Performance–Optimized Lattice Boltzmann

System Simulation Group
University of Erlangen–Nürnberg, Germany
Thomas Pohl

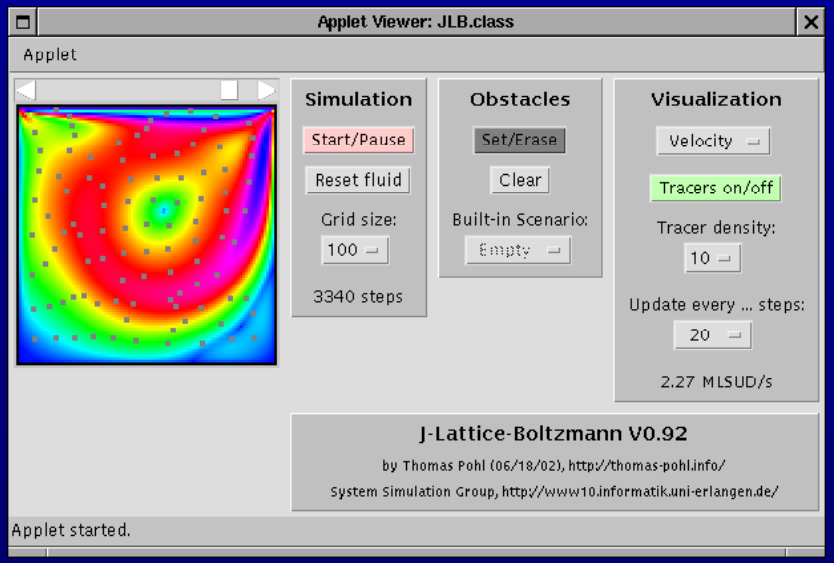
Outline

1. What is Lattice Boltzmann (LB)?
2. LB Implementation in Java
3. Cache Optimizations for LB
4. Applications in Material Science
5. Conclusions & Future Work

A. Grid Computing with LB

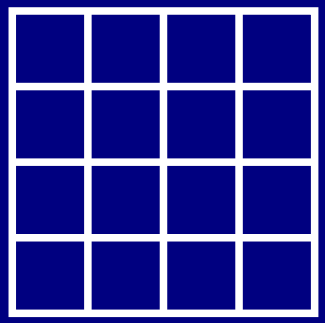
What is Lattice Boltzmann (LB)?

- First introduced by McNamara and Zanetti in 1988
- Another method to do computational fluid dynamics (CFD)
- Weakly compressible approximation of the Navier–Stokes equations
- Good for sufficiently small Mach numbers (< 0.1)
- Very easy to implement

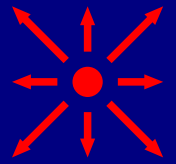


Basic Ingredients of LB

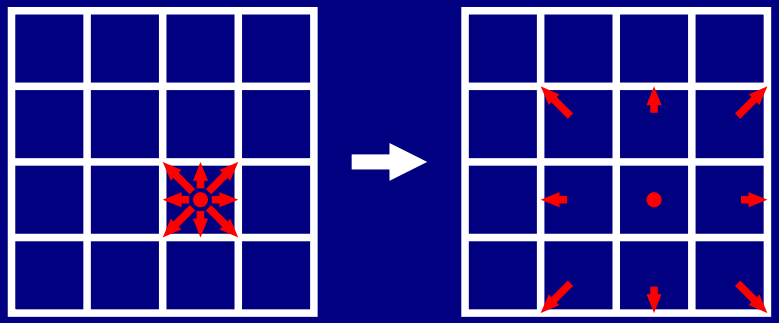
1. Lattice consisting of several cells



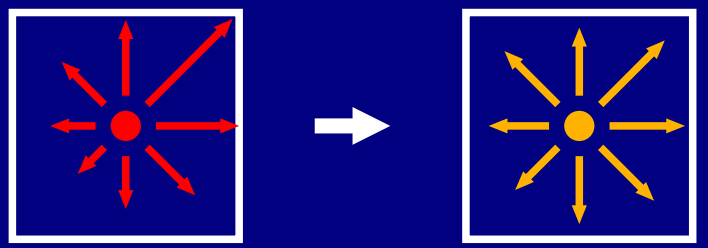
2. Distribution functions $f_i(x, y)$ for each cell



