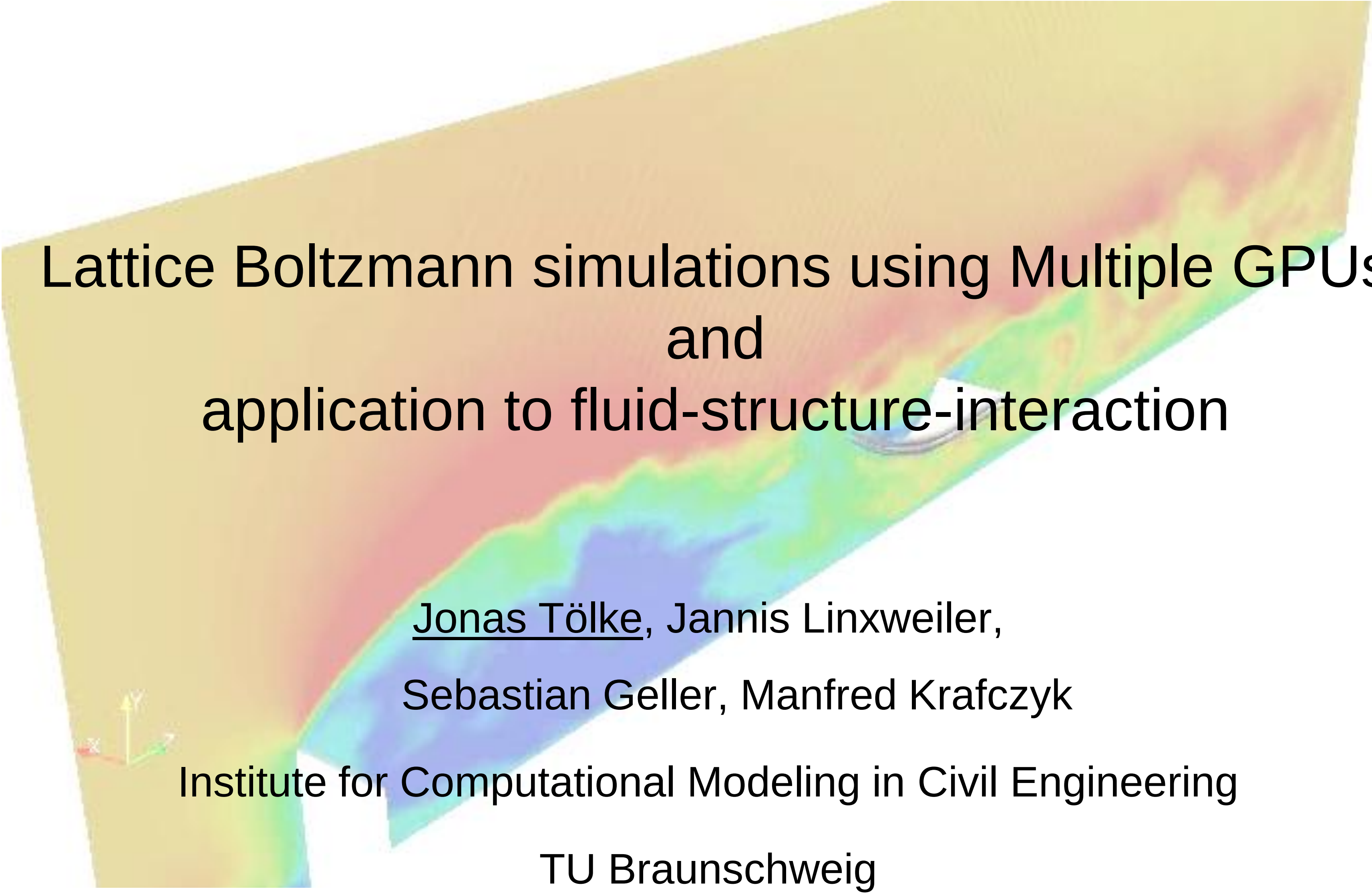


Lattice Boltzmann simulations using Multiple GPUs and application to fluid-structure-interaction

Jonas Tölke, Jannis Linxweiler,

Sebastian Geller, Manfred Krafczyk

Institute for Computational Modeling in Civil Engineering

TU Braunschweig

Outline

- D3Q13 Model
- GPU Programming
  - nVIDIA G80/G92/GT200 chip – the parallel stream processor
  - nVIDIA CUDA
  - Multiple GPUs
- Implementation of the D3Q13 model
- Results: Moving Sphere in a pipe
- Outlook

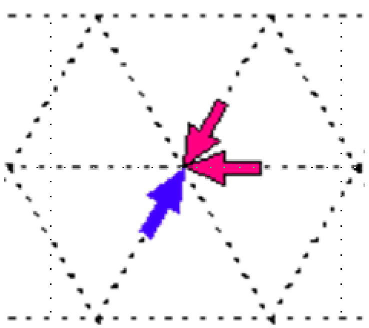
Lattice-Boltzmann Automata

Frisch, Hasslacher, Pomeau 86,
Wolfram 86,
Frisch, d'Humières, Hasslacher, Lallemand, Pomeau, Rivet 87

Lattice Boltzmann Equation (LBE)

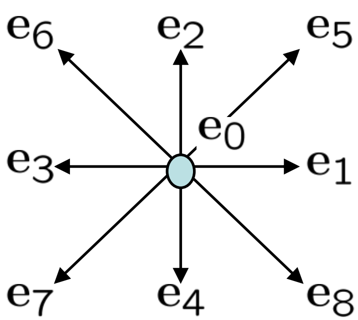
$$f_i(t+\Delta t, \mathbf{x}+\mathbf{e}_i\Delta t) = f_i(t,\mathbf{x})+\Omega_i, \quad i = 0,\ldots,b-1$$

f	Mass fractions
e	Microscopic velocity of the particles
t	Time
x	Space



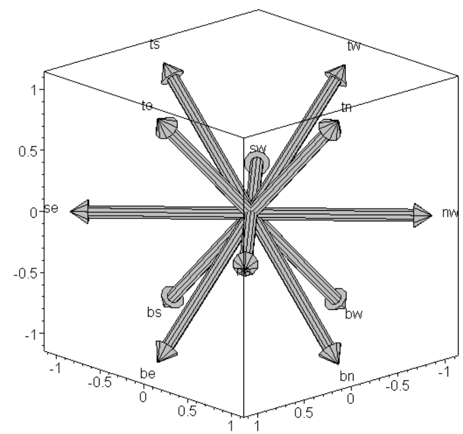
d2q6-Model

FHP 86, Wolfram 86



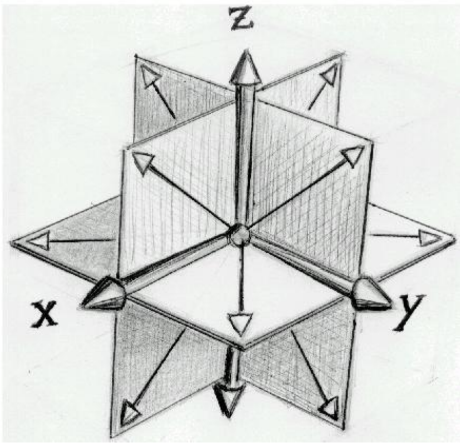
d2q9-Model

Qian, d'Humières, Lallemand 92



d3q13-Model

d'Humières, Bouzidi, Lallemand 01



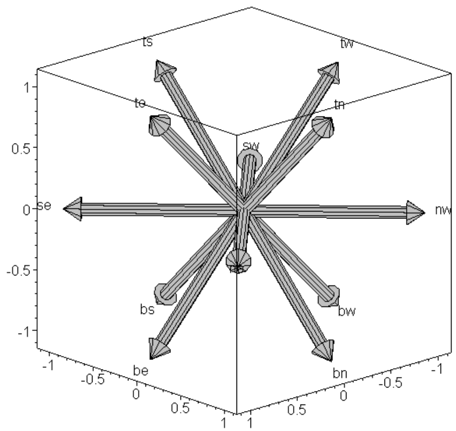
d3q19-Model

Qian, d'Humières, Lallemand 92

D3Q13-Model

d'Humières et al. 2001

Lattice:



$$\{\mathbf{e}_i, i = 0, \ldots, 12\} =$$

$$\{\mathbf{e}_r, \mathbf{e}_{ne}, \mathbf{e}_{sw}, \mathbf{e}_{se}, \mathbf{e}_{nw}, \mathbf{e}_{te}, \mathbf{e}_{bw}, \mathbf{e}_{be}, \mathbf{e}_{tw}, \mathbf{e}_{tn}, \mathbf{e}_{bs}, \mathbf{e}_{bn}, \mathbf{e}_{ts}, \}$$

$$= \left\{ \begin{array}{ccccccccccccc} 0 & c & -c & c & -c & c & -c & c & -c & 0 & 0 & 0 & 0 \\ 0 & c & -c & -c & c & 0 & 0 & 0 & 0 & c & -c & c & -c \\ 0 & 0 & 0 & 0 & 0 & c & -c & -c & c & c & -c & -c & c \end{array} \right\}$$

c: microscopic speed

Moments:

$$\boldsymbol{m} = \mathrm{M}\boldsymbol{f} := (\rho, \rho_0 u_x, \rho_0 u_y, \rho_0 u_z, e, p_{xx}, p_{ww}, p_{xy}, p_{yz}, p_{xz}, h_x, h_y, h_z)$$

Collision operator:

$$f_i(t+\Delta t, \mathbf{x}+\mathbf{e}_i \Delta t) = f_i(t, \mathbf{x}) + \Omega_i, \quad i = 0, \ldots, 12$$

$$\boldsymbol{\Omega} = \mathrm{M}^{-1} \boldsymbol{k}$$

Eigenvectors

$$Q_{0,i} = 1 = (1,1,1,1,1,1,1,1,1,1,1,1,1)$$

$$Q_{1,i} = e_{x,i} = c \cdot (0,1,-1,1,-1,1,-1,1,-1,0,0,0,0)$$

$$Q_{2,i} = e_{y,i} = c \cdot (0,1,-1,-1,1,0,0,0,0,1,-1,1,-1)$$

$$Q_{3,i} = e_{z,i} = c \cdot (0,0,0,0,0,1,-1,-1,1,1,-1,-1,1)$$

$$Q_{4,i} = \frac{13}{2}\,\mathbf{e}^2 - 12\,c^2 = c^2 \cdot (-12,1,1,1,1,1,1,1,1,1,1,1,1)$$

$$Q_{5,i} = 3\,e_{x,i}^2 - \mathbf{e}^2 = c^2 \cdot (0,1,1,1,1,1,1,1,1,-2,-2,-2,-2])$$

$$Q_{6,i} = e_{y,i}^2 - \ e_{z,i}^2 = c^2 \cdot (0,1,1,1,1,-1,-1,-1,-1,0,0,0,0)$$

$$Q_{7,i} = e_{x,i}\,e_{y,i} = c^2 \cdot (0,1,1,-1,-1,0,0,0,0,0,0,0,0)$$

$$Q_{8,i} = e_{y,i}\,e_{z,i} = c^2 \cdot (0,0,0,0,0,0,0,0,0,1,1,-1,-1)$$

$$Q_{9,i} = e_{x,i}\,e_{z,i} = c^2 \cdot (0,0,0,0,0,1,1,-1,-1,0,0,0,0)$$

$$Q_{10,i} = e_{x,i}\,(e_{y,i}^2 - e_{z,i}^2) = c^3 \cdot (0,1,-1,1,-1,-1,1,-1,1,0,0,0,0)$$

$$Q_{11,i} = e_{y,i}\,(e_{z,i}^2 - e_{x,i}^2) = c^3 \cdot (0,-1,1,1,-1,0,0,0,0,1,-1,1,-1)$$

$$Q_{12,i} = e_{z,i}\,(e_{x,i}^2 - e_{y,i}^2) = c^3 \cdot (0,0,0,0,0,1,-1,-1,1,-1,1,1,-1)$$

