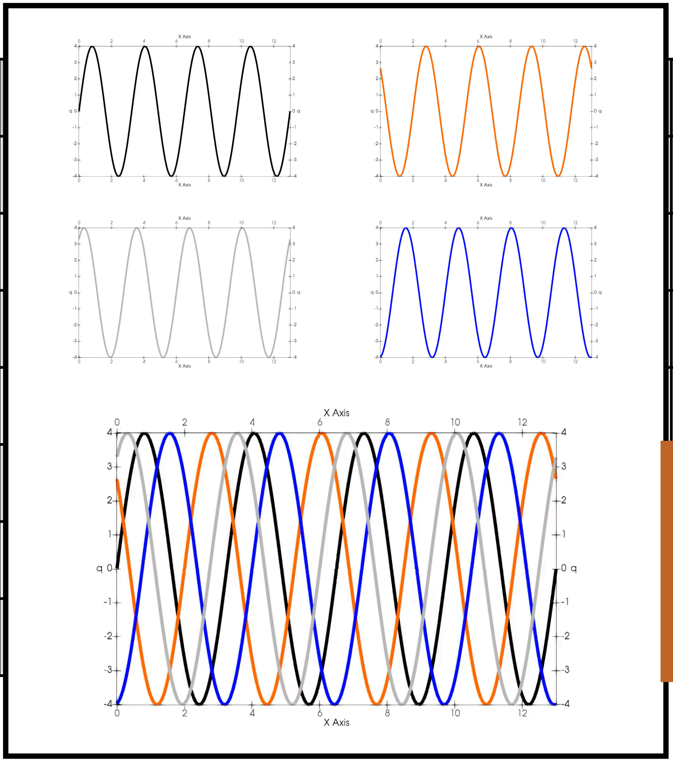
- System: Name of the system or cloud service (last row).
- Code-name of the used microarchitecture: Sandy Bridge (SNB), Ivy Bridge (IVB), Knights Corner (KNC), Haswell (HSW), Knights Landing (KNL), Skylake (SKX).
- Nodes: Used number of nodes in the run.
- Cores: Used number of cores in the run; includes host and accelerators cores for the heterogeneous runs.
- Order: Used order of convergence in the ADER-DG solver.
- Precision: Used floating point precision in the ADER-DG solver.
- HW-PFLOPS: Sustained Peta Floating-Point Operations Per Second (PFLOPS) in hardware.
- NZ-PFLOPS: Sustained Peta Floating-Point Operations Per Second (PFLOPS) if only non-zero operations are counted, i.e., ignoring artificial operations, introduced through dense matrix operators on sparse matrices.
- NZ-%Peak: Relative peak utilization, when comparing the machines' theoretical floating point performance to the sustained NZ-PFLOPS.

Sources:

- SuperMUC: [ISC14], [SC14]
- Stampede, Tianhe-2: [SC14]
- SuperMUC 2: [IPDPS16]
- Theta, Cori: [ISC17]
- AWS EC2: [ISC19]

Introduction of “Mini-batches” for PDEs

Year	System	Order	Precision	HW-PFLOPS	NZ-PFLOPS	NZ-%Peak
2014	SuperMUC	6	FP64	1.6	0.9	26.6
2014	Stampede	6	FP64	2.3	1.0	11.8
2014	Tianhe 2	6	FP64	8.6	3.8	13.5
2015	SuperMUC 2	6	FP64	2.0	1.0	27.6
2016	Theta	4	FP64	1.8	1.8	21.5
2016	Cori 2	4	FP64	5.0	5.0	18.1
2018	AWS EC2	5	FP32	1.1	1.1	21.2



A collection of weak scaling runs for elastic wave propagation with ADER-DG. The runs had similar but not identical configurations. Details are available from the given sources.

Explanation of the columns:

- System: Name of the system or cloud service (last row).
- Code-name of the used microarchitecture: Sandy Bridge (SNB), Ivy Bridge (IVB), Knights Corner (KNC), Haswell (HSW), Knights Landing (KNL), Skylake (SKX).
- Nodes: Used number of nodes in the run.
- Cores: Used number of cores in the run; includes host and accelerators cores for the heterogeneous runs.
- Order: Used order of convergence in the ADER-DG solver.
- Precision: Used floating point precision in the ADER-DG solver.
- HW-PFLOPS: Sustained Peta Floating-Point Operations Per Second (PFLOPS) in hardware.
- NZ-PFLOPS: Sustained Peta Floating-Point Operations Per Second (PFLOPS) if only non-zero operations are counted, i.e., ignoring artificial operations, introduced through dense matrix operators on sparse matrices.
- NZ-%Peak: Relative peak utilization, when comparing the machines’ theoretical floating point performance to the sustained NZ-PFLOPS.

Sources:

- SuperMUC: [ISC14], [SC14]
- Stampede, Tianhe-2: [SC14]
- SuperMUC 2: [IPDPS16]
- Theta, Cori: [ISC17]
- AWS EC2: [ISC19]

