Lattice-Boltzmann Simulations on GPUs

Multiple relaxation time LB with ESPResSo

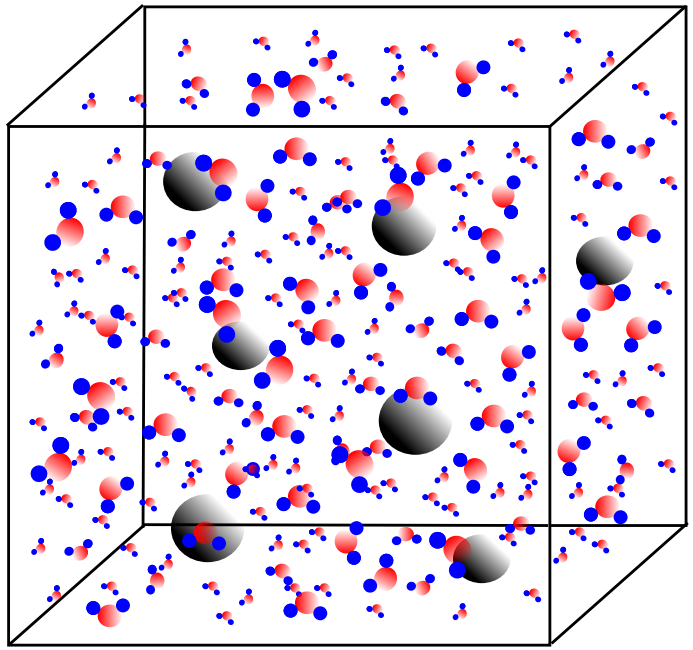
Dominic Röhm

Institute for Computational Physics

11.10.2012

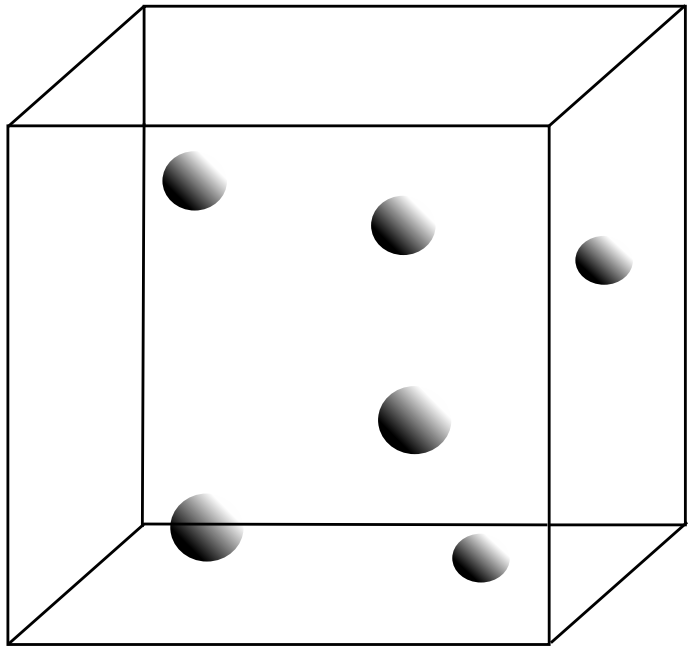
Motivation

- Molecular dynamics (MD) simulations $\Rightarrow$ simulating solvent explicitly is expensive



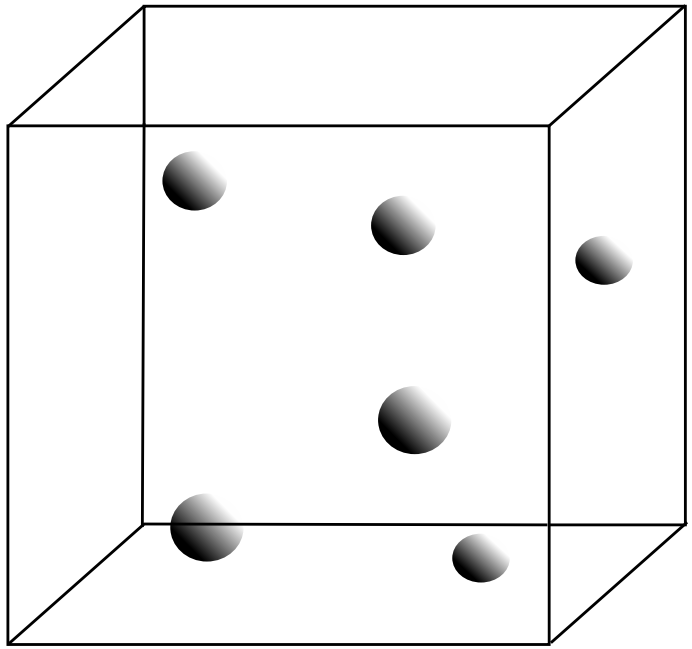
Motivation

- Molecular dynamics (MD) simulations $\Rightarrow$ simulating solvent explicitly is expensive
- Simple way $\Rightarrow$ use a thermostat (e.g. Langevin)



