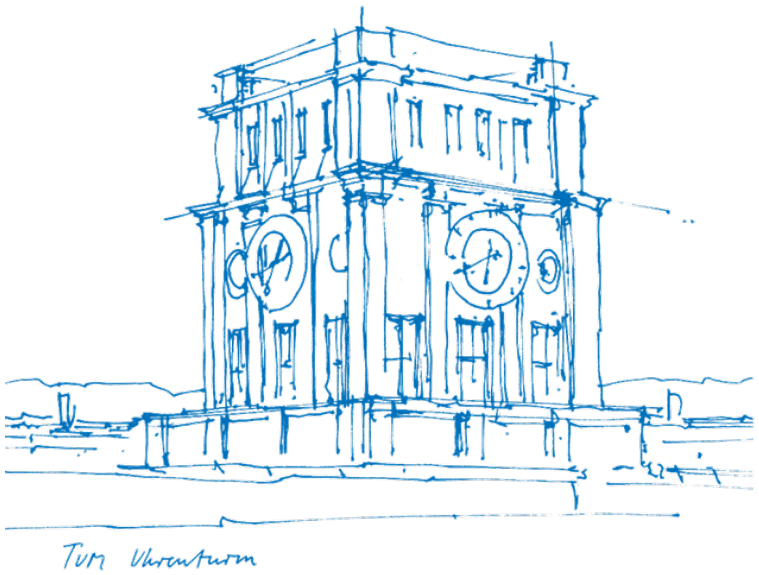
ASAGI – A Parallel Server for Adaptive Geoinformation

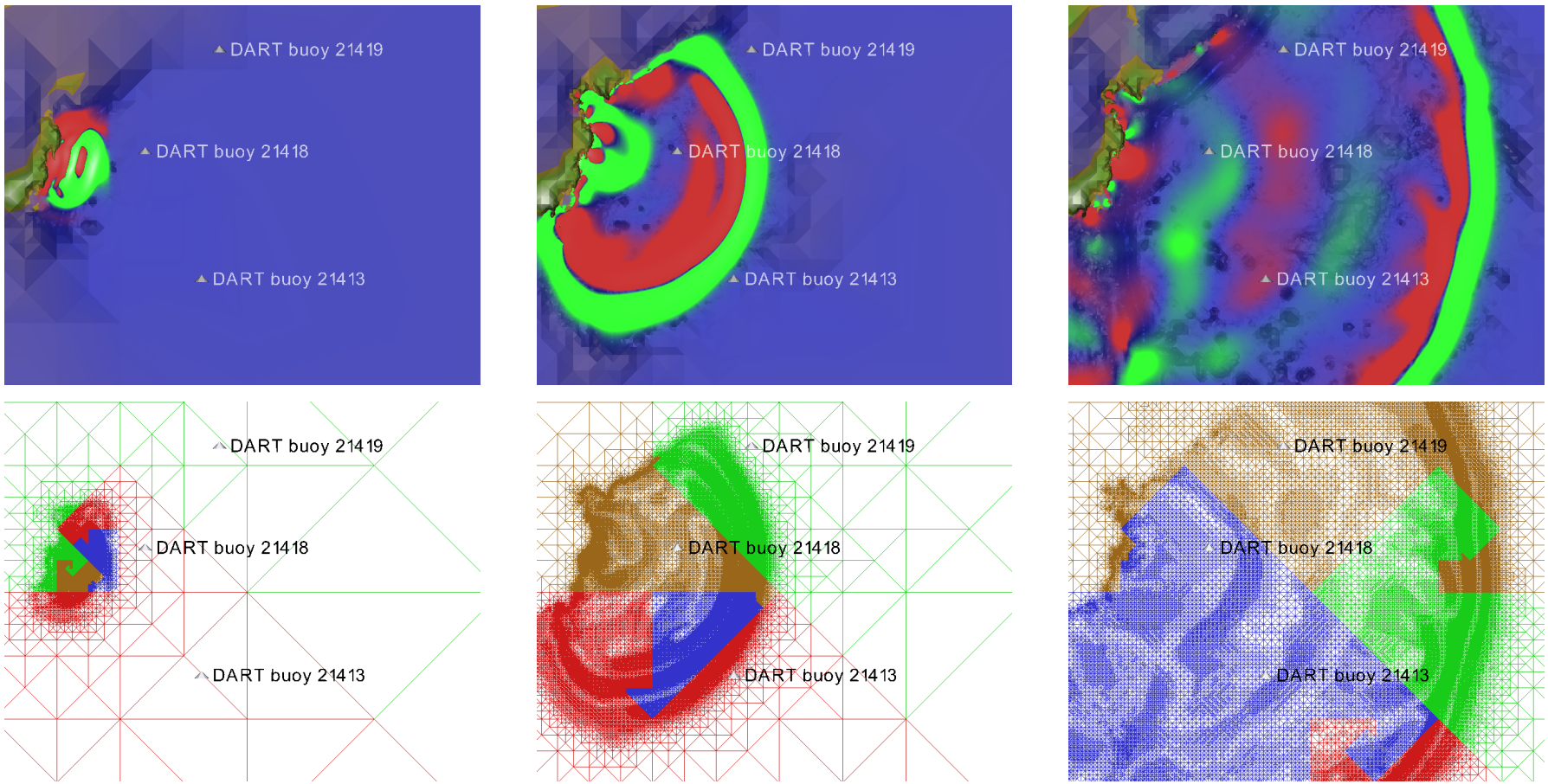
Exascale Applications & Software Conference (EASC2016)

Sebastian Rettenberger, Meister Oliver, Michael Bader,
Alice-Agnes Gabriel (Munich University)