Cloud Computing

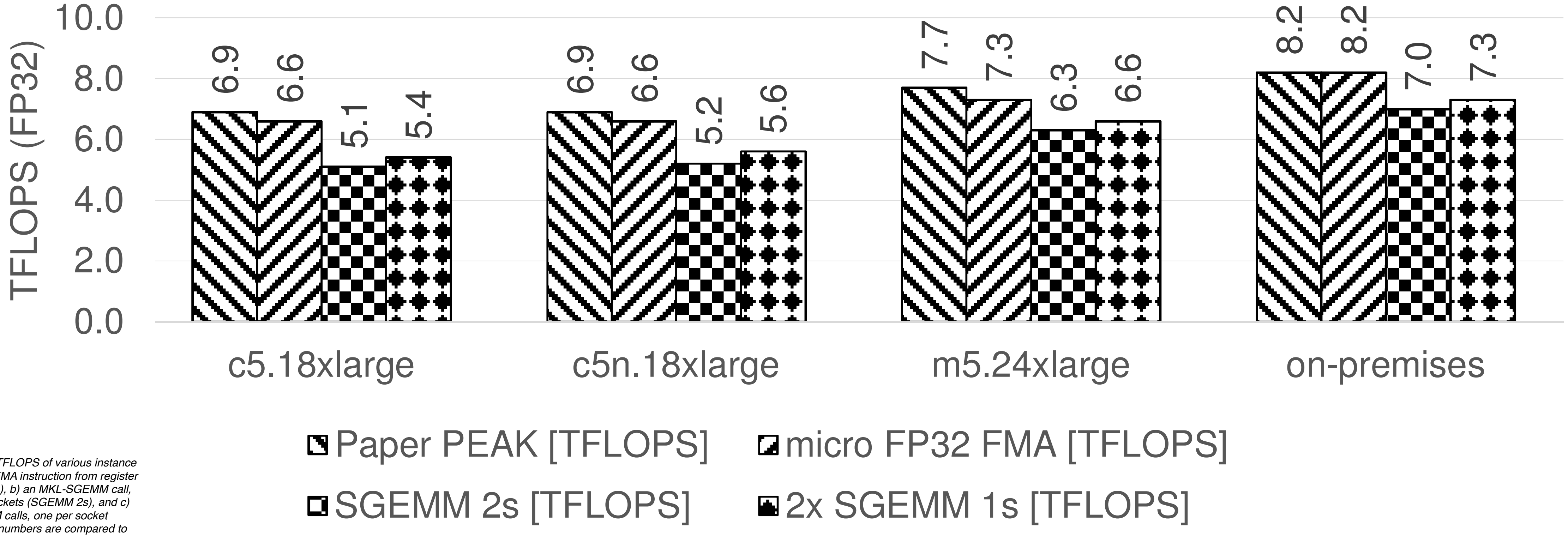
- *Micro-Benchmarks*
- *Machine Setup*
- *Performance Evaluation*

Key Performance Indicators (KPIs)

KPI	c5.18xlarge	c5n.18xlarge	m5.24xlarge	on-premises
CSP	Amazon	Amazon	Amazon	N/A
CPU name	8124M*	8124M*	8175M*	8180
#vCPU (incl. SMT)	2x36	2x36	2x48	2x56
#physical cores	2x18**	2x18**	2x24**	2x28
AVX512 Frequency	≤3.0GHz	≤3.0GHz	≤2.5GHz	2.3GHz
DRAM [GB]	144	192	384	192
#DIMMs	2x10?	2x12?	2x12/24?	2x12
spot $/h	0.7	0.7	0.96	N/A
on-demand $/h	3.1	3.9	4.6	N/A
interconnect [Gbps]	25***(eth)	25***/100***(eth)	25***(eth)	100(OPA)

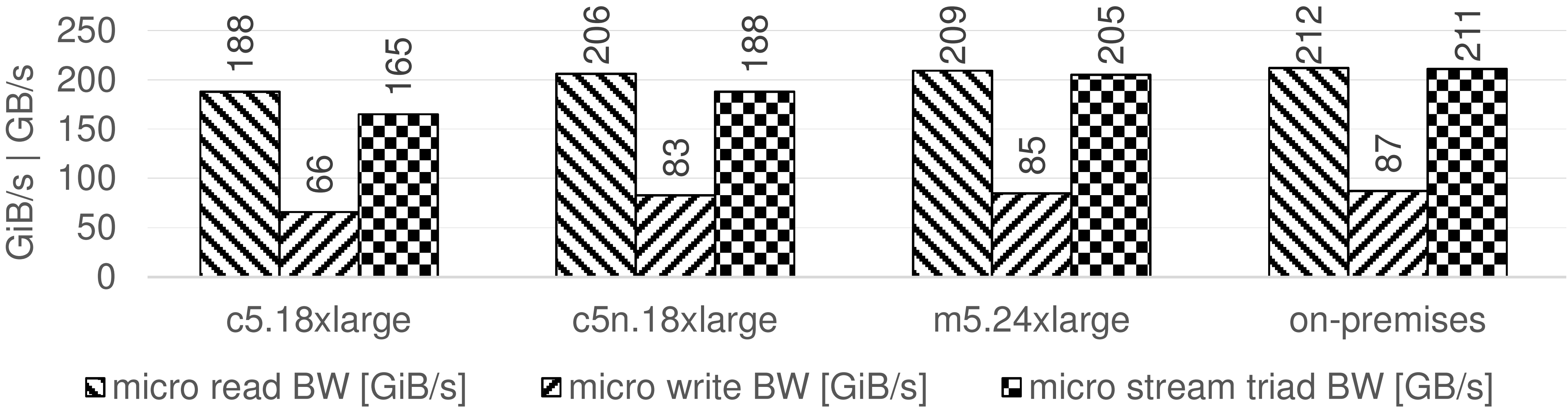
Publicly available KPIs for various cloud instance types of interest to our workload. Pricing is for US East at non-discount hours on Monday mornings (obtained on 3/25/19).
100Gbps for c5n.18xlarge reflects a recent update of the instance types (mid 2019).
*AWS CPU core name strings were retrieved using the "lscpu" command; **AWS physical cores are assumed from AWS's documentation, indicating that all cores are available to the user due to the Nitro Hypervisor; ***supported in multi-flow scenarios (means multiple communicating processes per host).