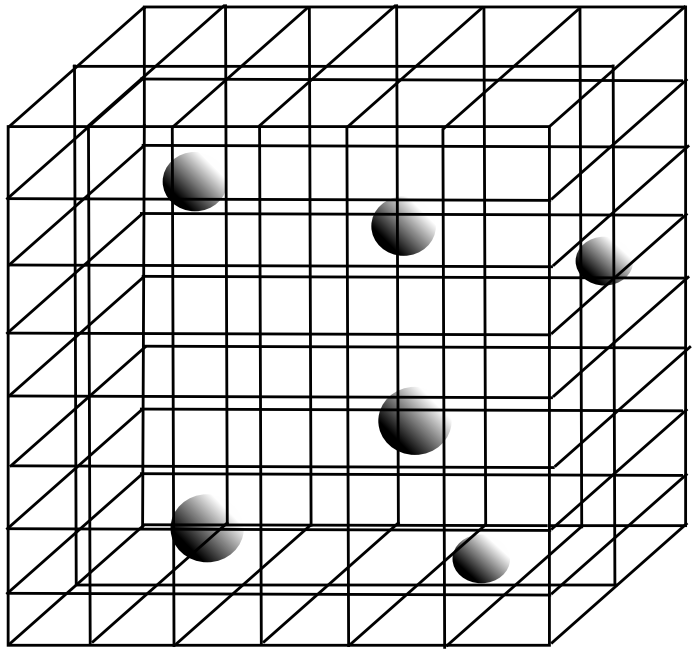
Motivation

- Molecular dynamics (MD) simulations $\Rightarrow$ simulating solvent explicitly is expensive
- Simple way $\Rightarrow$ use a thermostat (e.g. Langevin)
- **Problems**: confined geometries and no hydrodynamic interaction between particles



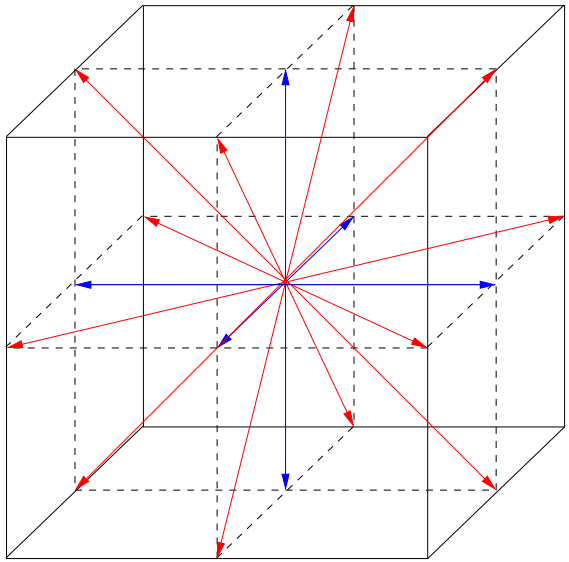
Motivation

- Molecular dynamics (MD) simulations $\Rightarrow$ simulating solvent explicitly is expensive
- Simple way $\Rightarrow$ use a thermostat (e.g. Langevin)
- **Problems**: confined geometries and no hydrodynamic interaction between particles
- **Solution**: Lattice Boltzmann method



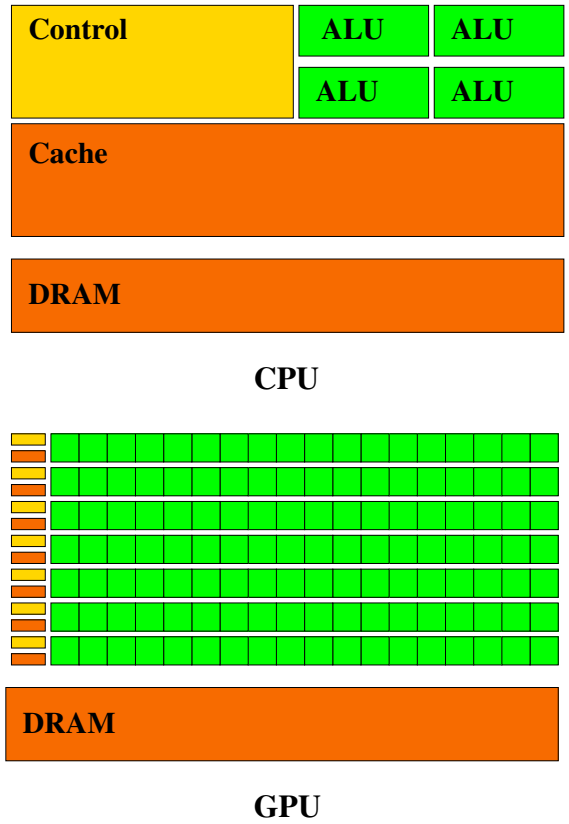
LB on GPUs

- Lattice based, well suited for **S**ingle **I**nstruction **M**ultiple **D**ata (SIMD) scheme



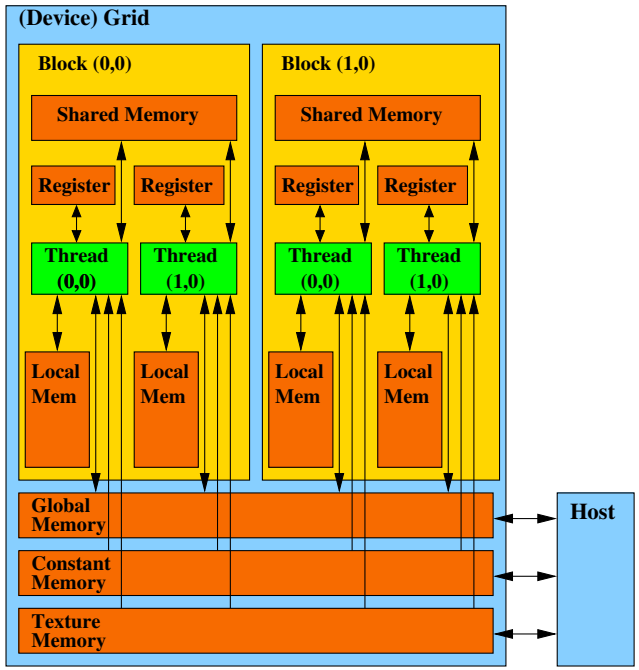
LB on GPUs

- Lattice based, well suited for **S**ingle **I**nstruction **M**ultiple **D**ata (SIMD) scheme
- GPUs $\Rightarrow$ Execute same code massively parallel



