technische universität dortmund

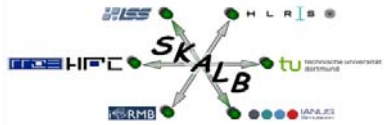
Modeling the propagation of elastic waves using spectral elements on a cluster of 192 GPUs

Dimitri Komatitsch, **Dominik Göddeke**
Gordon Erlebacher, David Michéa

ISC'10, Hamburg, Germany, May 31, 2010

Université de Pau, CNRS & INRIA Sud-Ouest Magique3D, France (http://www.univ-pau.fr/~dkomati1)
Angewandte Mathematik, TU Dortmund, Germany
Department of Scientific Computing, Florida State University, Tallahassee, USA
Bureau de Recherches Géologiques et Minières, Orléans, France

technische universität dortmund

Seismic wave propagation

- Application domains
  - Earthquakes in sedimentary basins and at the scale of a continent
  - Active acquisition experiments in the oil and gas industry
- High practical relevance
  - L‘Aquila (Italy), April 2009: 5.8 Richter scale
  - 260 dead
  - 1000 injured
  - 26000 homeless

- Good numerical and computational methods required!

Topography and sedimentary basins

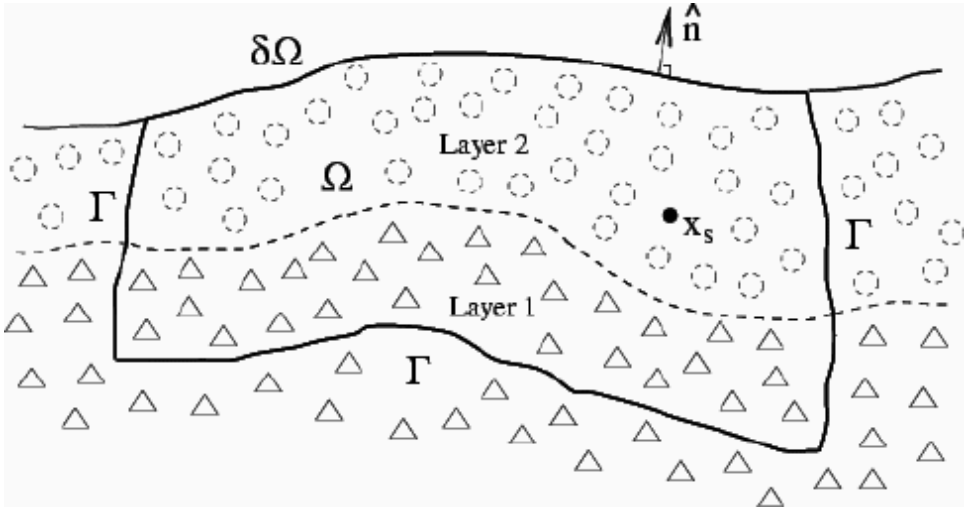
technische universität dortmund

- Topography needs to be honoured
  - Densely populated areas are often located in sedimentary basins
  - Surrounding mountains reflect seismic energy back and amplify it (think rigid Dirichlet boundary conditions)
  - Seismic shaking thus much more pronounced in basins

High resolution requirements

technische universität dortmund

- Spatially varying resolution
  - Local site effects and topography
  - Discontinuities between heterogeneous sedimentary layers in the Earth



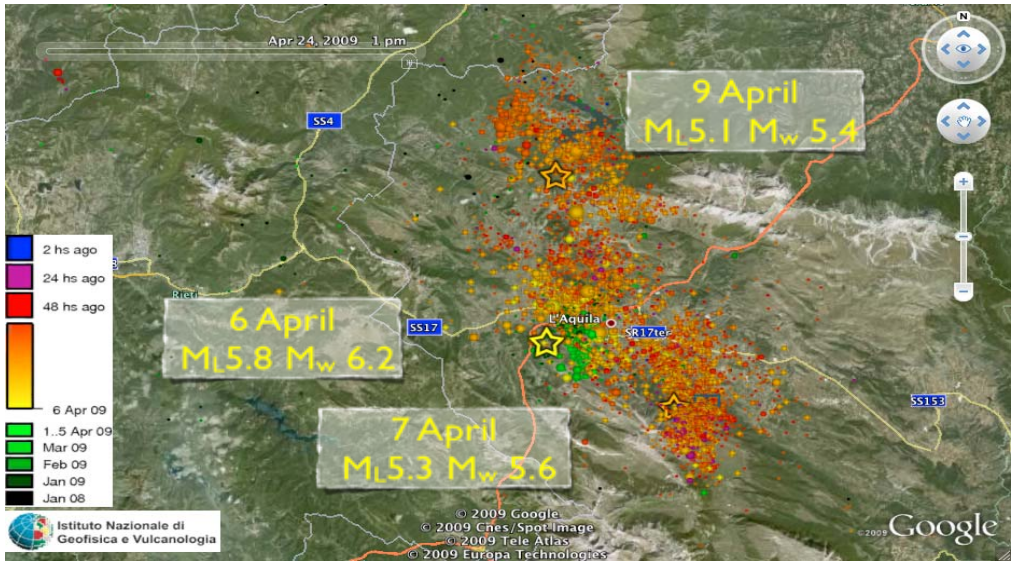
- High seismic frequencies need to be captured
  - High-order methods and finely-resolved discretisation required