3. Streaming step

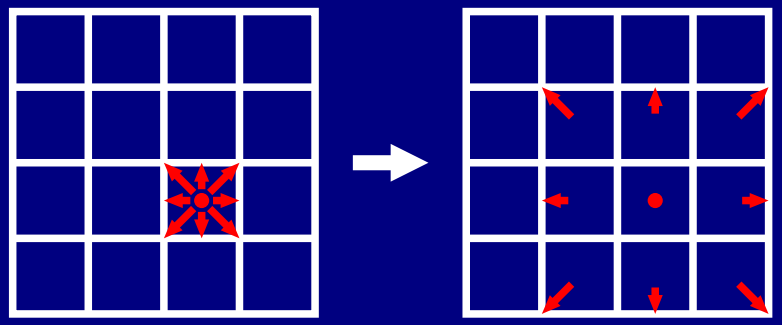


4. Collision step

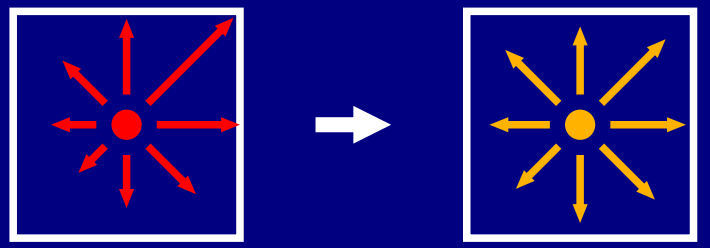


Fused Streaming/Collision Step

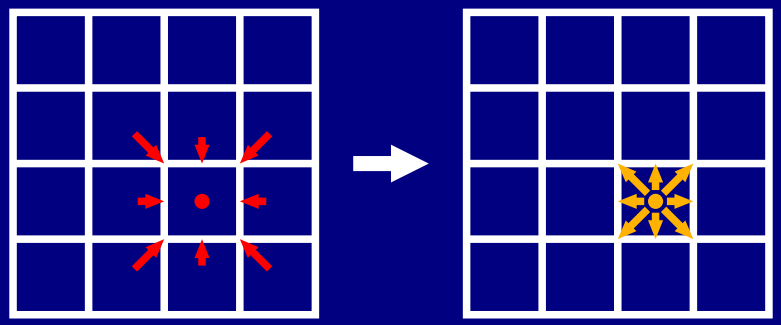
Streaming step



Collision step



Fused streaming/collision step



LB compared to Jacobi Relaxation

Similar:

- Complete sweep over the lattice
- Storage in two separate grids
- Data retrieval for a 9–point stencil

Different:

- Notion: time step $\longleftrightarrow$ iteration
- More data per cell
- More floating–point operations per cell
- Data storage

$\Longrightarrow$ **Optimization techniques for Jacobi should also work for LB**

Cache Optimizations for LB

Optimize performance

⇑

Put cache to work

⇑

Improve spatial and temporal
locality of data access

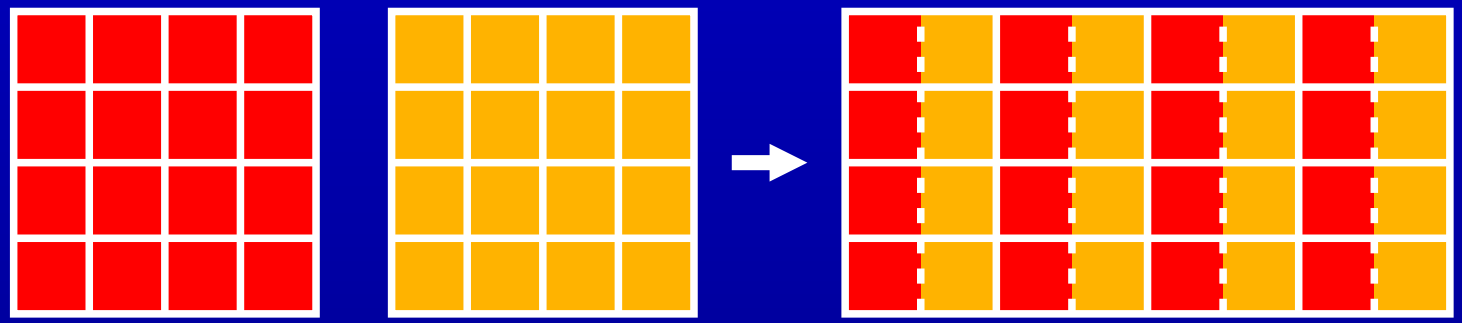
⇑

Techniques:

