Fifth European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
June 14th - 17th, 2010
Lisbon, Portugal

Performance Modeling and Optimization for 3D Lattice Boltzmann Simulations on Highly Parallel On-Chip Architectures: GPUs Vs. Multi-Core CPUs

MS11 GPU Computing in CFD

This work was supported by
BMBF, grant No 01IH08003A
(project SKALB)

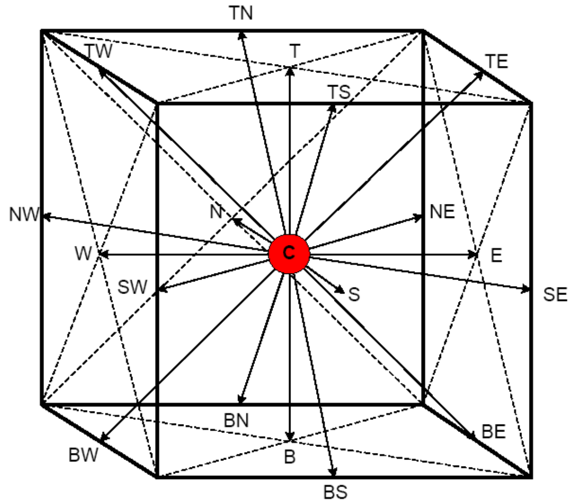
J. Habich[a], C. Feichtinger[b,c], Dr. T. Zeiser[a], Prof. Dr. G. Wellein[a,b], Dr. G. Hager[a]

[a]HPC Services – Regional ComputingCenter Erlangen
[b]Department of Computer Science
[C]Chair of System Simulation

Bundesministerium
für Bildung
und Forschung

The lattice Boltzmann method

- Explicit, fully discrete Boltzmann equation with BGK collision operator
- Physical discretization: D3Q19
- Push or Pull optimized layout
- Fullway/halfway bounce-back for obstacle treatment/boundary condition



PUSH

PULL

float / double f(0:xMax+1,0:yMax+1,0:zMax+1,0:18,0:1)

PUSH:

```
if( fluidcell(x,y,z) ) then
    LOAD f(x,y,z, 0:18,t)
    Relaxation (complex computations)
    SAVE f(x  ,y  ,z  , 0,t+1)
    SAVE f(x+1,y+1,z  , 1,t+1)
    SAVE f(x  ,y+1,z  , 2,t+1)
    SAVE f(x-1,y+1,z  , 3,t+1)
    …
    SAVE f(x  ,y-1,z-1,18,t+1)
endif
```

PULL:

```
if( fluidcell(x,y,z) ) then
    LOAD f(x  ,y  ,z  , 0,t)
    LOAD f(x+1,y+1,z  , 1,t)
    LOAD f(x  ,y+1,z  , 2,t)
    LOAD f(x-1,y+1,z  , 3,t)
    …
    LOAD f(x  ,y-1,z-1,18,t)
    Relaxation (complex computations)
    SAVE f(x,y,z, 0:18,t+1)
endif
```

Motivation

- **Why LBM → Easy to parallelize**

- **Why GPUs and CPUs:**

  → GPUs currently offer the highest peak performance
  → CPUs are available anyway on any GPU node

- **Why parallel:**
  → Parallelism will be the main contributor to future performance gain, and not single processor enhancements

Architectures

NVIDIA GT200

- **30 Multiprocessors (MP); each with:**
  - 8 processors SP driven by :
    Single Instruction Multiple Data (SIMD)
    Single Instruction Multiple Thread (SIMT)
  - Explicit in-order architecture
  - 16384 Registers
  - 16 KB of local on-chip memory
    (shared memory)
  - clock rate of 1.4 GHz

  1000 GFLOP/s (single precision
  84 GFLOP/s (double precision)

- **Up to 1.5 GB of global memory (DRAM)**
  - 1160 MHz DDR
  - 512 bit bus
  - Global gather/scatter possible → watch the latency
  - 148.6 GB/s bandwidth
  - 16 GB/s PCIe 2.0 x16 interface
    (bidirectional)

INTEL Xeon (node)

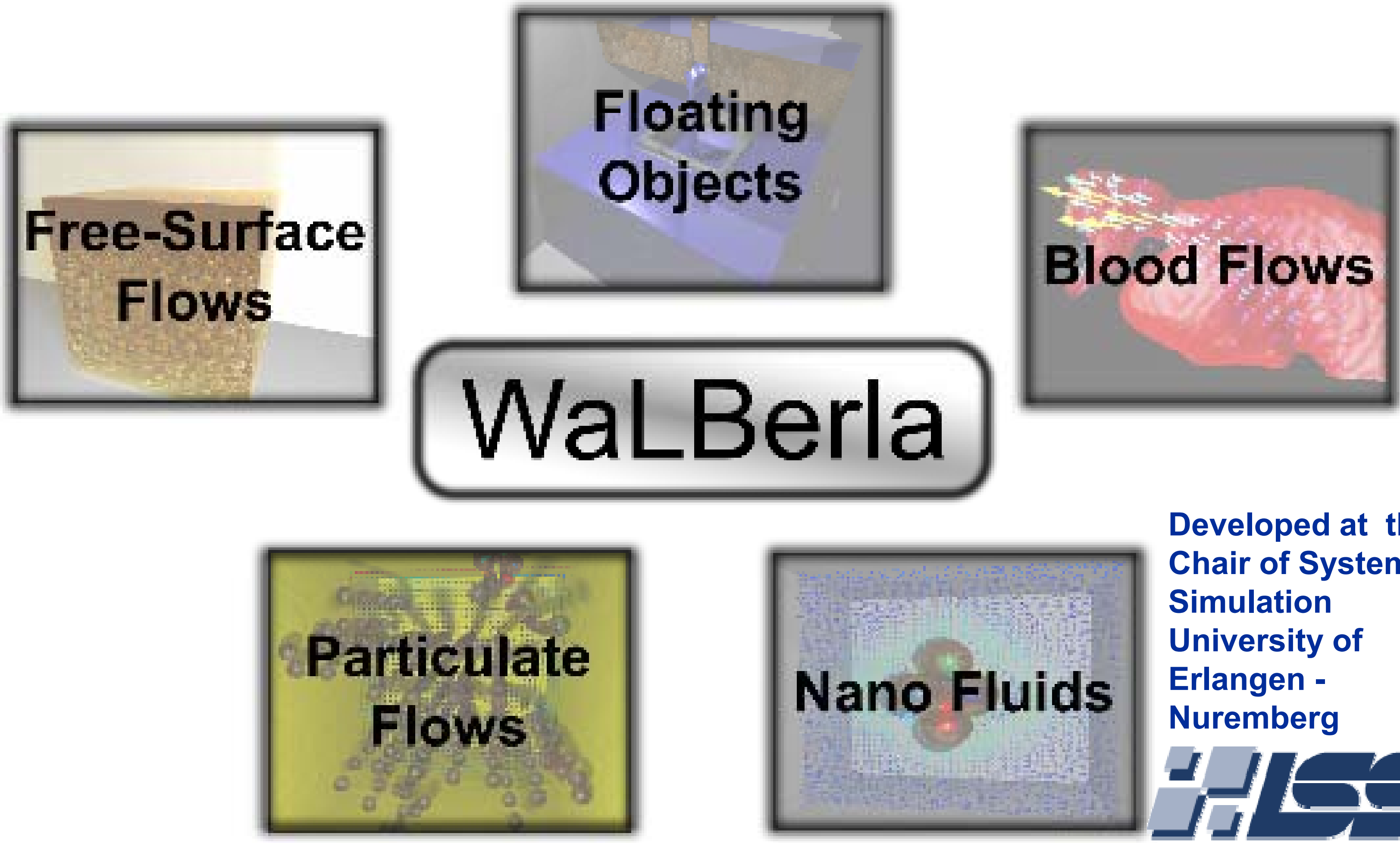
- 4 or 6 (8) cores per socket
- Up to 8 or 12 (16) SMT threads per socket
- 8 MB L3 cache
- Clock rates up to 3.33 GHz

200 GFLOP/s (single precision)
100 GFLOP/s (double precision)

Memory:

