Advanced, hardware-oriented Shallow Water simulations based on the Lattice-Boltzmann Method

Markus Geveler, Dirk Ribbrock

Inst. for Applied Mathematics
TU Dortmund University of Technology, Germany
markus.geveler@math.tu-dortmund.de

CMWR12: Advances in Heterogeneous Computing for Water Resources
Urbana-Champaign, June 20 2012

technische universität dortmund fakultät für mathematik

This work is...

...one outcome of the German SKALB project `www.skalb.de`

Numerical simulation of (technical) water flow

- In many cases: we are only interested in the (re-)construction of the free surface (or depth-averaged velocities)
    - coastal engineering (tidal flow, dam-break)
    - hydraulic engineering (bridges, dams, channels, canals, levees)
    - environmental flow (atmoshphere, river, ocean)
    - other (graphics, entertainment)

Problem: overall productivity of simulation (efficiency)

Motivation

Efficient yet stable and flexible simulation of free surface

- *Include many environmental and technical factors that interact with a fluid*

Impacts on fluid flow - feature list

- friction induced by solid surface / walls
- acceleration induced by surface slope / moving walls
- drying and wetting of the solid surface
- friction induced by atmospheric layer (wind)
- arbitrary geometry (solid stationary objects in the fluid)
- friction / acceleration / displacement induced by moving rigid objects (Fluid-Structure Interaction)
- pollution of the fluid (pollutant transport)

Motivation

Efficient yet stable and flexible simulation of free surface

- *Include many environmental and technical factors that interact with a fluid*

Impacts on fluid flow - feature list

- friction induced by solid surface / walls
- acceleration induced by surface slope / moving walls
- drying and wetting of the solid surface
- friction induced by atmospheric layer (wind)
- arbitrary geometry (solid stationary objects in the fluid)
- friction / acceleration / displacement induced by moving rigid objects (Fluid-Structure Interaction)
- pollution of the fluid (pollutant transport)

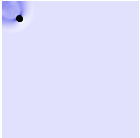
Motivation

Efficient yet stable and flexible simulation of free surface

- *Include many environmental and technical factors that interact with a fluid*

Impacts on fluid flow - feature list

- friction induced by solid surface / walls
- acceleration induced by surface slope / moving walls
- drying and wetting of the solid surface
- friction induced by atmospheric layer (wind)
- arbitrary geometry (solid stationary objects in the fluid)
- friction / acceleration / displacement induced by moving rigid objects (Fluid-Structure Interaction)
- pollution of the fluid (pollutant transport)



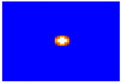
Motivation

Efficient yet stable and flexible simulation of free surface

- *Include many environmental and technical factors that interact with a fluid*

Impacts on fluid flow - feature list

- friction induced by solid surface / walls
- acceleration induced by surface slope / moving walls
- drying and wetting of the solid surface
- friction induced by atmospheric layer (wind)
- arbitrary geometry (solid stationary objects in the fluid)
- friction / acceleration / displacement induced by moving rigid objects (Fluid-Structure Interaction)
- pollution of the fluid (pollutant transport)



Introduction

Two key aspects of efficiency in Scientific Computing

- *All levels of hardware-parallelism have to be taken into account*
  - vectorisation on a single core (SIMD)
  - multi-core / many-core (CPUs + GPUs)
  - distributed memory
  - heterogeneous resources
    - on node-level: Hybrid CPU + GPU (common)
    - on chip-level: E.g. mobile SoCs / NVIDIA project Denver: high-performance ARM cores + GPU + DSP + ...**(new)**
- *All levels of the numerical method have to be taken into account*
  - discretization of time and space
  - *stabilisation*, linearization
  - numerically efficient solution pipeline
  - fitting to target hardware

→ Lattice-Boltzmann Method applied to 2D Shallow Water Equations

- SWE: Adjustment on the model level and usage of flexible numerical method
- LBM: chosen for better data locality, well suited for many applications, good fitting for (fine-grained) parallel hardware, flexible, extensible

SWE + LBM dissemination

SWE + stabilised LBM on heterogeneous systems

- bed slope force term stabilisation
- allowing dry-states
- 'depth-averaged' moving boundaries and FSI
- GPU / CELL / MPI / PThreads multi-core/ hybrid implementations
- $\rightarrow$ Geveler et al. 2009–2012 / SKALB project

SWE + pollutant transport

- coupled SWE / Convection Diffusion Equations
- hybrid MPI + CUDA
- heterogeneity on a chip: first LBM benchmarks on an ARM-based cluster
- $\rightarrow$ now