- Optimize data layout
- Optimize data access

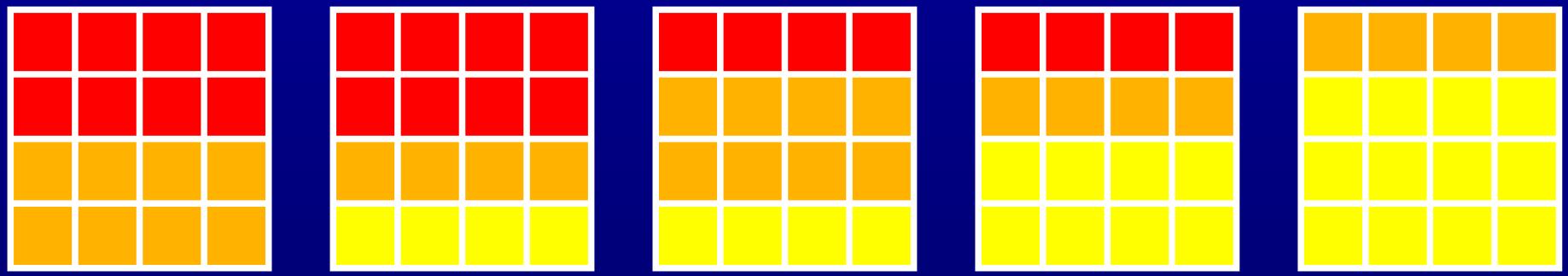
Two Examples for Cache Optimizations

Optimizing data layout: Array Merging



$\Longrightarrow$ **Improved spatial locality**

Optimizing data access: Loop Blocking

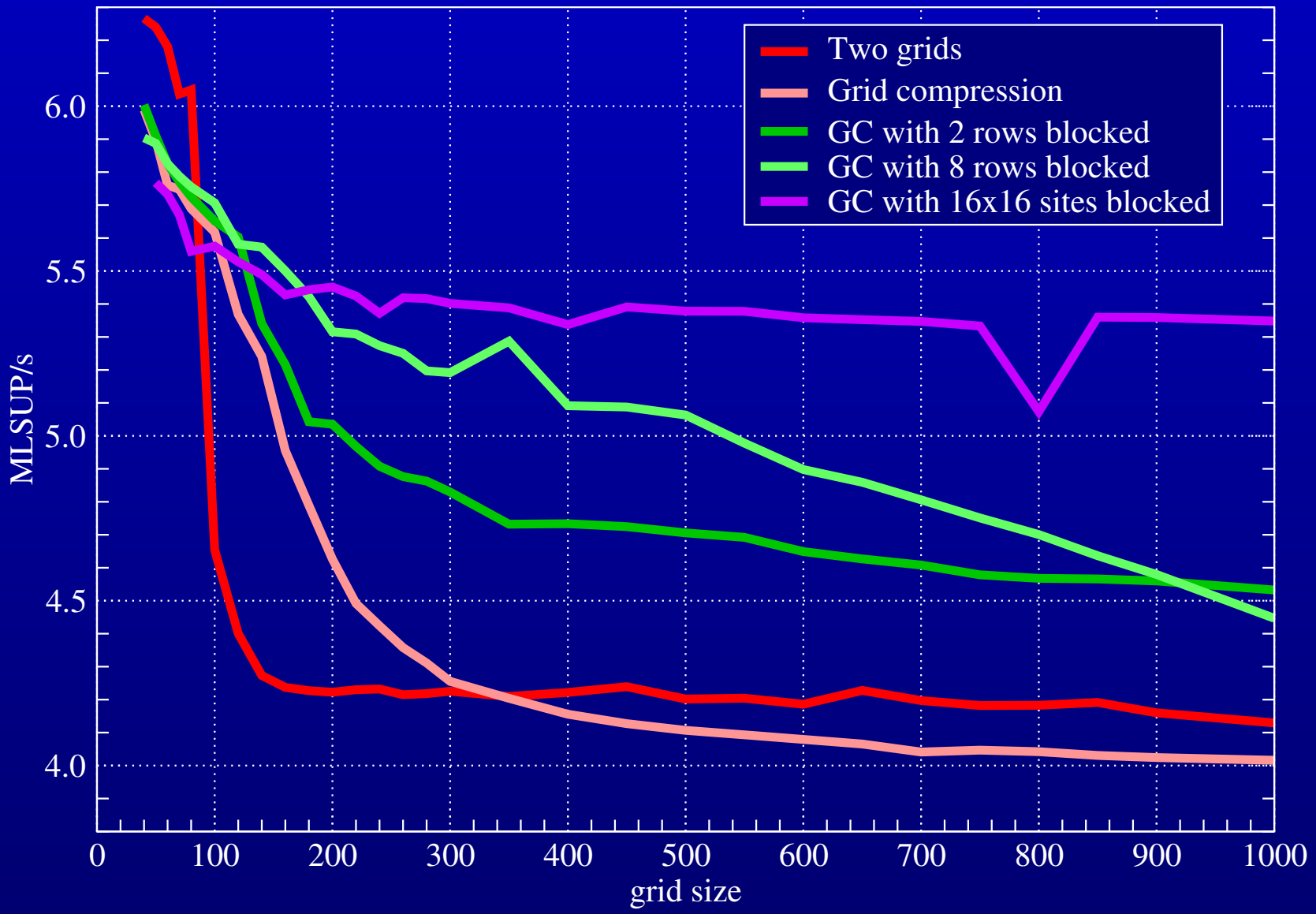


$\Longrightarrow$ **Improved temporal locality**