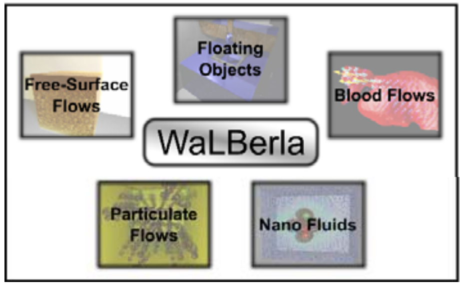
- 3x 1333 MHz DDR
- 64 bit bus
- 61 GB/s peak bandwidth

Widely applicable LB from Erlangen (WaLBerla)

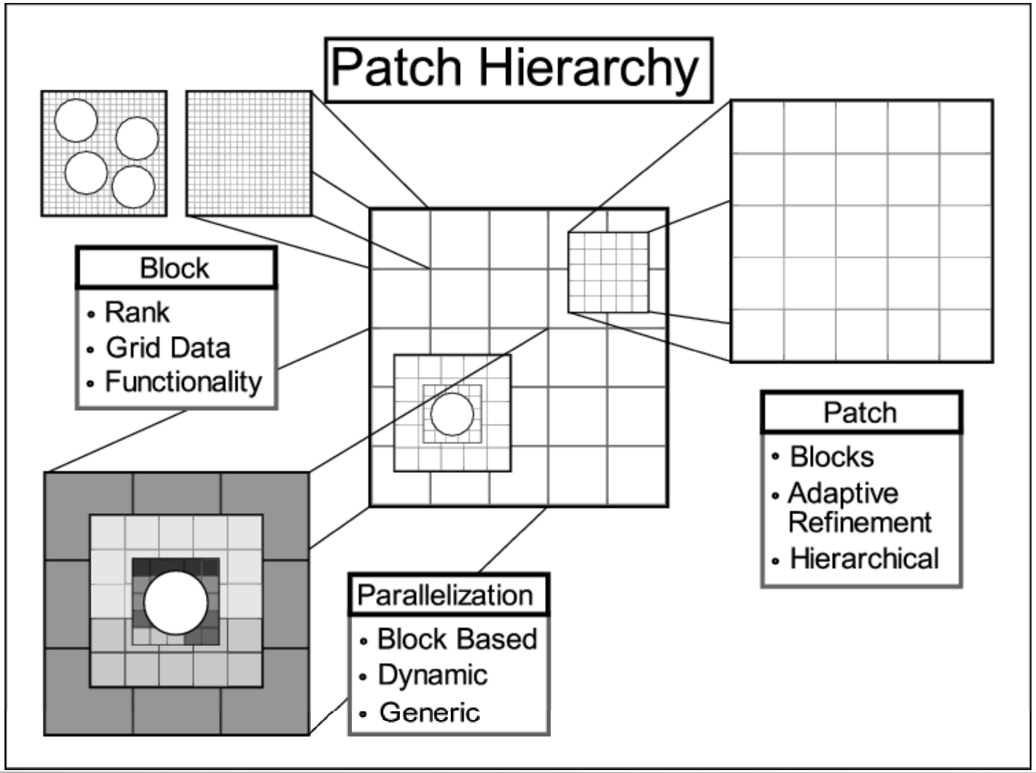


Developed at the Chair of System Simulation University of Erlangen - Nuremberg

Widely applicable LB from Erlangen (WaLBerla)



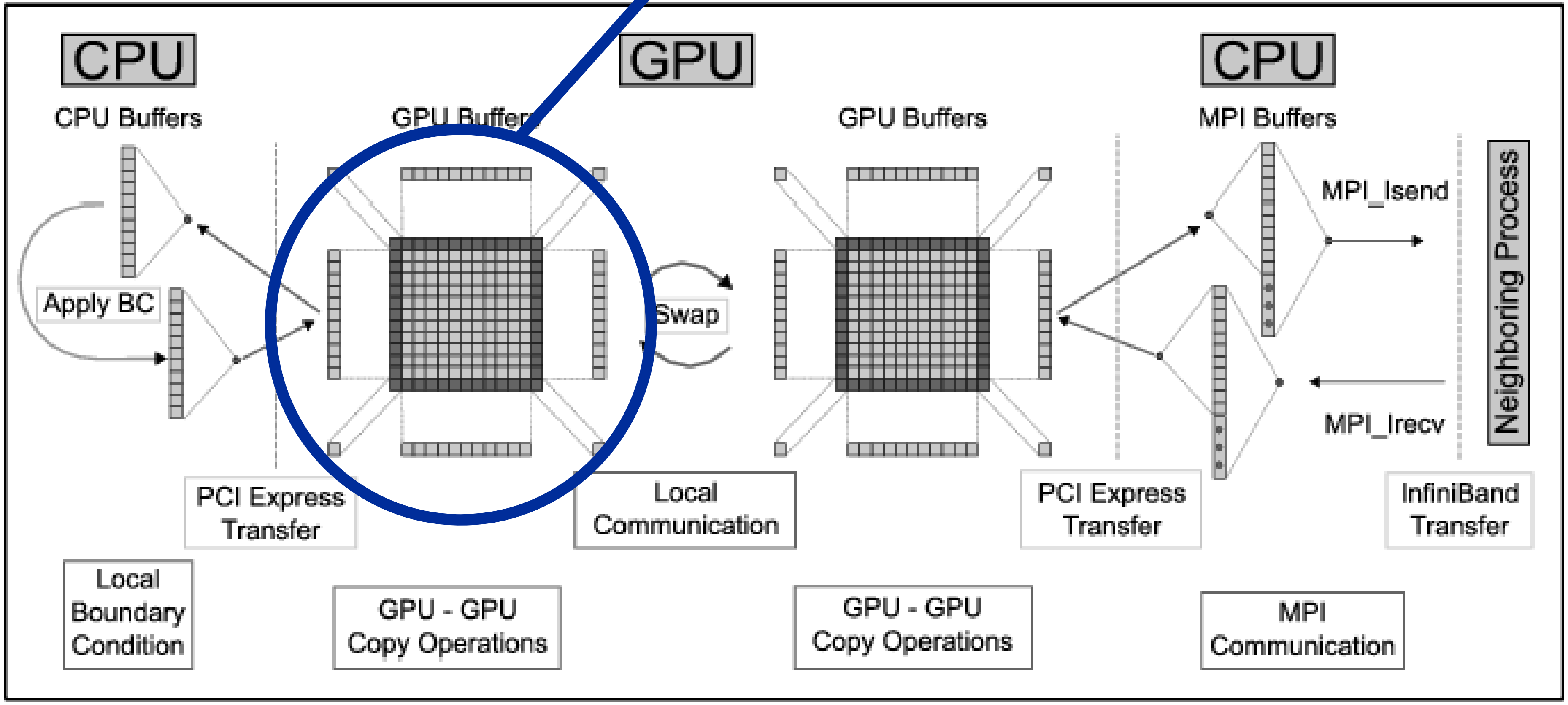
- Patch and Block based domain decomposition
- Block contains Simulation data and Meta data e.g. for parallelization, advanced models
- Block can be algorithm or architecture specific
- All Blocks are equal in spatial dimensions
- MPI processes can have one or multiple blocks