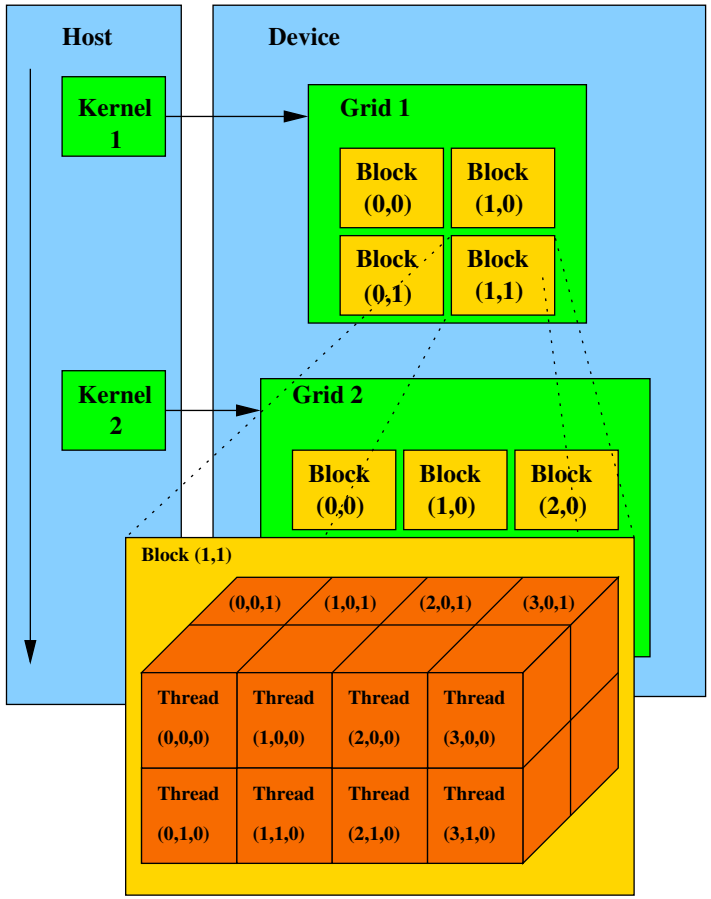
LB on GPUs

- Lattice based, well suited for **S**ingle **I**nstruction **M**ultiple **D**ata (SIMD) scheme
- GPUs $\Rightarrow$ Execute same code massively parallel
- Different types of memory explicitly accessed, atomic operations



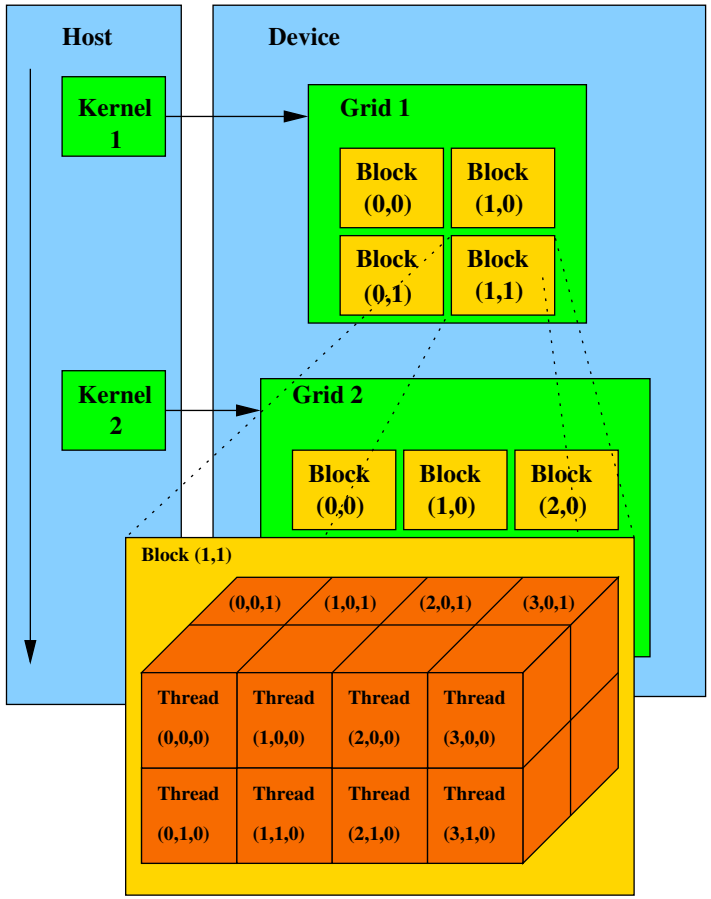
LB on GPUs

- Lattice based, well suited for **S**ingle **I**nstruction **M**ultiple **D**ata (SIMD) scheme
- GPUs $\Rightarrow$ Execute same code massively parallel
- Different types of memory explicitly accessed, atomic operations
- Hierarchical thread scheme