Collision operator

$$k_0 = 0,\; k_1 = 0\; k_2 = 0,\; k_3 = 0$$
$$k_4 = k_e = -s_e \left(e - (\frac{39}{2} c_s^2 - 12\, c^2)\, \rho + \frac{13}{2} \rho_0\, (u_x^2 + u_y^2 + u_z^2) \right)$$
$$k_5 = k_{xx} = -s_\nu \left(p_{xx} - \rho_0\, (2\, u_x^2 - u_y^2 - u_z^2) \right)$$
$$k_6 = k_{ww} = -s_\nu \left(p_{ww} - \rho_0\, (u_y^2 - u_z^2) \right)$$
$$k_7 = k_{xy} = -s'_\nu\, (p_{xy} - \rho_0\, u_x\, u_y)$$
$$k_8 = k_{yz} = -s'_\nu\, (p_{yz} - \rho_0\, u_y\, u_z)$$
$$k_9 = k_{xz} = -s'_\nu\, (p_{xz} - \rho_0\, u_x\, u_z)$$
$$k_{10} = k_{hx} = -s_h\, (h_x - \rho_0\, u_x (u_y^2 - u_z^2))$$
$$k_{11} = k_{hy} = -s_h\, (h_y - \rho_0\, u_y (u_z^2 - u_x^2))$$
$$k_{12} = k_{hz} = -s_h\, (h_z - \rho_0\, u_z (u_x^2 - u_y^2))$$

Relaxation rates:

$$s_\nu = \frac{2}{8 \frac{\nu}{c^2\, \Delta t} + 1} \qquad s'_\nu = \frac{2}{4 \frac{\nu}{c^2\, \Delta t} + 1} \qquad s_e = \frac{2}{6 \frac{\nu_B}{c^2\, \Delta t} + 1} \qquad s_h \in]0, 2[$$

Pressure:

$$p = c_s^2 \rho$$

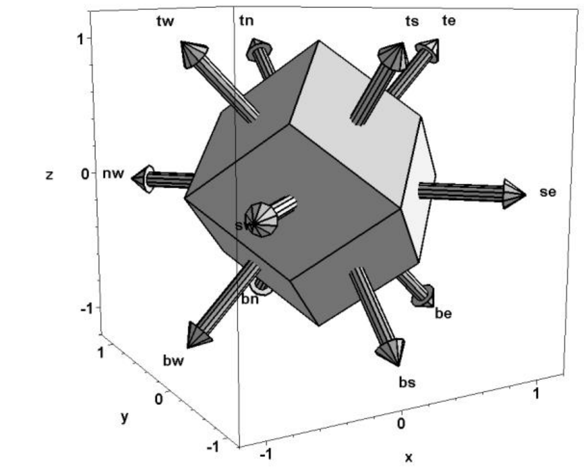
NO LBGK!

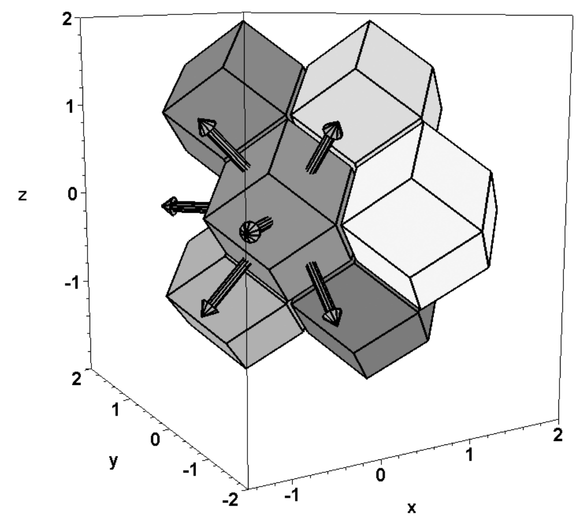
D3Q13 – Model – Unit Cell

Toelke et al. 2008

Two independent sub-lattices: delete one!
→ Basic Unit cell: Rhombic Dodecahedron

- It is a Catalan solid with 12 rhombic faces, 24 edges and 14 vertices
- First described by Johannes Kepler
- Vertex first projection of the 4d hypercube
- The rhombic dodecahedra honeycomb: space-filling tessellation, Voronoi diagram of the face-centered cubic sphere-packing,
- The honeycomb is cell-transitive, face-transitive and edge-transitive. It is ***not*** vertex-transitive
- $V=2h^3$





D3Q13 – Model: Connection graph

Toelke et al. 2008

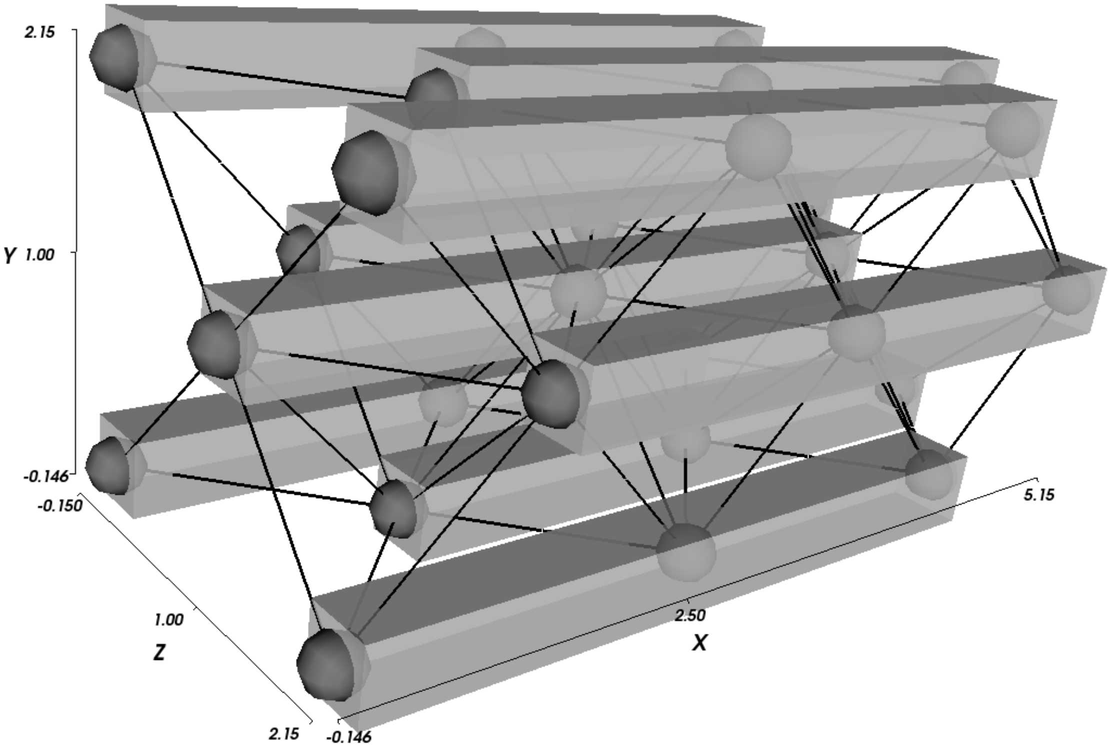
Mapping 1D - Vector
space location – ***memory*** location:

$$a = \begin{cases} 0 & \text{if } j \text{ even and } k \text{ even} \\ 0 & \text{if } j \text{ odd and } k \text{ odd} \\ 1 & \text{if } j \text{ odd and } k \text{ even} \\ 1 & \text{if } j \text{ even and } k \text{ odd} \end{cases}$$

$$x = h\,(a + 2\,i)$$

$$y = h\,j$$

$$z = h\,k$$



C-Code:

```
int m = nx*(ny*k + j) + i;
float x =  h * ( (j&0x1)^(k&0x1) + i*2 );
float y =  h * j;
float z =  h * k;
```

lattice size:

2 *nx* x *ny* x *nz* (128 x 128 x 512)

data structure:

nx x *ny* x *nz* (64 x 128 x 512)

Hardware – Stream Computing

- Vector Machines (Cray, NEC, ...)
- Cell Processor (IBM Blade Server, Sony Play Station 3)
- GPUs

GPU - Programming

Group of Arie Kaufman:

- Implementing Lattice Boltzmann Computation on Graphics Hardware *(Li/Wei/Kaufman 2003)*
- Dispersion simulation and visualization for urban security *(Qiu et al. 2004)* ,
- Simulation of soap bubbles *(Wei et al. 2004)*
- Melting and flowing in multiphase environment *(Zhao et al. 2006)*
- Visual simulation of heat shimmering and mirage *(Zhao et al. 2006)*
- GPU clusters for general-purpose computation *(Fan et al. 2004)*

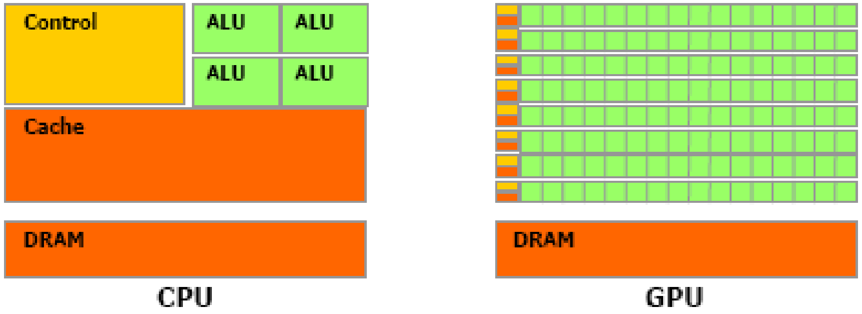
- Real-time ink dispersion in absorbent paper *(Chu/Tai 2005)*
- Simulation of miscible binary mixtures *(Zhu et al. 2006)*
- Implementation of a Navier-Stokes solver on a GPU *(Wu et al. 2004)*
- Hierarchical parallel processing of large scale data clustering on a PC cluster with GPU co-processing *(Takizawa 2006)*

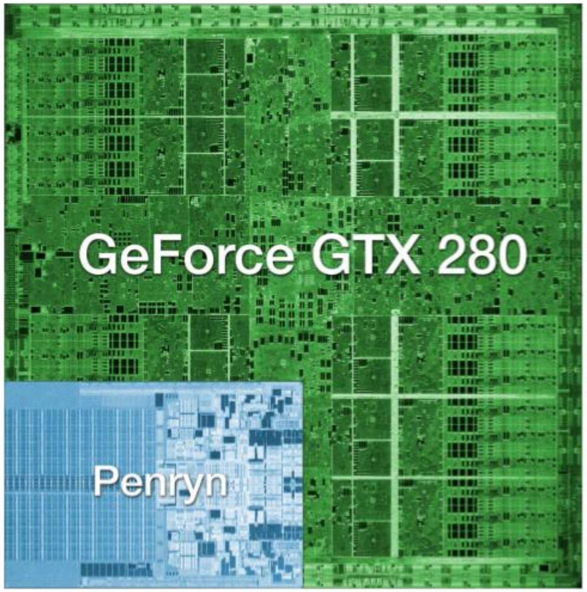
→ Programming style close to the hardware especially developed for graphics applications!