Heterogeneous LBM

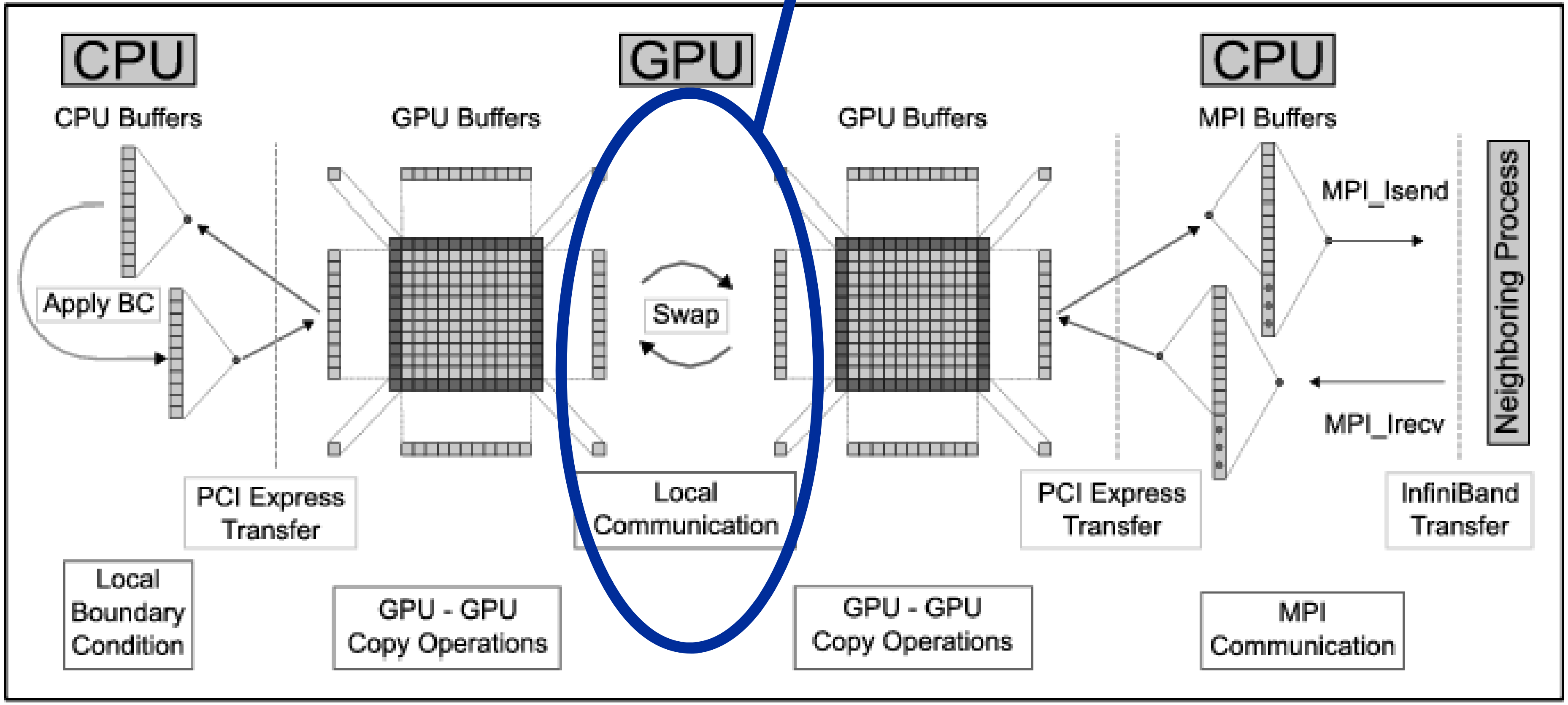
Copy to Buffers on CPU and GPU

After each iteration, boundary data is copied to Communication Buffers



Buffer swap on GPU

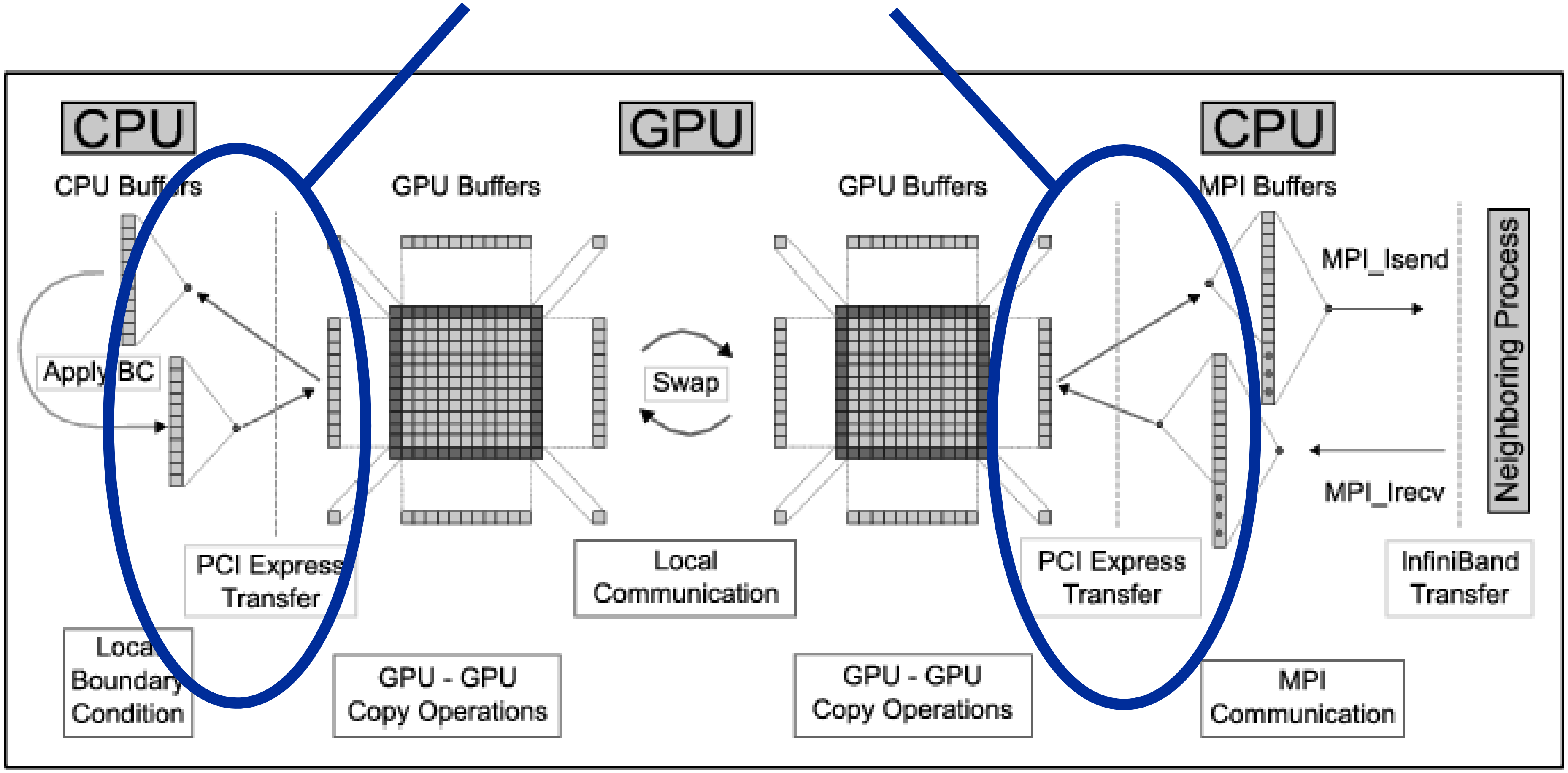
Local Communication Buffers are only swapped.
No Copy is done!