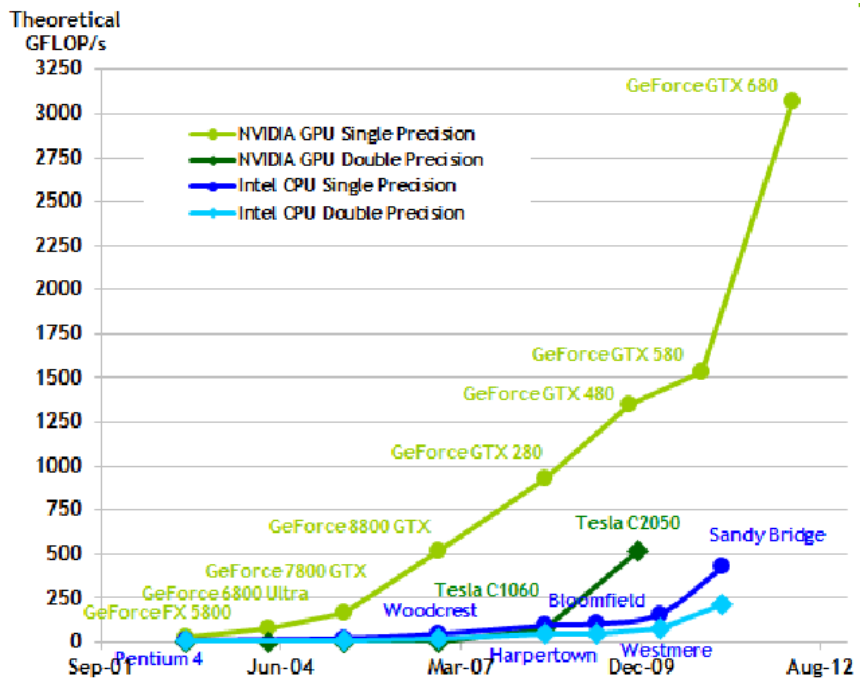
LB on GPUs

- Lattice based, well suited for **S**ingle **I**nstruction **M**ultiple **D**ata (SIMD) scheme
- GPUs $\Rightarrow$ Execute same code massively parallel
- Different types of memory explicitly accessed, atomic operations
- Hierarchical thread scheme
- Asynchronous memory copy, streams



Theoretical power of (NVIDIA) GPUs

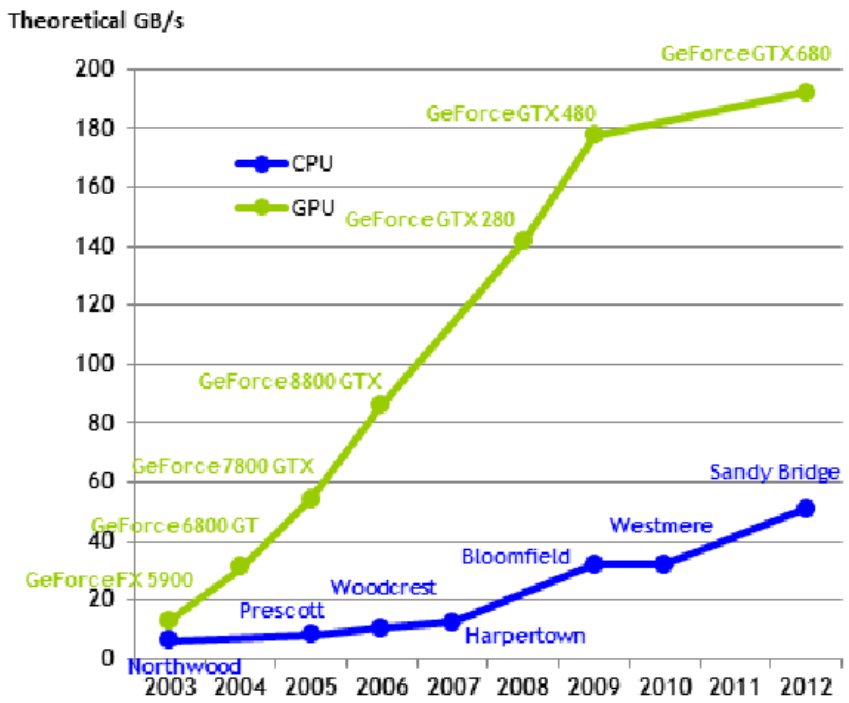
- Floating point ops per sec



Theoretical speedup almost 10x

Theoretical power of (NVIDIA) GPUs

- Floating point ops per sec
- Bandwidth of the VRAM



Theoretical speedup almost 10x

Lattice-Boltzmann on GPUs

- Velocity space resides in (large but “slow”) global memory
- Double buffering to avoid race conditions during streaming
- Memory layout is optimized for coalesced access to the fluid velocities
- Kernels for solvent update and particle-solvent interaction

Lattice-Boltzmann on GPUs

- Velocity space resides in (large but “slow”) global memory
- Double buffering to avoid race conditions during streaming
- Memory layout is optimized for coalesced access to the fluid velocities
- Kernels for solvent update and particle-solvent interaction

Solvent kernel

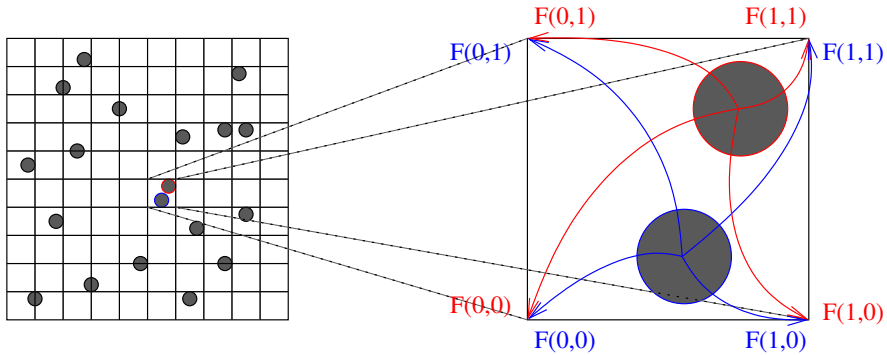
- One thread per lattice node
- Mode space transformation into registers
- Relax modes, thermalize modes, apply (external) forces, normalization
- Thermalization uses a dedicated Gaussian RNG per lattice node
- back transformation into velocity space and streaming step with periodic boundaries at once