Micro-Benchmarking: 32-bit Floating Point



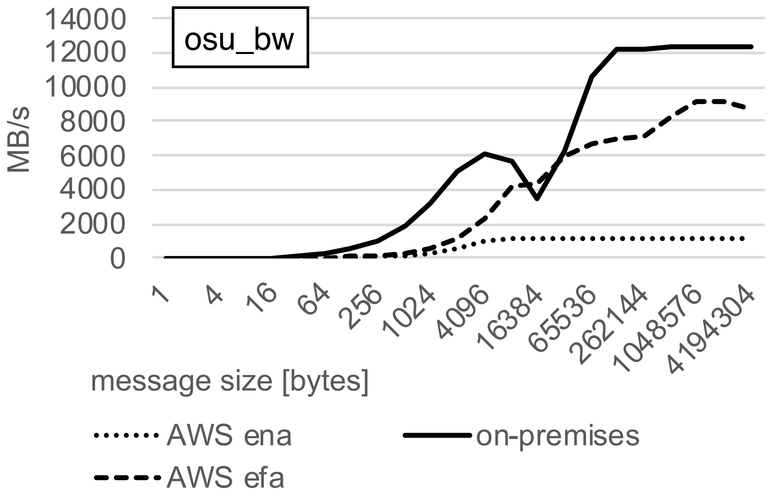
Sustained FP32-TFLOPS of various instance types: a) simple FMA instruction from register (micro FP32 FMA), b) an MKL-SGEMM call, spanning both sockets (SGEMM 2s), and c) two MKL-SGEMM calls, one per socket (SGEMM 1s). All numbers are compared to the expected AVX512 turbo performance (Paper PEAK).
on-premises: dual-socket Intel Xeon Platinum 8180, 2x12 DIMMs. [ISC19]

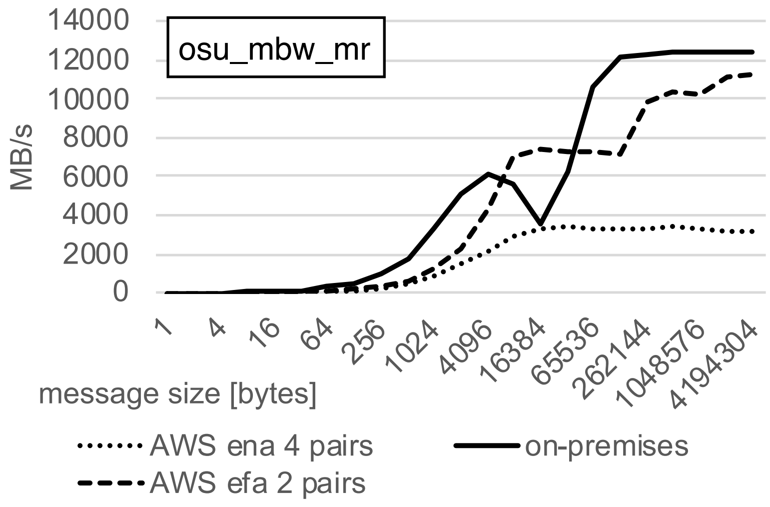
Micro-Benchmarking: Memory

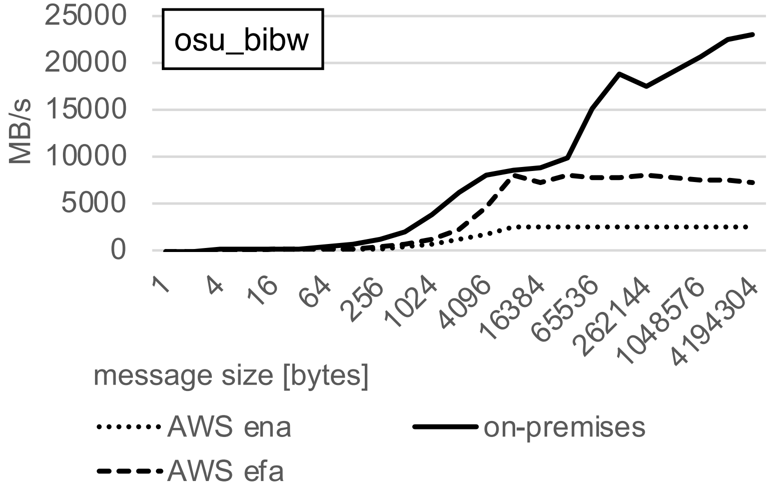


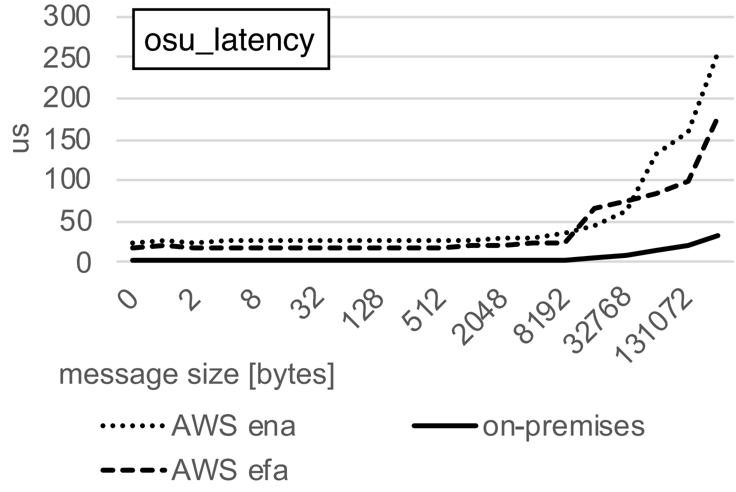
Sustained bandwidth of various instance types: a) a pure read-bandwidth benchmark (read BW), b) a pure write-bandwidth benchmark (write BW), and c) the classic STREAM triad with 2:1 read-to-write mix (stream triad BW).
on-premises: dual-socket Intel Xeon Platinum 8180, 2x12 DIMMS. [ISC19]