26 April 2016



Motivation – Parallel Simulations with AMR



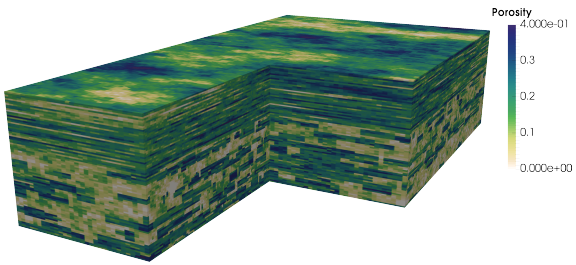
2011 Tohoku Tsunami simulated with sam(oa)^2

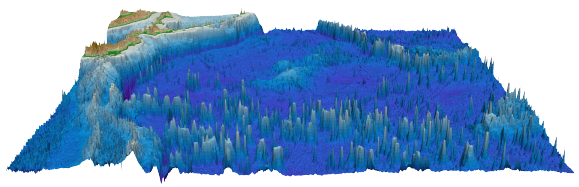
Bathymetry derived from: The GEBCO_2014 Grid

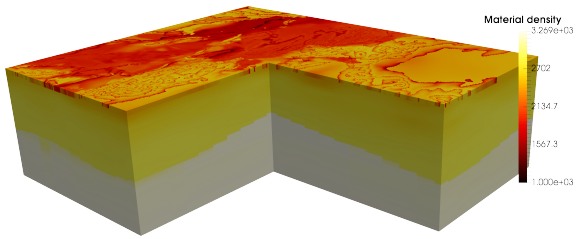
Displacement derived from: Shao, Li, Ji (UCSB) - Preliminary Result of the Mar 11, 2011 Mw 9.1 Honshu Earthquake

Geoinformation

- Material and geographic datasets
  $\rightarrow$ space- and time-dependent
- In our examples:
  - Permeability and porosity (porous media flows)
  - Bathymetry and sea floor displacement (Tsunami simulations)
  - Material velocity properties (Earthquake simulations)



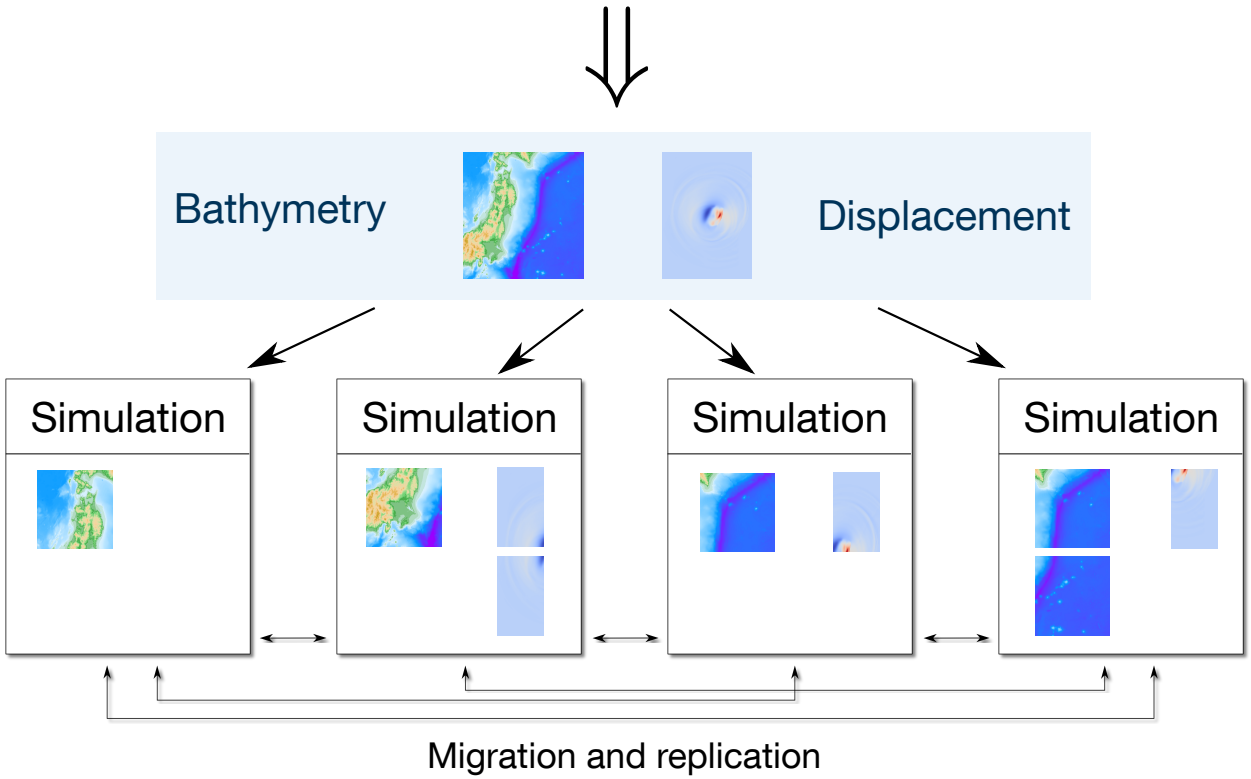




Geoinformation in Massively Parallel Simulations

How can we handle Geoinformation . . .