Johannes.Habich@rrze.uni-erlangen.de

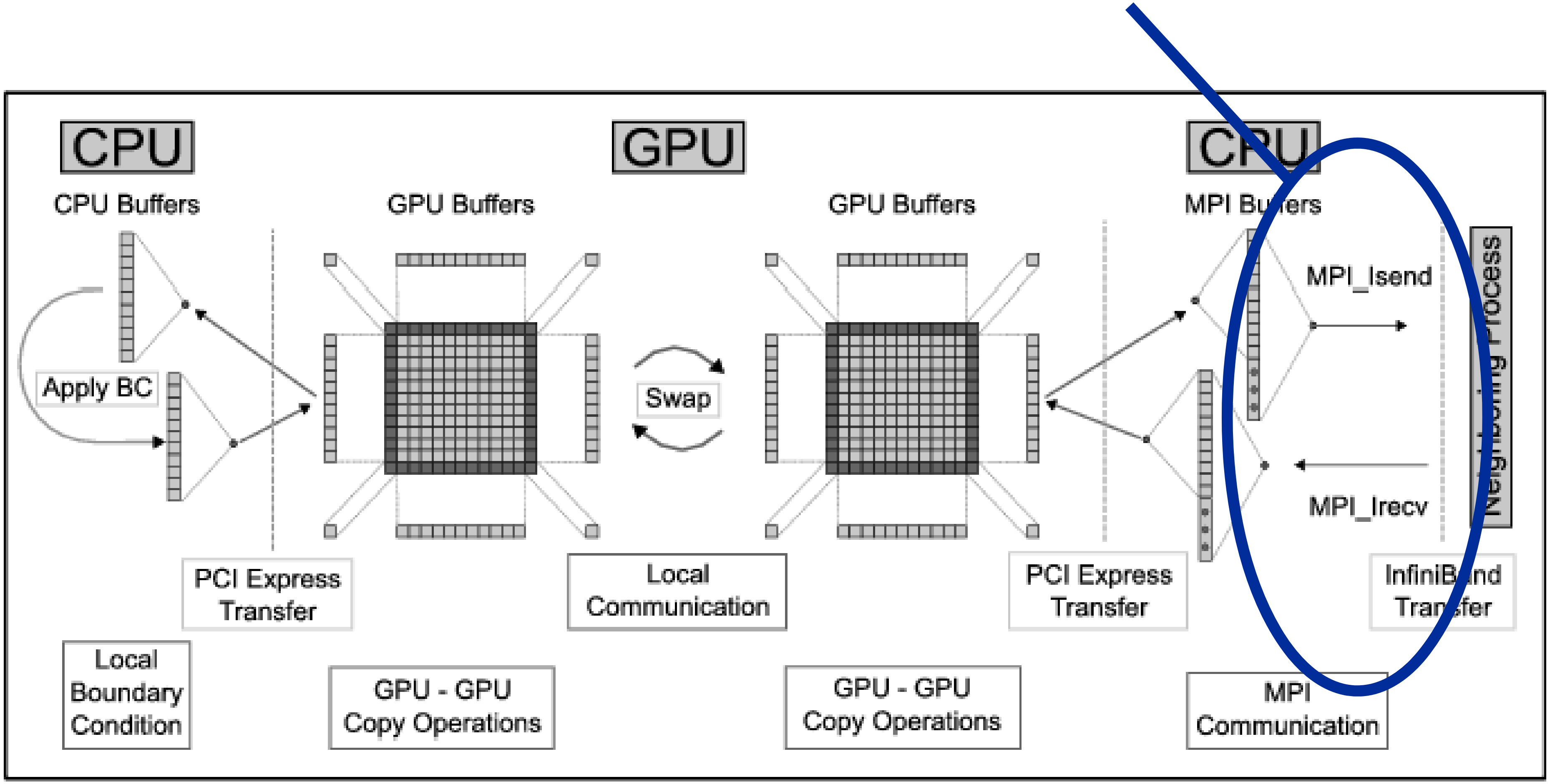
Transfer of buffers to the host

Data of GPU processes is transferred to the Host



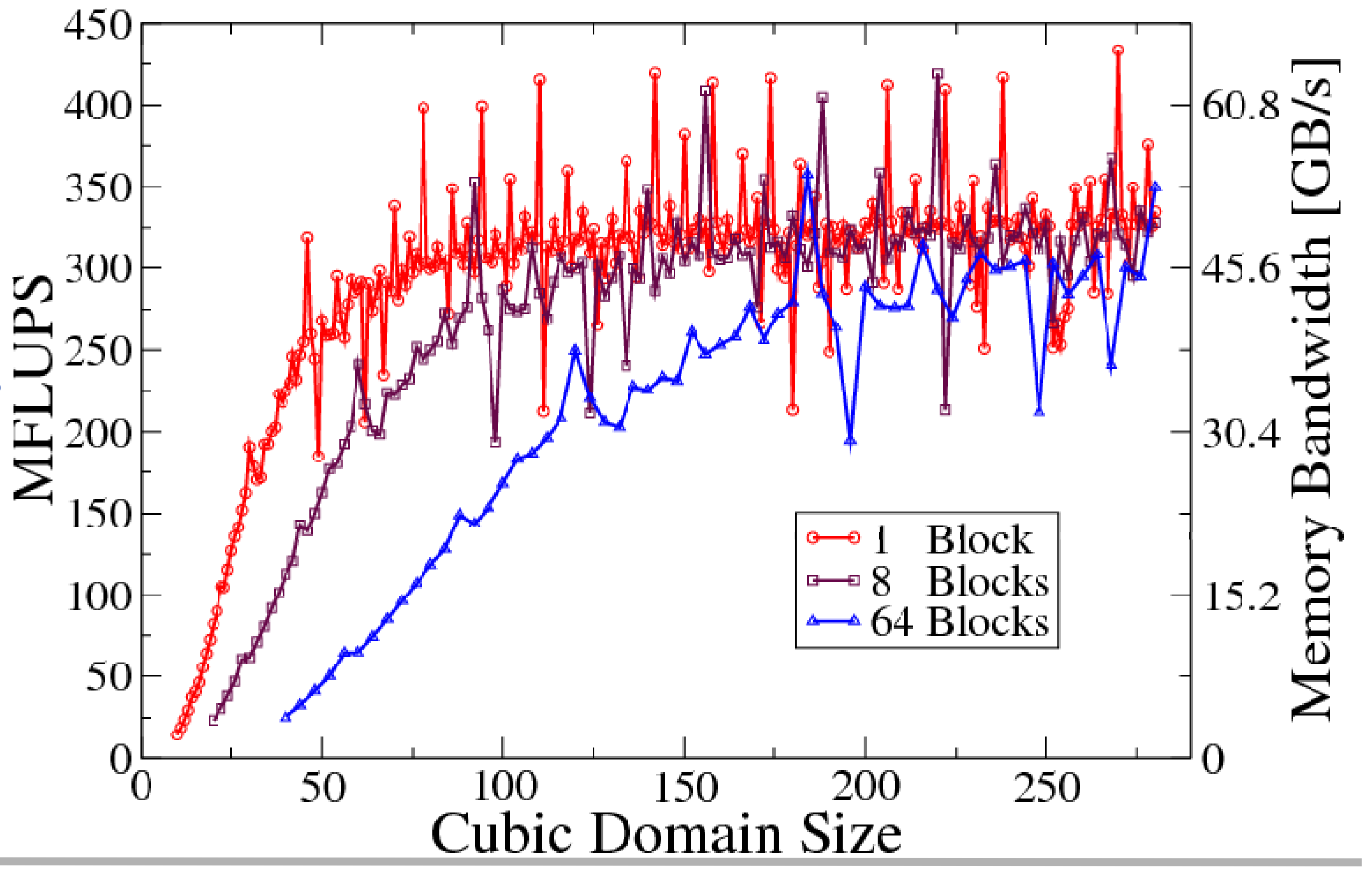
Transfer of buffers to the host

Buffers are transferred/received to/from other hosts



Pure kernel (SP), no PCIe/IB transfer

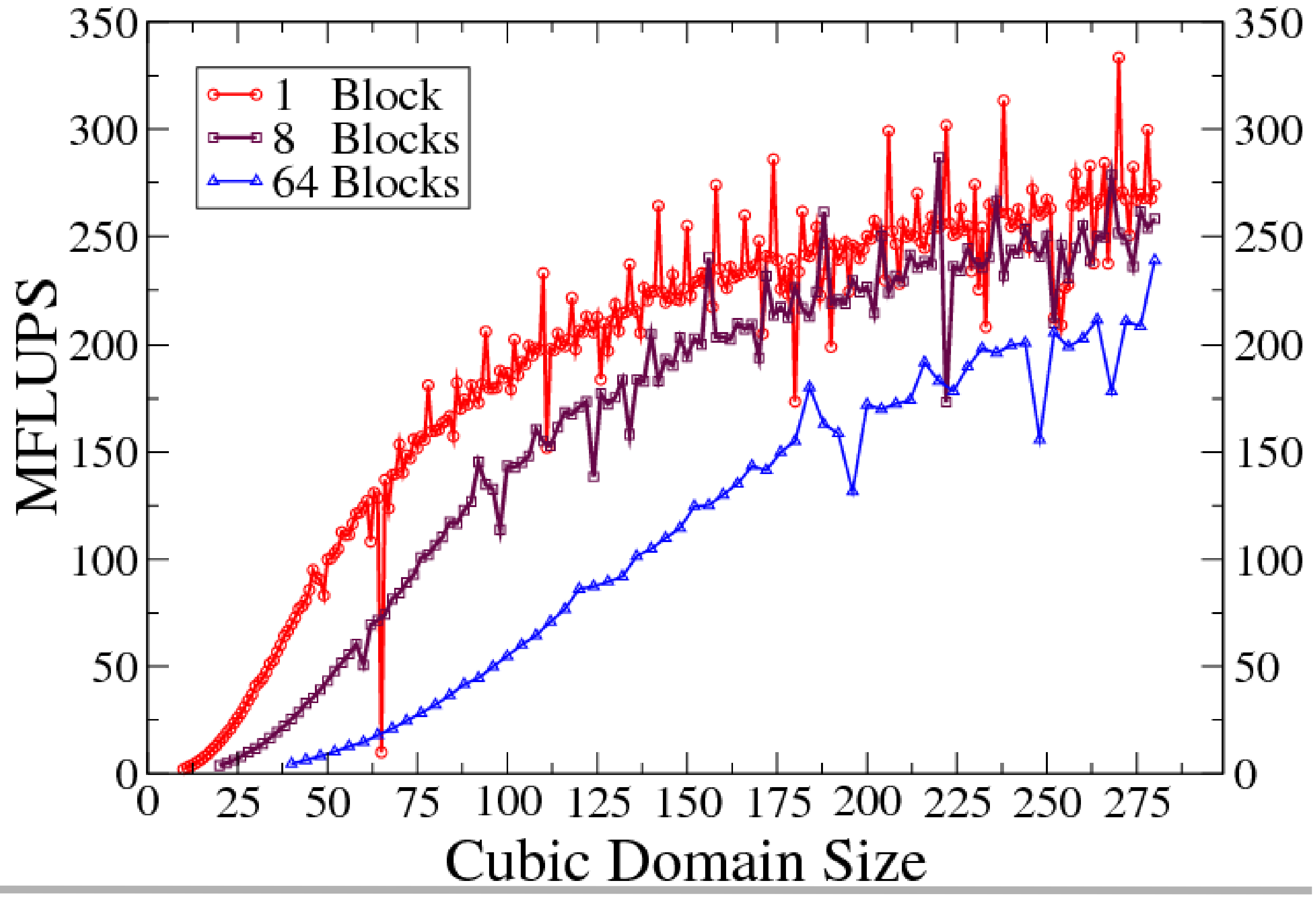
- **Maximum performance starting at 50x50x50**