Micro-Benchmarking: Network





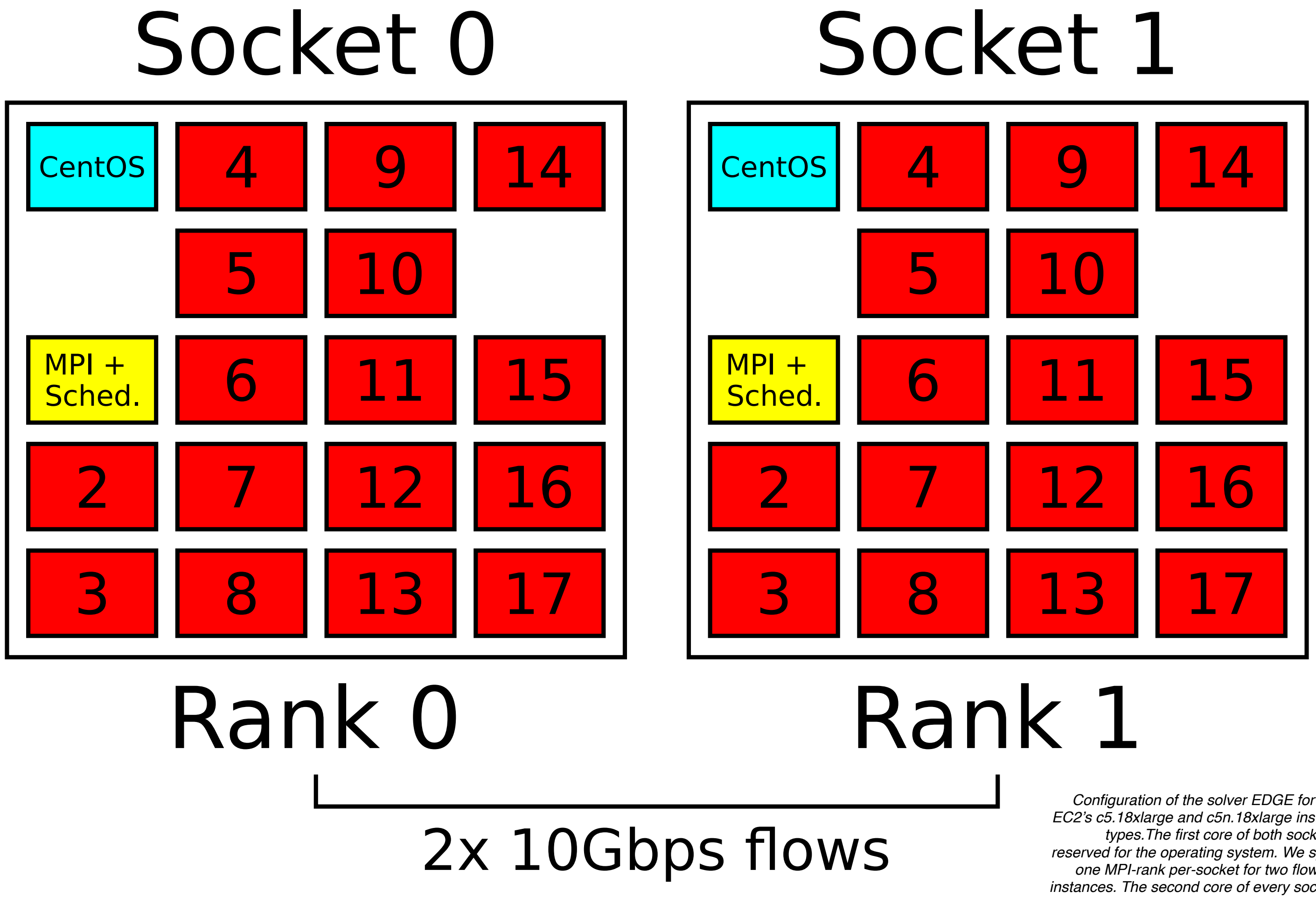




Interconnect performance of c5.18xlarge (AWS ena), c5n.18xlarge (AWS efa) and the on-premises, bare-metal system. Shown are results for the benchmarks osu_bw, osu_mbw_mr, osu_bibw and osu_latency (version 5.5). on-premises: dual-socket Intel Xeon Platinum 8180, 2x12 DIMMS, Intel OPA (100Gbps).

Machine Setup

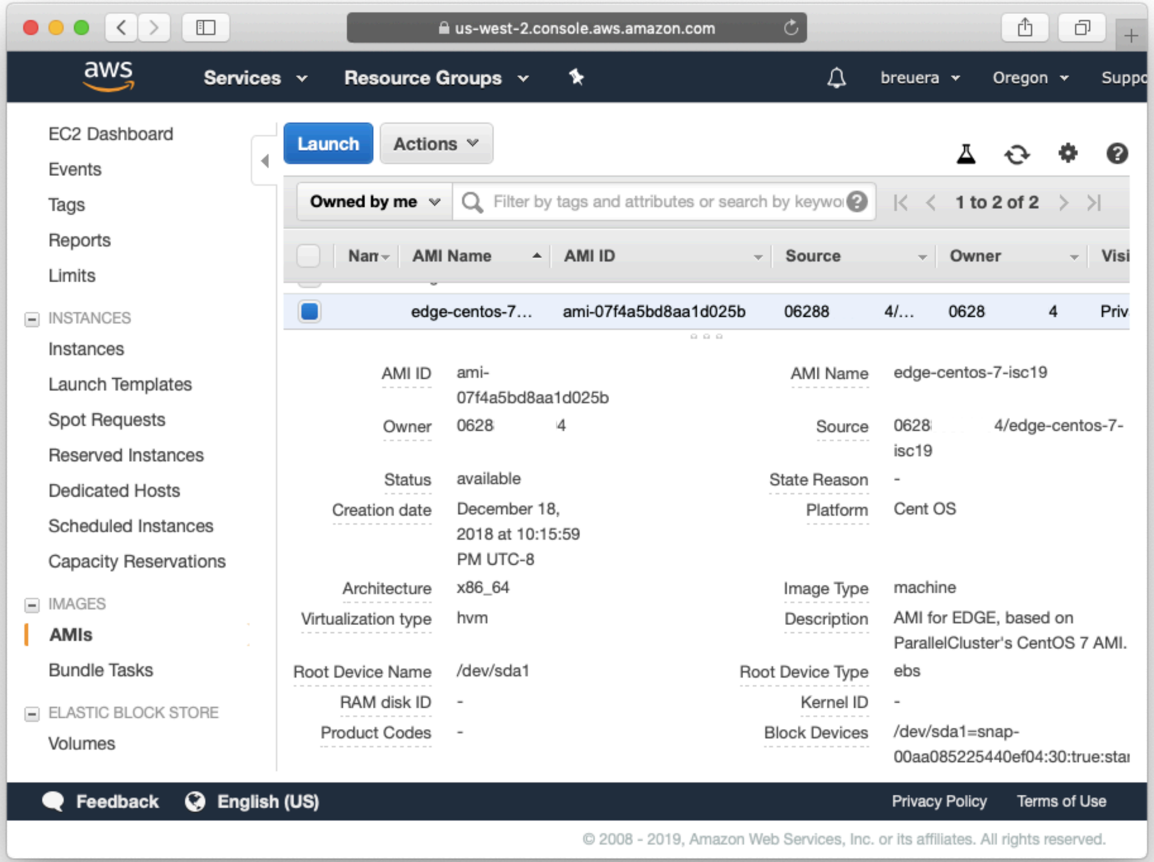
1. Select instance type
2. Create machine image:
   - OS customization: core specialization, C-states, huge pages, TCP tuning, ..
   - System-wide installation of tools and dependencies
3. Create Slurm-based cluster:
   - Compute nodes / instances boot customized machine image
4. Run jobs as on every other supercomputer