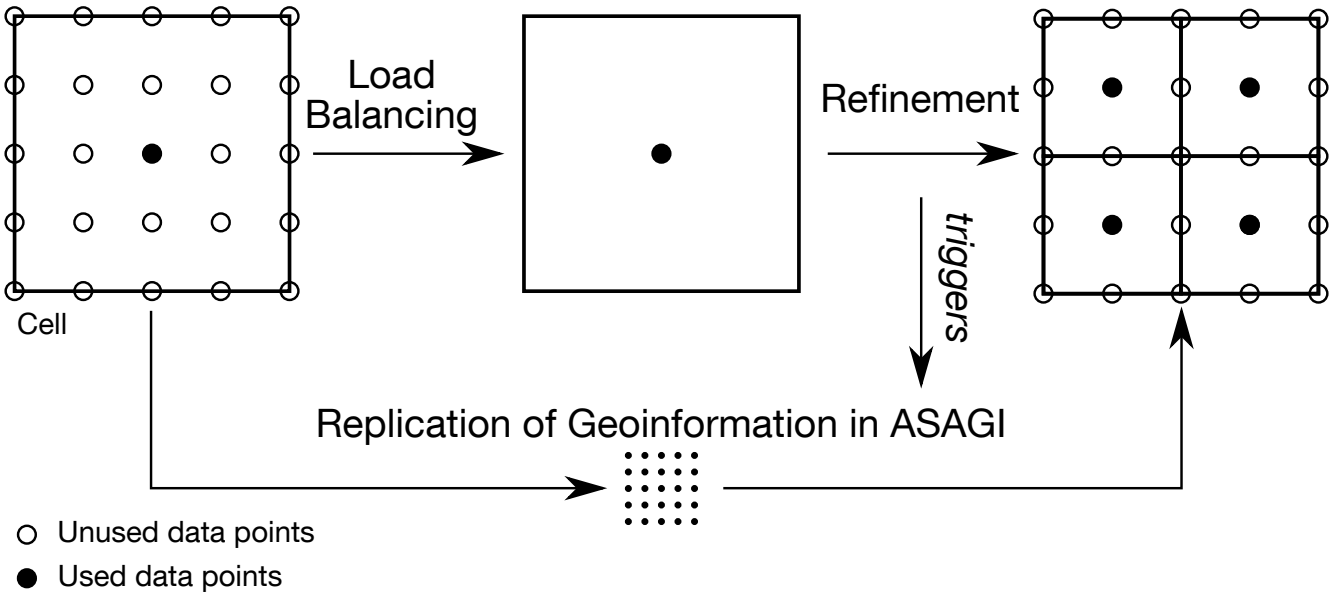
- in massively parallel simulations with adaptive mesh refinement?
- if it does not fit into the memory of a single node?
- if multiple resolutions are available?



ASAGI – A parallel Server for Adaptive Geoinformation

- Open-source library for Geoinformation
- Easy-to-use interface for integration into existing applications:
  - No knowledge about the mesh or partitioning/load-balancing required
  - Thread-safe and NUMA-aware for applications with hybrid parallelization (MPI+X)
- Interface for C, C++ and Fortran
- Datasets have to be stored as Cartesian grids
  $\rightarrow$ Is the case for many (sampled) Geoinformation datasets
- Support for multiple datasets and datasets with multiple resolutions

Replicate data on-demand

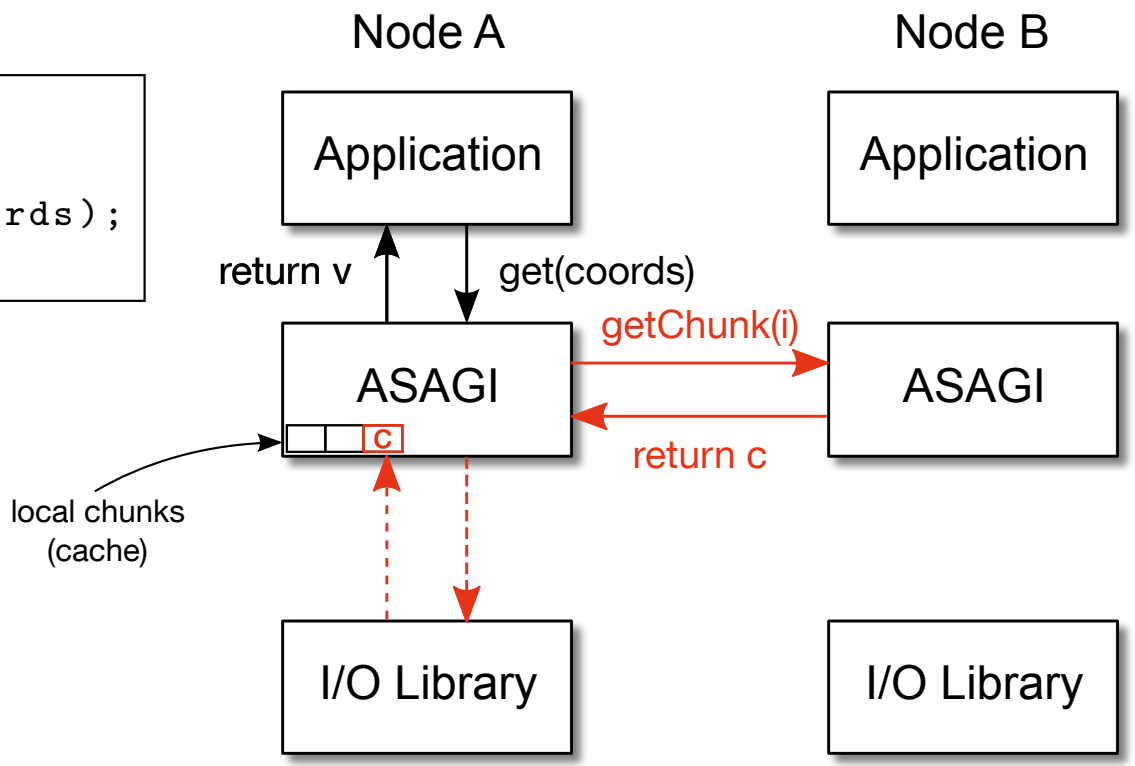


- Data is cached after replication $\rightarrow$ Additional accesses do not require communication
- Block-caching reduces the number of replications
  $\rightarrow$ Makes use of spatial and temporal locality of Geoinformation
- “Least recently” used chunks are removed from the cache