Timing constraints: Aftershocks

technische universität dortmund

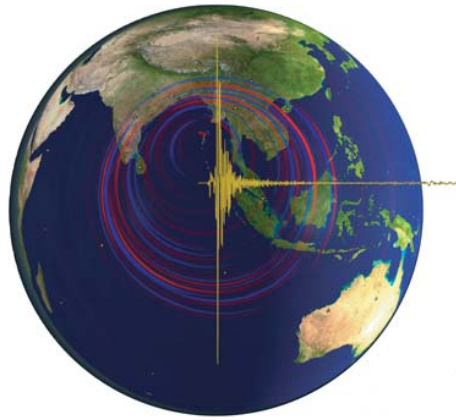
- Main shock typically followed by aftershocks
  - Predict effect of aftershocks within a few hours after an earthquake
  - Predict impact on existing faults (from previous earthquakes) that may break due to changed stress distribution in the area
  - Finish simulation ahead of follow-up shaking to issue detailed warnings

- L'Aquila earthquake aftershock prediction



Summary: Challenges

- Conflicting numerical goals
  - High-order methods
  - But: High flexibility and versatility required
  - Must be efficiently parallelisable in a scalable way
- Extremely high computational demands of typical runs
  - 100s of processors and 100s of GB worth of memory
  - Several hours to complete (100.000s of time steps)
- SPECFEM3D software package
  - http://www.geodynamics.org
  - Open source and widely used
  - Accepts these challenges and implements good compromises
  - Gordon Bell Prize 2003, finalist again 2008 (sustained 0.18 PFLOP/s)



Physical model and numerical solution scheme

Physical model: Elastic waves

technische universität dortmund

- Model parameters
  - Linear anisotropic elastic rheology for a heterogeneous solid part of the Earth, full 3D simulation

- Strong and weak form of seismic wave equation

$$\rho\ddot{\mathbf{u}} = \nabla\cdot\sigma + \mathbf{f}; \sigma = \mathbf{C}:\varepsilon; \varepsilon = \tfrac{1}{2}\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right]$$

$$\int_\Omega \rho\mathbf{w}\cdot\ddot{\mathbf{u}}d\Omega + \int_\Omega \nabla\mathbf{w}:\mathbf{C}:\nabla\mathbf{u}d\Omega = \int_\Omega \mathbf{w}\cdot f d\Omega + \int_\Gamma (\sigma\cdot\mathbf{n})\cdot\mathbf{w}d\Gamma$$

  - Displacement **u**, stress and strain tensors sigma and epsilon
  - Stiffness tensor **C** and density rho (given spatially heterogeneous material parameters)
  - External forces **f** (i.e., the seismic source), test function **w**

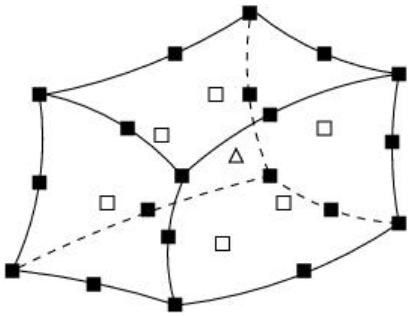
Numerical methods overview

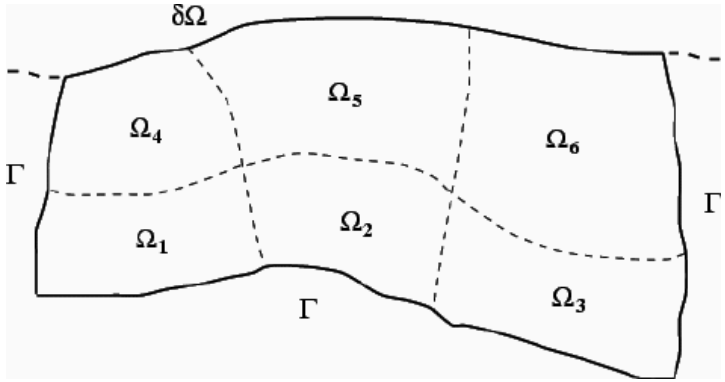
- Finite differences
  - Easy to implement, but difficult for boundary conditions, surface waves, and to capture nontrivial topography
- Boundary elements, boundary integral methods
  - Good for homogeneous layers, expensive in 3D
- Spectral and pseudo-spectral methods
  - Optimal accuracy, but difficult for boundary conditions and complex domains, difficult to parallelise
- Finite elements
  - Optimal flexibility and error analysis framework, but may lead to huge sparse ill-conditioned linear systems

Spectral element method (SEM)

technische universität dortmund

- Good compromise between conflicting goals
  - “Hybrid“ approach: Combines accuracy of pseudo-spectral methods with the geometric flexibility of finite element methods
  - Alleviates their respective difficulties and parallelises moderately easily
- Cover domain with large, curvilinear hexahedral “spectral“ elements
  - Mesh honours topography and interior discontinuities
- Approximate quantities in each cell with high-order interpolation