nVIDIA - G80/G92/GT200: the parallel stream processor

GTX 280: 1.4 billion transistors

Montecito: 1.7 (1.5 are L3 cache)





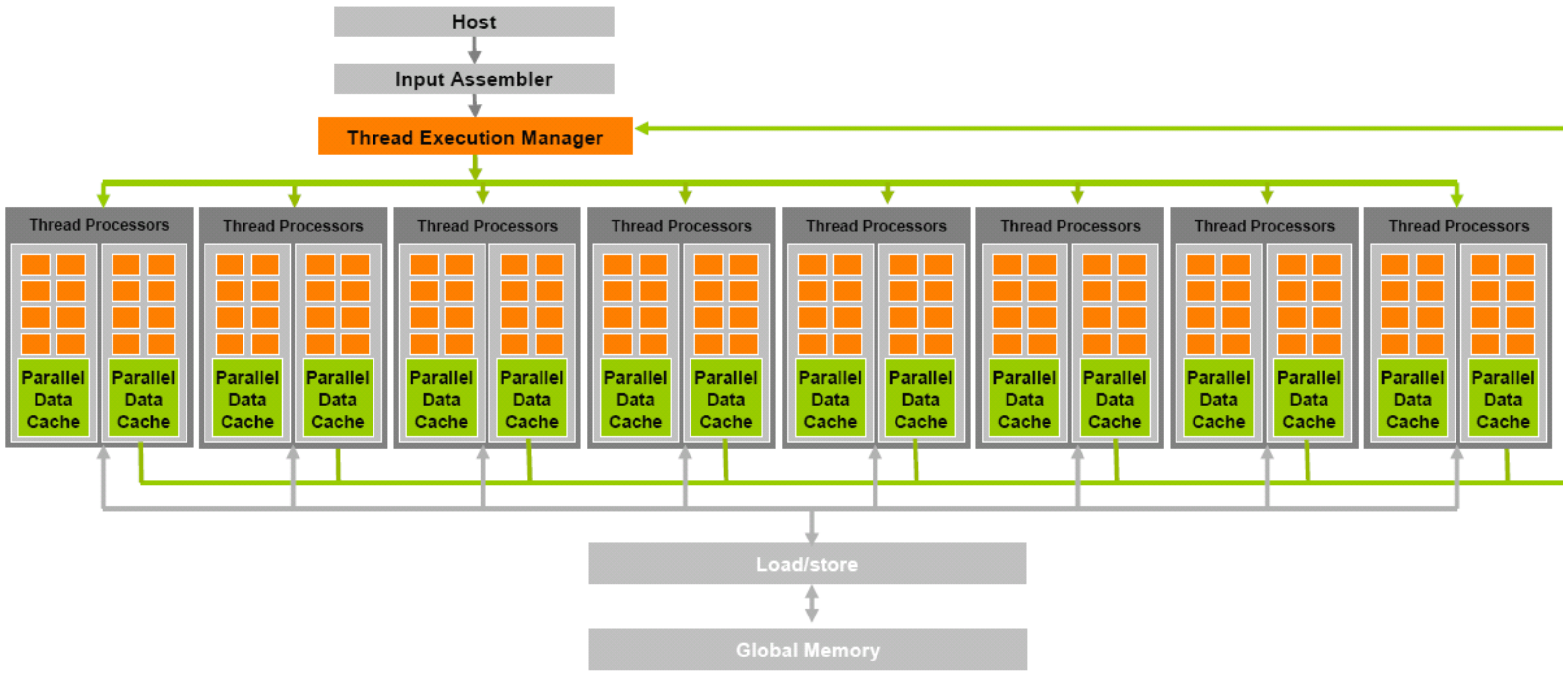
Hardware:

- GeForce
- Tesla
- Quadro

Software:

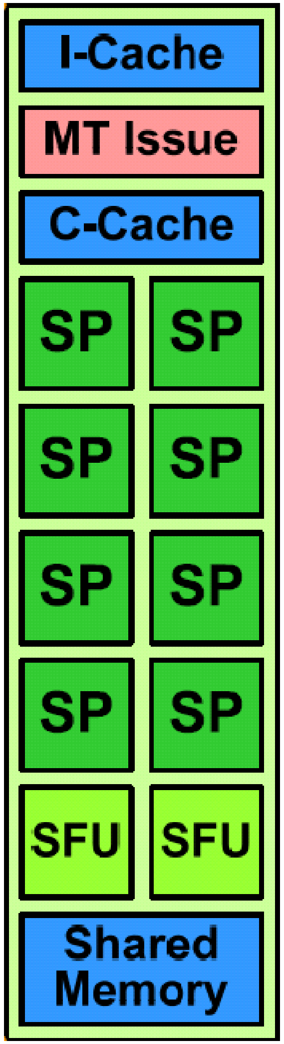
- Compute Unified Device Architecture (CUDA 2.0, Compiler+SDK)

nVIDIA - G80: the parallel stream processor



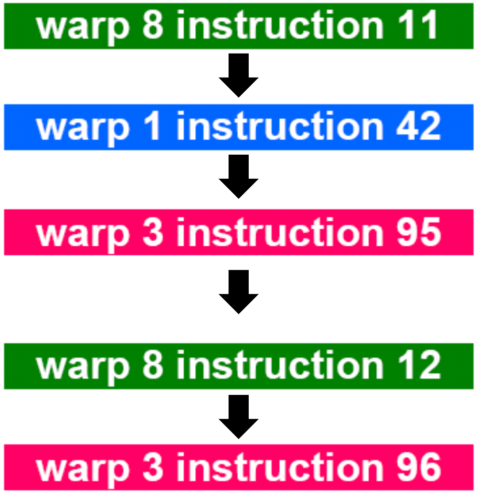
- 16 streaming multi-processors (SM) with 8 processors each, for a total of 128
- Floating-point processing power: 410 GFLOPs
- Memory Bandwidth: 104 GB/s

SM Multithreaded Multiprocessor



- SM has 8 SP Thread Processors
  - IEEE 754 32-bit floating point
  - 32-bit and 64-bit integer
  - 8K 32-bit registers
- Multithreaded Instruction Unit
  - 768 Threads, hardware multithreaded
  - 24 SIMT warps of 32 threads
  - Independent thread execution
  - Hardware thread scheduling
- 16KB Shared Memory
  - Concurrent threads share data
  - Low latency load/store

Single-Instruction Multi-Thread (SIMT) instruction scheduler



Comparison CPU-GPU

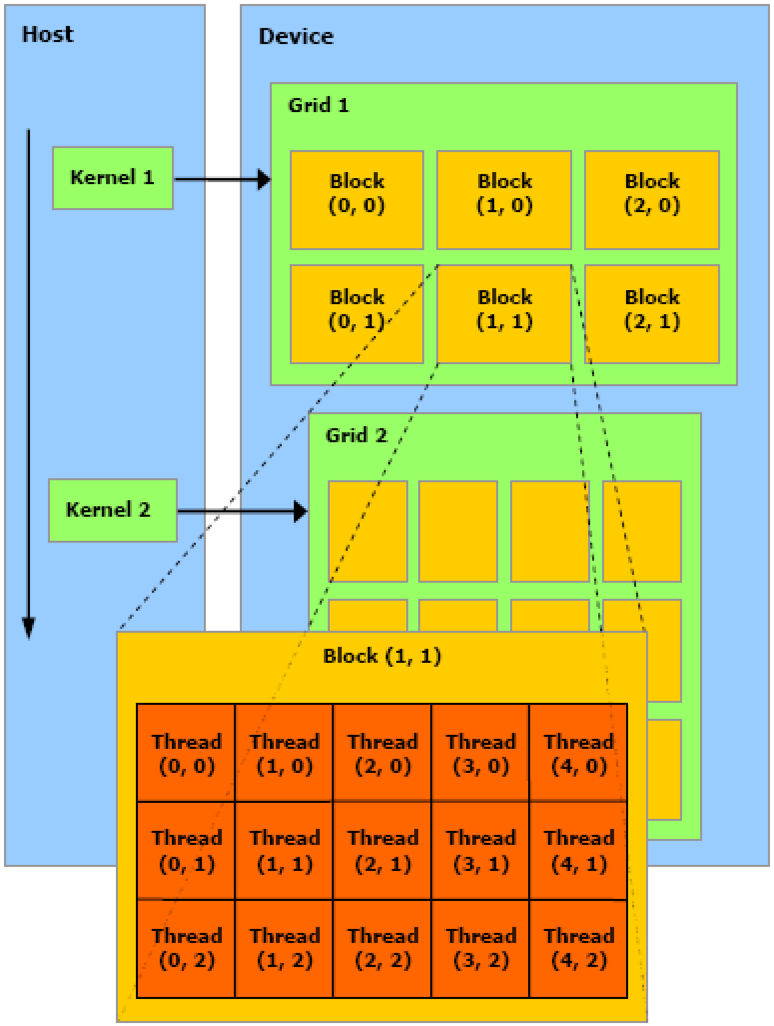
Intel Core 2 Duo | nVIDIA GTX280 | NEC SX-9A (16 CPUs)

Comparison CPU-GPU

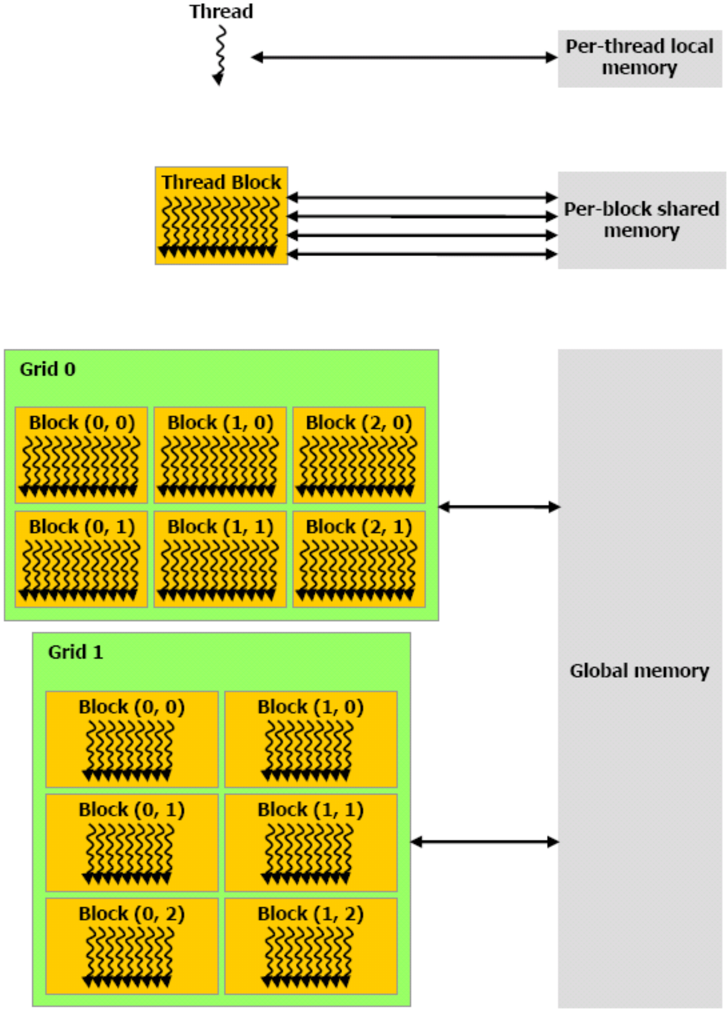
Platform	Memory [MB]	Peak [GFLOPS]	BW [GB/s]	price [Euro]
Intel Core 2 Duo (3.0 GHz)	4 000	48	7.0	1000
NEC SX-9A (16 CPUs)	1 000 000	1 600	4 000	ca. 600 000
nVIDIA GTX280	1 024	624	142	500