Accessing data from ASAGI

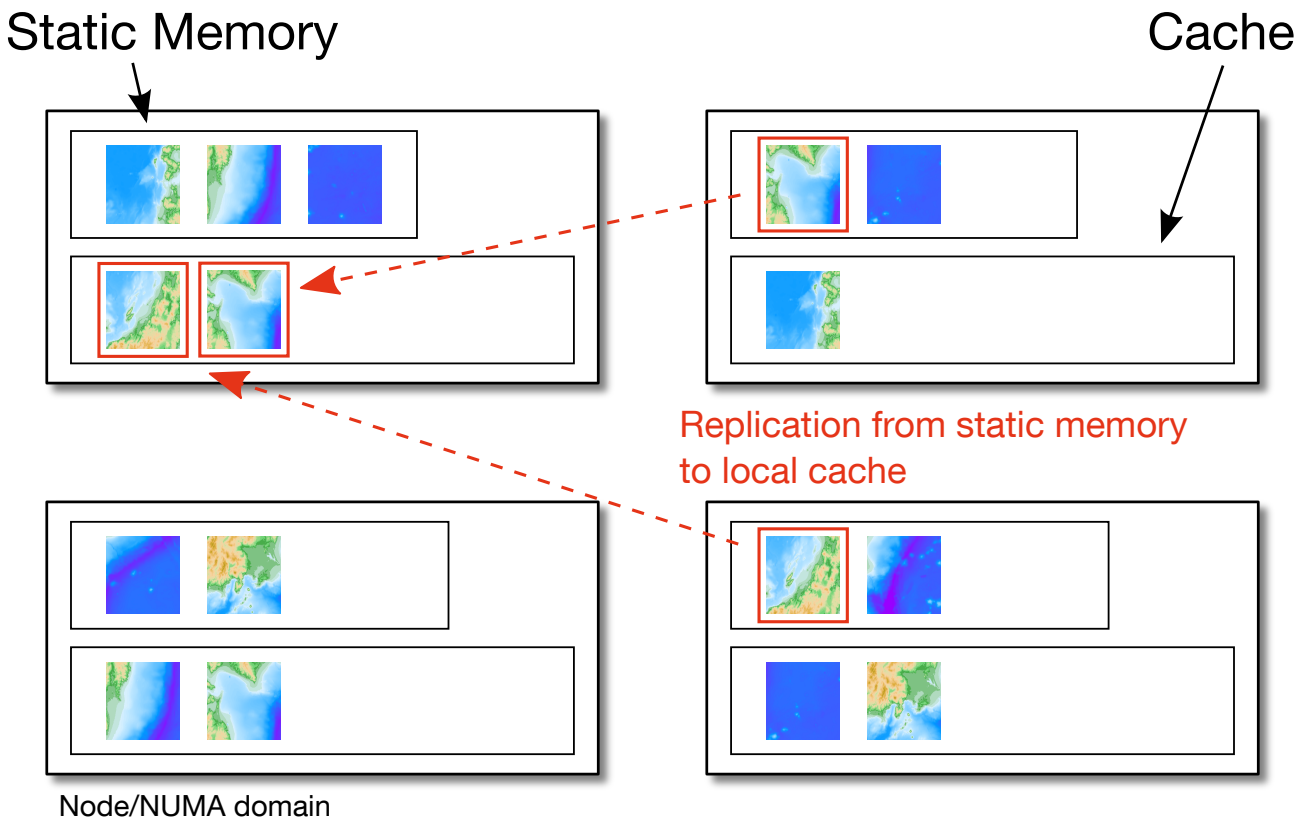
Application:

```
// [...]
float value
    = asagi->getFloat(coords);
// [...]
```



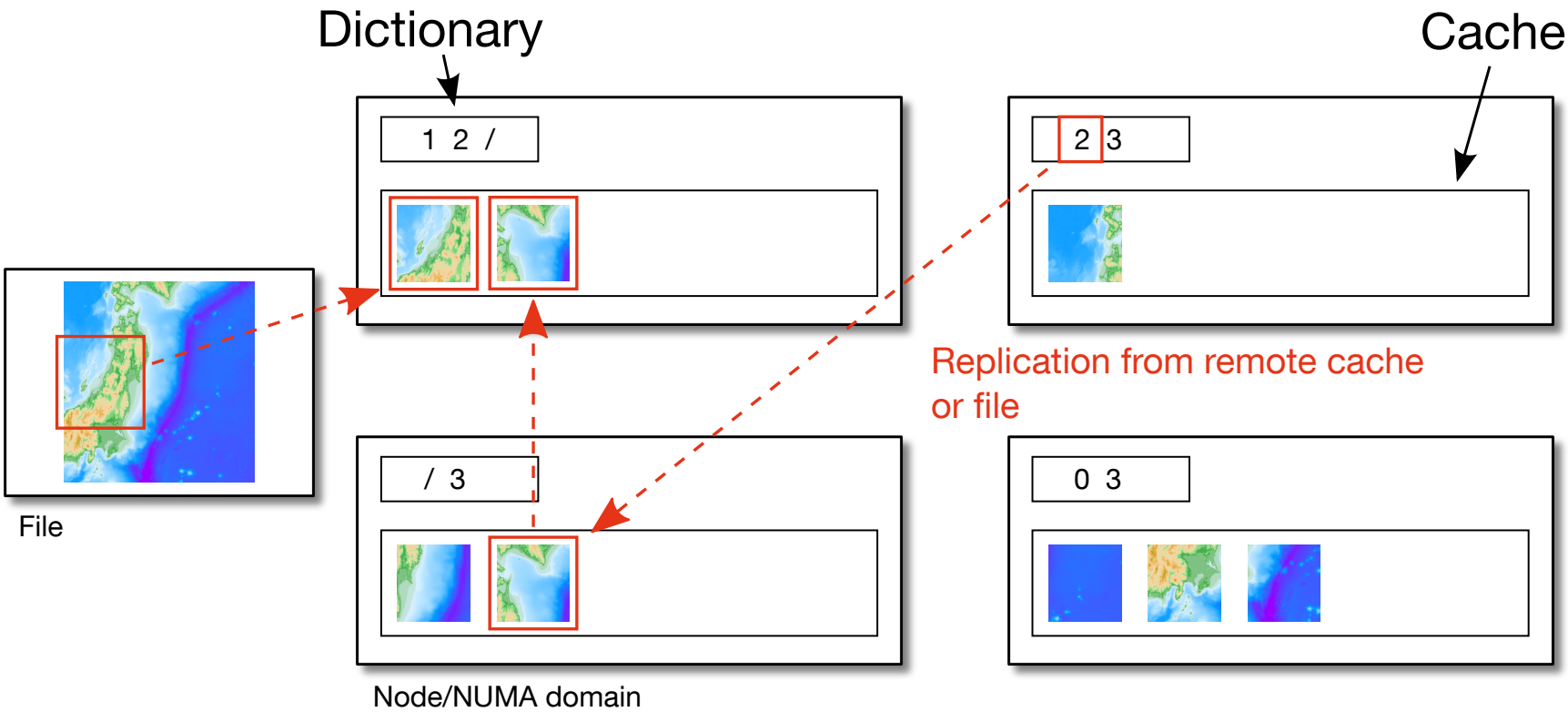
- Use local cache if possible
- Transfer chunk from other NUMA domains, MPI tasks or the I/O library