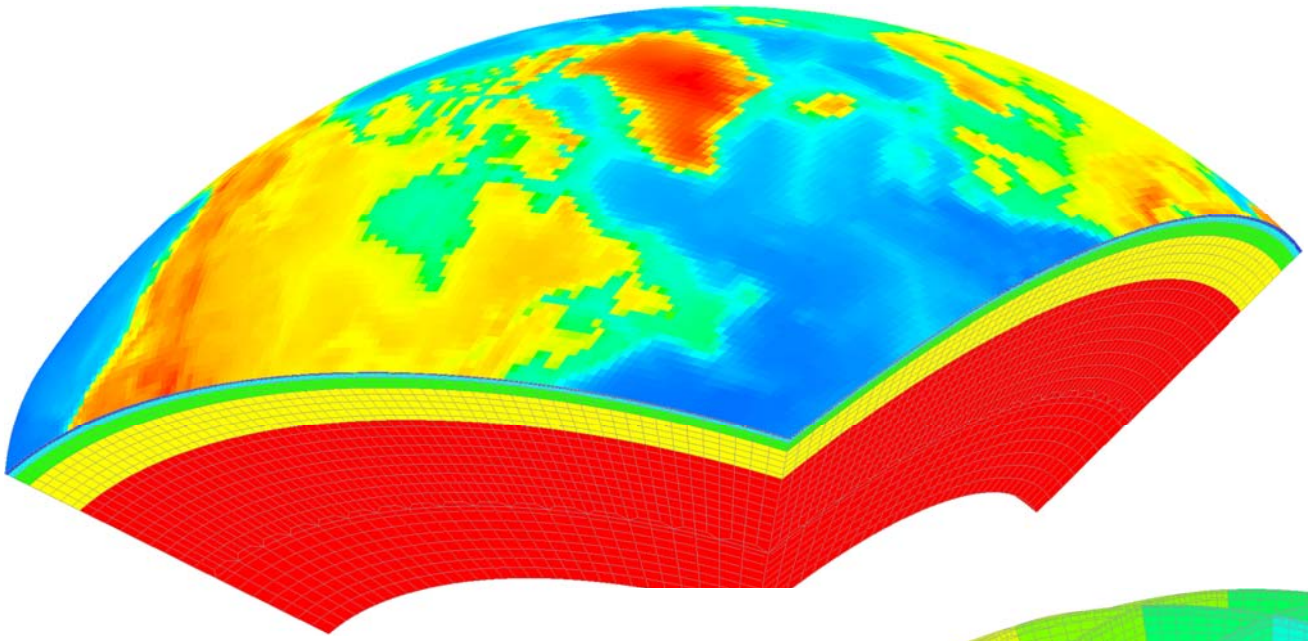


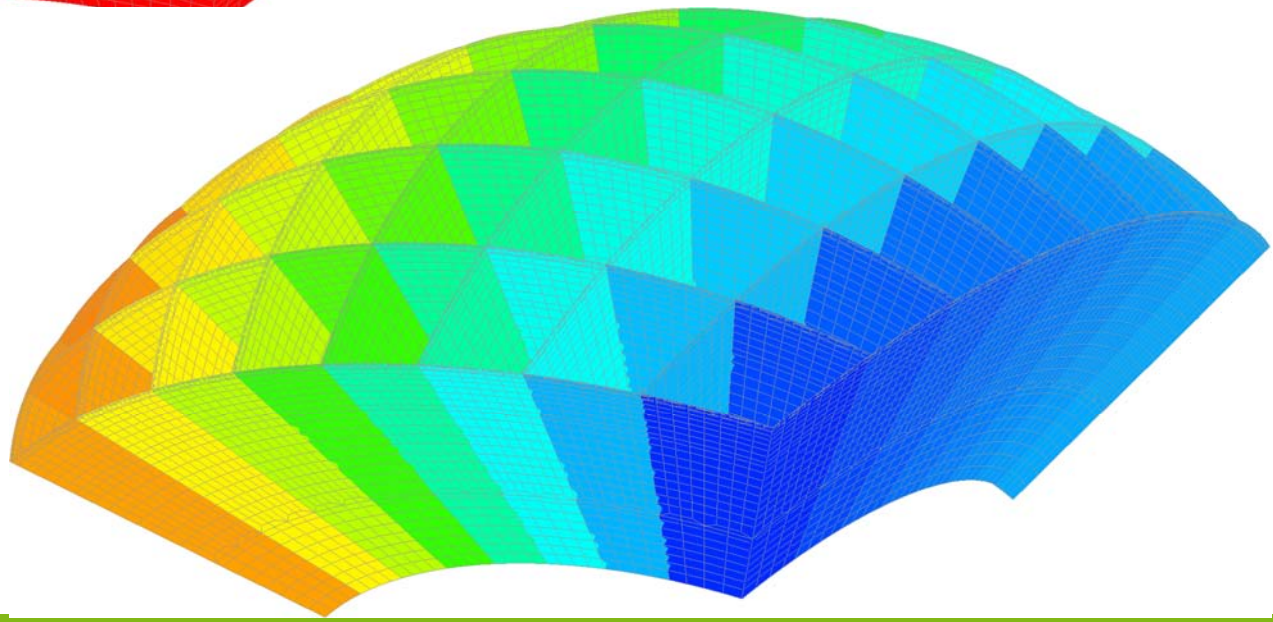
SEM details for seismic waves

technische universität dortmund

- Legendre polynomials for interpolation
  - Degree 4-10, 4 is good compromise between accuracy and speed
  - Gauß-Lobatto-Legendre control points (rather than Gauß points)
- Numerical quadrature to compute the integrals in the weak form
  - Use GLL points as well
  - Deg.+1 GLL points per element per spatial dimension (125 for deg. 4)
- Crucial benefit
  - Combination of these two leads to strictly diagonal mass matrix
  - Solving of linear systems becomes trivial
  - Permits use of fully explicit time stepping schemes (2^{nd} order centred finite difference Newmark in our case)