Common Unified Device Architecture (CUDA)

Programming model



Memory Model



Application Programming Interface (API)

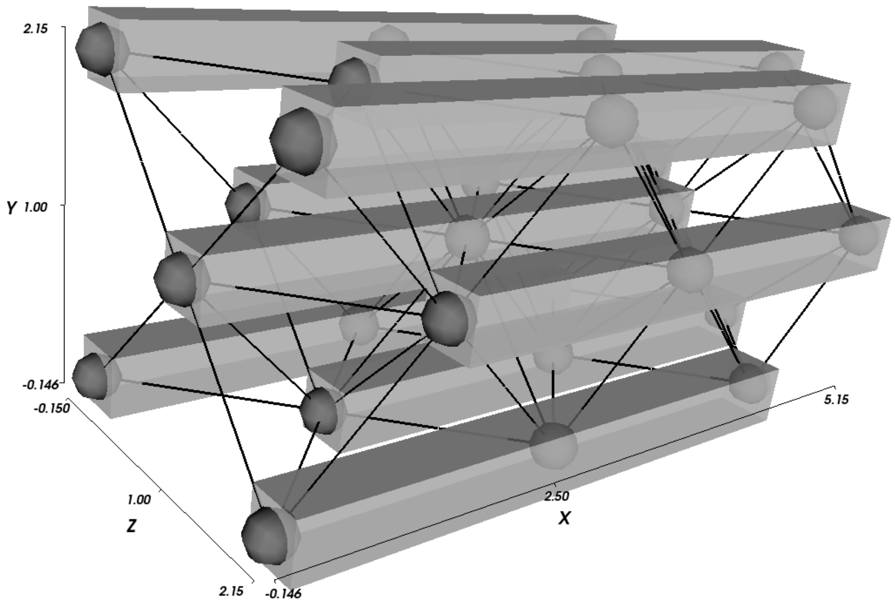
- Thread Block (typical size 64-256 threads)
- Grid of Thread Blocks (at least 16 blocks to run efficiently)
- Function Type Qualifiers (_device_, _global_, _host_)
- Variable Type Qualifiers (_device_,_shared_)
- Memory management (cudaMalloc, cudaMemcpy)
- Synchronization (_syncthreads())

Memory Bandwidth

- Effective bandwidth of each memory space depends significantly on the memory access pattern
- simultaneous memory accesses of one thread block can be coalesced into a single contiguous, aligned memory access if:
  - thread number N should access element N at address BaseAddress + sizeof(type)*N
  - sizeof(type)=4,8,16
  - BaseAddress has to be aligned to 16*sizeof(type) bytes (otherwise memory bandwidth performance breaks down to about 10 GB/sec)

Lattice Boltzmann kernel

- Load streams (13 streams for d3q13+ 1 stream geomat)
- Complex computations (collision)
- Write streams (13 streams for d3q13) to correct address (propagation)
- x-index mapped to Threads (16-256 (32-512 lattice size))
- y- and z-index mapped to grid (at least 16 in sum)



Lattice Boltzmann kernel

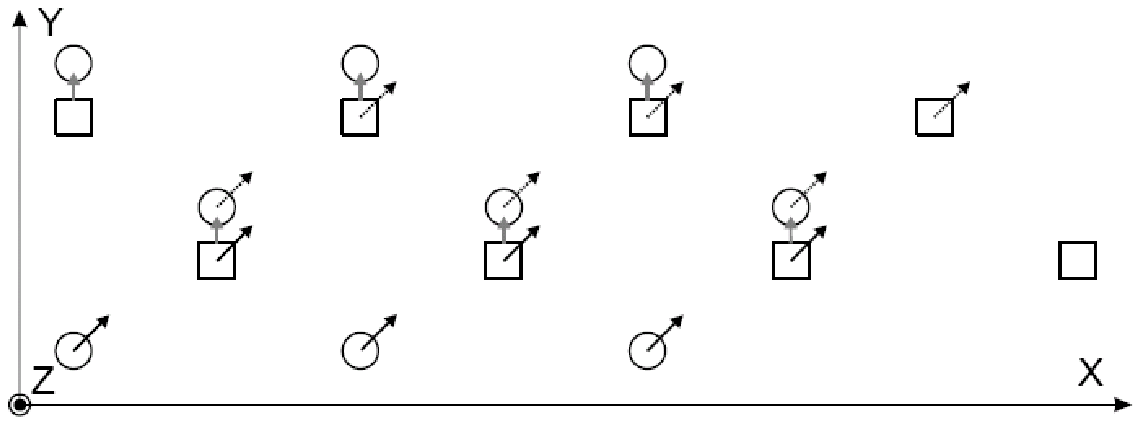
Problem:

- distribution with east- and west- (x-index) shift
- no alignment 16*sizeof(float) for high memory bandwidth!

Solution:

- Shared memory for distributions with east- and west- shift

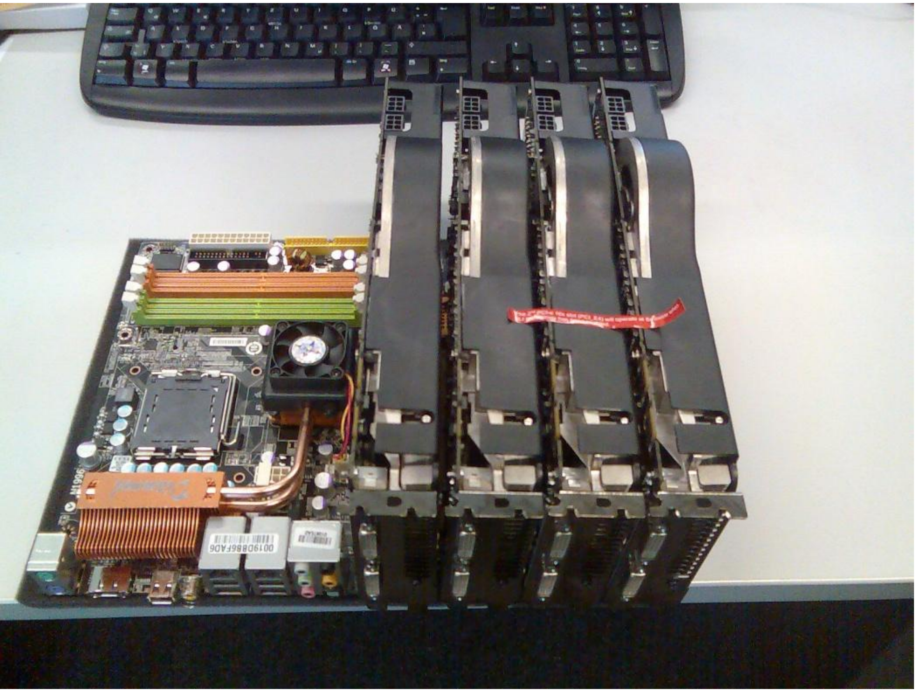
d3q13 staggered grid:



circles: device memory

squares: shared memory

Multi-GPU: Supercomputer on the Desktop -Teraflop Computing



Hardware cost:

- 4000 Euro

Communication between GPUs:

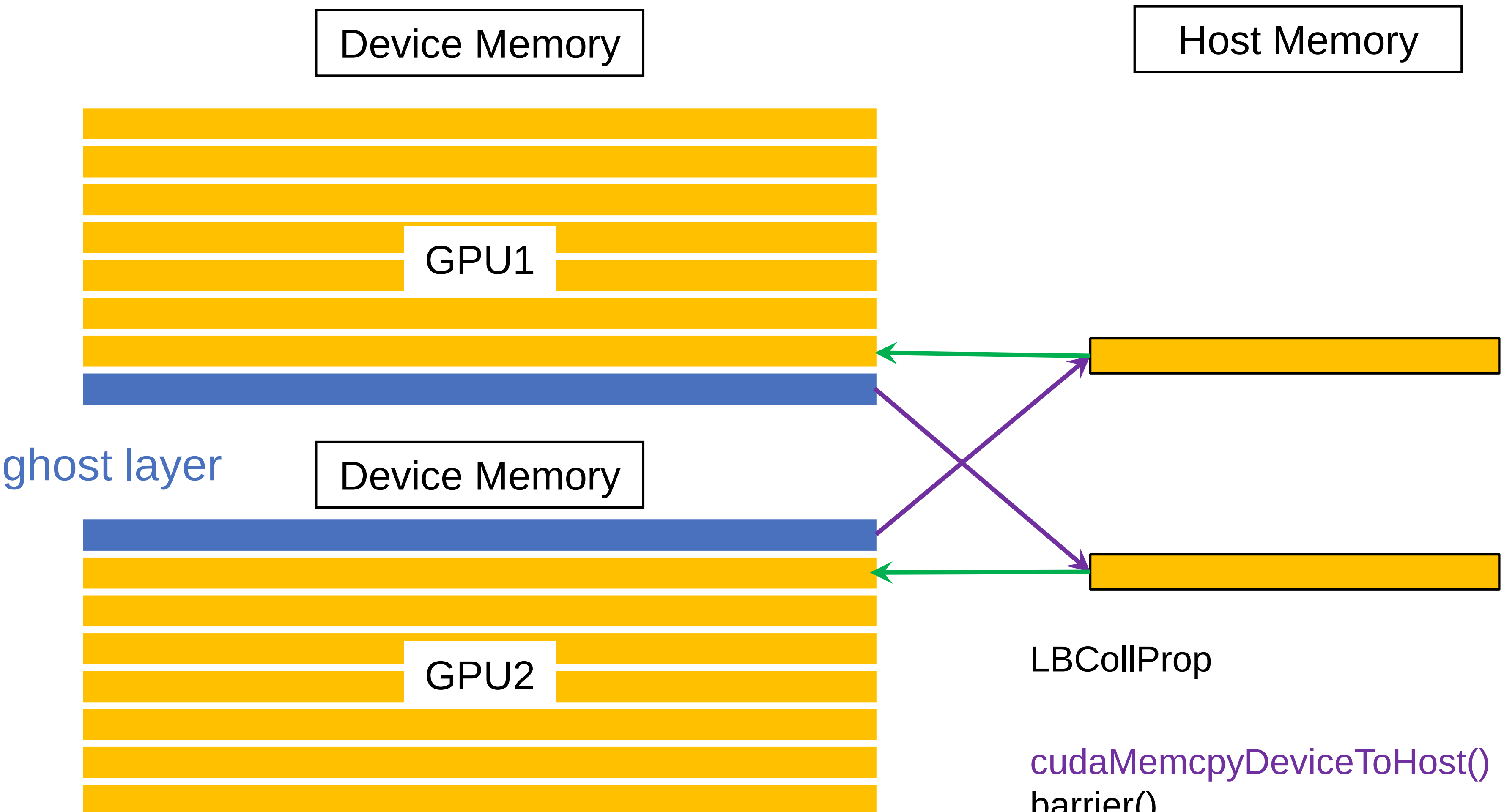
- 4 PCI Express slots
- Bandwidth Host↔Device 200-3000 MB/sec
- Latency like Front Side Bus (266 MHz)
- PThreads
- CUDA

Mainboard:
P6N Diamond MSI

→512 Cores!