Storage – Full Mode



- Static memory is set up at initialization and never changed
  $\rightarrow$ contains the complete dataset
- Missing chunks copied to local cache from static memory of other nodes/NUMA domains

Storage – Cache Mode



- Dictionary contains locations of chunks on other nodes/NUMA domains
  $\rightarrow$ is updated after every replication
- File is used if a chunk is not stored in any cache

Communication

Communication between nodes:

- Disabled
  $\rightarrow$ In full mode, each node stores the whole dataset
- Remote memory access (using MPI windows)
  $\rightarrow$ complex synchronization in cache mode with MPI mutexes
- An explicit communication thread (with `MPI_send`, `MPI_recv`)
  $\rightarrow$ One core is required by ASAGI for a communication thread (useful with hybrid parallelization)

NUMA detection:

- Automatic detection based on libnuma
  $\rightarrow$ One cache per NUMA domain
- Requires pinning of threads (automatically done in OpenMP)

SuperMUC Phase 1

- Ranked 23rd on Nov'15 TOP500 list
- 9216 dual socket Intel Xeon E5-2680 (Sandy Bridge) nodes with 16 cores and 32 GB memory each
- 18 islands connected via an 4:1 pruned tree with FDR10
- Peak performance: 3.2 PFlop/s
- GPFS file system with 180 GB/s
- IBM MPI and Intel MPI available

sam(oa)^2

- Parallel framework for Partial Differential Equations (PDEs)
- Dynamically adaptive triangular mesh based on the Sierpinski space-filling curve traversal

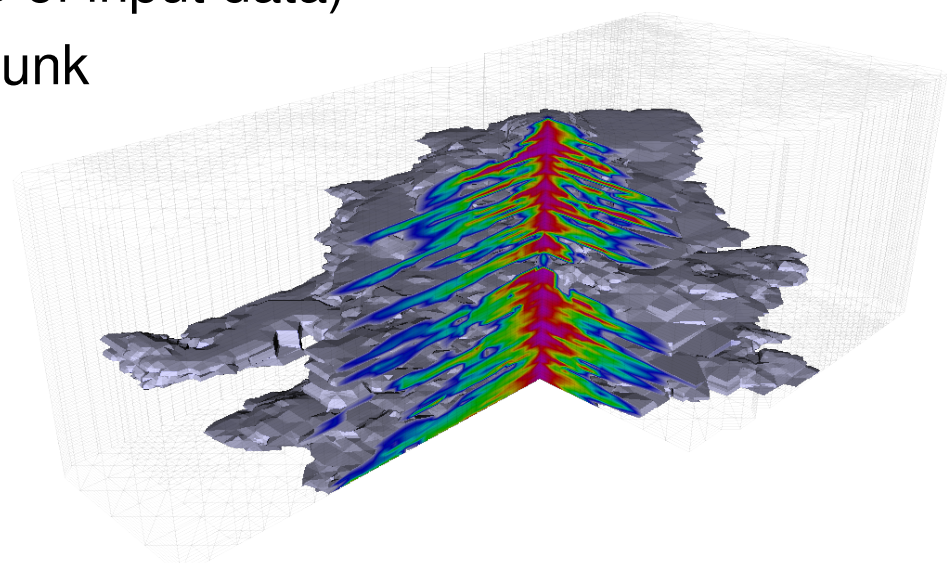
- 2 phases:
  - Initialization phase starts with a very small number of cells and successively refines and distributes the grid to multiple MPI tasks
  - Time stepping phase allows further refinement and coarsening in each time step

  $\rightarrow$ ASAGI provides Geoinformation in both phases for every refinement/coarsening
- ASAGI is tightly integrated into the refinement/coarsening step