Meshing





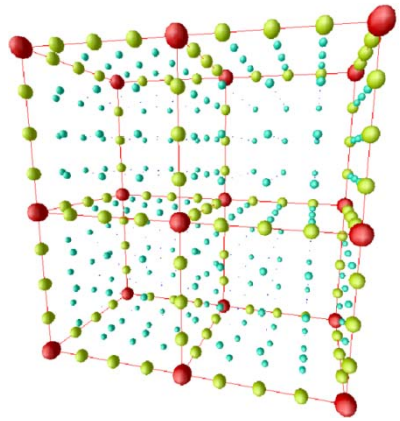
Block-structured mesh: each slice is unstructured, but all are topologically identical

Summary: Solution algorithm

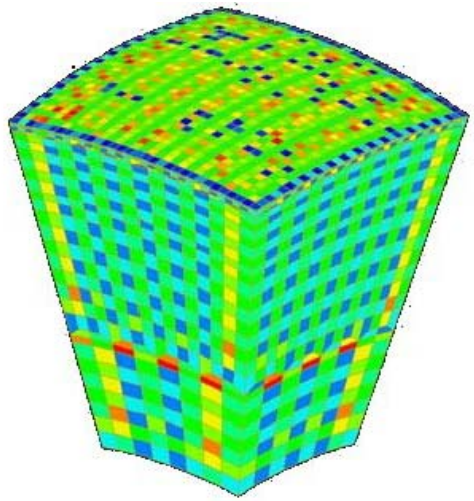
- Problem to be solved in algebraic notation: M**ü**+K**u**=**f**
  - Mass matrix M, Stiffness matrix K, displacement vector **u**, sources **f**
- Step 1 in the Newmark time loop
  - Update global displacement and velocity (1st time derivative of displacement) vectors using previous acceleration vector (2nd deriv.)
- Step 2
  - Compute element-local contributions in stiffness matrix term K**u** and gradient of **u** (basically, small dense matrix-matrix multiplications)
  - Assemble local contributions into global matrix
- Step 3
  - Compute global acceleration vector (remember: M is diagonal!)

GPU implementation and parallelisation

GPU implementation issues



- Steps 1 and 3 are trivially parallel
  - Only affect uniquely numbered global data
- Step 2 is tricky to parallelise
  - One thread for each of the 125 integration points, i.e., one thread block per element (128 threads to facilitate vectorisation)
  - Mesh is unstructured → indirect addressing challenging to coalesce
  - “Reduction“ of local contributions that are shared between elements into global matrix (must be atomic)
  - Solution: Mesh colouring
  - More sweeps, conflict-free within each sweep



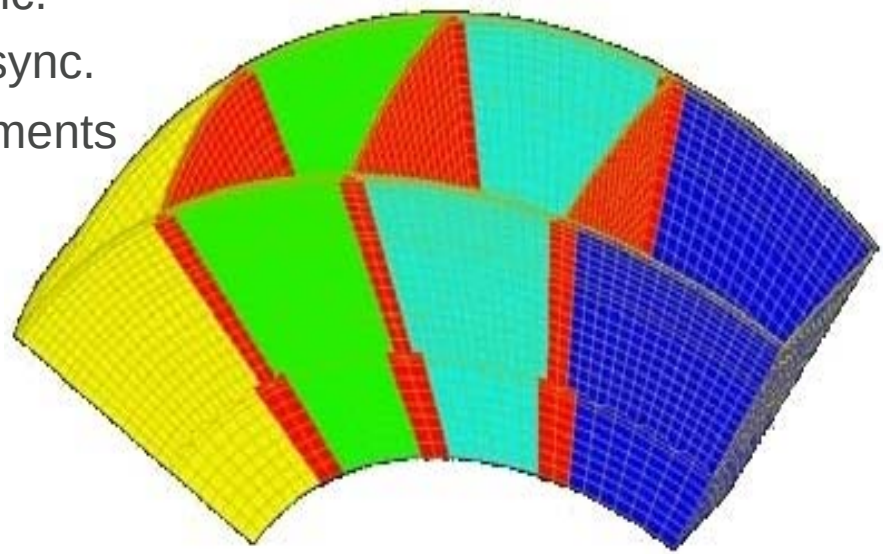
Details: D. Komatitsch et. al.: “Porting a high-order finite-element earthquake modeling application to NVIDIA graphics cards using CUDA “, IJPDC 69(5), p. 451-460 (2009).

Mapping to MPI clusters

technische universität dortmund

- Mapping and partitioning:
  - One mesh slice (100K-500K elements) associated with one MPI rank
  - Slice size always chosen to max out available memory per resource (weak scaling!)

- Overlap computation with communication (non-blocking MPI)
  - Separate outer (ie, shared via common cut plane) from inner elements
  - Compute outer elements, send async.
  - Compute inner elements, receive async.
  - Balance ratio of outer and inner elements



