High Order Seismic Simulations at Sustained Petascale

Alexander Heinecke, Intel Parallel Computing Lab

Alexander Breuer (TUM), Sebastian Rettenberger (TUM), Michael Bader (TUM), Christian Pelties (LMU), Alice-Agnes Gabriel (LMU)

Legal Disclaimer & Optimization Notice

Acknowledgements

- My colleagues at Intel
  - Karthikeyan Vaidyanathan, Mikhail Smelyanskiy, Pradeep Dubey
- Our colleagues around the world supporting us using their supercomputers
- Volkswagen Stiftung — Project ASCETE: Advanced Simulation of Coupled Earthquake-Tsunami Events
- Bavarian Competence Network for Technical and Scientific High Performance Computing (KONWIHR)
- SuperMUC Grant: pr45fi
- NSF Grant: OSI-1134872 (Stampede)
- Some materials in this presentation might be taken from other presentations of my colleagues. I did my very best to add citations☺!

References

a) A. Heinecke, A. Breuer, S. Rettenberger, M. Bader, A.-A. Gabriel, C. Pelties, A. Bode, W. Barth, K. Vaidyanathan, M. Smelyanskiy and P. Dubey: **Petascale High Order Dynamic Rupture Earthquake Simulations on Heterogeneous Supercomputers** [BibTeX].
In *Supercomputing 2014, The International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, New Orleans, LA, USA, November 2014. Gordon Bell Finalist.

b) A. Breuer, A. Heinecke, S. Rettenberger, M. Bader, A.-A. Gabriel and C. Pelties: **Sustained Petascale Performance of Seismic Simulations with SeisSol on SuperMUC** [pdf] [BibTeX].
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, p. 1–18. Springer, Heidelberg, June 2014. PRACE ISC Award 2014.

c) A. Breuer, A. Heinecke, M. Bader and C. Pelties: **Accelerating SeisSol by Generating Vectorized Code for Sparse Matrix Operators** [pdf] [BibTeX].
In *Parallel Computing - Accelerating Computational Science and Engineering (CSE)*, Volume 25 of Advances in Parallel Computing, p. 347–356. IOS Press, April 2014.

d) M. Bader: **Sustained Petascale Performance of Seismic Simulations with SeisSol** [BibTeX].
SIAM Workshop on Exascale Applied Mathematics Challenges and Opportunities (EX14), Chicago, July 2014.

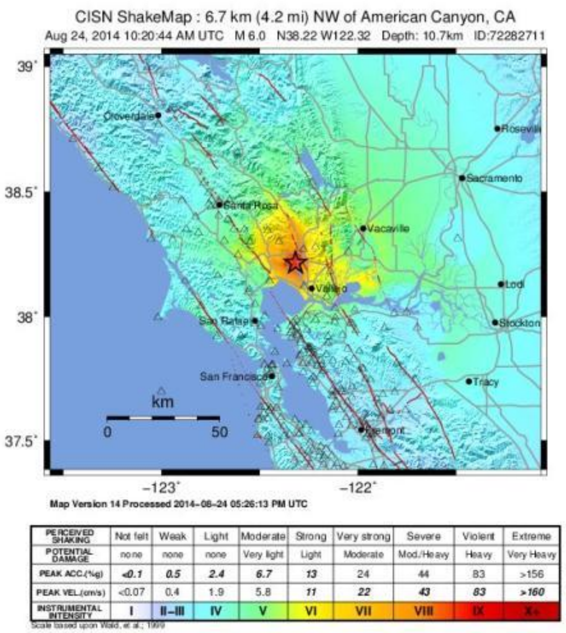
e) M. Bader: **On the Performance of Adaptive Mesh-Based Simulations on Modern HPC Architectures** [pdf] [BibTeX].
SIAM Conference on Parallel Processing in Scientific Computing - SIAM PP 2014, Portland, OR, USA, February 2014. Invited presentation.

f) A. Breuer: **Tuning Sparse and Dense Matrix Operators in SeisSol** [BibTeX].
SIAM Conference on Parallel Processing and Scientific Computing, Portland, Oregon, USA, February 2014. additional authors: Alex Heinecke, S. Rettenberger, Michael Bader, Alice Gabriel, Christian Pelties.

Motivation

“Development of more realistic implementations of dynamic or kinematic representations of fault rupture, including simulation of higher frequencies (up to 10+ Hz).”

2013 Science Collaboration Plan, Southern California Earthquake Center (SCEC).

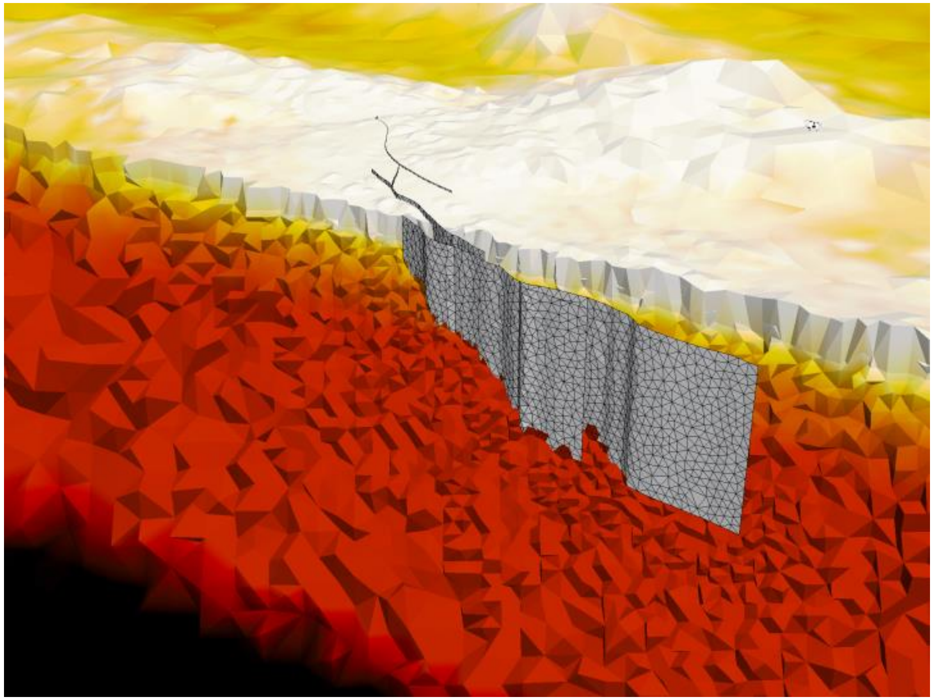


ShakeMap, M6.0, 2014-08-24, 3:20 am, American Canyon, CA, source: usgs.gov

Downtown Napa, Aug 24th, 2014. source: cnn.com

SeisSol in a Nutshell

- Full elastic wave equations in 3D and complex heterogeneous media
- Dynamic Rupture without artificial oscillations
- High order: ADER(time)-DG(space)
- Unstructured tetrahedral meshes
- Highly Optimized Compute Kernels
- Massively parallel



Discretization of the M7.2 Landers 1992 fault system, taken from a)

Outline

- Mathematical Background of SeisSol
- Optimizations of Compute-Kernels, Communication and I/O
- Application Scenarios:
  - “Cubes”-scenario: SuperMUC using IBM MPI, Stampede using MVAPICH: Paper a) + b)
  - Synthetic strong-scaling: SCEC LOH.1 benchmark
  - 7.2M Landers 1992 earthquake: SuperMUC using IBM MPI, Stampede using MVAPICH: Paper a)
- Conclusion