- **Fluctuations due to different thread numbers and influence of alignment**
- **Blocks influence kernel**
  Domains < 200x200x200
- **Comparison:**
  Xeon Node ~100 MLUPS
- **LUPS:**
  Lattice updates per second



Kernel with boundary transfer (SP), no IB

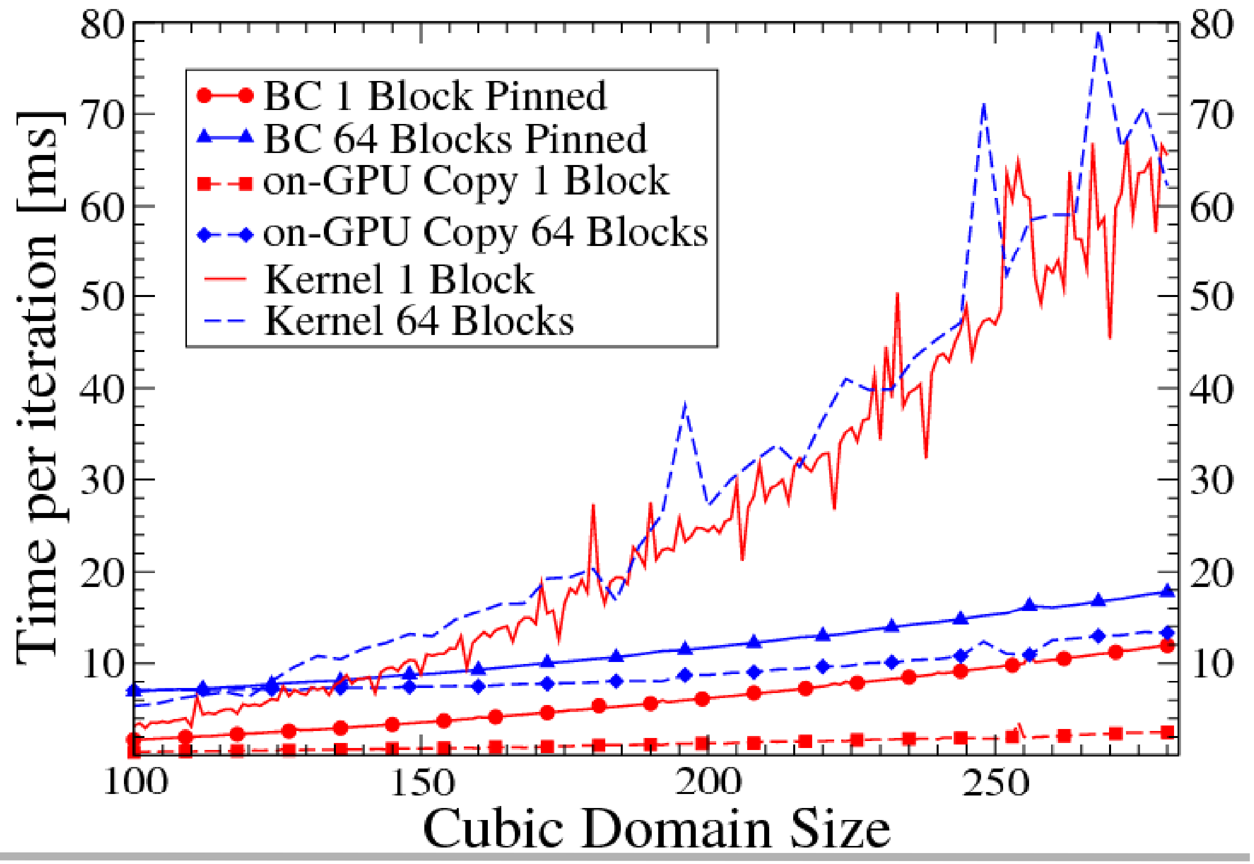
- **Maximum performance starting at 200x200x200**
  (64 times more than pure kernel (50x50x50)!!)
- **Blocks influence kernel with any domainsize**
- **28% is lost for 64 blocks**

Why?