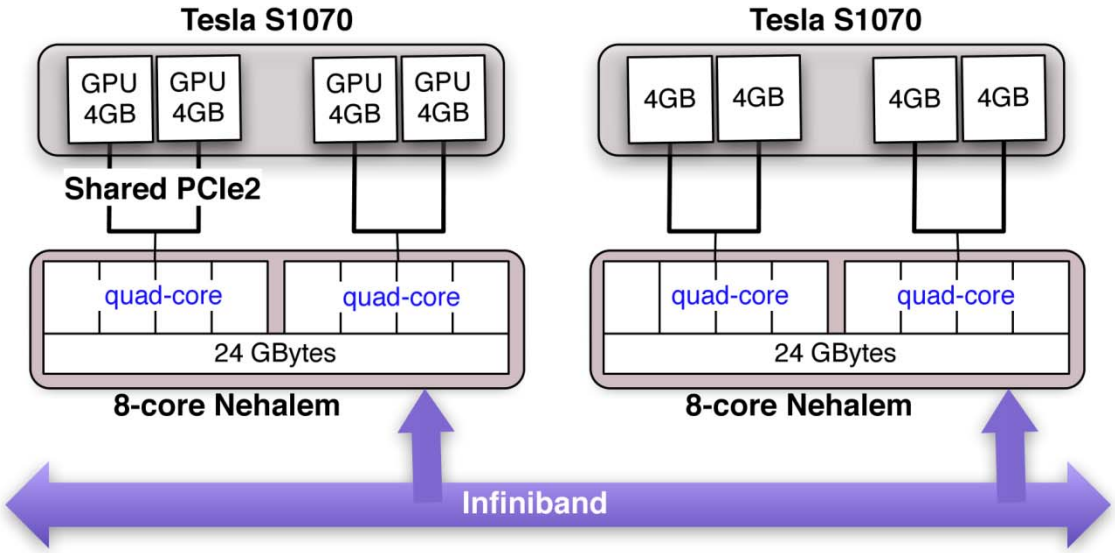
GPU cluster challenges

- Problem: PCIe bus
  - MPI buffers and communication remain on CPU
  - PCIe adds extra latency and bandwidth bottleneck
  - Found that merging cut planes on GPU, transfer in bulk to CPU, extract on CPU and send over interconnect to neighbours is faster than using asynchronous transfers (cudaMemcpyAsync()) or zero-copy because bookkeeping overhead is too high
- Problem: GPUs are too fast ☺
  - GPUs need higher ratio of inner to outer elements to achieve effective overlap of communication and computation
  - Speedup is higher than # of CPU cores per node
  - Ideal cluster for us: one GPU per CPU core
  - Future work: Map smaller slices to idle CPU cores and/or use OpenMP

Some results

Testbed

- „Titane“: Bull Novascale R422 E1 GPU cluster
  - Installed at CCRT/CEA/GENCI, Bruyères-le-Châtel, France
  - 48 nodes



- CPU reference code in SPECFEM3D is heavily optimised
  - In particular for cache misses (ParaVer tools, Barcelona)

Numerical validation

- Single vs. double precision
  - Single precision is sufficient for this problem class
  - So use single precision on CPU and GPU for a fair comparison
  - Same results except minimal floating point noise (last 2 digits)

- Application to a real earthquake
  - Bolivia 1994, Mw = 8.2
  - Lead to static offset (permanent displacement) several 100 km wide
  - Reference data from BANJO sensor array and quasi-analytical solution computed via summation of normal modes from sensor data

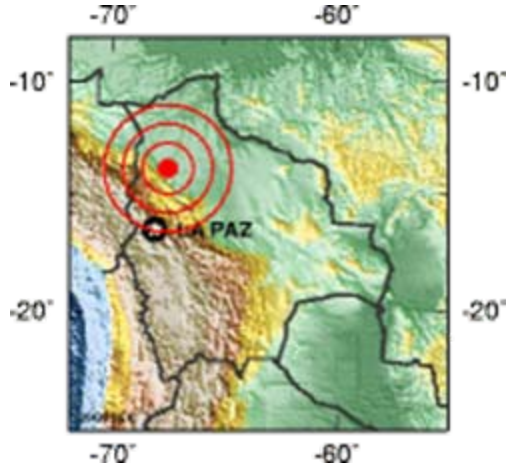
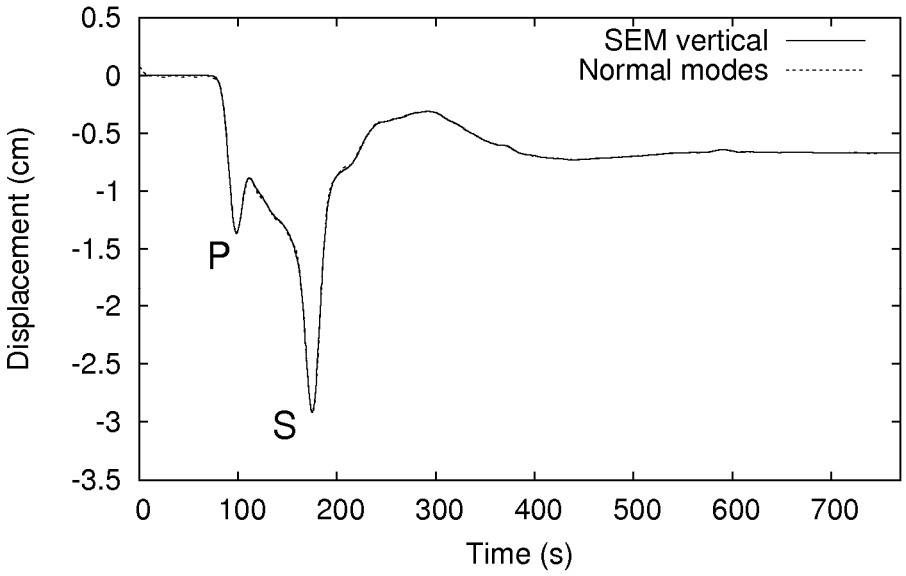