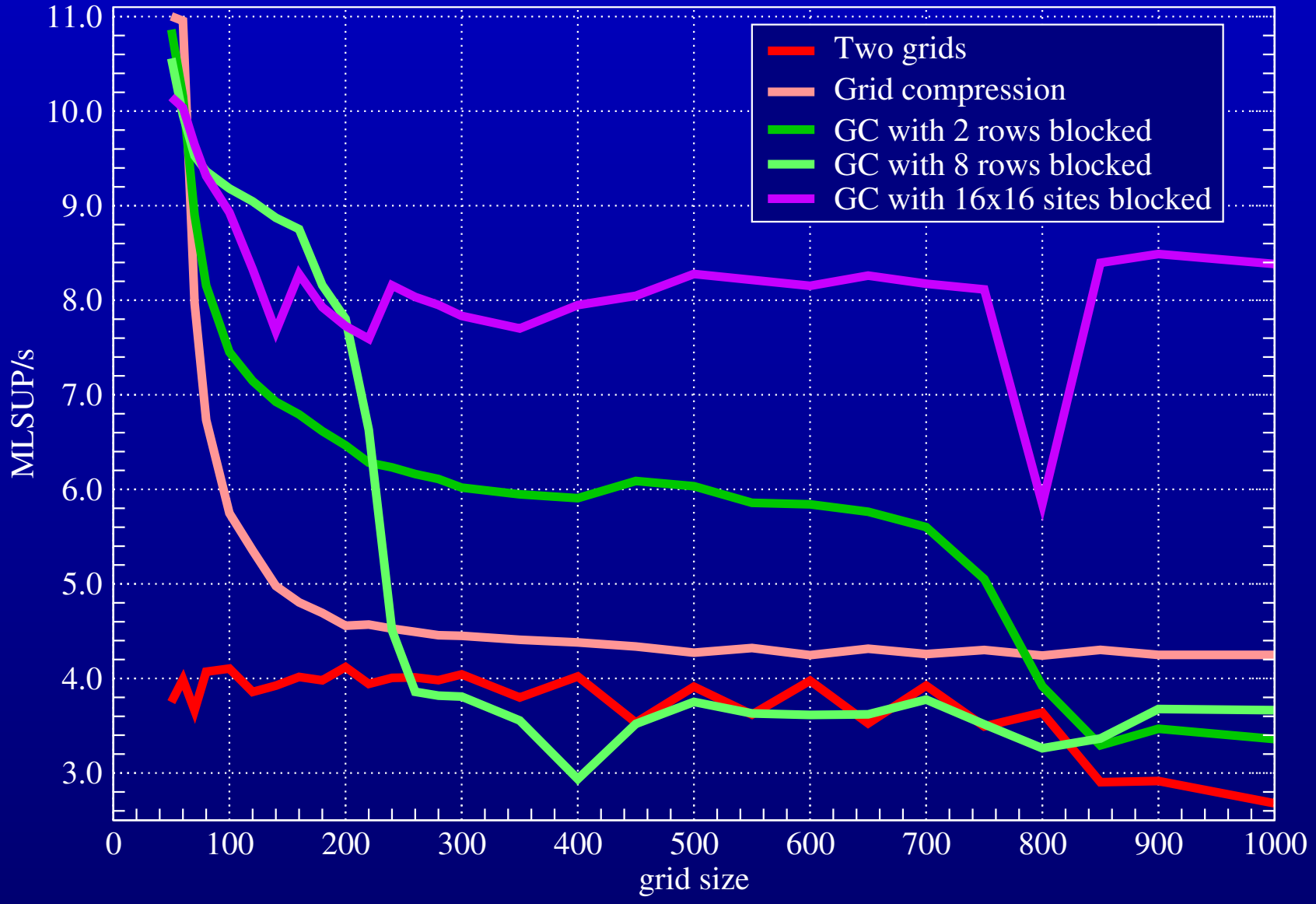
Cache–Optimized LB

- Jens Wilke's undergraduate thesis
- Supervised by Markus Kowarschik and Thomas Pohl
- Research and implementation of another data layout: grid compression (GC)
- Performance tests for 1–way and 2–way loop blocking
- Thesis available (in English) on `http://www10.informatik.uni-erlangen.de/` link "Research"