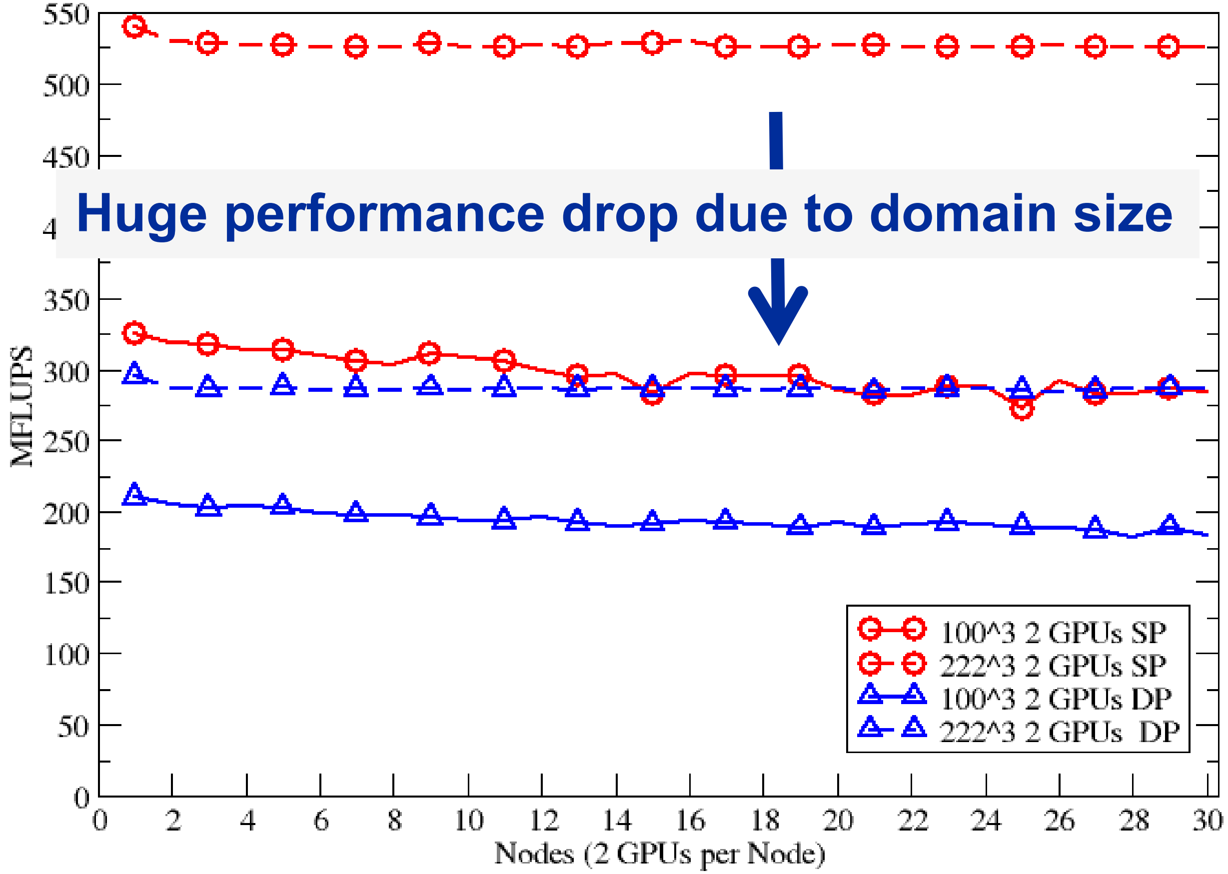
Time measurements of kernel with 1 and 64 blocks

- **Domains > 250^3 → about 50% of execution time is spent in non-kernel parts**
- **Kernel execution time is constant no matter how much blocks are used**
- **Domains < 150^3 non-kernel part becomes dominant**



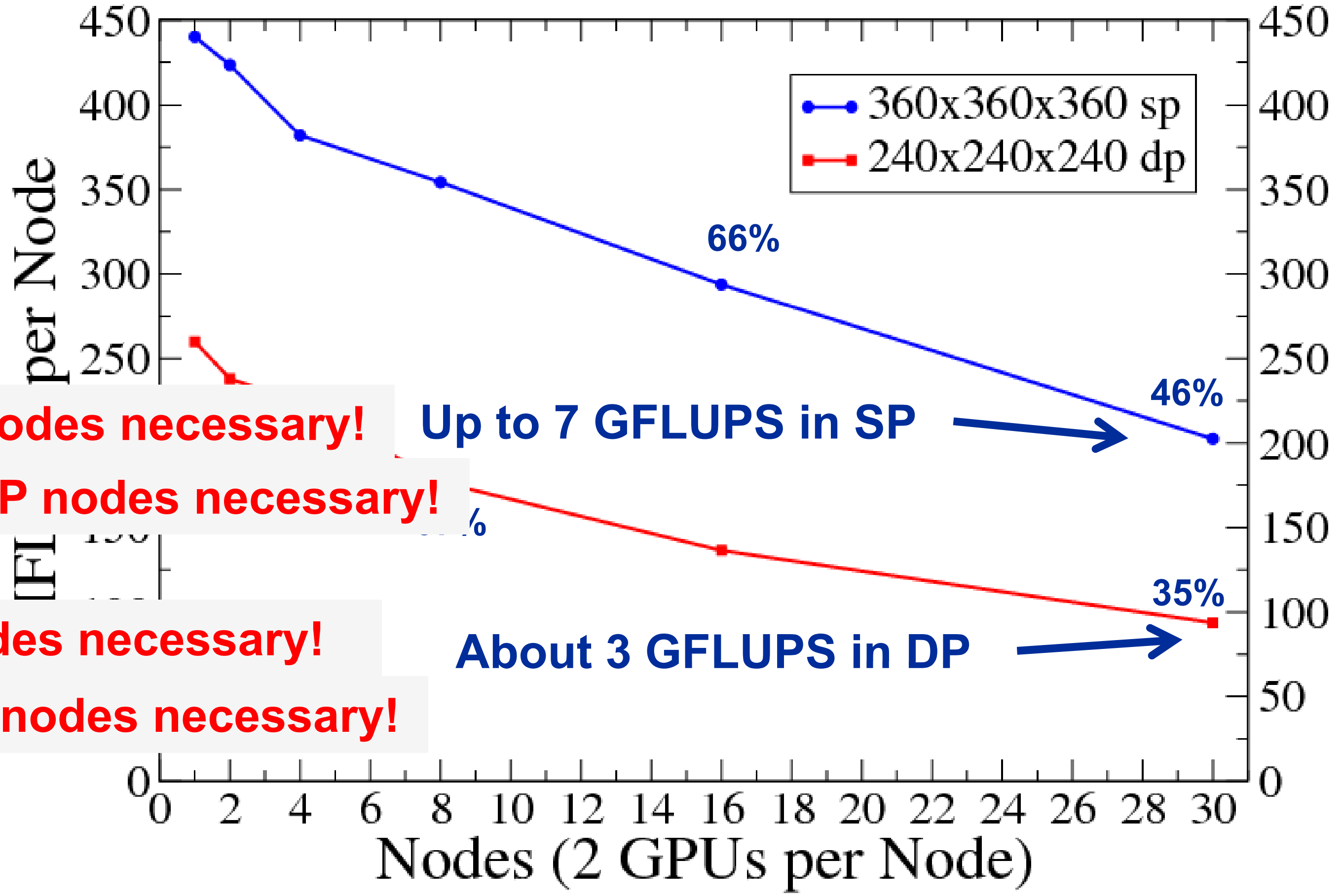
Weak scaling GPU per Node performance

- Weak scaling works as expected
- Initial performance drop from one to two cards per node
- Up to 16 GFLUPS max. performance



Strong Scaling GPU per Node performance

- Loss of 64% in SP on 30 Nodes (60 GPUs)
- Loss of 75% in DP on 30 Nodes (60 GPUs)



Up to 137 Intel Xeon nodes necessary!

Up to 1275 BlueGene/P nodes necessary!

Up to 70 Intel Xeon nodes necessary!

Up to 750 BlueGene/P nodes necessary!

Outlook

- **Implement grid refinement**
- **Implement dynamic load balancing for heterogeneous computations**

  static load Balancing already done:

  90 nodes: 60 GPUs and 660 CPUs: 17.8 GFLUPS

This work was supported by
BMBF, grant No 01IH08003A
(project SKALB)