Image courtesy wikipedia

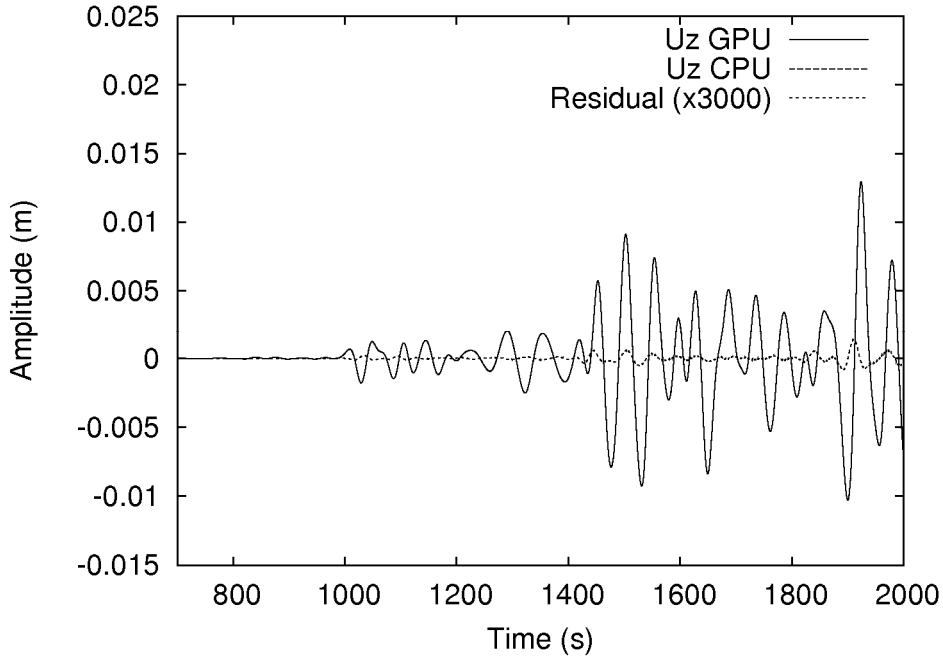
Numerical validation



Pressure and shear waves are accurately computed, static offsets are reproduced

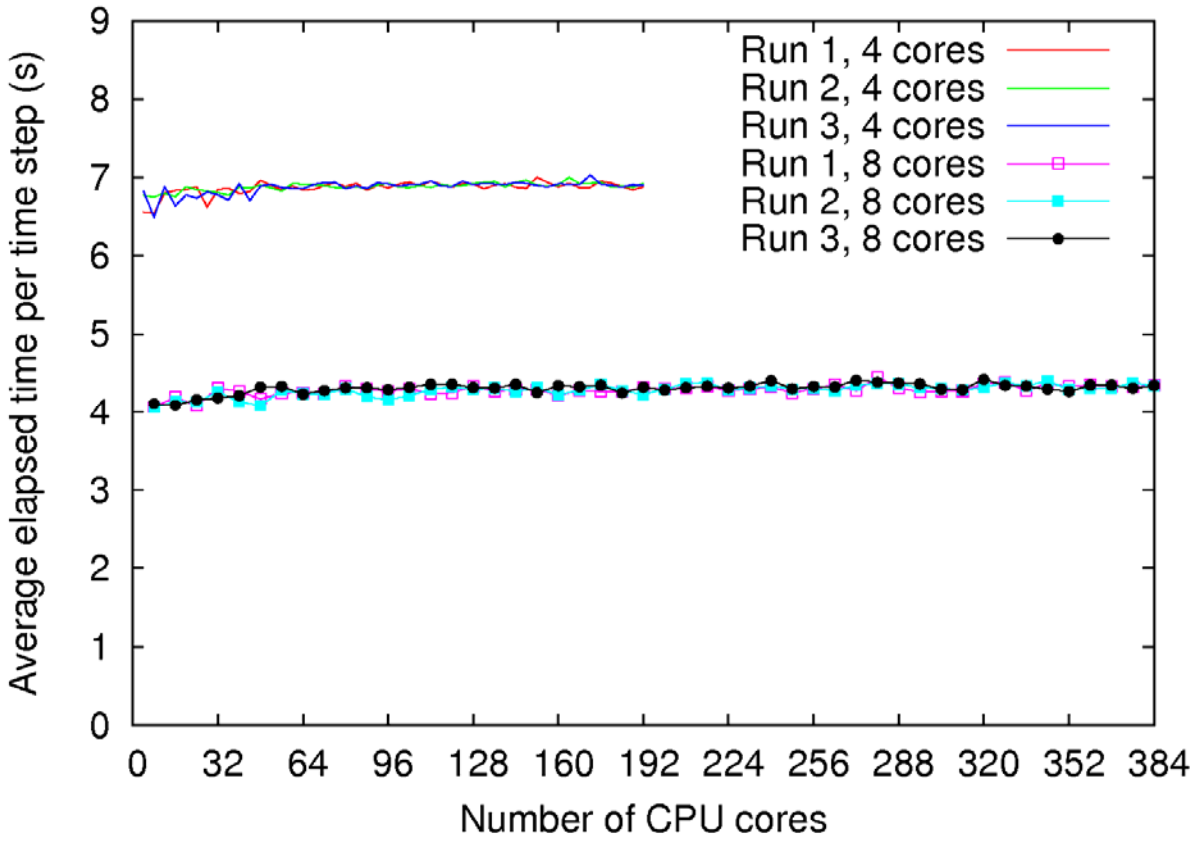
No difference between CPU and GPU solution