Configuration of the solver EDGE for AWS EC2's c5.18xlarge and c5n.18xlarge instance types. The first core of both sockets is reserved for the operating system. We spawn one MPI-rank per-socket for two flows per instances. The second core of every socket is reserved for our scheduling and MPI-progression thread. The remaining 16 cores of every socket are occupied by the 16 worker threads per rank.

Machine Setup

1. Select instance type — Center
2. Create machine image: — Vendor or Center
   - OS customization: core specialization, C-states, huge pages, TCP tuning, ..
   - System-wide installation of tools and dependencies
3. Create Slurm-based cluster: — Center
   - Compute nodes / instances boot customized machine image
4. Run jobs as on every other supercomputer — User

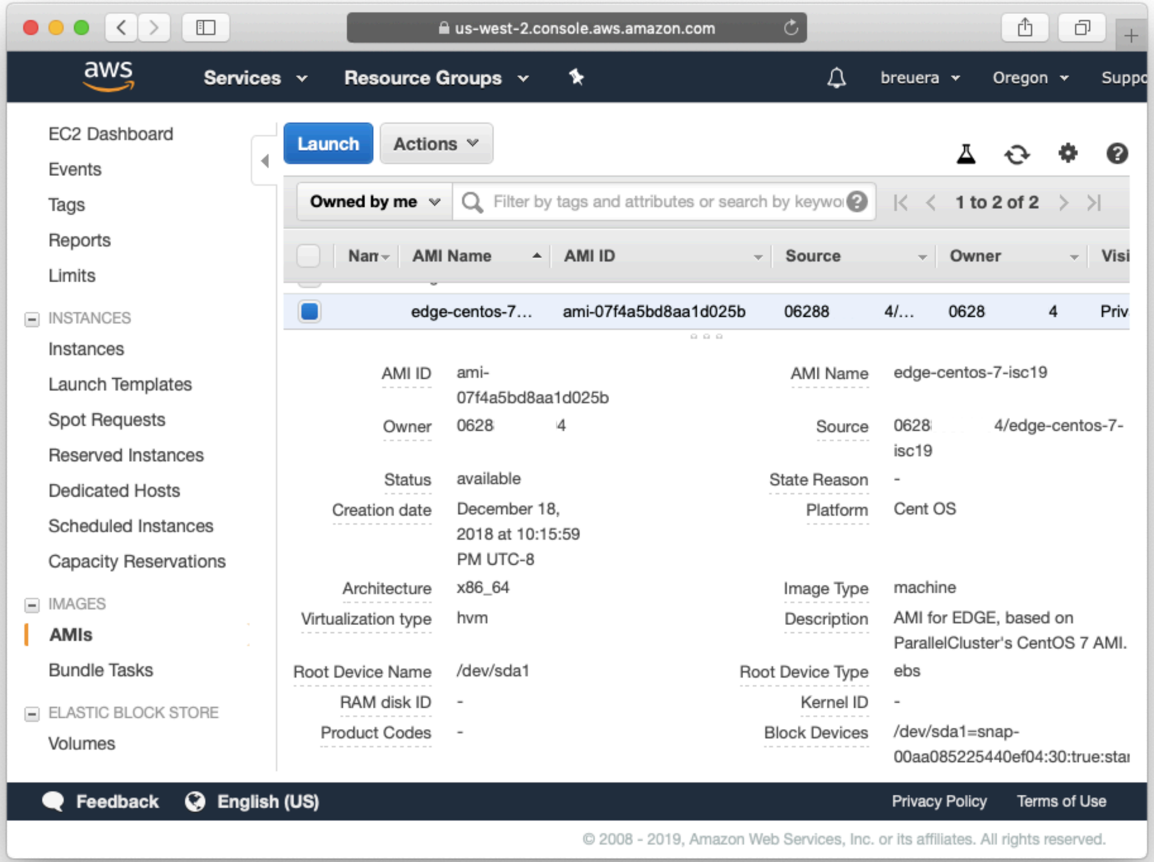


Screenshot showing the AWS Console for the Amazon Machine Image, used in [ISC19]'s large-scale simulations.