	Bandwidth [MB/s]	Latency [ns]
PCI-E/FSB	300-3000	10
Infiniband	312-7500	5 000
G-Ethernet	125	80 000

Multi-GPU

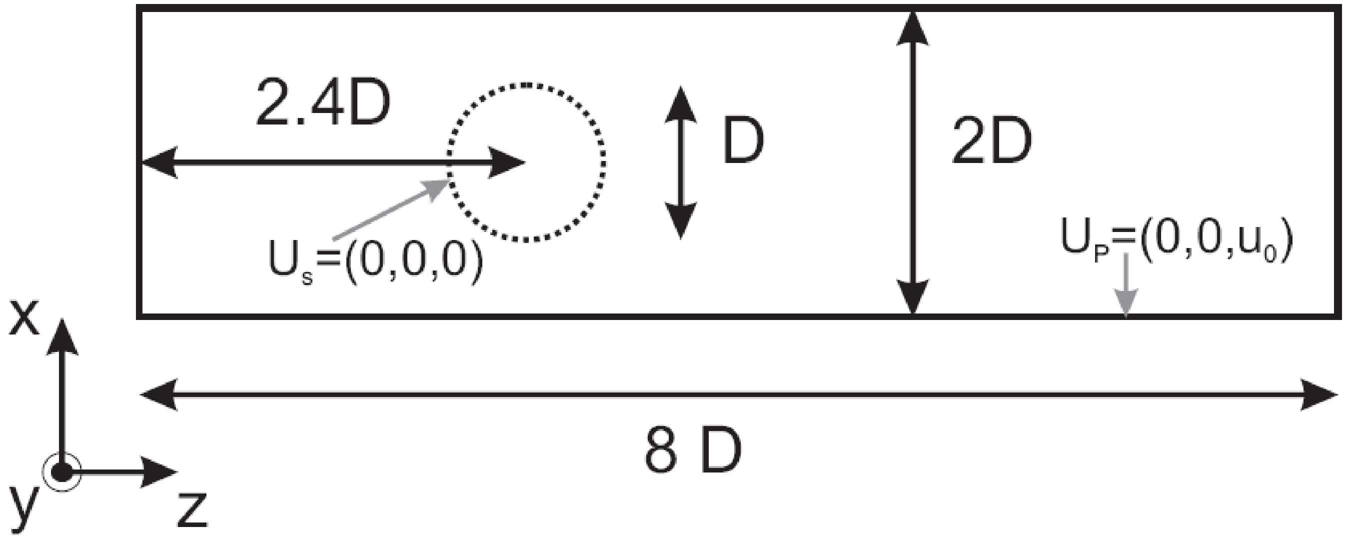


LBCollProp

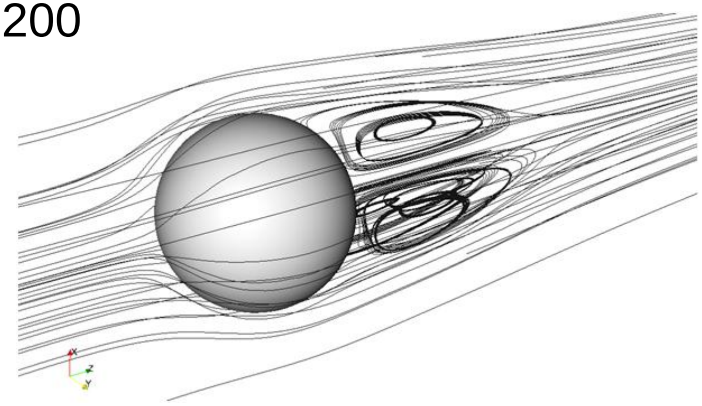
cudaMemcpyDeviceToHost()
barrier()

cudaMemcpyHostToDevice ()
barrier()

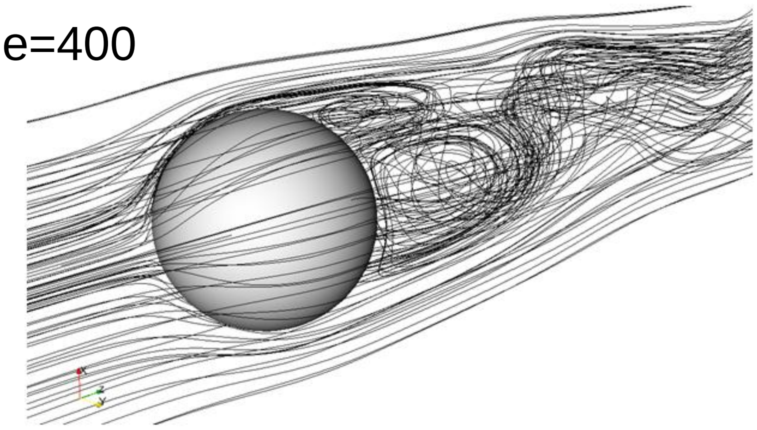
Example: Moving Sphere in a pipe



Re=200



Re=400



Moving Sphere in a pipe: Results (1 GPU)

R. Clift, J. R. Grace, M. E. Weber: Bubbles, Drops and Particles, Academic Press, 1978

(2nx,ny,nz) = 128x128x512

Re [-]	ν [$m^2 s^{-1}$]	WCT [s]	# iter[-]	$c_{d,W}$ [-]	$c_{d,W,Ref.}$ [-]	$\frac{p.drag}{v.drag}$	Rel. Err. [-]
10	0.121920	106	15 000	14.74	15.84	0.93	6.9%
50	0.024384	415	59 000	3.697	3.876	1.15	4.6%
100	0.012192	520	74 000	2.380	2.312	1.43	2.9%
200	0.006096	774	110 000	1.679	1.706	1.90	1.6%
300	0.004064	2100 [1]	300 000	1.440 [2]	1.448	2.35	0.6%
400	0.003048	2800 [1]	400 000	1.305 [3]	1.296	2.82	0.7%

[1] nonstationary flow field, time required to reach oscillatory state from initial uniform flow field (no disturbance imposed)

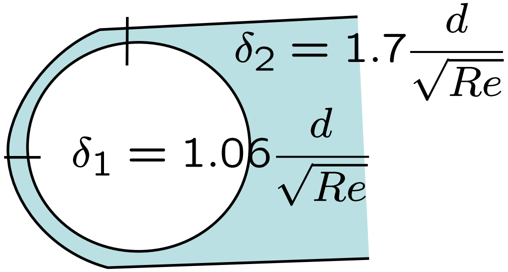
[2] average value, $t = 280 \ldots 2000\, T_{ref}$

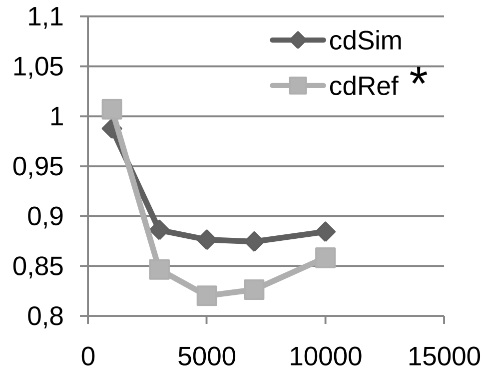
[3] average value, $t = 200 \ldots 3000\, T_{ref}$

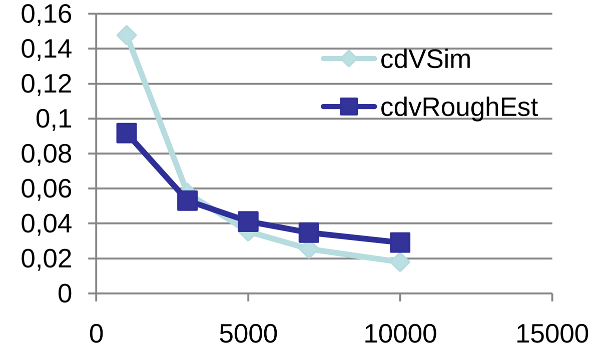
Moving Sphere in a pipe: Results LES (3 GPUs)

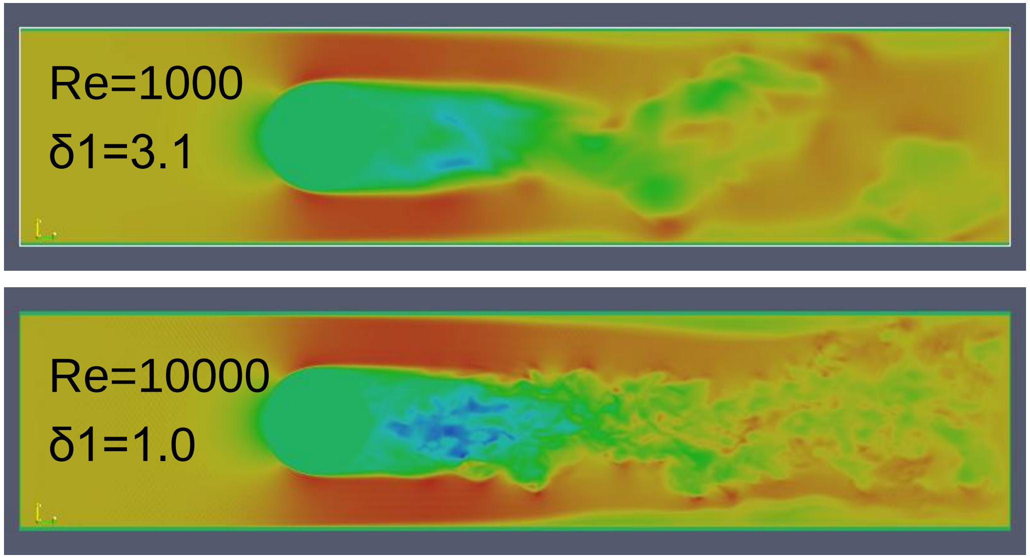
* R. Clift, J. R. Grace, M. E. Weber: Bubbles, Drops and Particles, Academic Press, 1978

(2nx,ny,nz) = 192x192x864, Re=1000,3000,5000,7000,10000

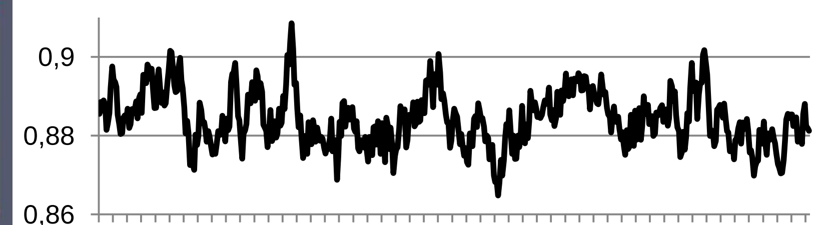








- 16 Mio grid points
- 1 Mio time steps
- 4h comput. time
- T_{ges} = 50 turnover times
- T_{ave}=0.5-1.0 T_{ges}



Achievable Performance of LB-Kernels

Wellein 2006

- LUPS: Lattice updates per second
- Limitation by ***Memory Bandwidth***:

  max LUPS = theoretical BW / [(14(read)+13(write)) * 4 byte]
  max LUPS = 3.5E9 Byte/s / (4*(14+13) Byte/lattice node) = 33 E6 LUPS

- Limitation by ***Performance of CPU***:

  max LUPS = theoretical FLOPS / (NCOLL /lattice node)

  NCOLL = 260 FLOP = 30+30 + 200 FLOPS (130 Additions / 30 Multiplications)
  max LUPS3 = 8.0E9 FLOPS / (260 FLOP/lattice node) = 31 E6 LUPS

This Notebook:

grid	MLUPS	BW	Perf.
8^3	15	-	48 %
64^3	9	27 %	29 %

Performance for D3Q13 model on GPU

- LUPS: Lattice updates per second
- Limitation by ***Memory Bandwidth***:

  max LUPS = theoretical BW / [(14(read)+13(write)) * 4 byte]
  max LUPS = 104E9 Byte/s / (4*(14+13) Byte/lattice node) = 963 E6 LUPS

- Limitation by ***Performance of CPU***:

  max LUPS = theoretical FLOPS / (NCOLL /lattice node)

  NCOLL = 260 FLOP = 30+30 + 200 FLOPS (130 Additions / 30 Multiplications)
  max LUPS3 = 410E9 FLOPS / (260 FLOP/lattice node) = 1 577 E6 LUPS

Moving Sphere in a pipe: Performance Single GPU

Tesla test sample (GT200)

- 192 cores (1.1GHz)
- 101 GB/s throughput
- supports double precision

Results for grid 64(128)x128x512 (single prec.)