Shallow Water Flows

Full featured interactive laminar flow

Inhomogeneous SWE

$$\frac{\partial h}{\partial t} + \frac{\partial (hu_j)}{\partial x_j} = 0 \quad \text{and} \quad \frac{\partial hu_i}{\partial t} + \frac{\partial (hu_i u_j)}{\partial x_j} + g\frac{\partial}{\partial x_i}(\frac{h^2}{2}) = S_i^{\text{bed}} + S_i^{\text{wind}}$$

$$S_i^{\text{bed}} = -g\left(h\frac{\partial b}{\partial x_i} + n_b^2 h^{-\frac{1}{3}} u_i \sqrt{u_j u_j}\right)$$

$$S_i^{\text{wind}} = (\rho_\alpha 10^{-3} \times (0.75 + 0.0067\sqrt{w_1^2 + w_2^2}))(w_1\sqrt{w_1^2 + w_2^2})$$

Shallow Water flows and pollutant transport

Explosion event: 'Deep Water Horizon' oil exploration platform in 2010

picture: greenpeace.org

- approx. $10,000km^2$ of oil spill in the Gulf of Mexico
- 35,000 to 60,000 Barrels of oil spill per day
- oil spill drift hit US coast causing massive damage to people, environment and economy

Shallow Water flows and pollutant transport

Couple SWE with Convection-Diffusion Equation

Inhomogeneous SWE + Convection-Diffusion

$$\frac{\partial h}{\partial t} + \frac{\partial (hu_j)}{\partial x_j} = 0 \quad \text{and} \quad \frac{\partial hu_i}{\partial t} + \frac{\partial (hu_i u_j)}{\partial x_j} + g\frac{\partial}{\partial x_i}(\frac{h^2}{2}) = S_i^{\mathsf{bed}} + S_i^{\mathsf{wind}}$$

$$\frac{\partial hc}{\partial t} + \frac{\partial (hu_j c)}{\partial x_j} = \frac{\partial}{\partial x_j}\left(Dh\frac{\partial c}{\partial x_j}\right) + S^{\mathsf{poll}}$$

$$S_i^{\mathsf{bed}} = -g\left(h\frac{\partial b}{\partial x_i} + n_b^2 h^{-\frac{1}{3}} u_i \sqrt{u_j u_j}\right)$$

$$S_i^{\mathsf{wind}} = (\rho_\alpha 10^{-3} \times (0.75 + 0.0067\sqrt{w_1^2 + w_2^2}))(w_1\sqrt{w_1^2 + w_2^2})$$

$$S^{\mathsf{poll}} = -Khc + S_0 h$$

Lattice-Boltzmann essentials

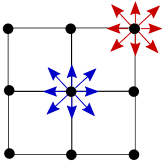
Modification on the discrete and model levels and adjustment to hardware

- discrete microscopic level: simplified particle movement
- continuum macroscopic level: gives correct description of fluid

Fluid motion is determined by distribution functions related to fixed spatial directions

Lattice velocities D2Q9

$$\mathbf{e}_\alpha = \begin{cases} (0,0) & \alpha = 0 \\ e(\cos\frac{(\alpha-1)\pi}{4}, \sin\frac{(\alpha-1)\pi}{4}) & \alpha = 1,3,5,7 \\ \sqrt{2}e(\cos\frac{(\alpha-1)\pi}{4}, \sin\frac{(\alpha-1)\pi}{4}) & \alpha = 2,4,6,8, \end{cases}$$



Lattice-Boltzmann essentials

Fluid behaviour is determined by a collision operator on these functions

Linear approximation to collision

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t) + Q(f_\alpha, f_\beta), \qquad \beta = 1, ..., k \ .$$

$$f_\alpha^{\text{temp}}(\mathbf{x}, t) = f_\alpha(\mathbf{x}, t) - \frac{1}{\tau}(f_\alpha - f_\alpha^{eq})$$

Plugin of governing equations

$$f_\alpha^{\text{eq}} = \begin{cases} h(1 - \frac{5gh}{6e^2} - \frac{2}{3e^2} u_i u_i) & \alpha = 0 \\ h(\frac{gh}{6e^2} + \frac{e_{\alpha i} u_i}{3e^2} + \frac{e_{\alpha j} u_i u_j}{2e^4} - \frac{u_i u_i}{6e^2}) & \alpha = 1, 3, 5, 7 \\ h(\frac{gh}{24e^2} + \frac{e_{\alpha i} u_i}{12e^2} + \frac{e_{\alpha j} u_i u_j}{8e^4} - \frac{u_i u_i}{24e^2}) & \alpha = 2, 4, 6, 8 \end{cases}$$

Lattice-Boltzmann essentials

Source terms in the model become cummulative terms along each direction

Forces in LBM

$$f_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = f_\alpha(\mathbf{x},t) - \frac{1}{\tau}(f_\alpha - f_\alpha^{eq}) + \frac{\Delta t}{6e^2} e_{\alpha i} S_i, \ \alpha = 0, \ldots, 8.$$

Recovery of waterdepth, velocity

LBM extraction

$$h(\mathbf{x},t) = \sum_\alpha f_\alpha(\mathbf{x},t) \quad \text{and} \quad u_i(\mathbf{x},t) = \frac{1}{h(\mathbf{x},t)} \sum_\alpha e_{\alpha i} f_\alpha,$$