Amplification shows that the only difference is floating point noise



CPU weak and strong scalability

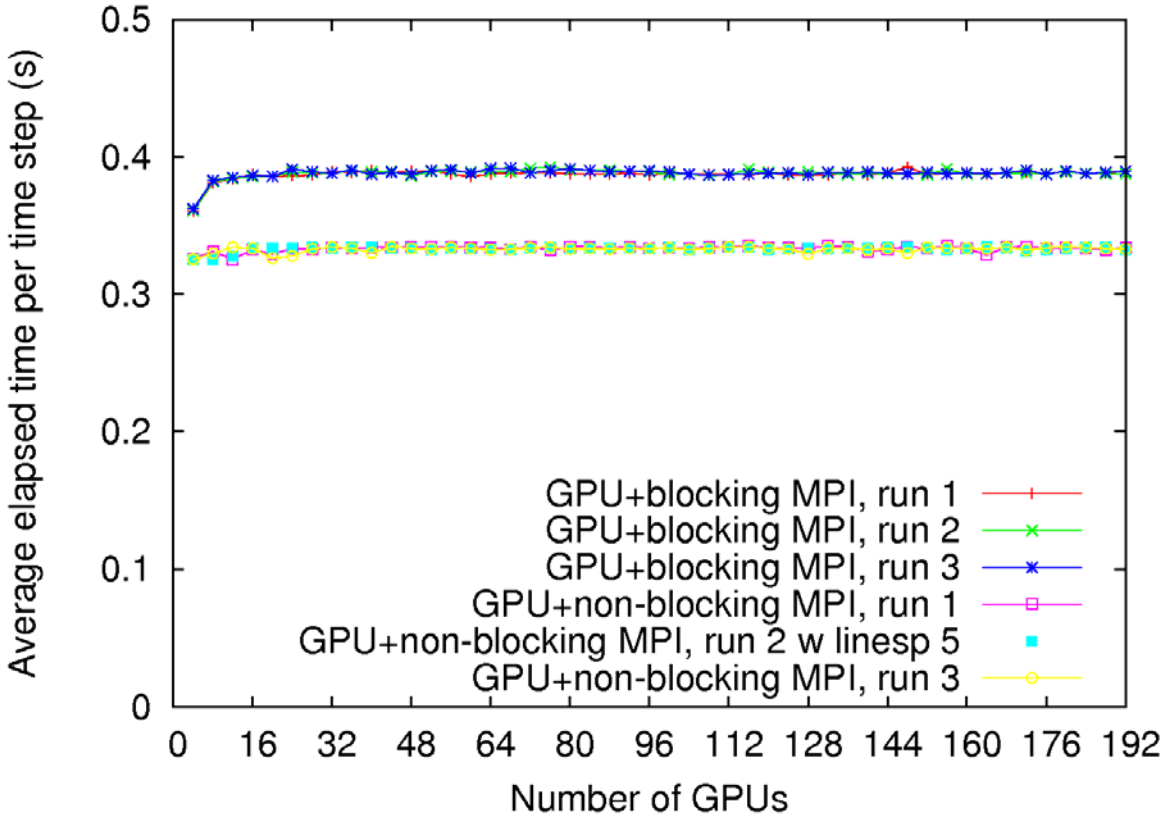
technische universität dortmund



- Constant problem size per node (4*3.6 GB or 8*1.8 GB)
- Weak scaling excellent up to 17 billion unknowns
- 4-core case uses 2+2 cores per node (proper process pinning)
- Strong scaling only 60% gain due to memory bus and network contention