SuperMUC: 9216 Xeon E5 nodes, LRZ Germany, 3PF
Stampede: 6400 Xeon E5 nodes + 1 Xeon Phi, TACC USA, 9+ PF

Deriving SeisSol's Compute Kernels

$$Q_k^{n+1} = Q_k^n - \mathcal{B}_k\left(\mathcal{J}_k^{n,n+1}, \mathcal{J}_{k(1)}^{n,n+1}, \ldots, \mathcal{J}_{k(4)}^{n,n+1}\right) + \mathcal{V}_k\left(\mathcal{J}_k^{n,n+1}\right)$$

SeisSol's Compute Kernels

$$\hat{q}_b^{n+1} = \hat{q}_b^n - \frac{1}{|J| m_b}\left(\int_{t^n}^{t^{n+1}} \int_{\partial T_k} \phi_b f(q) \cdot n \, d\vec{x} dt - \int_{t^n}^{t^{n+1}} \int_{T_k} \nabla \phi_b \cdot f(q) \, d\vec{x} dt\right)$$

DG-Formulation

$$q_t + A(\vec{x}) q_x + B(\vec{x}) q_y + C(\vec{x}) q_z = 0$$

Elastic Wave Equations

Taken from: f)

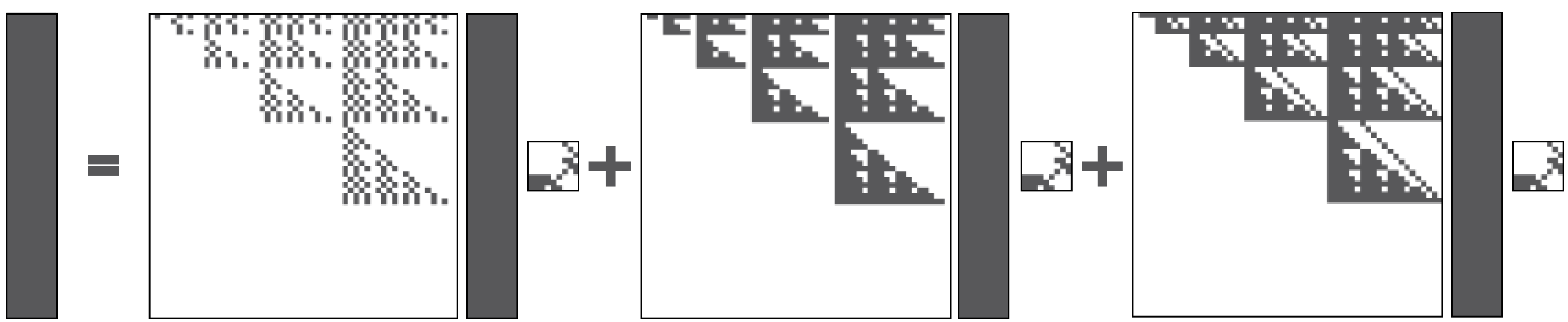
Time Integration Kernel

$$\mathcal{I}_k^{n,n+1}$$

$\mathcal{I}_k^{n,n+1}$ can be compute by recursive scheme:

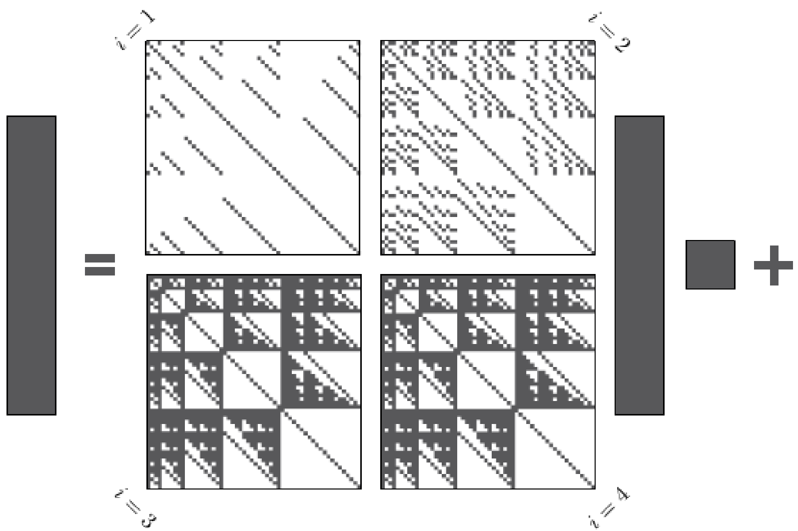
$$\mathcal{I}_k^{n,n+1} := \mathcal{I}_k(t^n, t^{n+1}, Q_k^n) = \sum_{j=0}^{\mathcal{O}-1} \frac{(t^{n+1} - t^n)^{j+1}}{(j+1)!} \frac{\partial^j}{\partial t^j} Q_k(t^n)$$

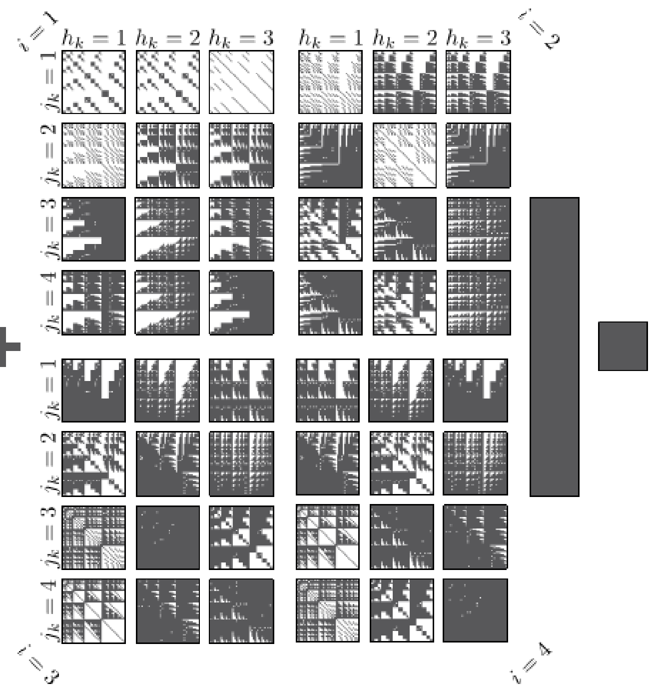
$$\frac{\partial^{j+1}}{\partial t^{j+1}} Q_k = -\hat{K}^{\xi} \left(\frac{\partial^j}{\partial t^j} Q_k \right) A_k^{\star} - \hat{K}^{\eta} \left(\frac{\partial^j}{\partial t^j} Q_k \right) B_k^{\star} - \hat{K}^{\zeta} \left(\frac{\partial^j}{\partial t^j} Q_k \right) C_k^{\star}$$