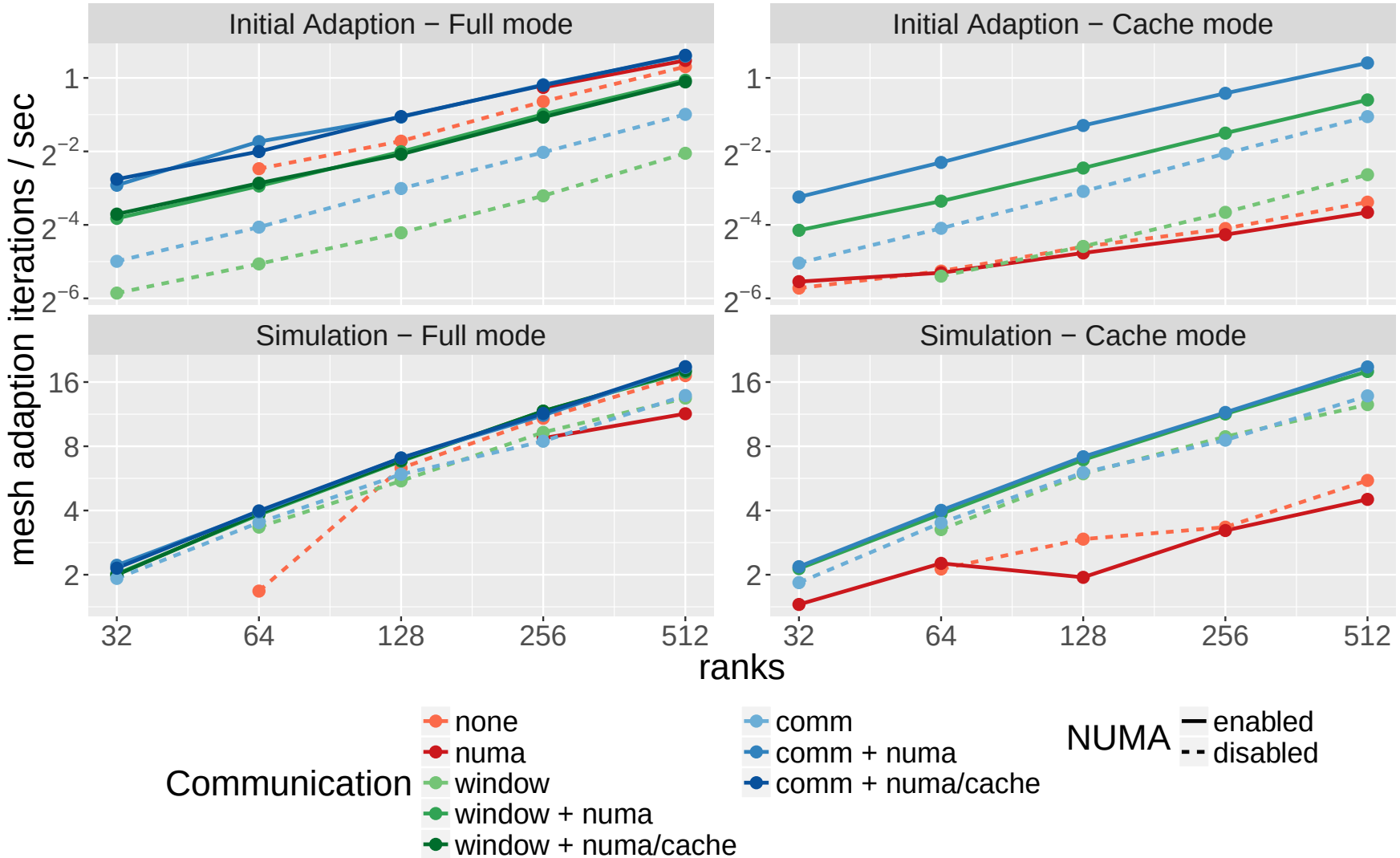
- Hybrid Intel MPI+OpenMP parallelization with one task per node

Two-Phase Porous Media Flow

- 2.5D oil reservoir simulation
  $\rightarrow$ AMR in horizontal dimensions and uniform refinement in vertical dimension
- 8,000 to 33 million cells with 340 layers
- ASAGI provides permeability tensor and cell porosity
  (4$\times$ 1.1 billion data points; 17 GB of input data)
- 64 $\times$ 64 $\times$ 340 data points per chunk
- 5.3 GB cache