Machine Setup

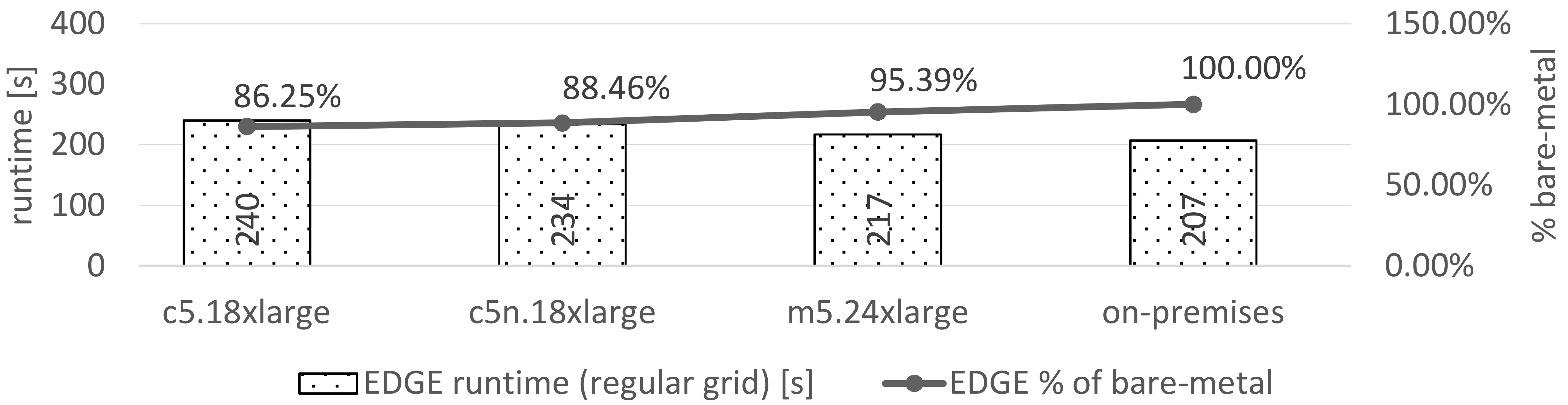
1. Select instance type
2. Create machine image:
    - OS customization: core specialization, C-states, huge pages, TCP tuning, ..
    - System-wide installation of tools and dependencies
3. Create Slurm-based cluster:
    - Compute nodes / instances boot customized machine image
4. Run jobs as on every other supercomputer

100% Open Source



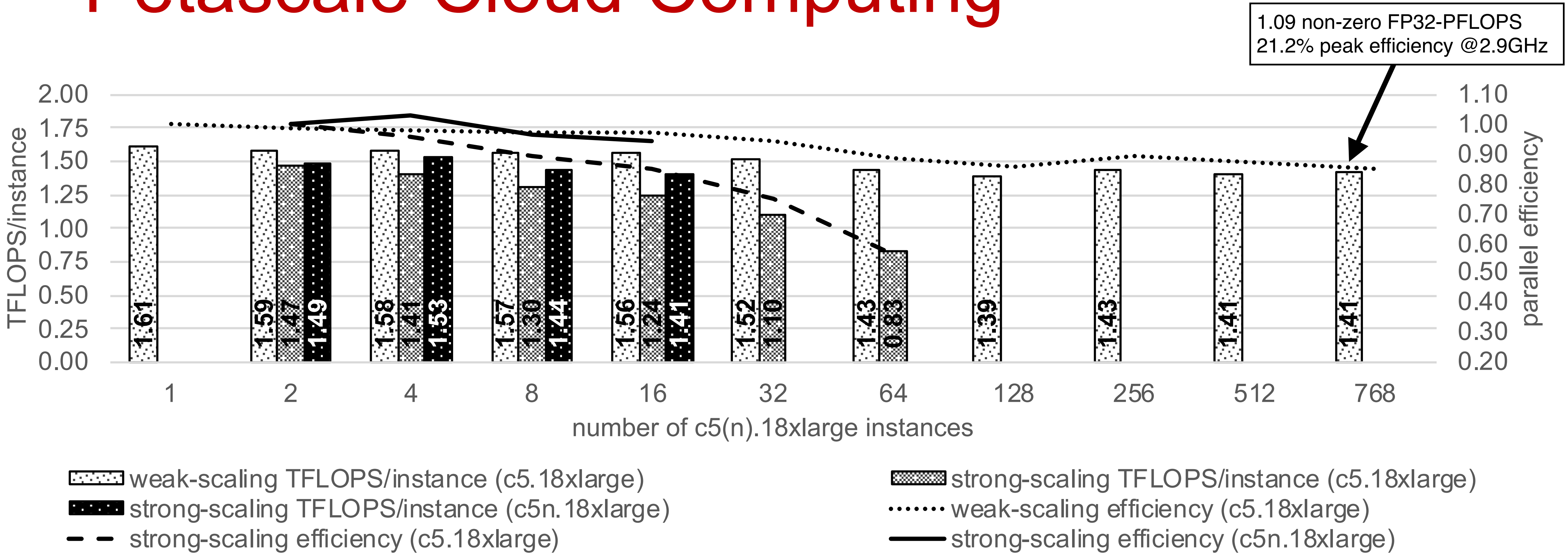
Screenshot showing the AWS Console for the Amazon Machine Image, used in [ISC19]'s large-scale simulations.

Cloud Virtualization vs. Bare Metal



Runtime of a regular setup of EDGE. As expected, all cloud instances are slower than the top-bin bare-metal machine. AWS instances are within 85% of the on-premises performance. on-premises: dual-socket Intel Xeon Platinum 8180, 2x12 DIMMS, Intel OPA (100Gbps). [ISC19]

Petascale Cloud Computing



Weak and strong scalability of EDGE in AWS EC2 on c5.18xlarge and c5n.18xlarge instances. We sustained 1.09 PFLOPS in weak-scaling on 768 c5.18xlarge instances. This elastic high performance cluster contained 27,648 Skylake-SP cores with a peak performance of 5 PFLOPS.