Flux Computation – Boundary Kernel

$$\mathcal{B}_k\left(\mathcal{J}_k^{n,n+1}, \mathcal{J}_{k(1)}^{n,n+1}, \ldots, \mathcal{J}_{k(4)}^{n,n+1}\right) = \sum_{i=1}^{4} \left(M^{-1}F^{-,i}\right) I_k^{n,n+1} \left(\frac{|S_k|}{|J_k|} N_{k,i} A_k^{+} N_{k,i}^{-1}\right)$$
$$+ \sum_{i=1}^{4} \left(M^{-1}F^{+,i,j_k(i),h_k(i)}\right) I_{k(i)}^{n,n+1} \left(\frac{|S_k|}{|J_k|} N_{k,i} A_{k(i)}^{-} N_{k,i}^{-1}\right)$$

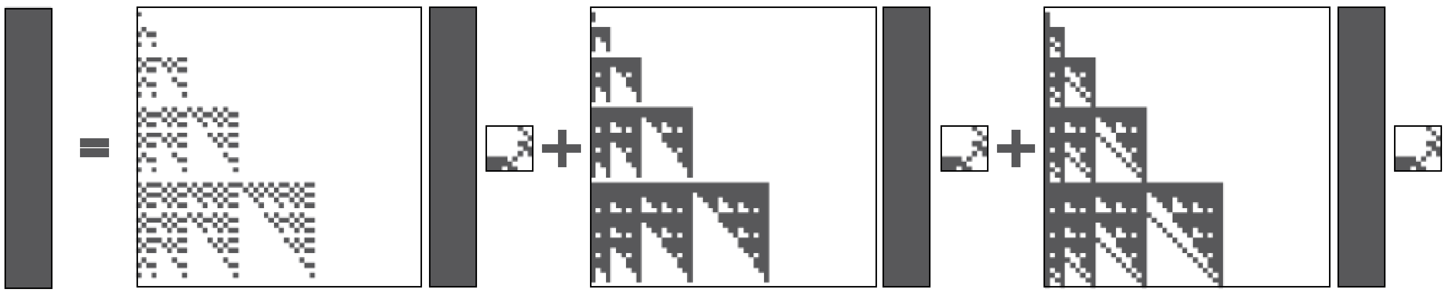




Taken from a)

Volume Integration Kernel

$$\mathcal{V}_k\left(\mathcal{I}_k^{n,n+1}\right) = \tilde{K}^{\xi}\left(\mathcal{I}_k^{n,n+1}\right) A_k^{\star} + \tilde{K}^{\eta}\left(\mathcal{I}_k^{n,n+1}\right) B_k^{\star} + \tilde{K}^{\zeta}\left(\mathcal{I}_k^{n,n+1}\right) C_k^{\star}$$



Taken from a)

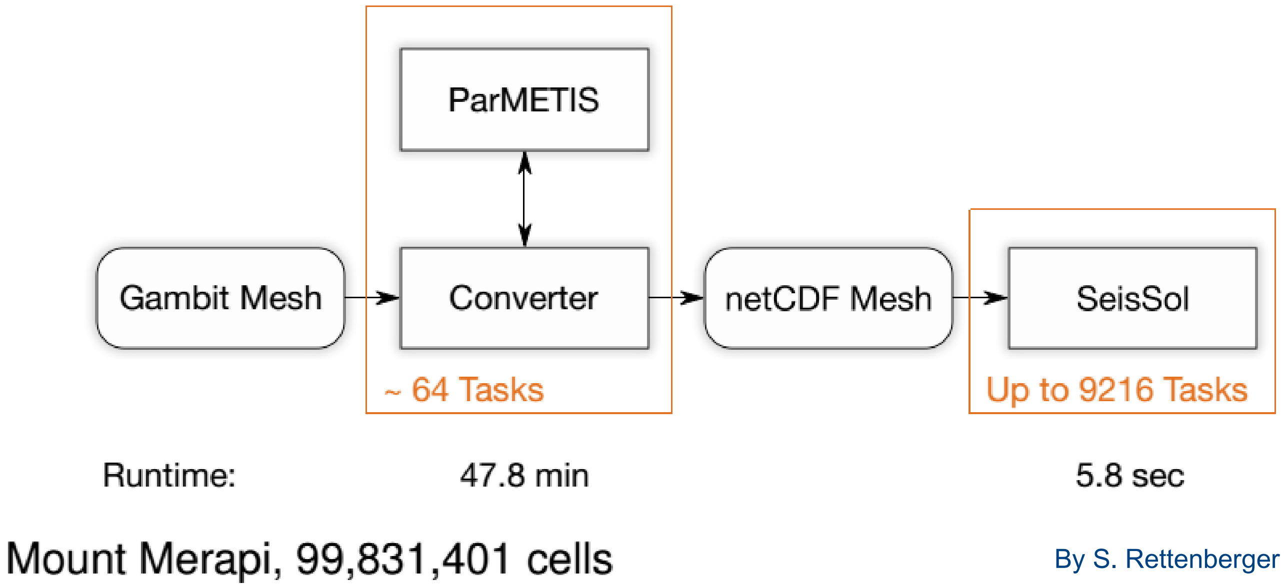
Dynamic Rupture Kernel

- Not part of the elastic wave equations discretization
- → multi-physics formulation
- Dynamic Rupture is implemented as a boundary condition, so we omit these faces during the flux computation!

Kernel Routines

- Highly optimized sparse and dense matrix kernels for by offline code generation and auto-tuning:
  - Intel SSE3
  - Intel AVX
  - Intel Xeon Phi
- Xeon E5 node (2x 8 cores Sandy Bridge) speed-up > 5X
- 1 Xeon Phi coprocessor ~ 1.85X faster than a Xeon E5 node

Mesh Partitioning and I/O Optimizations

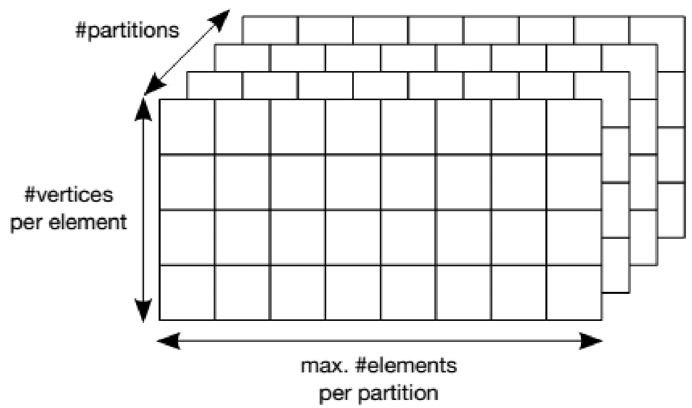


- Reduce complexity to O(#cells/partitions)
- 3-D padded netCDF file:

#partition X

#vertices X

#elements per partition



By S. Rettenberger

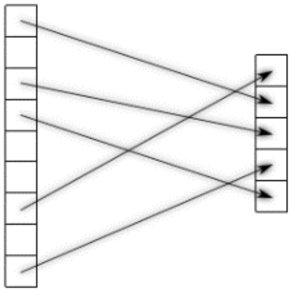
Optimization Notice

MPI Optimizations

- unstructured mesh → unstructured communication patterns
- No global communication in solver phase
- At large scale: 3-30 neighbors per rank
- 20-10K elements
- SeisSol was known to scale very well due to very high amount of compute (we will come back to this ☺)