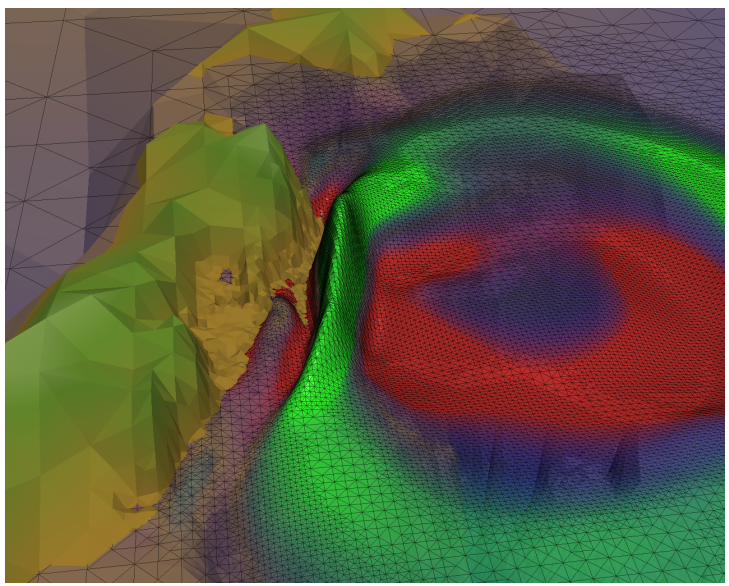


Two-Phase Porous Media Flow – Results

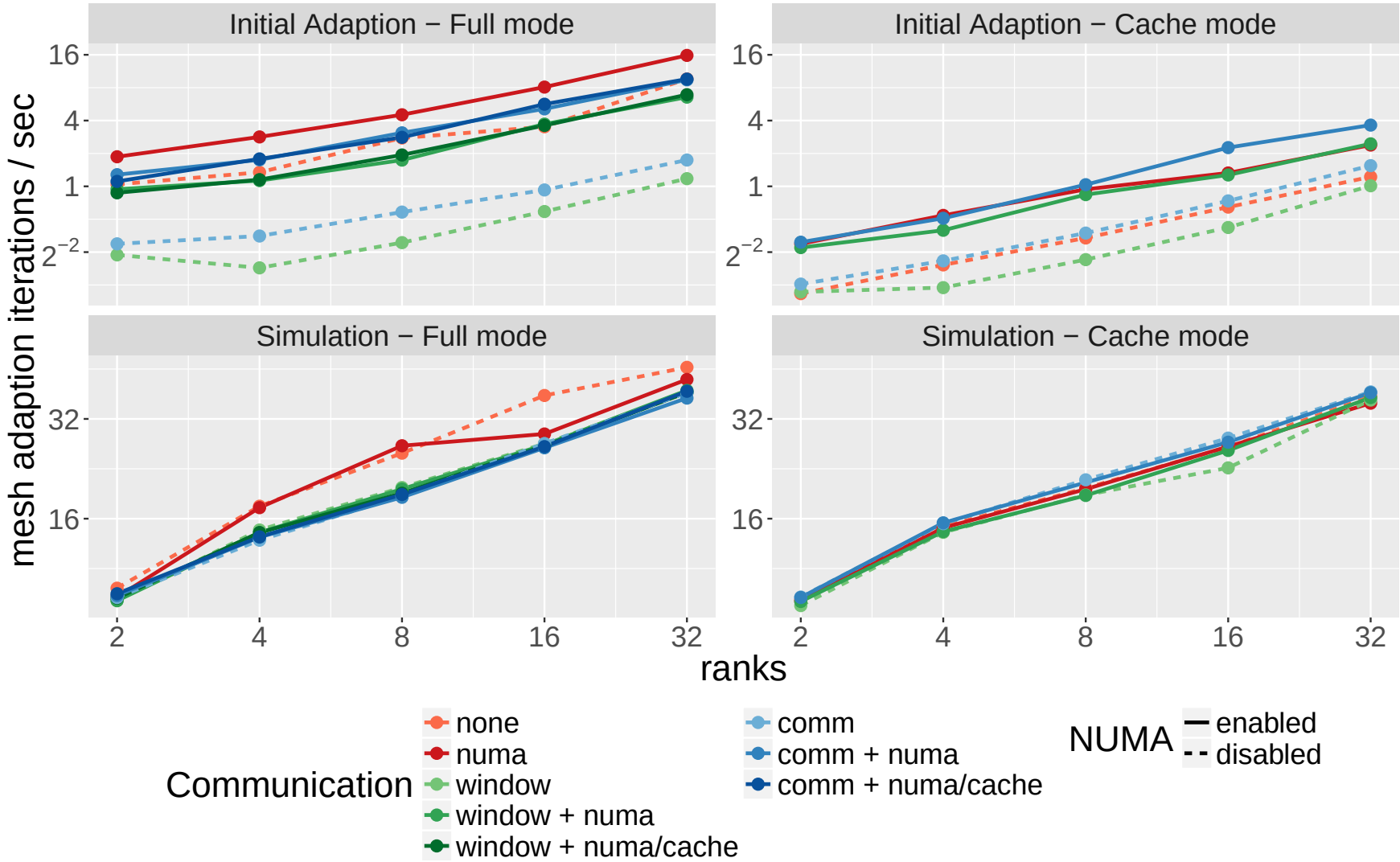


Tsunami Simulation with Time-Dependent Displacement

- Simulation based on the 2011 Tohoku tsunami with time-dependent displacements
- 427 MB bathymetry data and
  2.8 GB displacement data with 80 time steps
- $128 \times 128 \times 4$ data points per chunk
- 32 + 128 MB cache

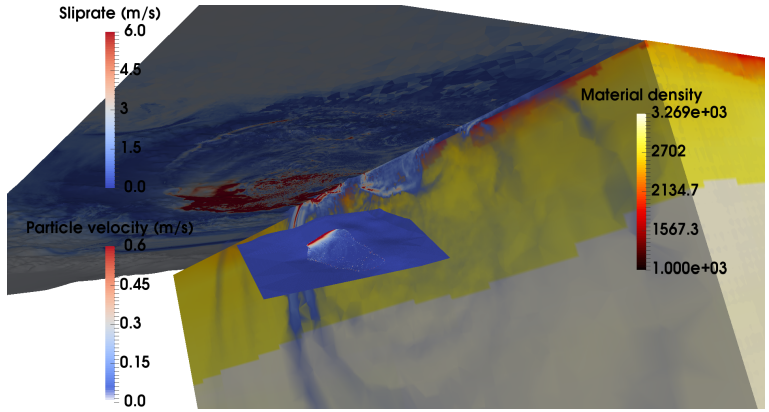


Tsunami Simulation – Results



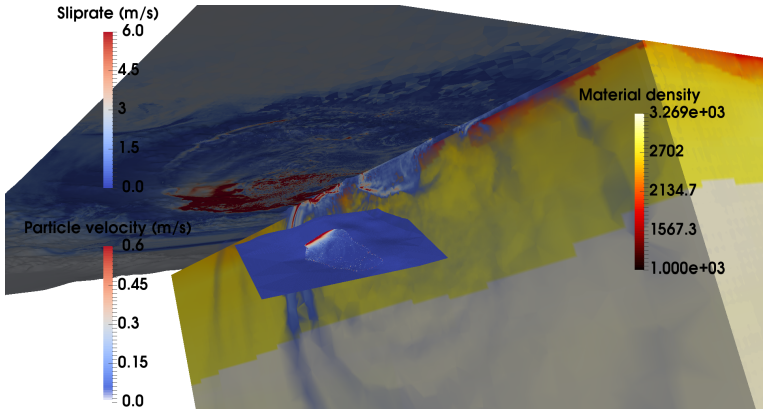
SeisSol

- Earthquake simulation code that couples wave propagation to dynamic rupture simulations
- Based on static but fully unstructured tetrahedral meshes
  $\rightarrow$ ASAGI only required for the initialization
- 3D velocity models (density, shear modulus, first Lamé parameter) provided by ASAGI