Lattice-Boltzmann on GPUs

- Velocity space resides in (large but “slow”) global memory
- Double buffering to avoid race conditions during streaming
- Memory layout is optimized for coalesced access to the fluid velocities
- Kernels for solvent update and particle-solvent interaction

Particle-solvent interaction kernel

- One thread per particle
- Interpolate fluid velocity at the position of the particle (registers)
- Fluid force acting on the particle
- Distribute reactio back to lattice nodes (requires atomic operations)



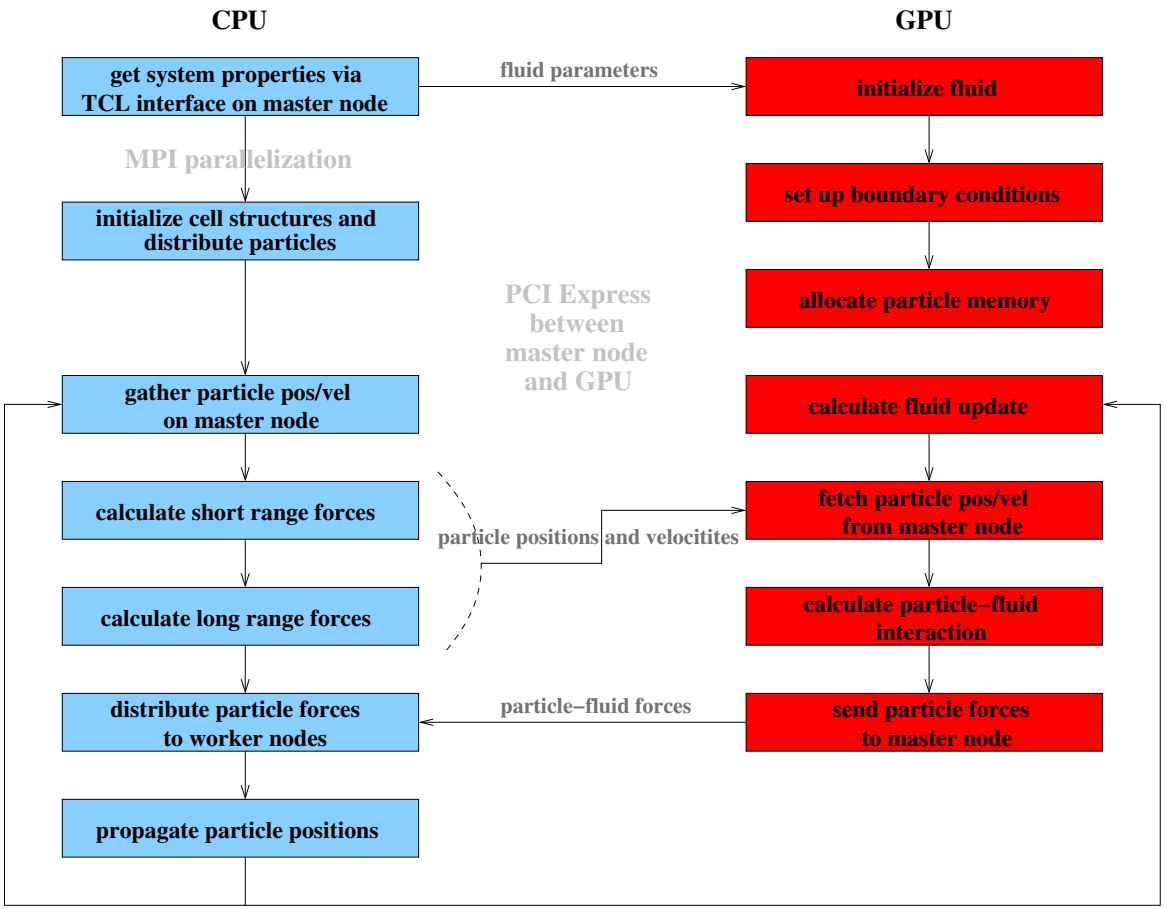
Hydrodynamics in confined geometries

movie

Hydrodynamic (long range) interactions

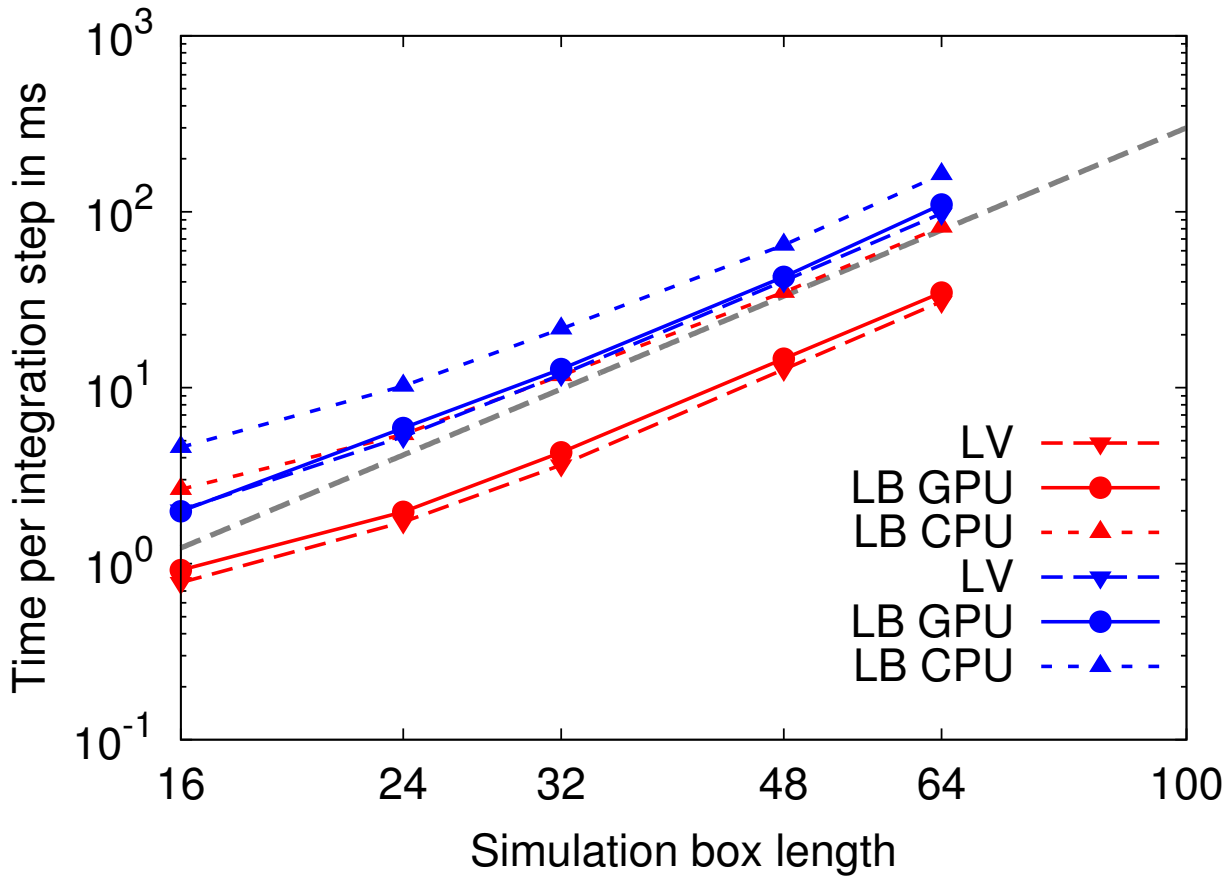
movie

Parallel execution scheme



Performance benchmark: Suspension

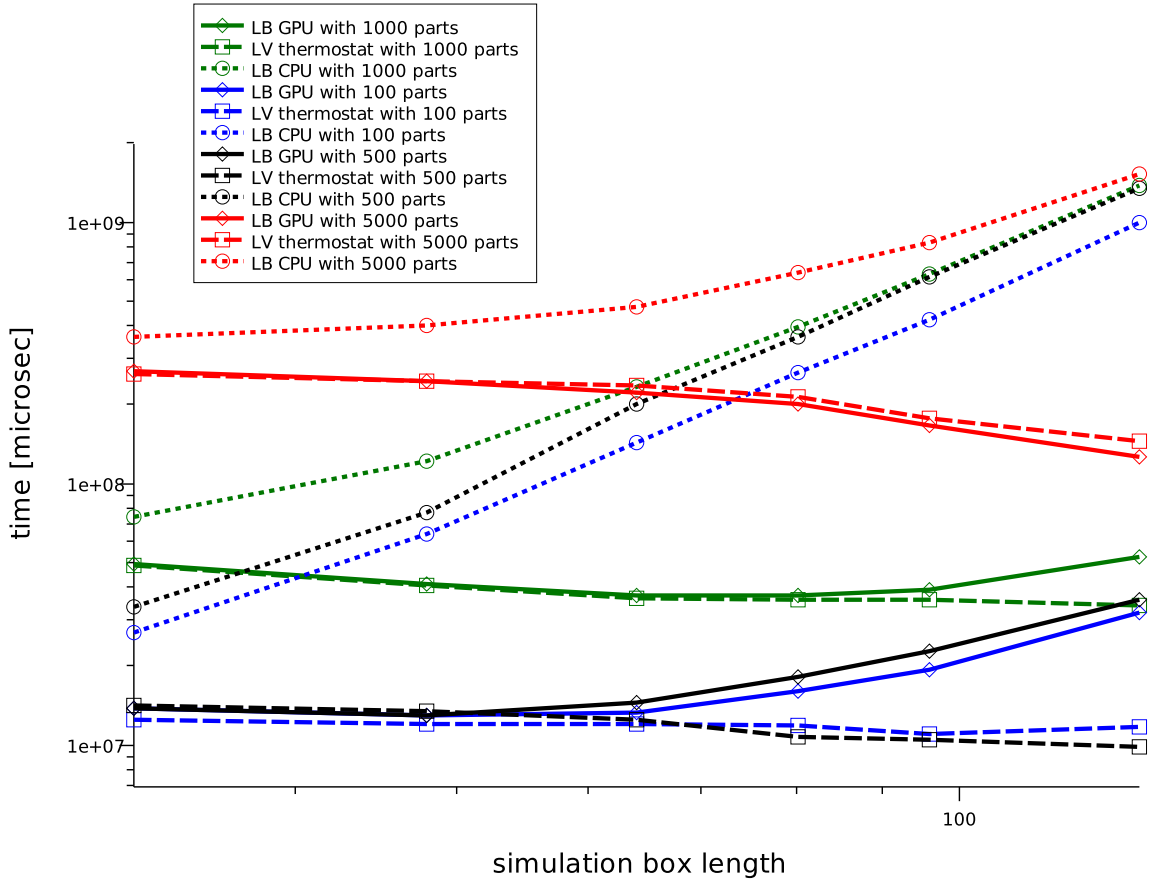
Particle interaction: Lennard-Jones radius 2.5 on INTEL XEON Quadcore@2.4GHz + NVIDIA Tesla C2050