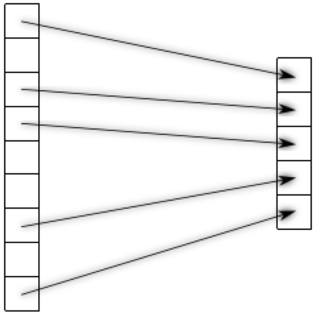
Old SeisSol (per time step):

1. Allocate MPI buffer
2. Gather data
3. Send/Receive
4. Scatter data
5. Deallocate MPI Buffer



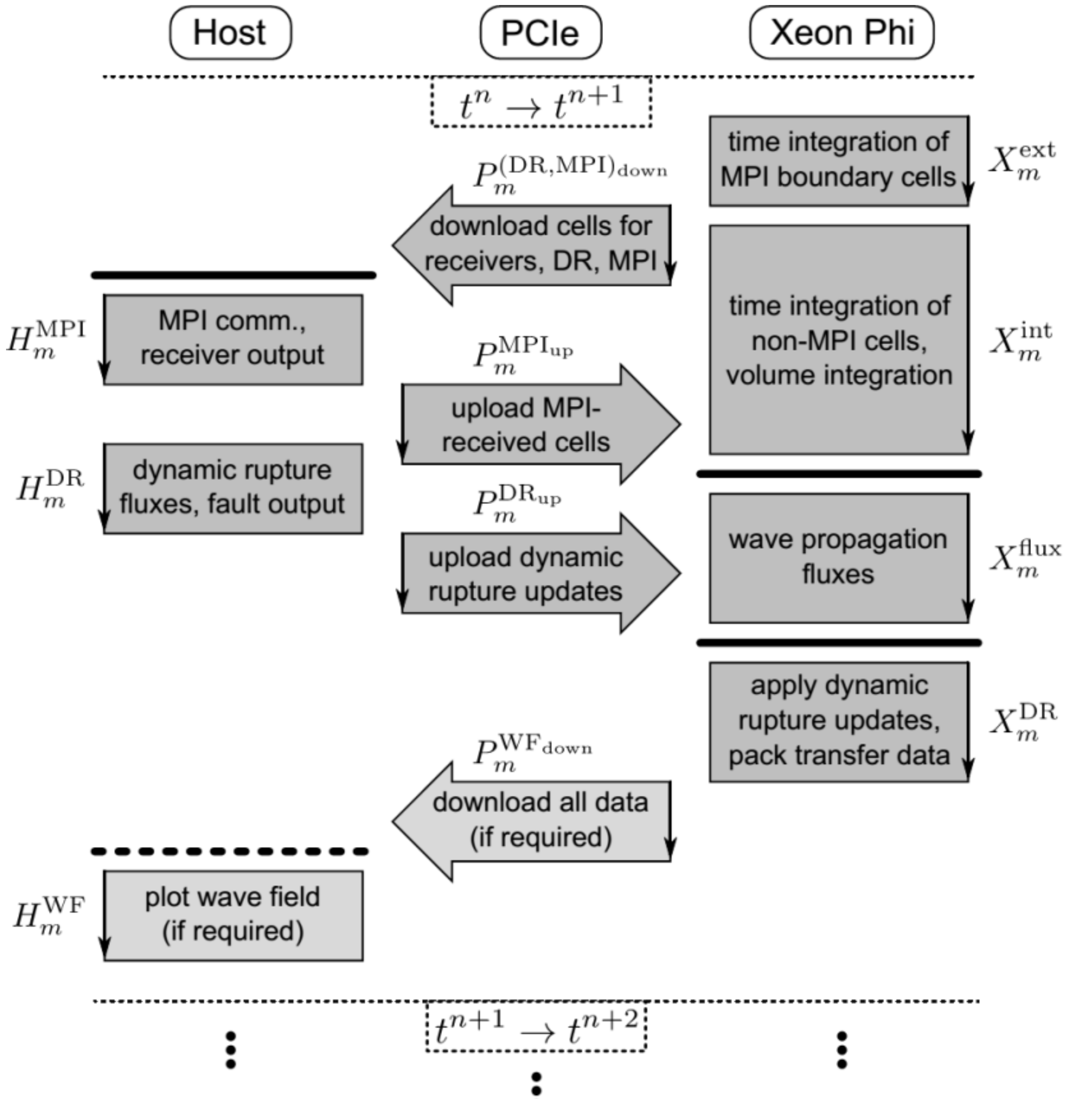
Refactored SeisSol (per time step):

1. Gather data (parallel)
2. Send/Receive (persistent)
3. Scatter data



Last but not least: Xeon Phi Offload

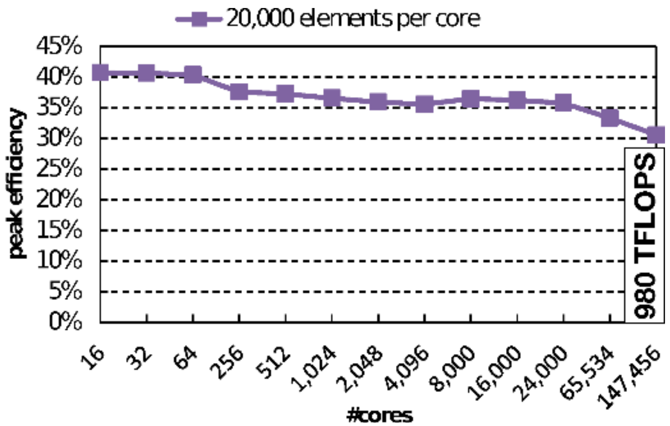
- We need to keep Xeon Phi as busy as possible
- We have to overlap communication
- We have to overlap dynamic rupture computations

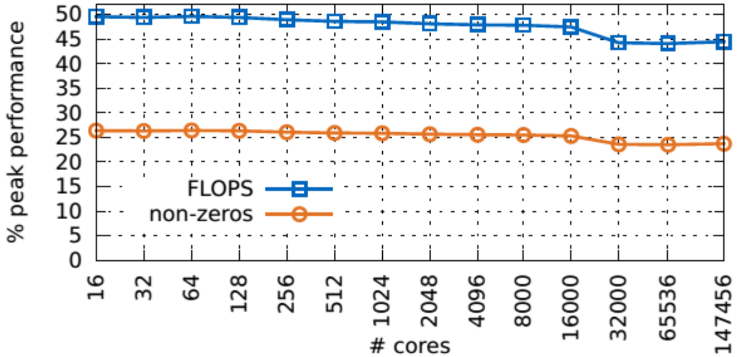


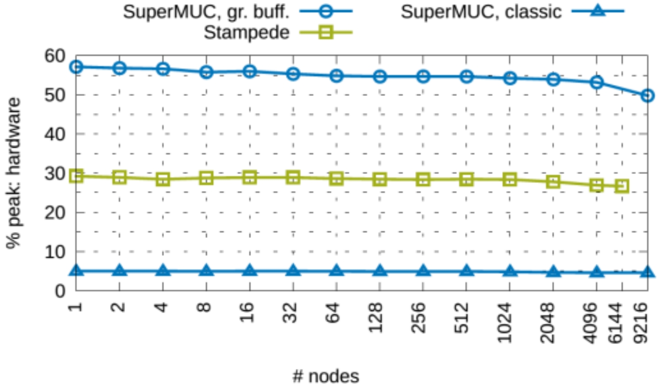
Taken from a)