Part of a Comprehensive Approach

- Machine:
    - Hardware selection
    - OS customization
    - HPC Environment

- Single Node:
    - Kernels
    - Custom OpenMP and load balancing
    - Memory Layout
- Multi Node:
    - Overlapping communication and computation
    - Prioritization of crucial work
    - Communication “as is”, no additional MPI-buffers
- Algorithmic: Clustered Local Time Stepping (LTS), fused simulations
- Software Engineering:
    - CI/CD, continuous verification
    - Workflow automation
    - Software and data sharing
- Modeling and Simulation:
    - Model extensions
    - Surface meshing
    - Volume meshing
    - Mesh annotations
- Data Analysis:
    - Verification
    - SGT assembly and processing

Outlook: Beyond Petascale

Year	System	Architecture	Nodes	Cores	Order	Precision	HW-PFLOPS	NZ-PFLOPS	NZ-%Peak
2018	AWS EC2	SKX	768	27648	5	FP32	1.1	1.1	21.2
2016	Cori 2	KNL	9000	612000	4	FP64	5.0	5.0	18.1
2015	SuperMUC 2	HSW	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2016	Theta	KNL	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2014	Tianhe 2	IVB+KN	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2014	Stampede	SNB+KN	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2014	SuperMUC	SNB	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Current:
- 25Gbps c5.18xlarge (limited to 20Gbps in our configuration)
- Spot-instances and us-west-2 (Oregon)

Outlook:
- 100Gbps network closes gap to on-premises solutions
- Cloud is (much) bigger than our run (general purpose CPUs); what is the limit?

A collection of weak scaling with ADER-DG. The runs had similar but not identical configurations. Details are available from the given sources.

Explanation of the columns:
- System: Name of the system or cloud service (first row).
- Code-name of the used microarchitecture: Sandy Bridge (SNB), Ivy Bridge (IVB), Knights Corner (KNC), Haswell (HSW), Knights Landing (KNL), Skylake (SKX).
- Nodes: Used number of nodes in the run.
- Cores: Used number of cores in the run; includes host and accelerators cores for the heterogeneous runs.
- Precision: Used floating point precision in the ADER-DG solver.
- HW-PFLOPS: Sustained Peta Floating-Point Operations Per Second (PFLOPS) in hardware.
- NZ-PFLOPS: Sustained Peta Floating-Point Operations Per Second (PFLOPS) if only non-zero operations are counted, i.e., ignoring artificial operations, introduced through dense matrix operators on sparse matrices.
- NZ-%Peak: Relative peak utilization, when comparing the machines' theoretical floating point performance to the sustained NZ-PFLOPS.

- SuperMUC: [ISC14], [SC14]
- Stampede, Tianhe-2: [SC14]
- SuperMUC 2: [IPDPS16]
- Theta, Cori: [ISC17]
- AWS EC2: [ISC19]

References

- [ISC19] A. Breuer, Y. Cui, A. Heinecke. Petaflop Seismic Simulations on Elastic Cloud Clusters. In International Conference on High Performance Computing. Springer, Cham, 2019.
- [ISC17] A. Breuer, A. Heinecke, Y. Cui. EDGE: Extreme Scale Fused Seismic Simulations with the Discontinuous Galerkin Method. In High Performance Computing. ISC 2017. Lecture Notes in Computer Science, volume 10266, pp. 41-60. Springer, Cham.
- [ISC16] A. Heinecke, A. Breuer, M. Bader: High Order Seismic Simulations on the Intel Knights Landing Processor In High Performance Computing. ISC 2016. Lecture Notes in Computer Science, volume 9697, pp. 343-362. Springer, Cham.
- [IPDPS16] A. Breuer, A. Heinecke, M. Bader: Petascale Local Time Stepping for the ADER-DG Finite Element Method In 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 854-863. IEEE.
- [ISC15] A. Breuer, A. Heinecke, L. Rannabauer, M. Bader: High-Order ADER-DG Minimizes Energy- and Time-to-Solution of SeisSol. In 30th International Conference, ISC High Performance 2015, Frankfurt, Germany, July 12-16, 2015, Proceedings
- [SC14] A. Heinecke, A. Breuer, S. Rettenberger, M. Bader, A.-A. Gabriel, C. Pelties, A. Bode, W. Barth, X.-K. Liao, K. Vaidyanathan, M. Smelyanskiy and P. Dubey: Petascale High Order Dynamic Rupture Earthquake Simulations on Heterogeneous Supercomputers. In Supercomputing 2014, The International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, New Orleans, LA, USA, November 2014. Gordon Bell Finalist.
- [ISC14] A. Breuer, A. Heinecke, S. Rettenberger, M. Bader, A.-A. Gabriel and C. Pelties: Sustained Petascale Performance of Seismic Simulations with SeisSol on SuperMUC. In J.M. Kunkel, T. T. Ludwig and H.W. Meuer (ed.), Supercomputing — 29th International Conference, ISC 2014, Volume 8488 of Lecture Notes in Computer Science. Springer, Heidelberg, June 2014. 2014 PRACE ISC Award.
- [PARCO13] A. Breuer, A. Heinecke, M. Bader and C. Pelties: Accelerating SeisSol by Generating Vectorized Code for Sparse Matrix Operators. In Parallel Computing — Accelerating Computational Science and Engineering (CSE), Volume 25 of Advances in Parallel Computing. IOS Press, April 2014.

Acknowledgements

This work was supported by the Southern California Earthquake Center (SCEC) through contribution #18211. This work was supported by SCEC through contribution #16247. This research was supported by the AWS Cloud Credits for Research program. This research used resources of the Google Cloud. This work was supported by the Intel Parallel Computing Center program. 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 research used resources of the Argonne Leadership Computing Facility (ALCF), which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575.