Intel Itanium2 0.9 GHz



AMD Athlon XP 2.0 GHz



Applications in Material Science

- Cooperation with Michael Thies and Carolin Körner, Material Science and Technology of Metals, University of Erlangen–Nürnberg
- **Goal:** Simulation of the fabrication process of metal foam on the supercomputer Hitachi SR8000–F1 at the Leibniz Computing Center in Munich
- Supported by the Competence Network for Technical, Scientific High Performance Computing in Bavaria **KONWIHR**
- **Results so far:** 3D LB code with free surfaces and surface tension implemented by Nils Thürey
- Ray tracer and animations by Nils Thürey

Conclusions & Future Work

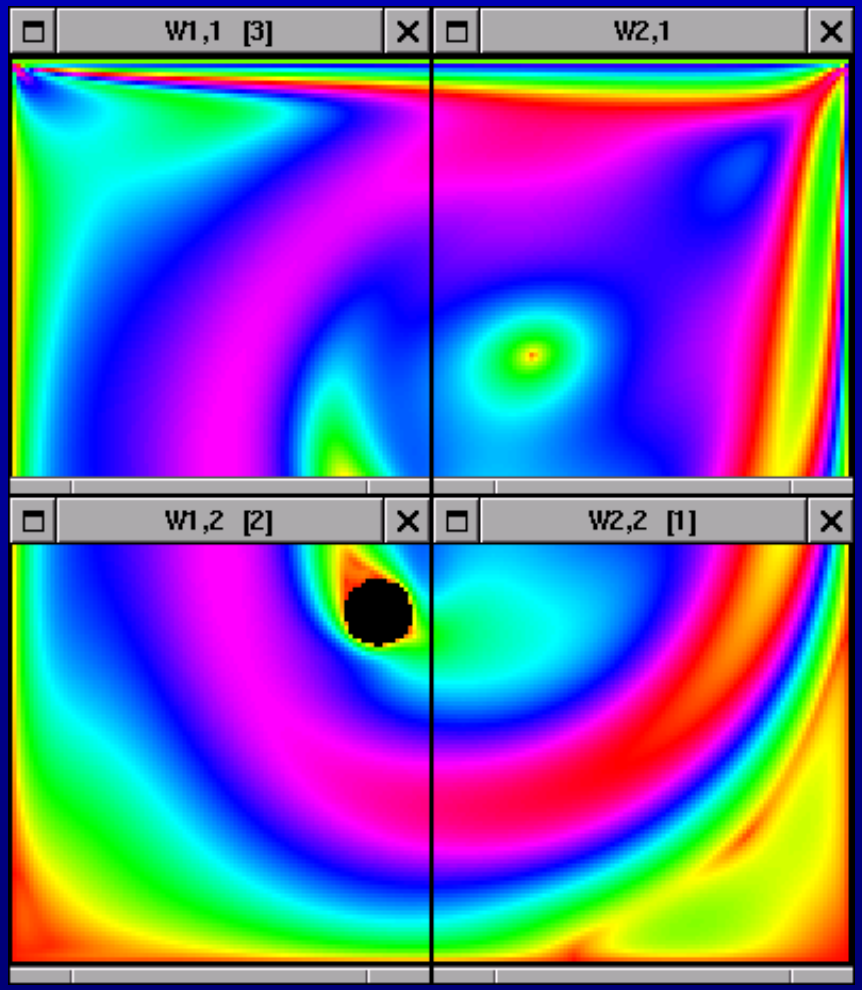
Cache Optimizations:

- Presented techniques do work for LB
- Performance gain strongly depends on the architecture
- Work in progress: extension to 3D

Metal Foam Simulation:

- Lattice Boltzmann with free surface extension is a reasonable choice for this task
- Work in progress: parallelization, more accurate calculation of surface tension, correct handling of bubble segmentation, ...
- Extensions to the model: heat transfer, solidification

Grid Computing with LB



Cooperation with Hans De Sterck and Robert Markel, Department of Applied Mathematics, University of Colorado:

- Implemented in Java
- Simple and small worker tasks
- Upload of the byte code for the simulation to the worker tasks
- Communication is done via bags