Cubes – (Burn-In Test) on SuperMUC

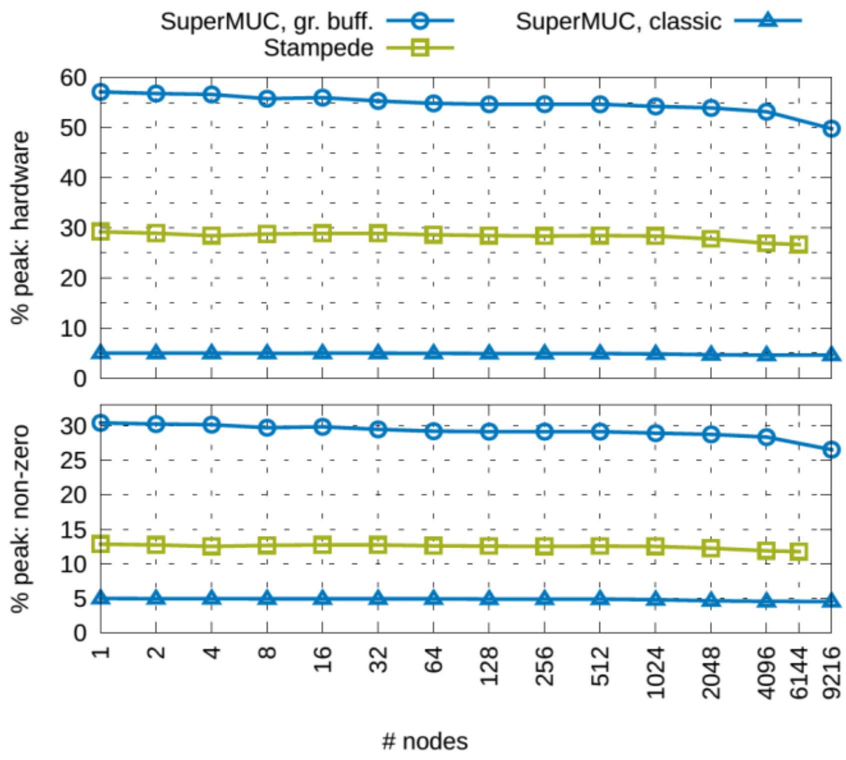
- SC'13 (980 TFLOPs)
  - first release of Kernel Lib
  - no MPI optimizations
- ISC'14 (1.42 PFLOPs)
  - second release of kernel lib
  - MPI optimizations
- SC'14 (1.6 PFLOPs)
  - third release of kernel lib
  - physics optimizations

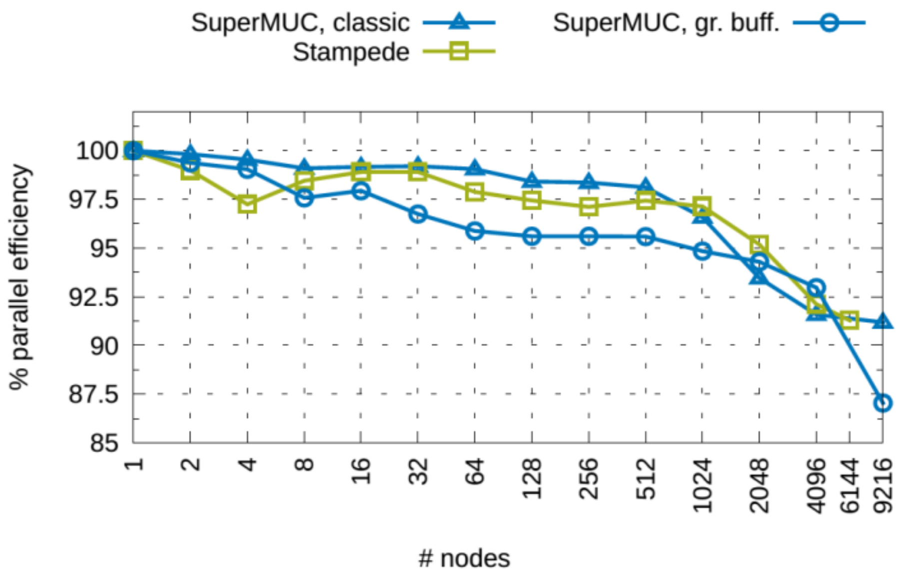






Detailed SuperMUC – Stampede results

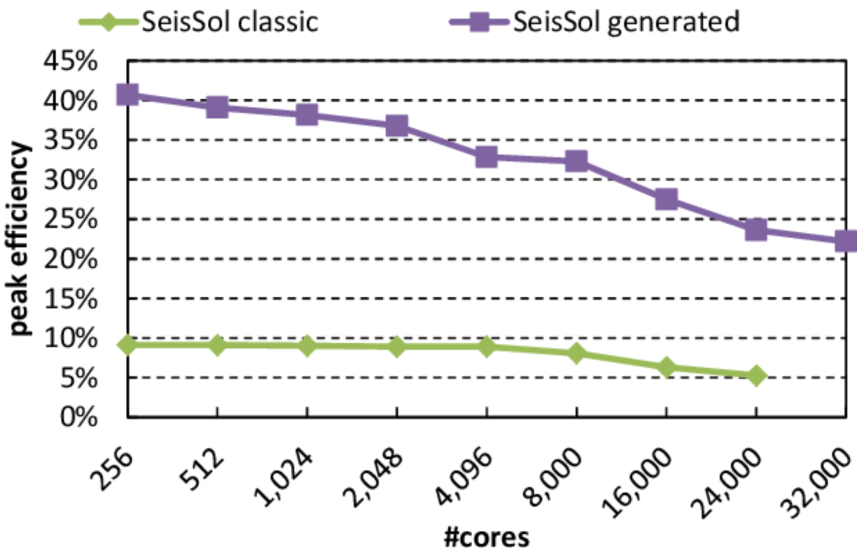




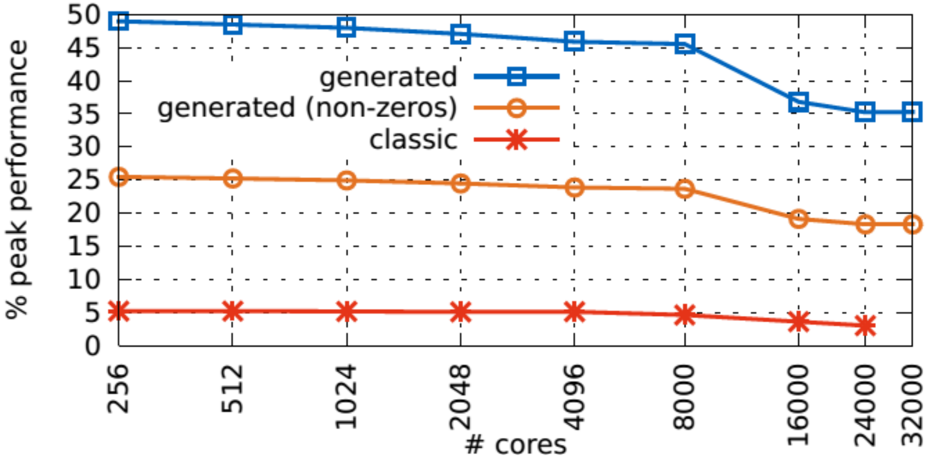
Optimization Notice

Strong Scaling the SCEC LOH.1 benchmark (SC'13 vs. ISC'14)

- SCEC LOH.1: 7,252,482 elements
- Simulation-Time: 100 time steps
- 6th order in space and time



Note: SC'13 classic Flops where calculated using padded FLOPs! -> We move to non-zero FLOPs for all later publications since this is the right way to go!!