- 690 MLUPS
- ***72 % Throughput (!)*** (83 % pure MemCpy)
- 43 % peak perf.

Moving Sphere in a pipe: Performance Multi-GPU (Ultra 8800)

Increasing Problem Size (number of threads 64):

Partition in z-direction

# cards	nx,ny,nz	P [MLUPS]	Eff.[%]	th. Eff. [%]
1	128 x 128 x 512	545	-	-
2	128 x 128 x 1024	1029	94	96
3	128 x 128 x 1536	1460	89	91

SGI at Munich: d3q19, 4096 cores, 12E9 LUPS, 30 Mio Euro

Fixed Problem Size (128 x 128 x 512, number of threads 64):

# cards	P [MLUPS]	Eff.[%]	th. Eff. [%]
1	545	-	-
2	963	88	93
3	1184	72	77

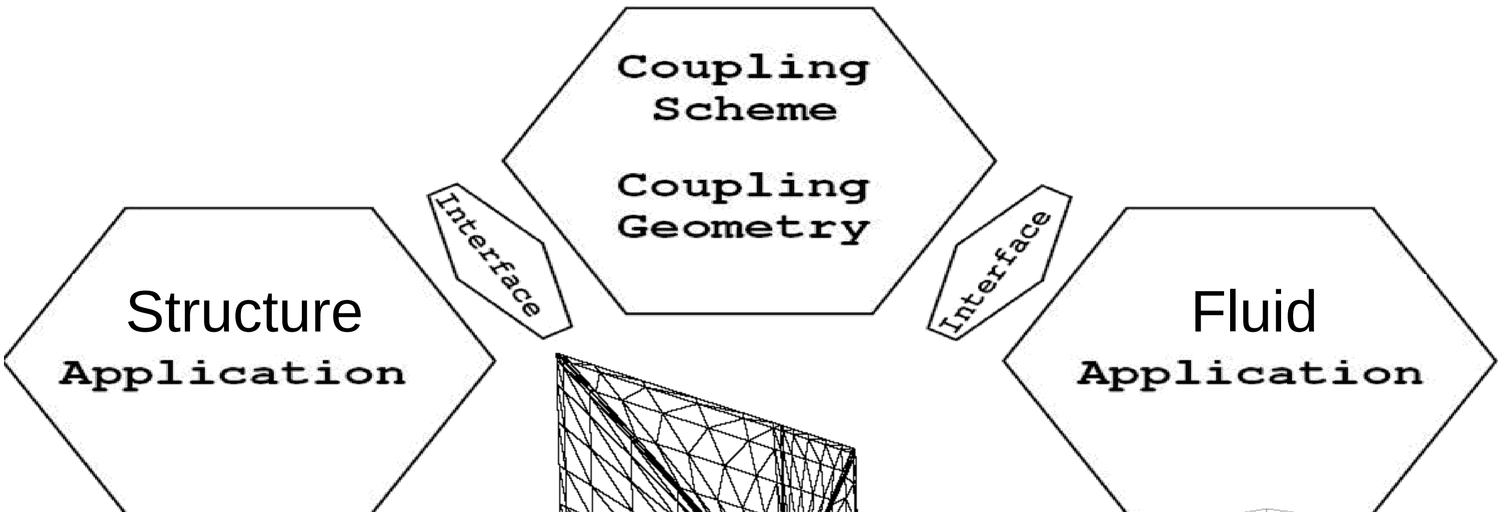
No Communication Hiding with Comp. Cap. <1.1

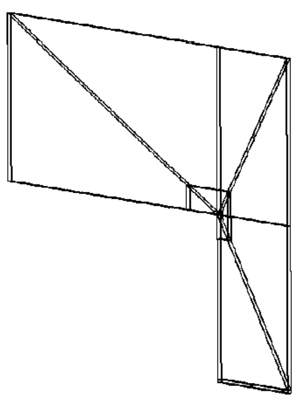
$$EN = \frac{0.5 \times Throughput}{nzloc \times MBW_{net}}$$

$$th.Eff. = \frac{1}{1 + EN}$$

card 1,2: $MBW_{net} = 1.5GB/sec$, card 3: $MBW_{net} = 0.55GB/sec$

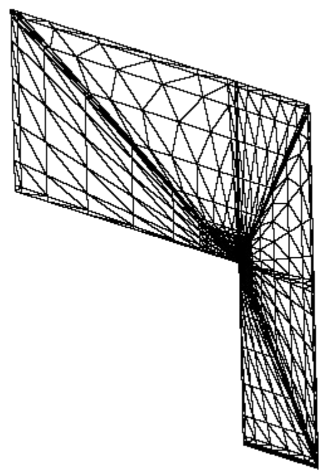
FSI: Coupling Scheme / Interface mesh



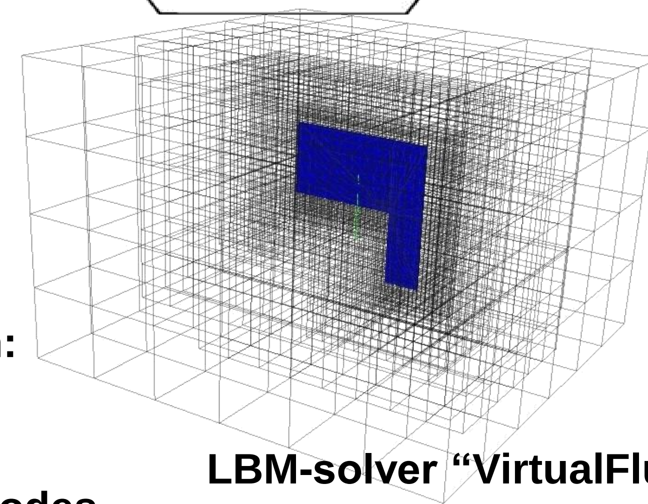


p-FEM solver "AdhoC"

(TU München)



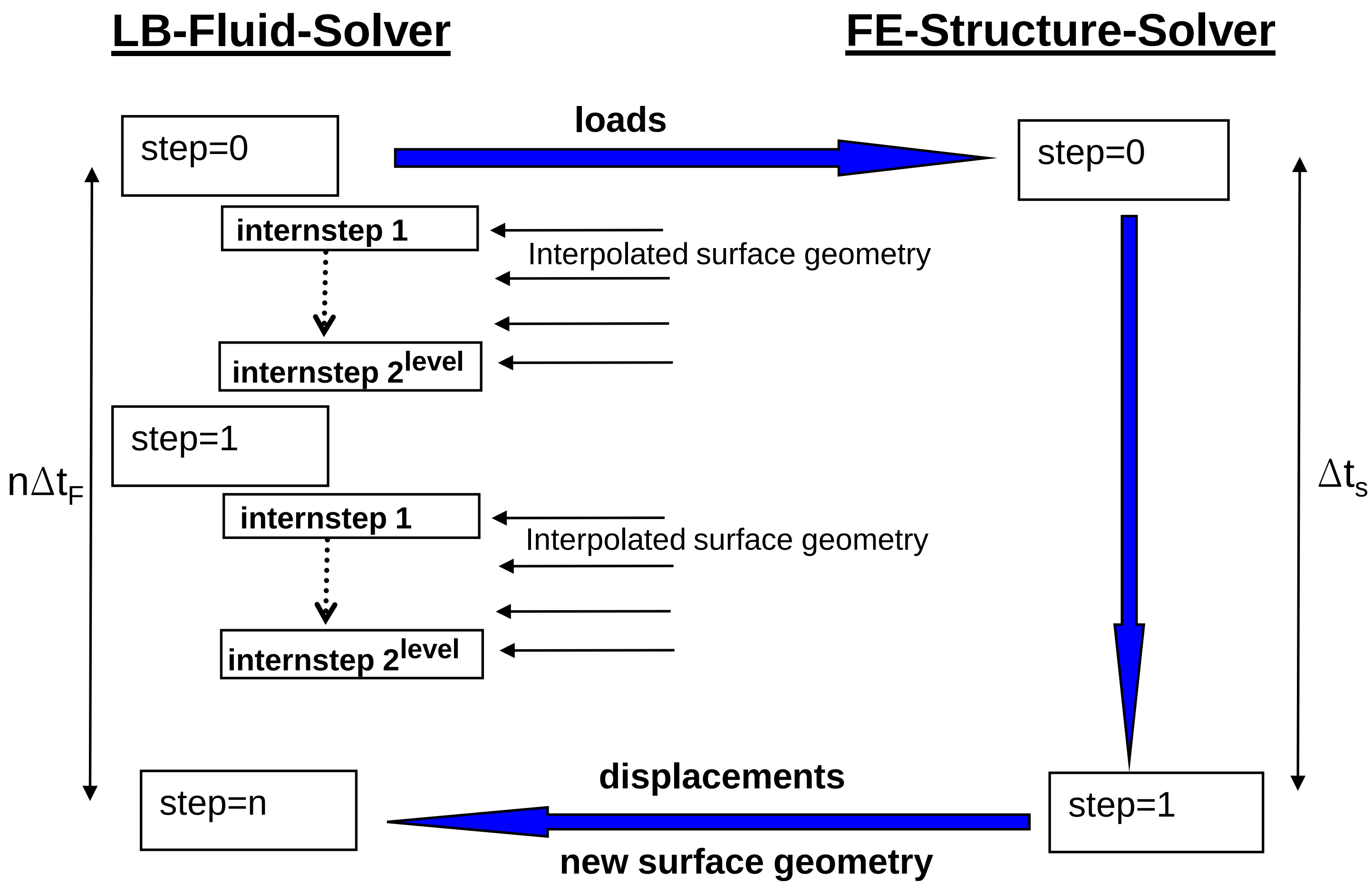
Moving Surface mesh:
- **flat triangles**
- **loads on nodes**
- **displacements on nodes**
- **linear interpolation**



LBM-solver "VirtualFluids"

(TU Braunschweig)

