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)
INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Copyright © 2014, Intel Corporation. All rights reserved. Intel, Pentium, Xeon, Xeon Phi, Core, VTune, Cilk, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804
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


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

\[
Q_{k}^{n+1} = Q_{k}^{n} - B_{k} \left( J_{k}^{n,n+1}, J_{k(1)}^{n,n+1}, \ldots, J_{k(4)}^{n,n+1} \right) + V_{k} \left( J_{k}^{n,n+1} \right)
\]

\[
\hat{q}_{b}^{n+1} = \hat{q}_{b}^{n} - \frac{1}{|J|m_{b}} \left( \int_{t_{n}}^{t_{n+1}} \int_{\partial T_{k}} \phi_{b} f(q) \cdot n \, d\vec{x} \, dt - \int_{t_{n}}^{t_{n+1}} \int_{T_{k}} \nabla \phi_{b} \cdot f(q) \, d\vec{x} \, dt \right)
\]

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

\[ j^{n,n+1}_k \text{ can be compute by recursive scheme:} \]

\[ j^{n,n+1}_k := J_k(t^n, t^{n+1}, Q^n_k) = \sum_{j=0}^{\Omega-1} \frac{(t^{n+1} - t^n)^{j+1}}{(j + 1)!} \frac{\partial^j}{\partial t^j} Q_k(t^n) \]

\[ \frac{\partial^{j+1}}{\partial t^{j+1}} Q_k = -\hat{K}_\xi \left( \frac{\partial^j}{\partial t^j} Q_k \right) A_k^* - \hat{K}_n \left( \frac{\partial^j}{\partial t^j} Q_k \right) B_k^* - \hat{K}_\zeta \left( \frac{\partial^j}{\partial t^j} Q_k \right) C_k^* \]
Flux Computation – Boundary Kernel

\[ \mathcal{B}_k \left( j_{k,1}^{n,n+1}, j_{k,2}^{n,n+1}, \ldots, j_{k,4}^{n,n+1} \right) = \sum_{i=1}^{4} \left( M^{-1} F^-, i \right) I_{k}^{n,n+1} \left( \frac{|S_k|}{|J_k|} N_{k,i} A_{k}^{+} N_{k,i}^{-1} \right) + \sum_{i=1}^{4} \left( M^{-1} F^+, i, j_{k(i)}, h_{k(i)} \right) I_{k(i)}^{n,n+1} \left( \frac{|S_k|}{|J_k|} N_{k,i} A_{k(i)}^{-} N_{k,i}^{-1} \right) \]

Taken from a)
Volume Integration Kernel

\[ \mathcal{V}_k \left( j_{k}^{n,n+1} \right) = \tilde{K}^\xi \left( j_{k}^{n,n+1} \right) A_k^* + \tilde{K}^\eta \left( j_{k}^{n,n+1} \right) B_k^* + \tilde{K}^\zeta \left( j_{k}^{n,n+1} \right) C_k^* \]

Dynamic Rupture Kernel

- Not part of the elastic wave equations discretization
- \( \rightarrow \) 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 $O(\#\text{cells}/\#\text{partitions})$
- 3-D padded netCDF file:
  - $\#\text{partition} \times$  
  - $\#\text{vertices} \times$  
  - $\#\text{elements per partition}$

Mount Merapi, 99,831,401 cells

Runtime: 47.8 min

ParMETIS

Gambit Mesh

Converter

netCDF Mesh

SeisSol

~ 64 Tasks

Up to 9216 Tasks

By S. Rettenberger
MPI Optimizations

- unstructured mesh $\rightarrow$ 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
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 were calculated using padded FLOPs! -> We move to non-zero FLOPs for all later publications since this is the right way to go!!
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

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

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

<table>
<thead>
<tr>
<th><strong>avg. runtime</strong></th>
<th><strong>Stampede</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{time_{outer}}$</td>
<td>4</td>
</tr>
<tr>
<td>$E_{comm+PCle}$</td>
<td>56</td>
</tr>
<tr>
<td>$E_{time_{inner}+volume}$</td>
<td>47</td>
</tr>
<tr>
<td>$E_{DR+PCle}$</td>
<td>22</td>
</tr>
<tr>
<td>$E_{flux}$</td>
<td>52</td>
</tr>
<tr>
<td>$E^{1000\cdot \Delta t}$</td>
<td>112</td>
</tr>
<tr>
<td>comm. exposed</td>
<td>(∼7%) 9</td>
</tr>
<tr>
<td>DR exposed</td>
<td>0</td>
</tr>
<tr>
<td>model misfit for $E^{1000\cdot \Delta t}$</td>
<td>≈ 1%</td>
</tr>
</tbody>
</table>
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.
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)!
Exploiting Taylor Series during Computation of Time Integration Kernel

\[
\begin{align*}
\dot{\kappa}^\xi & \quad Q_k^\eta & \quad \Lambda_k^\eta \\
\frac{\partial^1}{\partial t^1} Q_k & \quad \Lambda_k & \\
\frac{\partial^2}{\partial t^2} Q_k & \quad \Lambda_k^\eta
\end{align*}
\]

\[
\begin{align*}
\dot{\kappa}^\eta & \quad Q_k & \quad B_k^\eta \\
\frac{\partial^1}{\partial t^1} Q_k & \quad B_k & \\
\frac{\partial^2}{\partial t^2} Q_k & \quad B_k^\eta
\end{align*}
\]

\[
\begin{align*}
\dot{\kappa}^\zeta & \quad Q_k^\eta & \quad C_k^\eta \\
\frac{\partial^1}{\partial t^1} Q_k & \quad C_k & \\
\frac{\partial^2}{\partial t^2} Q_k & \quad C_k^\eta
\end{align*}
\]

Taken from b)