Taken from b)

The M7.2 Landers 1992 Earthquake

IRL

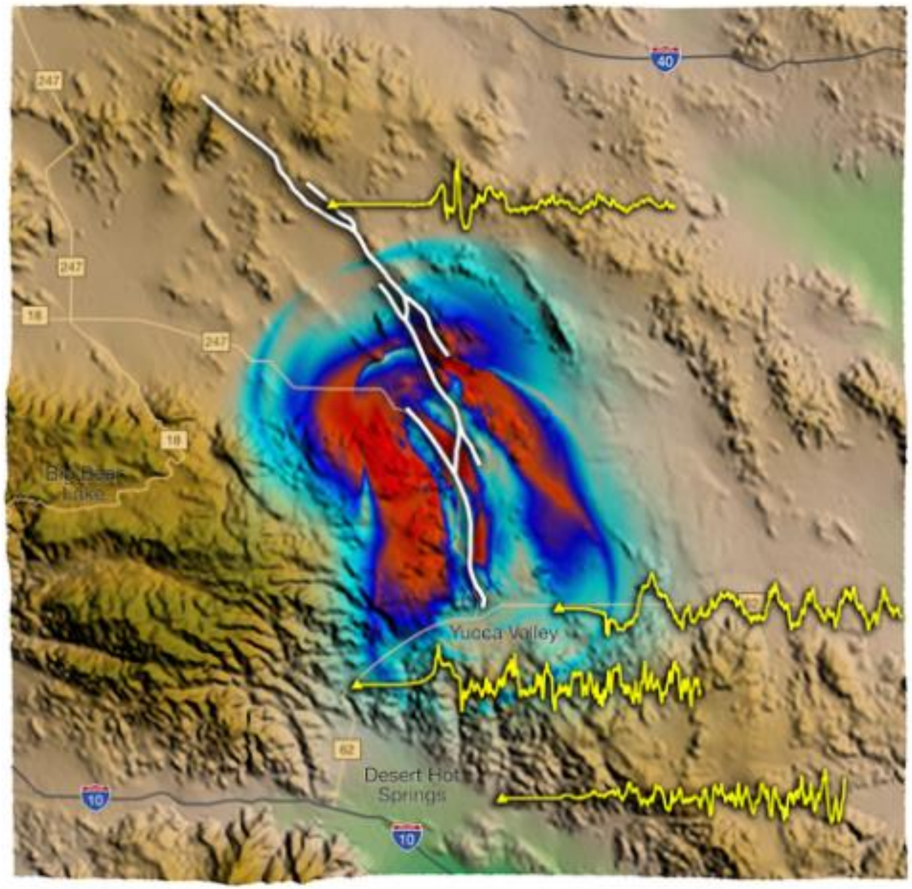
- Type: lateral strike-slip
- Time: June 28, 1992, 4:57 am PDT
- Magnitude: 7.2
- Rupture Length: 85 km
- Faults Ruptured: Johnson Valley, Landers, Homestead Valley, Emerson, and Camp Rock
- Average Slip: 3 to 4 meters, max. 6 meters
- Depth: 1.1 km

Pictures taken from: http://www.data.scec.org/significant/landers1992.html

The M7.2 Landers 1992 Earthquake

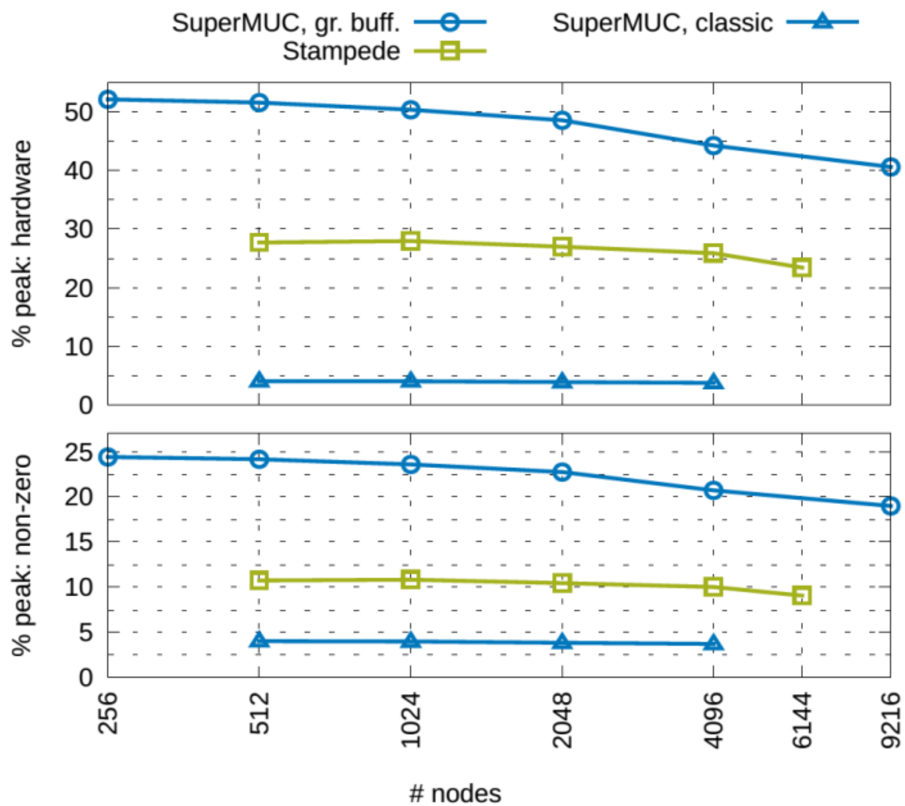
SeisSol Simulation

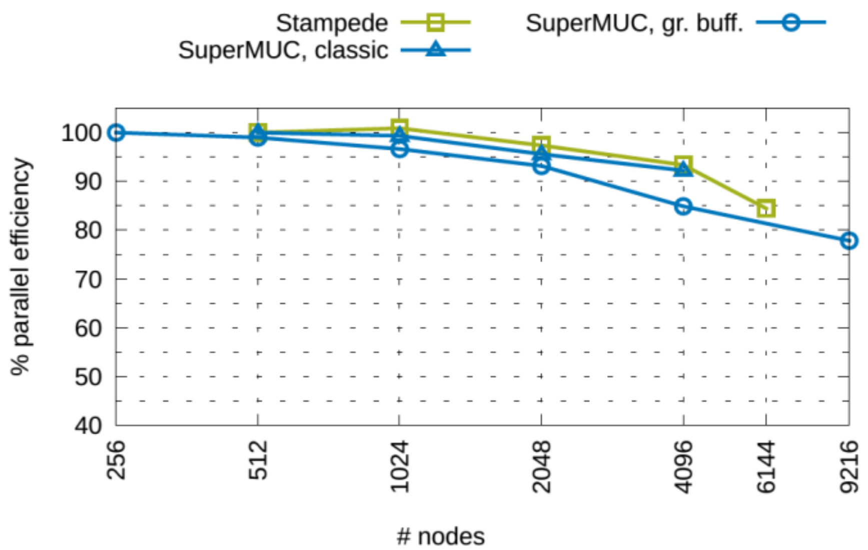
- 191,098,540 tetrahedrons (~1300 per core of SuperMUC, ~130 per thread of Xeon Phi on Stampede)
- Production run SuperMUC:
  - 234,567 time steps equaling 42s simulated time
  - Output: 23 pick-points + high-res fault
  - 7h 15m @ 147,456 SNB-EP cores
  - 1.25 PFLOPs incl. setup and output!! (96.7% of scaling without setup and output)
  - Frequencies up to 10Hz