Bundesministerium
für Bildung
und Forschung

Thank you very much for your attention

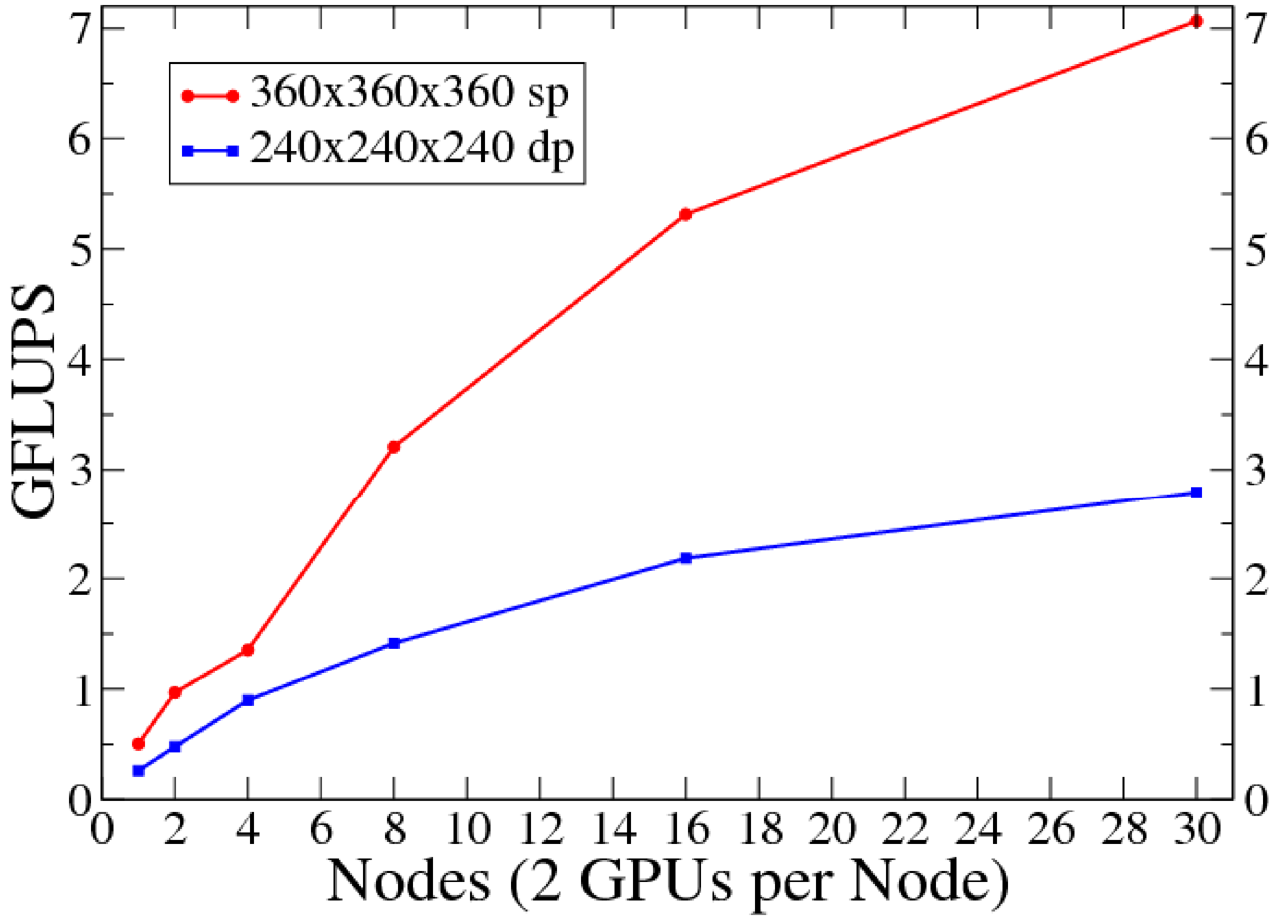
Johannes Habich

Regional Computing Center,
University Erlangen-Nuremberg
HPC Services
Martenstrasse 1
D-91058 Erlangen

Johannes.Habich@rrze.uni-erlangen.de

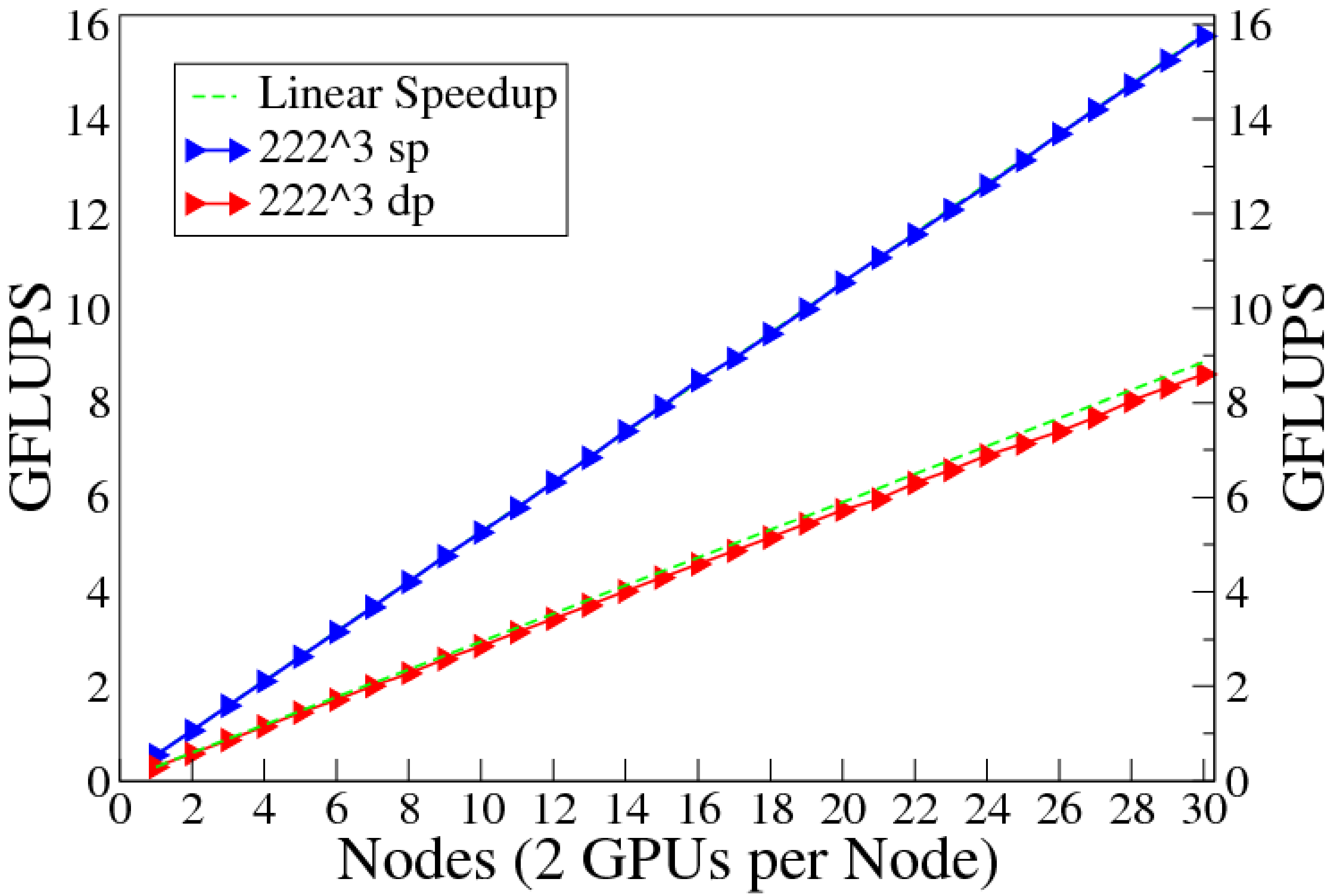
Strong Scaling GPU

- **Up to 7 GFLUPS in SP and nearly 3 GFLUPS in DP on 60 GPUs**
- **Communication bound starting at 16 Nodes**