LB cost negligible for densities > 0.3 due to interleaving

Performance benchmark: Electro-osmotic flow

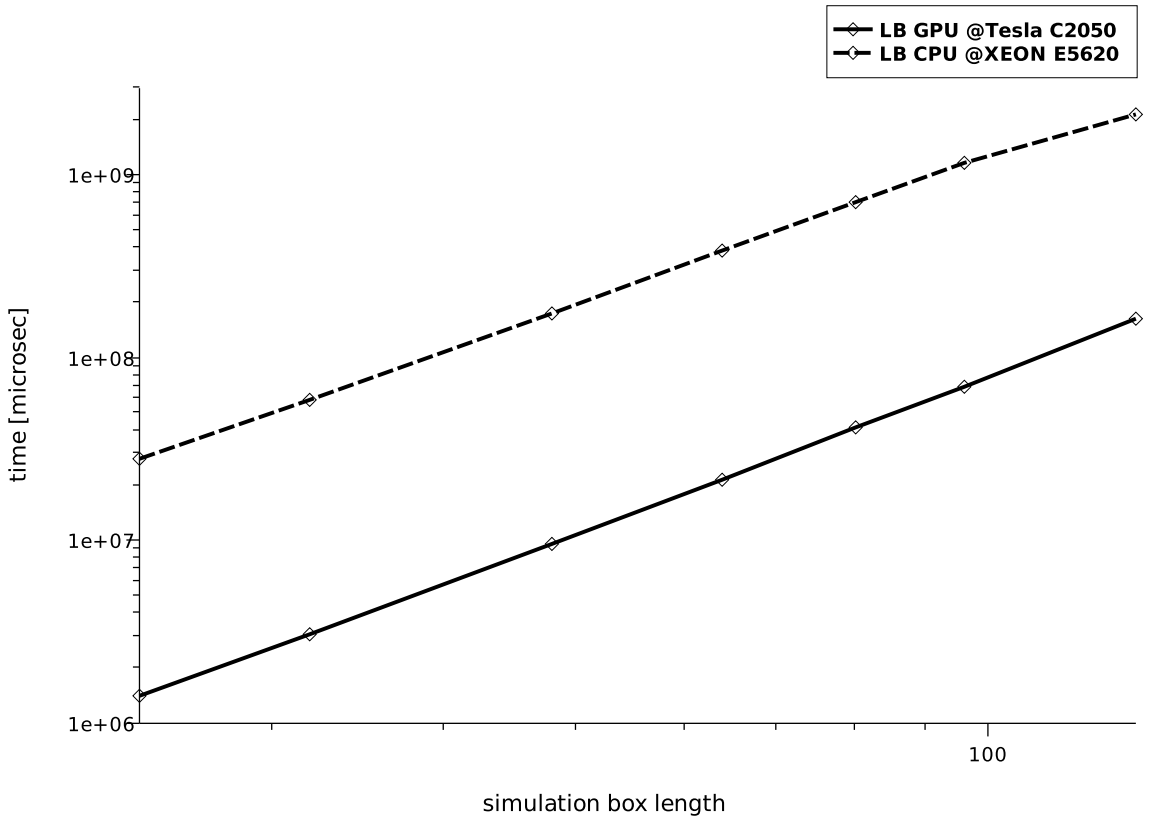
Particle interaction: L-J radius 1.1225, P3M+ELC, constraints, fluid boundaries on INTEL XEON Quadcore@2.4GHz + NVIDIA Tesla C2050