Taken from a)

Detailed Scaling Data of Landers





- 1000 time steps of the Landers scenario, no output
- MPI communication can be hidden on Stampede
- Scalability on Stampede is equal to SuperMUC

Performance Breakdown and Model for 6144 Stampede Nodes

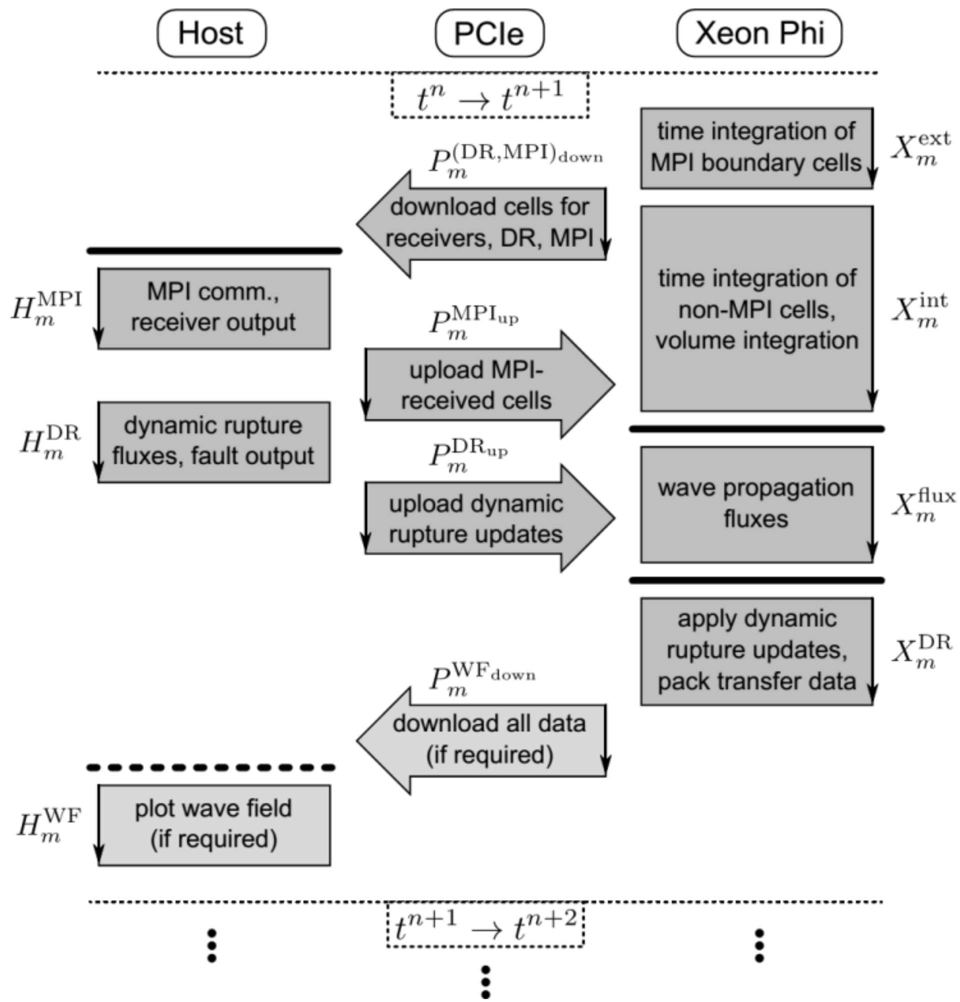
$$E_m^{\Delta t} = E_m^{\text{time}_\text{outer}} + \max\left(E_m^{\text{comm+PCIe}}, E_m^{\text{time}_\text{inner}\text{+volume}}\right) + \max\left(E_m^{\text{DR+PCIe}} - O_m, E_m^{\text{flux}}\right),$$

with

$$\begin{aligned} E_m^{\text{time}_\text{outer}} &= X_m^{\text{ext}}, \\ E_m^{\text{comm+PCIe}} &= P_m^{(\text{MPI,DR})_\text{down}} + H_m^{\text{MPI}} + P_m^{(\text{MPI})_\text{up}}, \\ E_m^{\text{time}_\text{inner}\text{+volume}} &= X_m^{\text{int}}, \\ E_m^{\text{DR+PCIe}} &= H_m^{\text{DR}} + P_m^{(\text{DR})_\text{up}}, \\ E_m^{\text{flux}} &= X_m^{\text{flux}} \end{aligned}$$

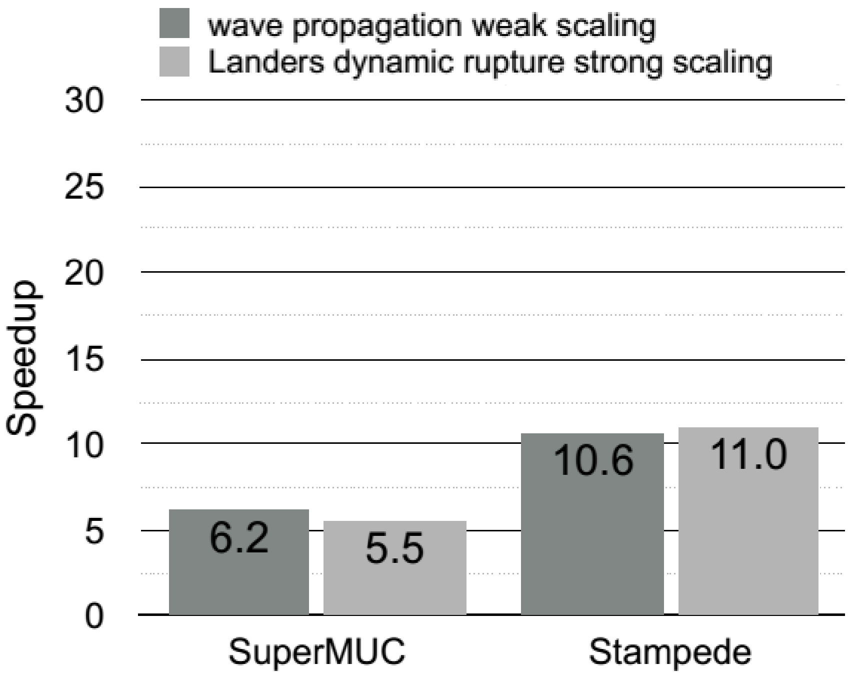
$$O_m = \max\left(E_m^{\text{time}_\text{inner}\text{+volume}} - E_m^{\text{comm+PCIe}}, 0\right).$$