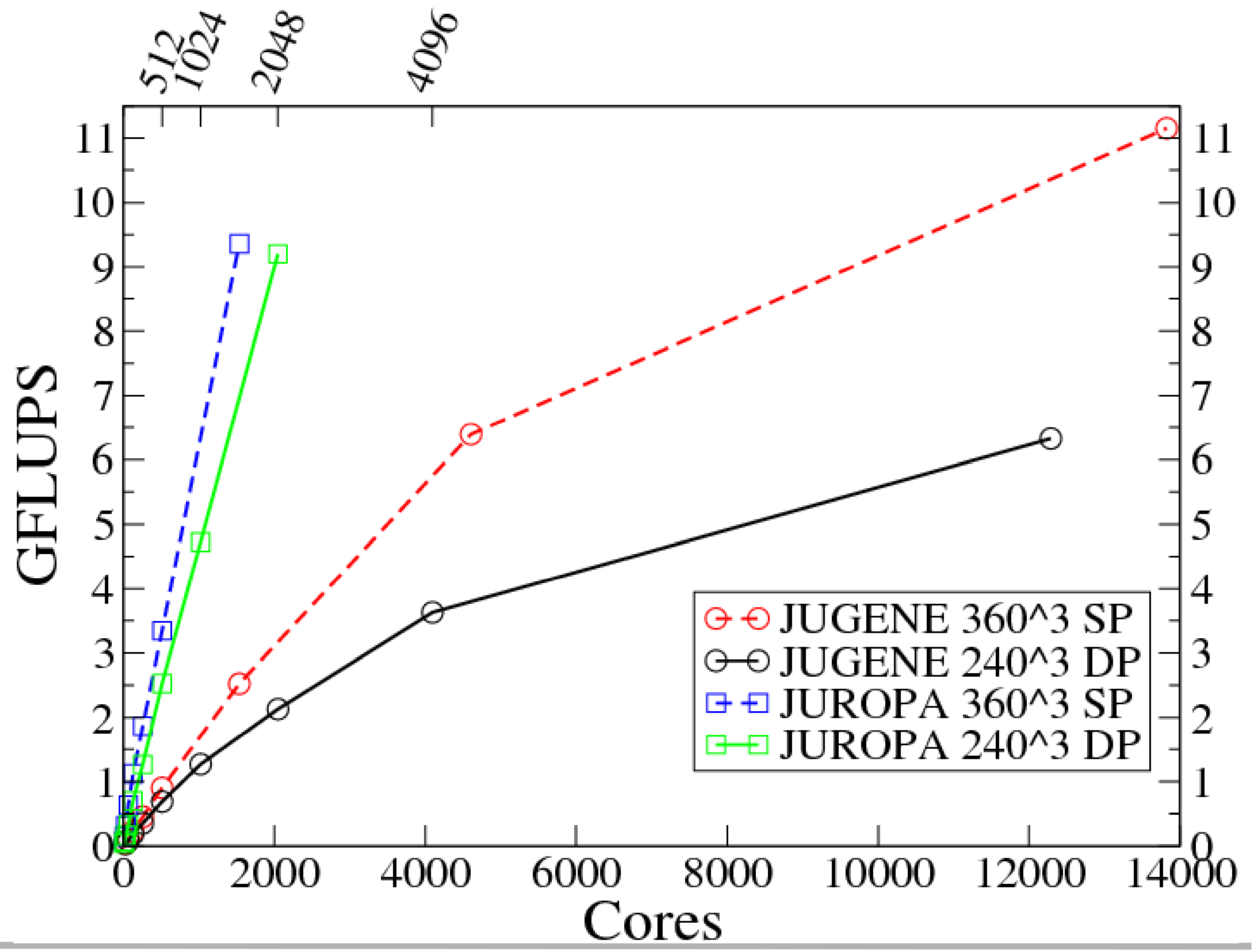


Johannes.Habich@rrze.uni-erlangen.de

Weak Scaling on GPUs



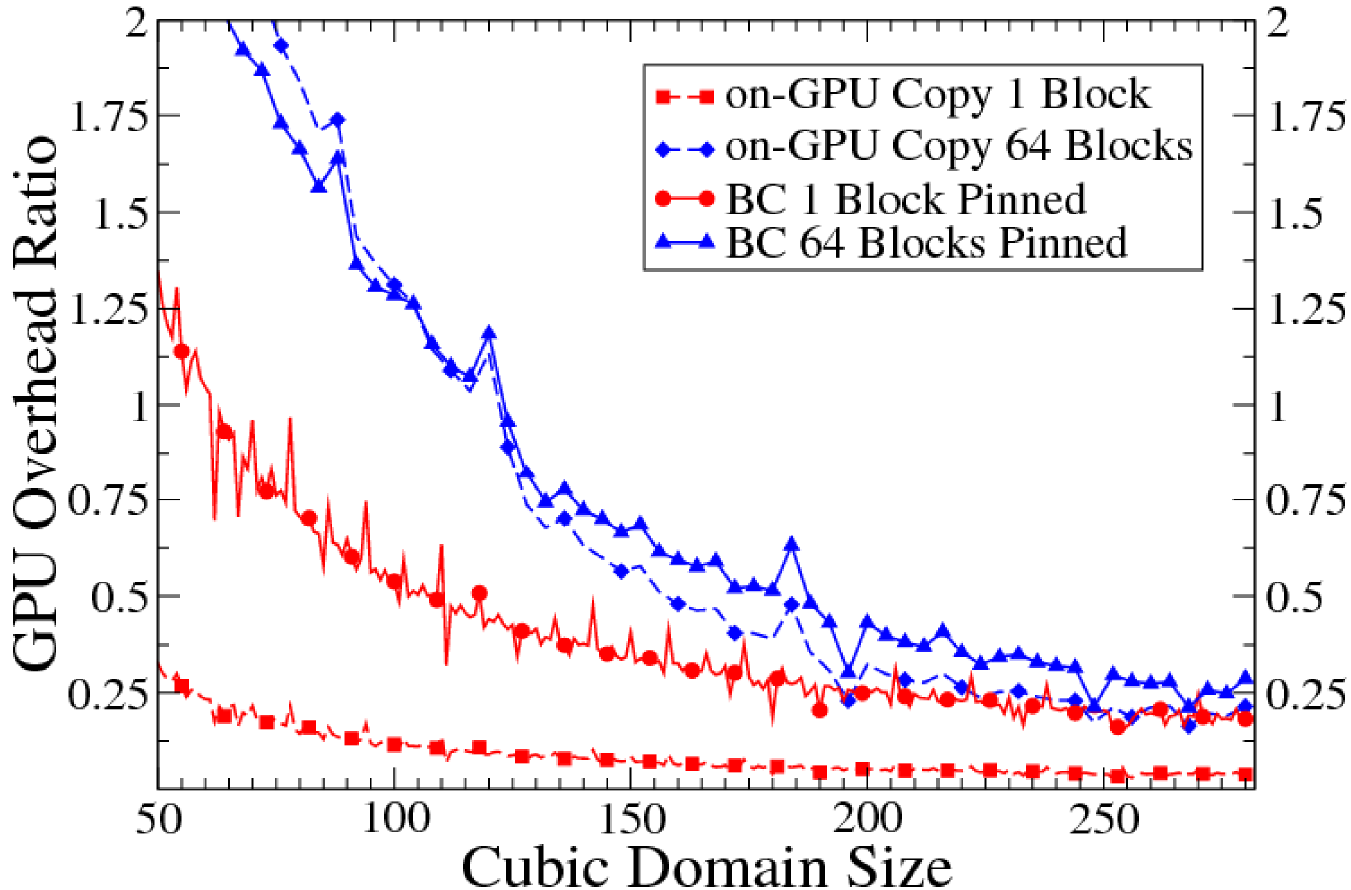
Weak Scaling on CPUs



Johannes.Habich@rrze.uni-erlangen.de

Dominant part in small domain scenarios

- The fraction of BC treatment and Communication and kernel time is shown
- Domains > 250x250x250 → about 25% for 64 Blocks



From „Boltzmann” to„Lattice-Boltzmann“ and „Navier-Stokes“

Boltzmann Equation

$$\frac{\partial f}{\partial t} + \xi \frac{\partial f}{\partial x} + K \frac{\partial f}{\partial \xi} = Q(f, f)$$

small Knudsen-Number

Chapman-Enskog-Expansion

Continuity Equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0$$

Conservation of momentum (j=1,2,3)

$$\rho \left(\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} \right) = -\frac{\partial p}{\partial x_j} - \frac{\tau_{ij}}{\partial x_i} + \rho g_j$$

Chapman-Enskog-Expansion

Negelect volumetric forces

Bhatnagar-Gross-Krook-Approximation

$$\frac{\partial f}{\partial t} + \xi \frac{\partial f}{\partial x} = \frac{1}{\tau}(f - f^0)$$

Small Knudsen-Number
Small Mach-Number

Discrete Velocity Space

Velocity-discrete BGK-Eq.

$$\frac{\partial f_i}{\partial t} + c_i \frac{\partial f_i}{\partial x} = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right)$$

Dicretizing Space and Time

Lattice-Boltzman-Equation (LBGK) (i=0,…,N-1)

$$f_i(t_* + \Delta t, x_* + c_i \Delta t) = f_i(t_*, x_*) - \frac{\Delta t}{\tau}\left(f_i(t_*, x_*) - f_i^{eq}(\rho(x_*, t_*), u(x_*, t_*))\right)$$

Darstellung: nach einer Idee von M. Krafczyk