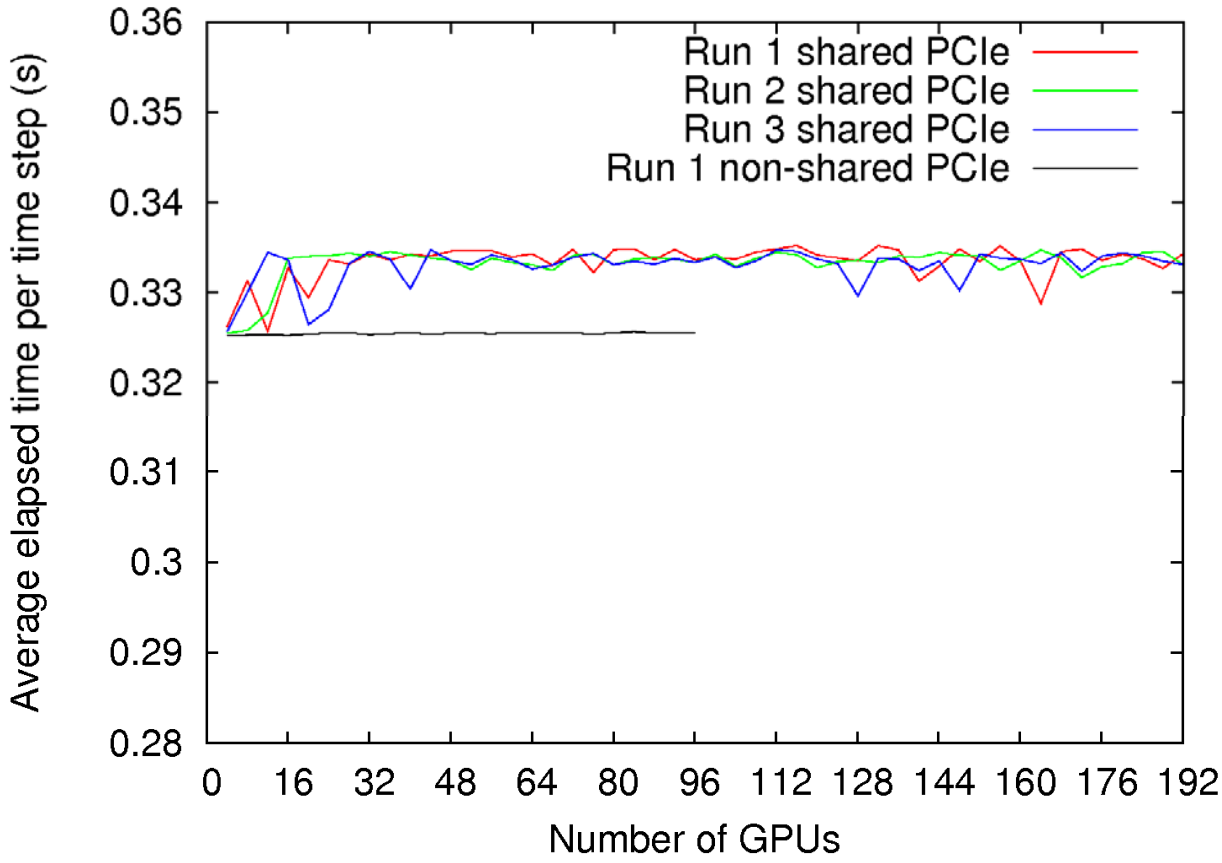
GPU weak scalability

technische universität dortmund



- Constant problem size of 3.6 GB per GPU
- Weak scaling excellent up to 17 billion unknowns
- Blocking MPI results in 20% slowdown

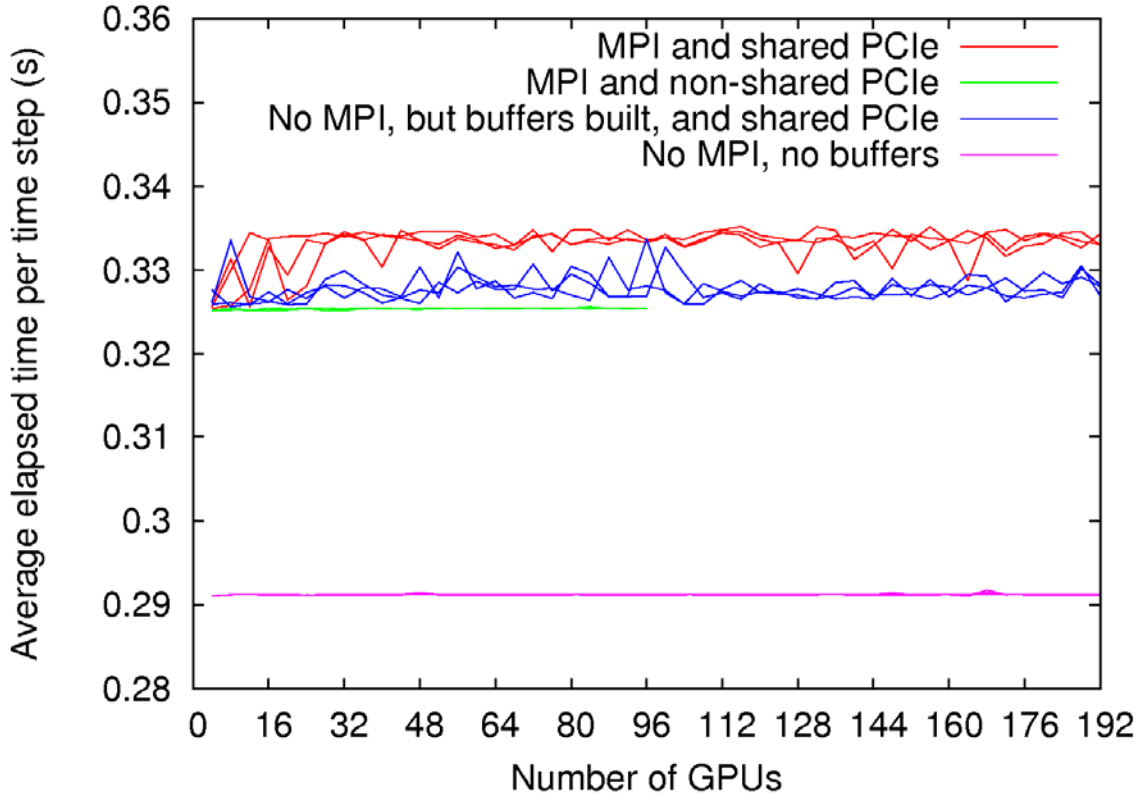
Effect of bus sharing



- 2 GPUs share one PCIe bus in the Tesla S1070 architecture
- This is a potentially huge bottleneck!
- Bus sharing introduces fluctuations between runs and a slowdown ≤ 3%

GPU performance breakdown

technische universität dortmund



- Effect of overlapping (no MPI = replace send/receive with zeroing)
- Red vs. blue curve: Difference ≤ 2.8%, i.e., very good overlap
- Green vs. magenta: Total overhead cost of running this problem on a cluster is ≤ 12% (for building, processing and transmitting buffers)

Summary

- Excellent agreement with analytical and sensor data results
  - Double precision is not necessary (neither on CPU nor on GPU)
- Excellent weak scalability
  - On CPUs and GPUs
  - Full CPU nodes suffer from memory bus and interconnect contention
  - GPUs minimally suffer from PCIe bus sharing in the S1070 architecture
  - Very good overlap between computation and communication
- Speedup
  - 25x serial
  - 20.6x vs. half the cores, 12.9x vs. all cores per node, up to 192 GPUs
  - Results for the practically relevant case in geophysics to fill up a given machine as good as possible

Acknowledgements

- French ANR grant NUMASIS ANR-05-CICG-002
- French CNRS, INRIA, IUF
- German DFG grant TU102/22-2
- German BMBF: „HPC Software für skalierbare Parallelrechner“, SKALB project (01IH08003D)