Coupling algorithm (explicit) / Mapping of surface mesh

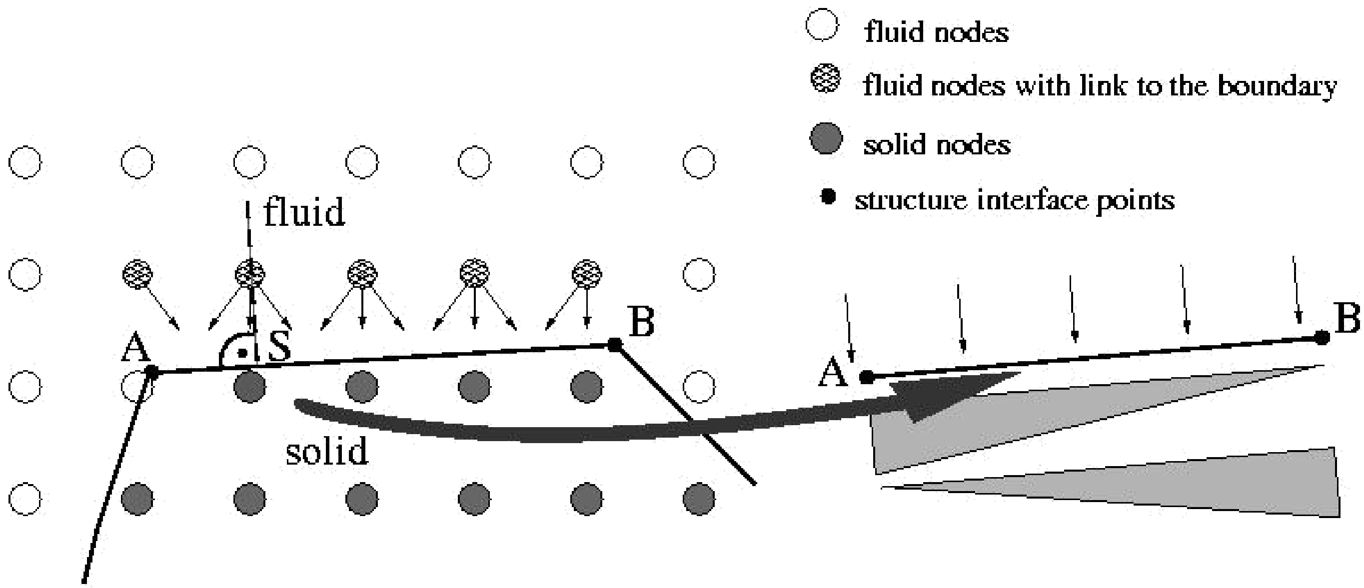


Mapping of loads on surface mesh / method1 (conservative)

momentum exchange (Ladd, 1992, 2002) :

$$dI = c_0 \,(f_{inversDir}(x, t+\Delta t) + \tilde{f}_{Dir}(x,t))$$

forces: $$\vec{F}_i = \sum_i \vec{v}_i \cdot [\, f_{inversDir}(x, t+\Delta t) + \tilde{f}_{Dir}(x,t)] \cdot \frac{\Delta x^2}{\Delta t}$$

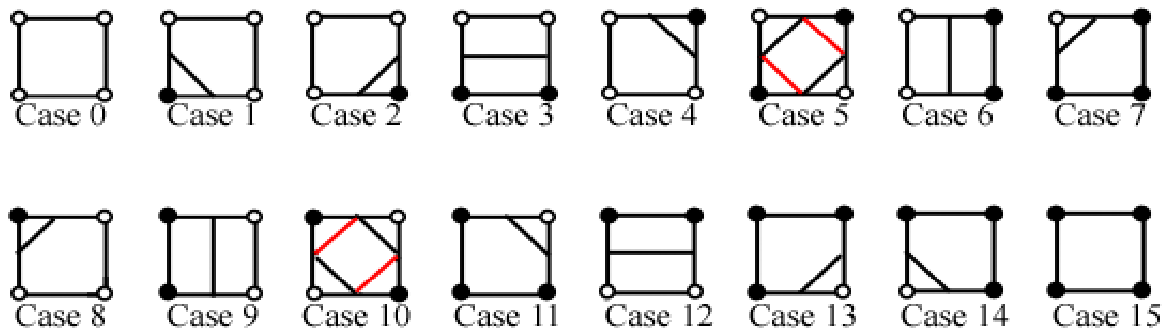


Mapping of loads on surface mesh / method 2 (profile preserving)

stress tensor (local):

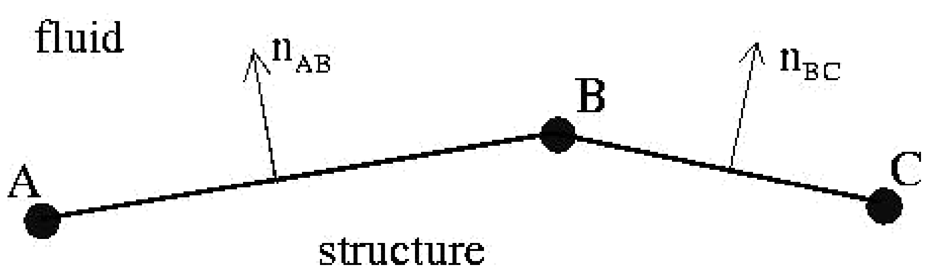
$$P_{\alpha\beta} = c_s^{\,2} \rho\, \delta_{\alpha\beta} + \left(1 - \frac{s_{vis}}{2}\right) \sum_k f_k^{neq} v_{k,\alpha} v_{k,\beta}$$

linear extrapolation of stress tensor depending on configuration of active nodes:



forces:

$$\vec{F}_B = \left(\frac{1}{4} \underline{\underline{P}}_A + \frac{3}{4} \underline{\underline{P}}_B\right) \vec{n}_{AB} \frac{1}{2} l_{AB} + \left(\frac{3}{4} \underline{\underline{P}}_B + \frac{1}{4} \underline{\underline{P}}_C\right) \vec{n}_{BC} \frac{1}{2} l_{BC}$$



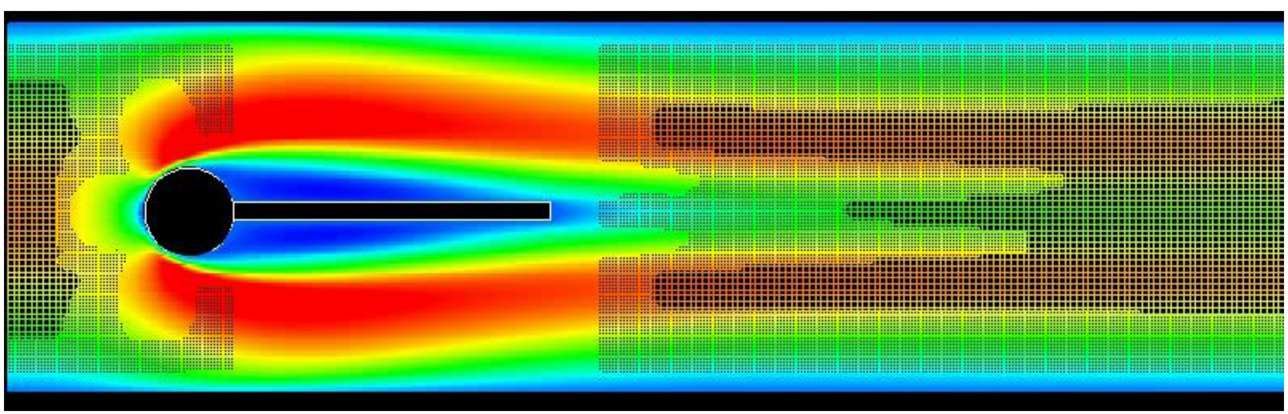
Benchmark FSI 1/2/3 Results

- FE-Solver Adhoc :
  - ca. 1000 DOF
  - Newmark: □=0.49 γ=0.9
- Explicit Coupling
- 3 Grid Levels Fluid solver
- Coupling 2*dt_f=dt_s
- approx. 1000 time steps per periode

	FSI1	FSI2	FSI3
#nodes Fluid	125553	160170	275646
err. Ux [%]	**0.9**	**2.7 ± 2.9 (0.1)**	**6.7 ± 7.1 (0.9)**
err. Uy [%]	**1.4**	**6.5 ± 0.2 (5.3)**	**0.2 ± 1.7 (3.8)**

Computational Time FSI2:

1h/Periode

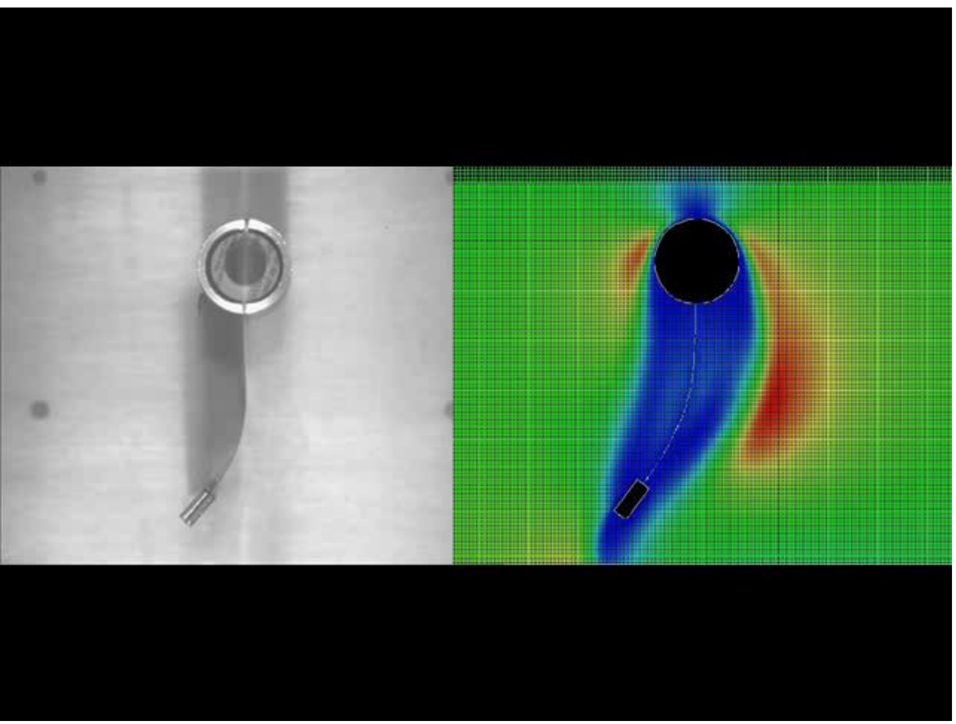


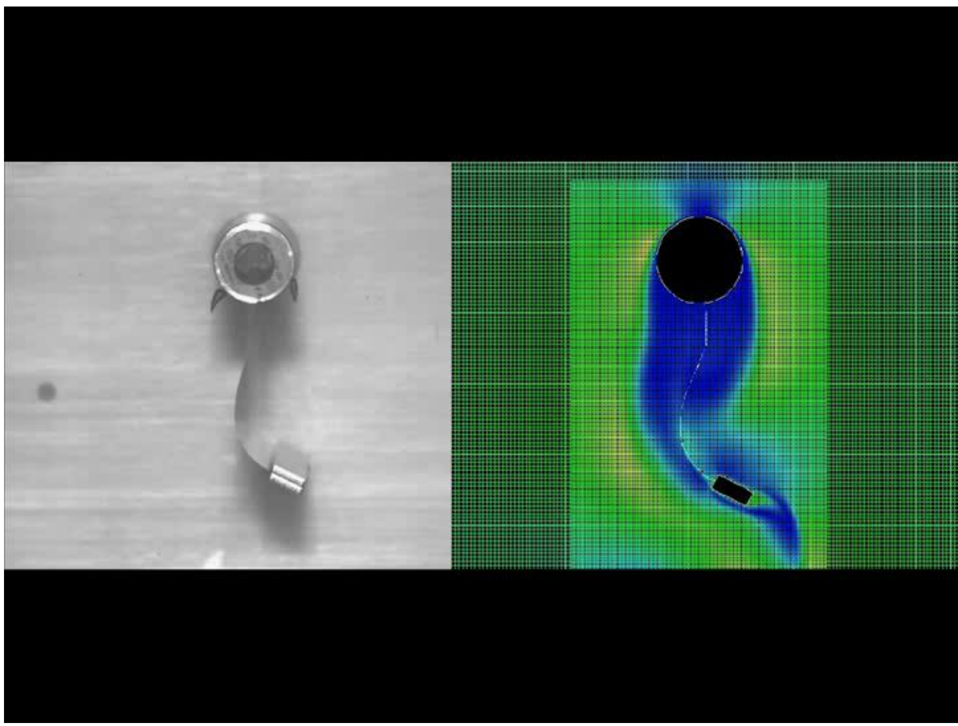
Experimental Benchmark

EXPERIMENTAL STUDY ON A FLUID-STRUCTURE INTERACTION REFERENCE TEST CASE
Jorge P. Gomes and Hermann Lienhart
Fluid-Structure Interaction: Modelling, Simulation, Optimisation
Lecture Notes in Computational Science and Engineering , Vol. 53, pages 356 - 370

