Institut für Informatik — TU München
Scientific **C**omputing in **C**omputer **S**cience
M. Bader, A. Breuer

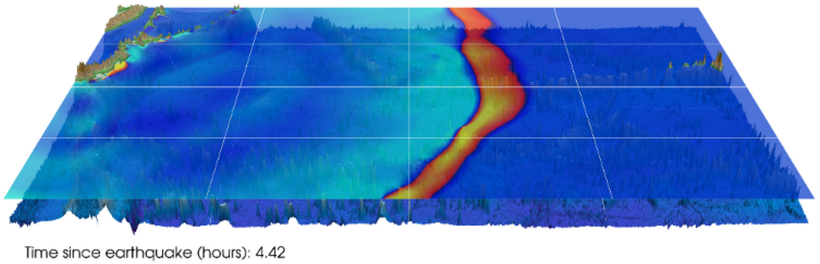
WS 2014/15

Masterpraktikum Scientific Computing (High Performance Computing) Project: Shallow Water Equations

12/15/2014

Optimization of SWE

The project is about optimization of the shallow water code SWE, which is freely available at `https://github.com/TUM-I5/SWE`. SWE uses a Finite Volume discretization of the shallow water equations, which fall into the general of group non-linear hyperbolic partial differential equations.
While the shallow water equations are typically used for tsunami or dam-break simulations, the numerical discretization techniques and resulting computational challenges for other hyperbolic PDEs are usually very similar, further prominent hyperbolic examples are: Earthquake simulations, weather and climate simulations or aerodynamics.



In principle all concepts of the previous sessions are applicable for optimization of SWE, however we consider the following approaches to be "promising":

- Manual instrumentation of the entire code to identify bottlenecks and performance improvements, i.e. with `http://scalasca.org/`
- Use of intrinsics for vectorization throughout the entire code (and checking the current implementation for non-vectorized loops); also consider the potential performance gain vs. working time for your implementation! Usually initialization routines are sufficiently optimized by the compiler!
- Implementation of different blocking strategies to improve cache behavior and resulting performance of the code: for example, using the existing `SWE_Blocks` for blocking vs. implementation of blocking within each `SWE_Block`;
- Optimization of the hybrid MPI/OpenMP-parallelization of the code (incl. strategies to overlap communication and computation);
- "Fusion" of the two main kernels to compute numerical fluxes and to update all unknowns into one single loop over all cells (to reduce the number of accesses to main memory);
- Use of emerging architectures is highly encouraged: Xeon Phi (SuperMIC), AMD FirePro W8000 GPUs (MAC Cluster) or NVIDIA M2090 GPUs (MAC Cluster). For the Phis *Intel Cilk Plus Array Notation* or *OpenCL* are promising additional approaches, for the AMD GPUs *OpenCL* is required, for the NVIDIA GPUs *CUDA* is recommended.

It is your job to optimize SWE, which includes choosing an appropriate optimization strategy, a detailed documentation of your steps and thorough analysis & evaluation of the resulting improvements (performance, scalability). Be aware that the above approaches are suggestions and your own optimization strategies are welcome.
Note that it will not be possible to tackle all of the above approaches in the given time. You should discuss possible routes with your supervisor before starting the project.

Deliverables

- Optimization strategy (1-2 pages)
- Git-repository, which documents all of your code-changes
- Documentation how to build and execute your code
- All slides, animations, graphs, . . . created throughout the project phase

Roadmap

- 01/05/2015, 8AM: Hand in of your optimization strategy
- 01/12/2015, 4PM: Milestone meeting: Presentation and discussion of your strategy and status
- 01/19/2015, 8AM: Hand in of all final results
- 01/19/2015, 4PM: Final presentation and closing

Have fun!