This work heavily used contributions of many authors to open-source software.
This software includes, but is not limited to: ASan (https://clang.llvm.org/docs/AddressSanitizer.html, debugging), AWS Parallel Cluster (https://github.com/aws/aws-parallelcluster, clusters in AWS), Catch (https://github.com/philsquared/Catch, unit tests), CentOS (https://www.centos.org, cloud OS), CGAL (http://www.cgal.org, surface meshes), Clang (https://clang.llvm.org/, compilation), Cppcheck (http://cppcheck.sourceforge.net/, static code analysis), Easylogging++ (https://github.com/easylogging/, logging), ExprTk (http://partow.net/programming/exprtk, expression parsing), GCC (https://gcc.gnu.org/, compilation), Git (https://git-scm.com, versioning), Git LFS (https://git-lfs.github.com, versioning), Gmsh (http://gmsh.info/, volume meshing), GoCD (https://www.gocd.io/, continuous delivery), HDF5 (https://www.hdfgroup.org/HDF5/, I/O), jekyll (https://jekyllrb.com, homepage), LIBXSMM (https://github.com/hfp/libxsmm, matrix kernels), METIS (http://glaros.dtc.umn.edu/gkhome/metis/metis/overview, partitioning), MOAB (http://sigma.mcs.anl.gov/moab-library/, mesh interface), NetCDF (https://www.unidata.ucar.edu/software/netcdf/, I/O), ObsPy (https://github.com/obspy/obspy/wiki, signal analysis), OpenMPI (https://www.open-mpi.org, cloud MPI), ParaView (http://www.paraview.org/, visualization), pugixml (http://pugixml.org/, XML interface), Read the Docs (https://readthedocs.org, documentation), SAGA-Python (http://saga-python.readthedocs.io/, automated remote job-submission), Scalasca (http://www.scalasca.org, performance measurements), Score-P (https://www.vi-hps.org/projects/score-p/, instrumentation), SCons (http://scons.org/, build scripts), Singularity (https://www.sylabs.io/docs/, container virtualization), Slurm-GCP (https://github.com/SchedMD/slurm-gcp, clusters in GCP), TF-MISFIT GOF CRITERIA (http://www.nuquake.eu, signal analysis), UCVMC (https://github.com/SCECcode/UCVMC, velocity model), Valgrind (http://valgrind.org/, memory debugging), Visit (https://wci.llnl.gov/simulation/computer-codes/visit, visualization).

ISC17

- Theta (ALCF @ ANL)
    - Cray XC40 (early science)
    - 3,200 Intel Xeon Phi 7230 at 1.3 GHz (with Intel Turbo Boost enabled)
    - 16 GB of in-package HBM and 192GB of DDR4 RAM
- Cori Phase II (NERSC @ LBNL)
    - Cray XC40
    - 9,304 Intel Xeon Phi 7250 68-core processors at 1.4 GHz (with Intel Turbo Boost enabled)
    - 16 GB of in-package HBM and 96 GB of DDR4 RAM

Time

$$\mathcal{T}_k(t_0, \hat{t}, \Delta t) = \int_{\hat{t}}^{\hat{t}+\Delta t} Q_k(t)\,\mathrm{d}t = \sum_{d=0}^{\mathcal{O}-1} \frac{(\hat{t}+\Delta t)^{d+1} - \hat{t}^{d+1}}{(d+1)!} \cdot \frac{\partial^d}{\partial t^d} Q_k(t_0)$$

Volume

$$\mathcal{V}_k(\mathcal{T}_k) = \sum_{c=1}^{3} A_k^{\xi_c} \mathcal{T}_k K^{\xi_c} M^{-1}$$

Flux

$$\mathcal{F}_{k,i}^{-}(\mathcal{T}_k) = \frac{|S_i|}{|J_k|} \hat{A}_{k,i}^{-} \mathcal{T}_k F^{-,i} M^{-1} \qquad \mathcal{F}_{k,i}^{+}(\mathcal{T}_{k_i}) = \frac{|S_i|}{|J_k|} \hat{A}_{k,i}^{+} \mathcal{T}_{k_i} F^{+,i,j,k} M^{-1}$$

Local Update

$$Q_k^{*,n_k+1} = Q_k^{n_k} + \mathcal{V}_k(\mathcal{T}_k) - \sum_{i=1}^{4} F_{k,i}^{-}(\mathcal{T}_k)$$

Neighboring Update

$$Q_k^{n_k+1} = Q_k^{*,n_k+1} - \sum_{i=1}^{4} \mathcal{F}_{k,i}^{+}(\mathcal{T}_{k_i})$$

Time

$$\mathcal{T}_k(t_0, \hat{t}, \Delta t) = \int_{\hat{t}}^{\hat{t}+\Delta t} Q_k(t)\,\mathrm{d}t = \sum_{d=0}^{\mathcal{O}-1} \frac{\left(\hat{t}+\Delta t\right)^{d+1} - \hat{t}^{d+1}}{(d+1)!} \cdot \frac{\partial^d}{\partial t^d} Q_k(t_0)$$

Volume

$$\mathcal{V}_k(\mathcal{T}_k) = \sum_{c=1}^{3} \hat{A}_k^{\xi_c} \mathcal{T}_k K^{\xi_c} M^{-1}$$

Flux

$$\mathcal{F}^-_{k,i}(\mathcal{T}_k) = \frac{|S_i|}{|J_k|} \hat{A}^-_{k,i} \mathcal{T}_k F^{-,i} M^{-1} \qquad \mathcal{F}^+_{k,i}(\mathcal{T}_{k_i}) = \frac{|S_i|}{|J_k|} \hat{A}^+_{k,i} \mathcal{T}_{k_i} F^{+,i,j,k} M^{-1}$$

Local Update

$$Q_k^{*,n_k+1} = Q_k^{n_k} + \mathcal{V}_k(\mathcal{T}_k) - \sum_{i=1}^{4} F^-_{k,i}(\mathcal{T}_k)$$

Neighboring Update

$$Q_k^{n_k+1} = Q_k^{*,n_k+1} - \sum_{i=1}^{4} \mathcal{F}^+_{k,i}(\mathcal{T}_{k_i})$$