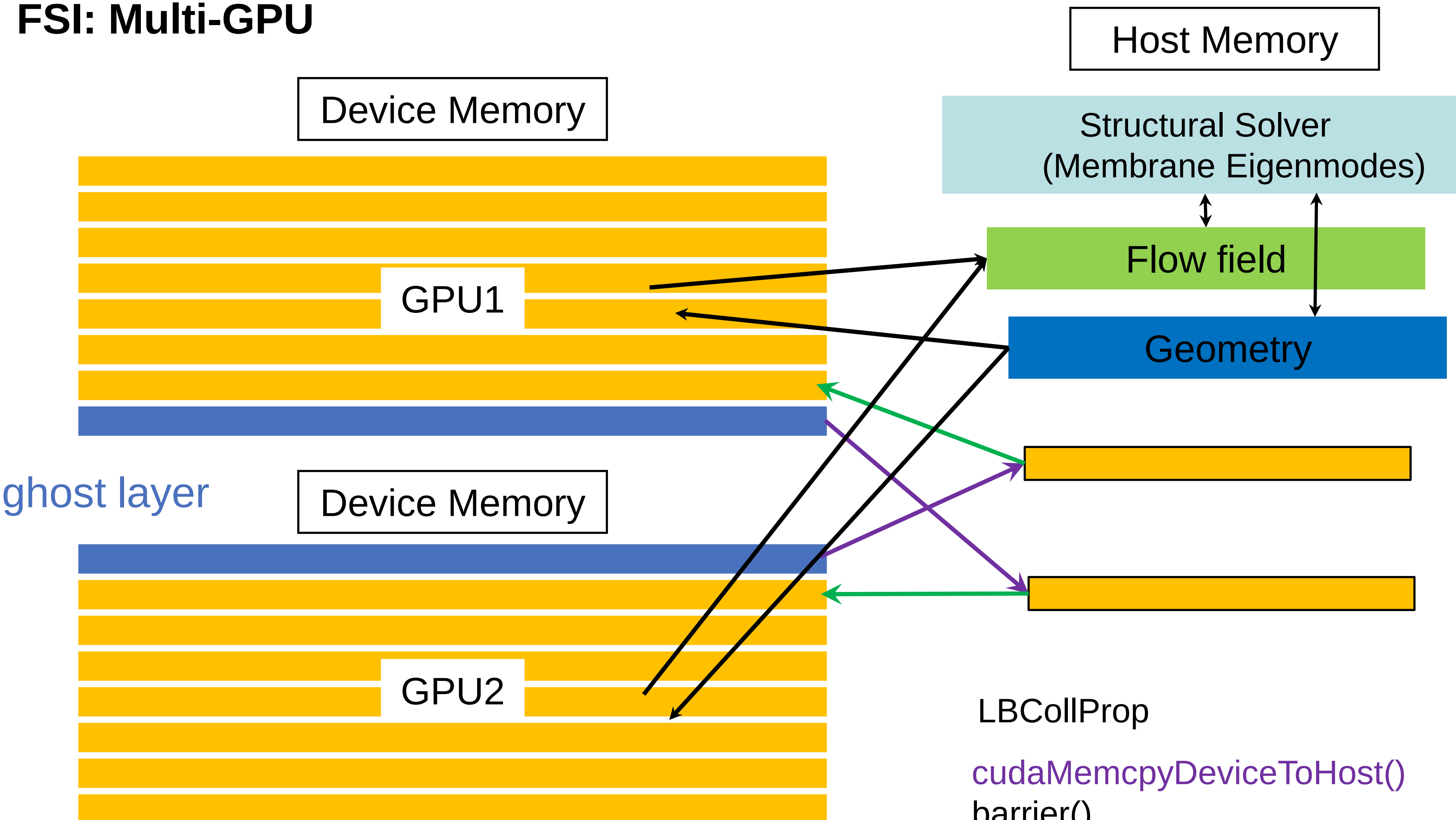
Re=140

Re=190





FSI: Multi-GPU



Transfer if needed for FSI or Postprocessing

LBCollProp

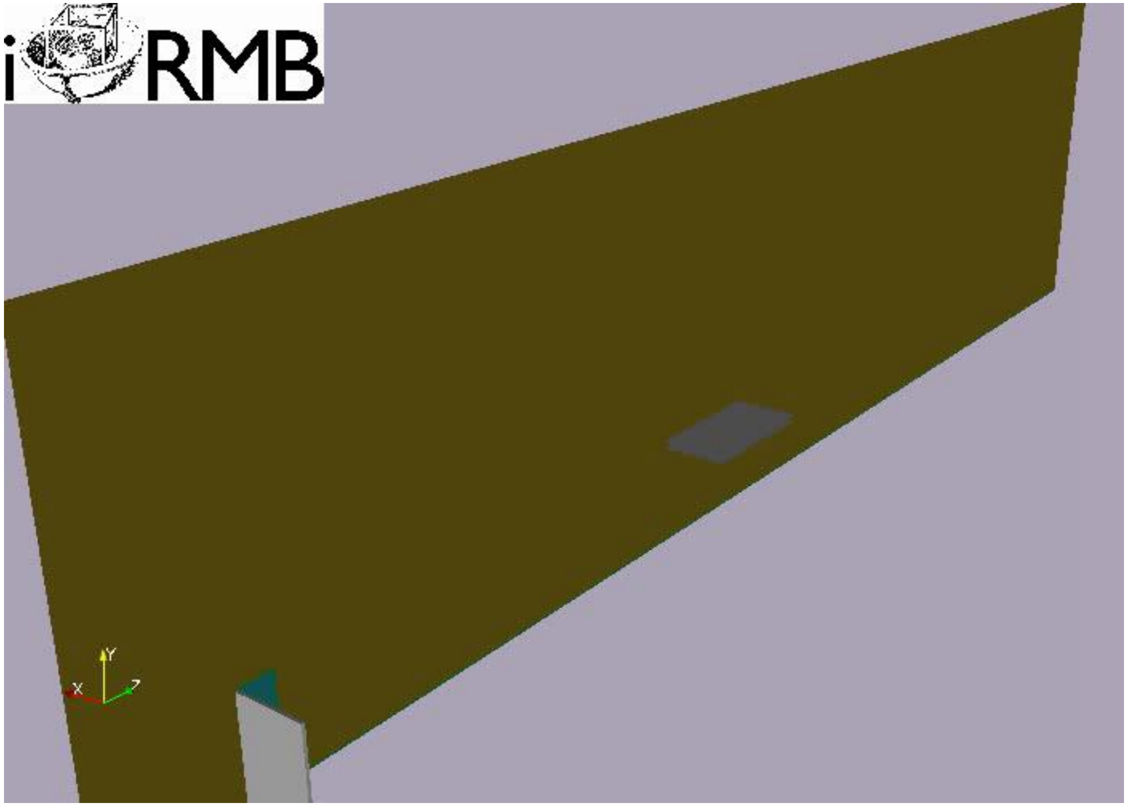
cudaMemcpyDeviceToHost()
barrier()

cudaMemcpyHostToDevice ()
barrier()

Oscillating Membrane in flow field

$$\mathrm{Re} = \frac{u_0 L}{\nu} = 6000 \qquad \mathrm{Ae} = \frac{T}{\rho_f \; u_0^2 \; L} = 0.111 \qquad \frac{\rho_s}{\rho_f} = 100$$

- lattice size: 128x128x512
- 3 GPUs
- > 1E9 LUPS
- 250 time steps in 1 sec
- LES
- Compt. Steering
- 500 000 time steps
- 2000 sec total time
- dt_s=10 dt_f
- Membrane: Eigenmodes



Conclusions

- EURO/FLOP (CHEAP) and WATT/FLOP (Green Computing) very favorable
- No tedious access to supercomputers (data transfer)
- Inverse Moore's Law (Good for complex collision operators!)

Fluid-Structure-Interaction
- no remeshing for moving or deformed obstacles (Eulerian grid)
- explicit time stepping scheme for FSI is feasible

Further Applications:
- Computational Steering
- Interactive Shape Design, Optimization
- Aeroacoustics
- ...

Outlook

Developments Hardware nVIDIA

- Overlap of computation and copies device↔host (Comp. Cap. >= 1.1)
- Double precision cards
- Tesla products

Further Developments:

- Communication Hiding (Comp. Cap. >= 1.1)
- Grid Refinement
    - Tree type data structure on the CPU
    - Leaves are Matrices
    - Compute-intensive Leaves are loaded to GPU

References

- Jonas Tölke (2008): ***Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture***, Computing and Visualization in Science, DOI 10.1007/s00791-008-0120-2

- Tölke, J. and Krafczyk, M. (2008): ***TeraFLOP computing on a desktop PC with GPUs for 3D CFD***, International Journal of Computational Fluid Dynamics, 22:7, 443 — 456

3D driven cavity: Performance on a single card

GeForce 8800 Ultra (G80), MLUPS

ny × nz \ nx	16	32	64	80	128	192	256
32 ×32	231	392	570	446	523	444	476
64 ×64	239	378	565	472	546	454	483
128×128	230	384	592	478	549	452	483

throughput: 63GB/sec ^= ***61 %*** of Max. Bandwidth (75GB/sec pure MemCpy)

computational performance: 38 % of peak perf.

Moving Sphere in a pipe: Performance Single GPU

Tesla test sample (GT200)

- 192 cores (1.1GHz)
- 101 GB/s throughput
- supports double precision

Results for grid 64(128)x128x512 (single prec.)

- 690 MLUPS
- ***72 % Throughput (!)*** (83 % pure MemCpy)
- 43 % peak perf.

Performance of Products based on G80/G92/GT200 chips

Product	# Cores	Processor Clock [MHz]	Performance [GFLOP]	Bandwidth [GB/sec]	Memory [MB]
GeForce GTX 280	240	1300	624	142	1024
GeForce 9800 GX2	256	1500	768	128	1024
GeForce 8800 Ultra	128	1600	410	104	768
Quadro FX 5600	128	1300	333	77	1500
Tesla C1060	240	1300	624	102	4000

Comparison CPU-GPU

Platform	**Memory [MB]**	**Peak [GFLOPS]**	**BW [GB/s]**	**price [Euro]**
Intel Core 2 Duo (3.0 GHz)	4 000	48	7.0	1000
NEC SX-8R A (8 CPUs)	128 000	281	563	expensive
nVIDIA GTX280	1 024	624	142	500