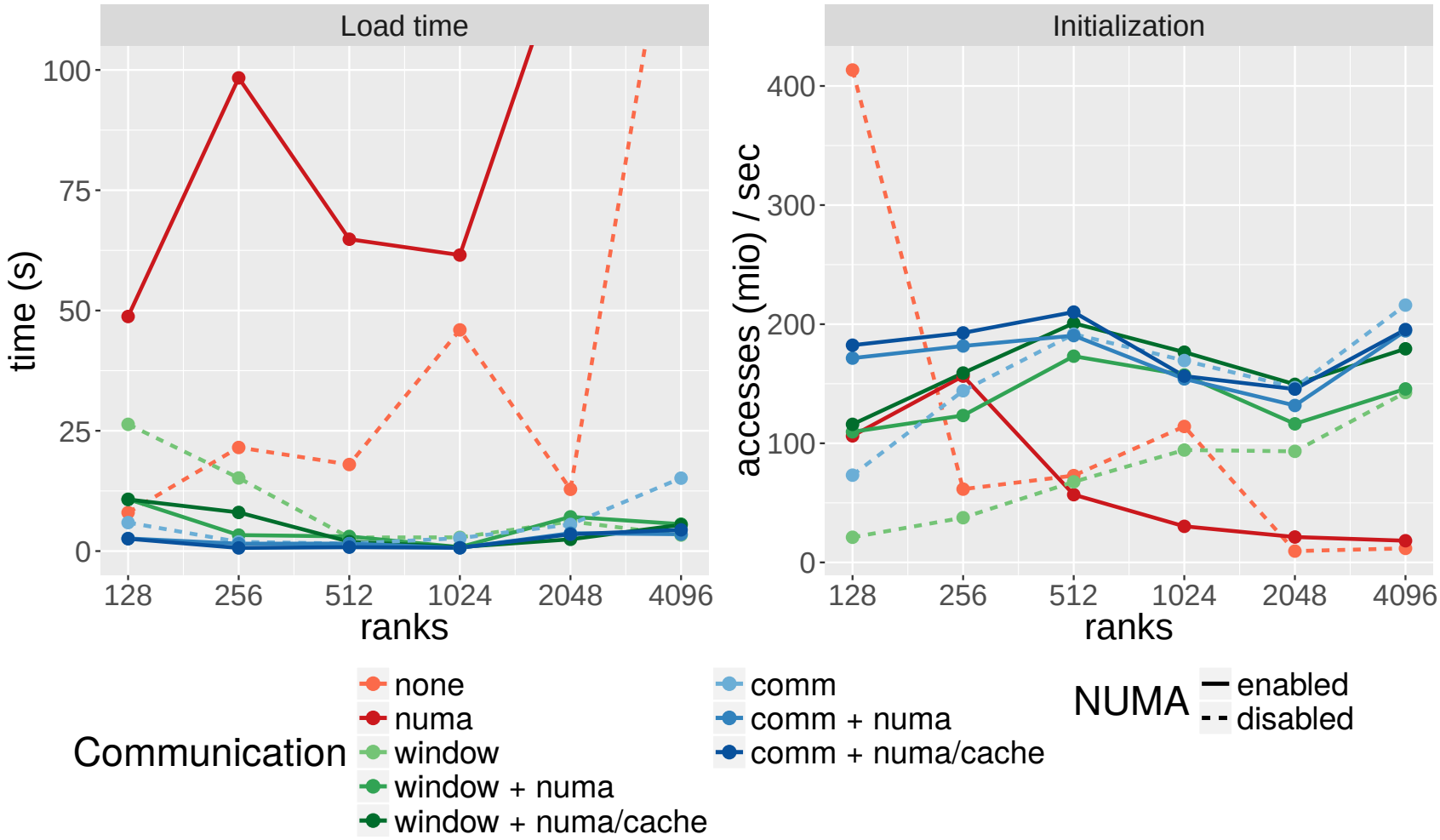


SeisSol – 1994 Northridge Earthquake

- Mesh with 75 million cells
- ASAGI provides 3D velocity with 527 million data points (5.9 GB) derived from SCEC community velocity model Harvard (CVM-H)
- $32 \times 32 \times 32$ data points per chunk
- 48 MB cache
- Hybrid IBM MPI+OpenMP parallelization with one task per node



1994 Northridge – Results (full mode)



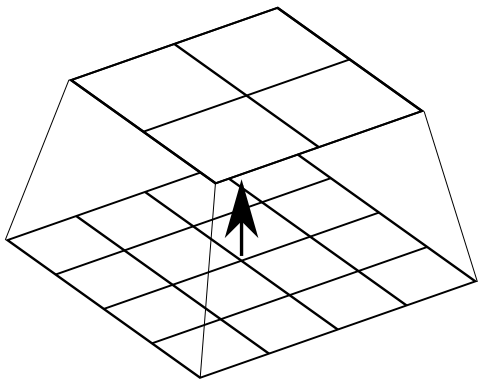
Outlook

New query interfaces:

- List of data points, range queries
- Toolbox for interpolation and averaging

```
// Range query
asagi->getRangeFloat(start, end, &values);
// Averaging
value = getAvgFloat(asagi, start, end);
```

Support for generation of
coarse resolutions on the fly



Integration into the European
Horizon 2020 project “ExaHype”

https://github.com/TUM-I5/ASAGI

or contact

rettenbs@in.tum.de