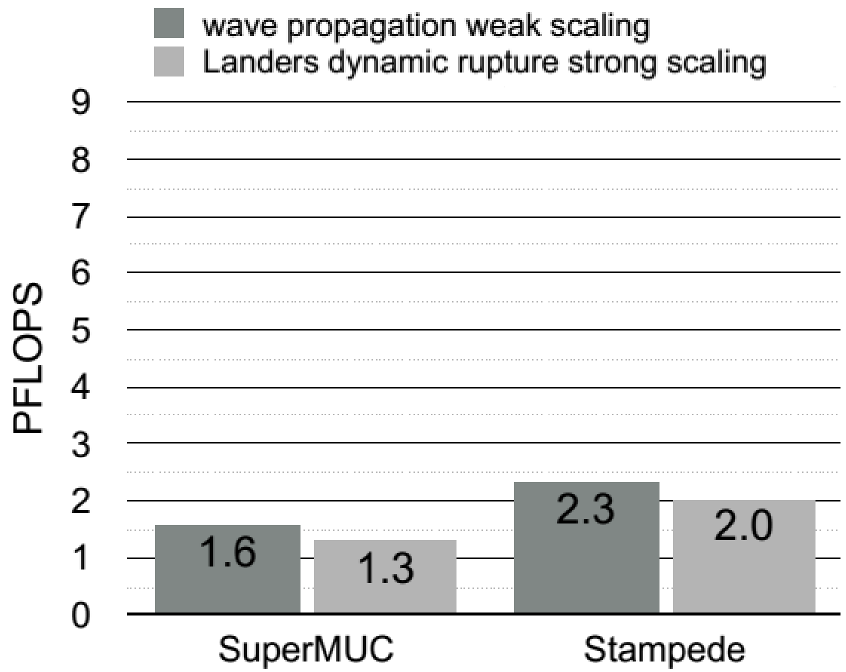
avg. runtime	Stampede
$E^{\text{time}_\text{outer}}$	4
$E^{\text{comm+PCIe}}$	56
$E^{\text{time}_\text{inner}\text{+volume}}$	47
$E^{\text{DR+PCIe}}$	22
E^{flux}	52
$E^{1000\cdot\Delta t}$	112
comm. exposed	(≈7%) 9
DR exposed	0
model misfit for $E^{1000\cdot\Delta t}$	≈ 1%



We saw ~1 GB/s bandwidth between processes -> topology aware mapping!

Performance Summary





Speed-up over SeisSol classic:

Xeon + Xeon Phi clusters can boost science performance by factor of 2.
Even more important: tripling the FLOPS (3 -> 9 PFLOPS)
Results in close to doubled application-level performance.

Optimization Notice

Future Work

- Local Time Stepping (LTS)
  - Even more unstructured communication schemes
  - RDMA one-sided seems to be promising, neighbor collectives?
- Improved partitioning reflecting LTS requirements
- Topology-aware process mapping (e.g. what happens on a Cascade or newer?)
- Improved compute kernels leveraging new processors architectures, e.g. Xeon E5 v3 (code-named Haswell) and Xeon Phi successor (code-named Knights Landing).

Conclusion

- Significant speed-ups due to kernel and communication optimizations

→ Sustained multi-petaflop application

- I/O optimizations allow SeisSol to run production scenarios at full machine size

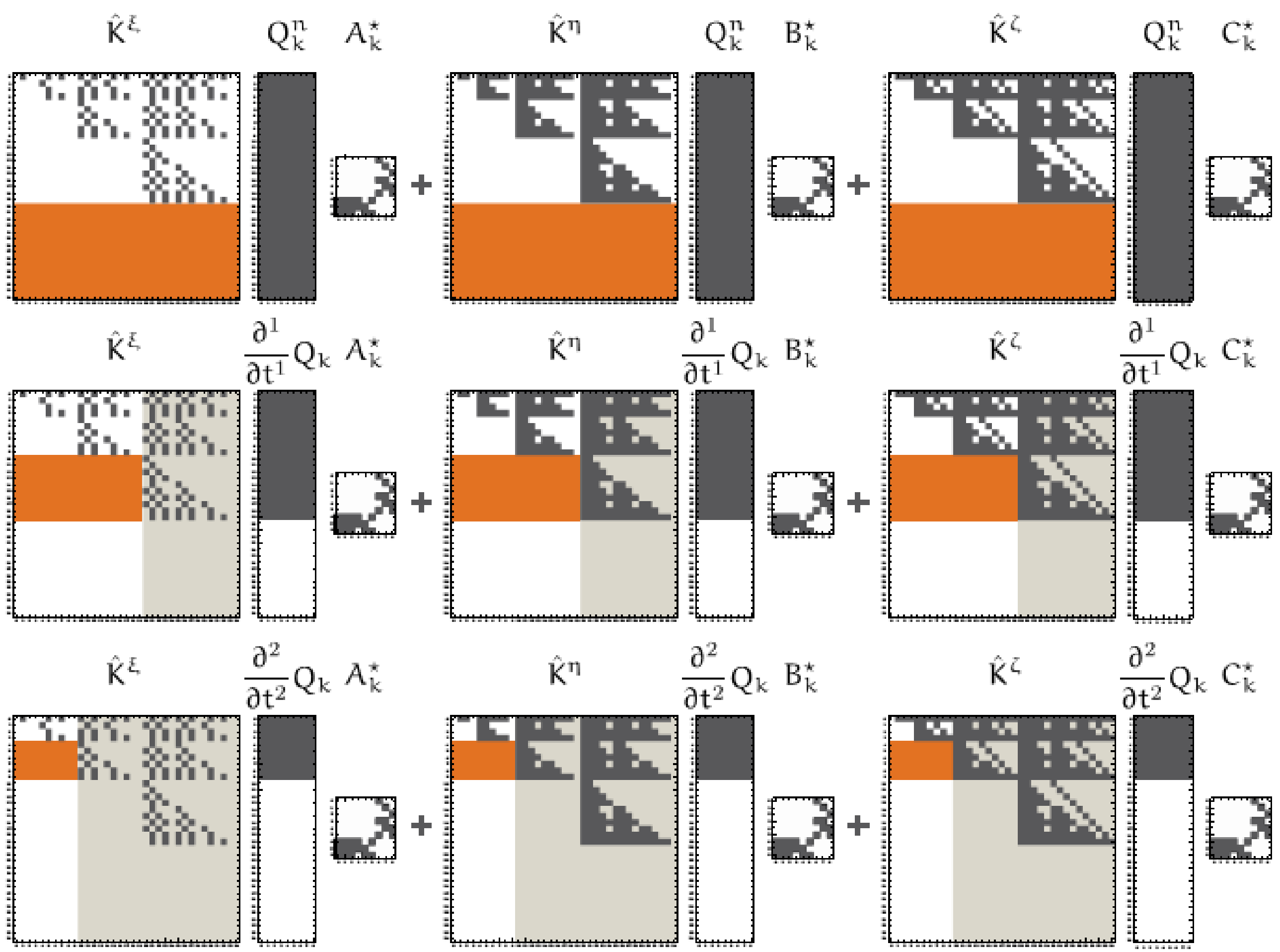
→ New science, see a)

- Support for heterogeneous cluster nodes in multi-physics scenarios
- Proof-by-example ☺:

→ For best performance on today's systems we have to tune the entire simulation pipeline (and not just kernels)!

intel®

Exploiting Taylor Series during Computation of Time Integration Kernel



Taken from b)