LB cost negligible for > 100 particles due to interleaving

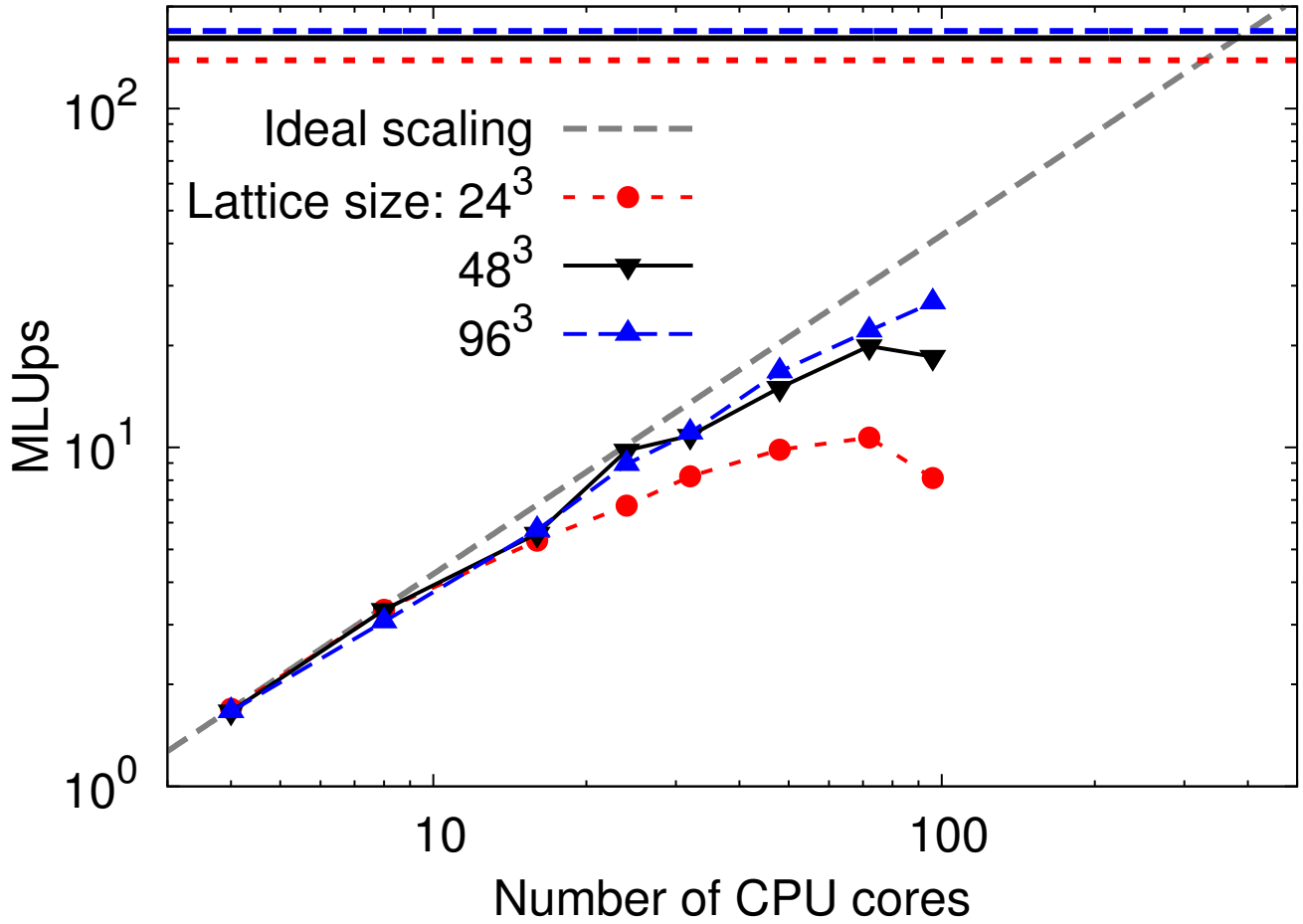
Performance benchmark: Thermalized fluid

two INTEL XEON Quadcore@2.4GHz + NVIDIA Tesla C2050



Thermalized fluid $\Rightarrow$ 50x

Performance benchmark: Thermalized fluid



CPU cluster can not achieve single GPU performance!

Lattice Boltzmann with ESPResSo

- `lbfluid (cpu/gpu) tau 0.1 agrid 1.0 visc 0.8`

Lattice Boltzmann with ESPResSo

- `lbfluid (cpu/gpu) tau 0.1 agrid 1.0 visc 0.8`
- `lbfluid friction 10.0 ext_force $fx $fy $fz`

Lattice Boltzmann with ESPResSo

- `lbfluid (cpu/gpu) tau 0.1 agrid 1.0 visc 0.8`
- `lbfluid friction 10.0 ext_force $fx $fy $fz`
- Ratio of

  `setmd time_step / (lbfluid tau)`

Lattice Boltzmann with ESPResSo

- `lbfluid (cpu/gpu) tau 0.1 agrid 1.0 visc 0.8`
- `lbfluid friction 10.0 ext_force $fx $fy $fz`
- Ratio of

  `setmd time_step / (lbfluid tau)`
- Parameter range e.g.

```
lbfluid agrid 0.5
lbfluid tau $tau/8
```

Lattice Boltzmann with ESPResSo

- `lbfluid (cpu/gpu) tau 0.1 agrid 1.0 visc 0.8`
- `lbfluid friction 10.0 ext_force $fx $fy $fz`
- Ratio of

  `setmd time_step / (lbfluid tau)`
- Parameter range e.g.

  ```
  lbfluid agrid 0.5
  lbfluid tau $tau/8
  ```
- ```
  lbboundary wall normal -1 0 0 dist [expr -$box_l_x+1]
  lbboundary wall normal 1 0 0 dist [expr +1]
  sphere cyclinder rhomboid
  ```
  ```

Lattice Boltzmann with ESPResSo

- `lbfluid (cpu/gpu) tau 0.1 agrid 1.0 visc 0.8`
- `lbfluid friction 10.0 ext_force $fx $fy $fz`
- Ratio of

  `setmd time_step / (lbfluid tau)`
- Parameter range e.g.

```
lbfluid agrid 0.5
lbfluid tau $tau/8
```

- `lbboundary wall normal -1 0 0 dist [expr -$box_l_x+1]`
  `lbboundary wall normal 1 0 0 dist [expr +1]`
  `sphere cyclinder rhomboid`
- `lbfluid print (vtk) velocity/boundary $filename`

HI + confined geometries

movie

Conclusions

- GPU code on Tesla C2050 up to **50** times faster than CPU code on two XEON E7620
- Get hydrodynamics for **free** or calculate pure hydrodynamic faster than on a **CPU Cluster**
- D. Roehm and A. Arnold, EPJ -ST 210 (89-100), 2012

Outlook

- GPU-accelerated Coulomb solver (P3M, ELC, MMM...)
- GPU-accelerated Poisson-Boltzmann solver, included in ESPResSo (G. Rempfer)