Boundary conditions: simple - define at microscopic layer

Pollutant transport in the LBM

General algorithm

- two solvers: one for the flow / pollutant each
- both share the velocity fields
- alternate both solvers

step 1: add a proper collision operator for the pollutant solver

Pollutant solver collision

$$g_\alpha^{\text{temp}}(\mathbf{x},t) = g_\alpha(\mathbf{x},t) - \frac{1}{\tau_{\text{poll}}^{\text{var}}}(g_\alpha - g_\alpha^{eq})$$

$$\tau_{\text{poll}}^{\text{var}} = 1/2 + h(\mathbf{x},t) \times (\tau_{\text{poll}} - 1/2)$$

$$g_\alpha^{\text{eq}} = \begin{cases} c(h - 5/9) & \alpha = 0 \\ c(1/9 + \frac{h}{3e^2} e_{\alpha i} u_i) & \alpha = 1,3,5,7 \\ c(1/36 + \frac{h}{12e^2} e_{\alpha i} u_i) & \alpha = 2,4,6,8 \end{cases}$$

Pollutant transport in the LBM

General algorithm

- two solvers: one for the flow / pollutant each
- both share the velocity fields
- alternate both solvers

step 2: add the force

Pollutant solver force

$$g_\alpha^{\text{temp}}(\mathbf{x},t) \leftarrow g_\alpha^{\text{temp}}(\mathbf{x},t) + \Delta t \omega_\alpha S^{\text{poll}}$$

step 3: add an extraction module

Pollutant solver extraction

$$c(\mathbf{x},t) = \sum_\alpha \frac{1}{h(\mathbf{x},t)} g_\alpha(\mathbf{x},t)$$

Implementation: A word on HPC software

HONEI

- primary design goals: abstract from target-hardware, exploit all resources in a node
- support natively:
  - x86 SIMD (handcrafted SSE2 using intrinsics)
  - x86 multi-core (pthreads)
  - distributed memory clusters (MPI)
  - NVIDIA GPUs (CUDA 3) + multiple GPUs + hybrid computations
  - Cell BE (libspe2)
- unittest + benchmark frameworks, visualisation
- build system to create SPE kernels for Cell BE, CUDA-kernels
- thread-management, MPI
- support for RTTI and exception-handling, memory transfers
- RPC system to call SPE programs from the PPE
- templates to facilitate development of new callable SPE functions and registering them with the RPC system
- automatic job scheduling
- ...

Implementation: packed lattice

One datastructure fits all architectures

- compress lattice: stationary obstacles
- store contiguously in memory
- instationary obstacles (moving solids): colouring
- cut domain in one dimension
- synchronisation is simple

Implementation: GPU

Shared memory: general

- cache lattice velocities $\mathbf{e}_{\alpha i}$

Memory arbitration

- external device memory management
- works 'behind the scenes'
- transfers only if necessary

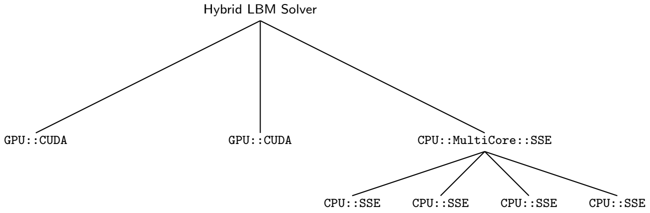
Complex kernels (force, FSI)

- avoid branch-divergence
- cache factors in force terms
- cache extrapolation weights

Implementation: hybrid compute nodes

Composition of hybrid solver: example

- 3 patches of domain initially
- 1 patch subdivided for the `MultiCore backend`
- each of the four MC-threads use the SSE backend
- resulting in utilisation of two GPUs and all cores of the (quadcore) CPU



Results

Sample results

- convection/diffusion: cavity flow driven pollution
- instant pollutant discharge
- high local concentration (but smooth: 2D Gaussian)
- $\rightarrow$ video

Performance results

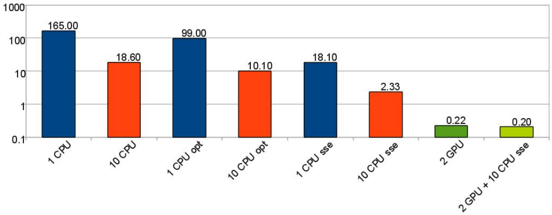
Always beware of scaling analysis alone! – Perfect scaling almost always means bad serial code (or slow single cores/nodes/devices)!

Single node performance

- 2 6-core Westmeres and 2 NVIDIA Tesla GPUs
- hybrid MPI+SSE/core+CUDA solver (flow solver + full force terms)
- Ultra high res. (7000×7000) lattice sites; metrics: time per timestep
- Load balancing
  - homogeneous: uniform
  - hybrid: 2 $\times$ 46% (2 GPUs) + 10 $\times$ 0.8% (10 CPU cores)
- Optimisation levels (CPU)
  - 'CPU': no optimisation concerning vectorisation
  - 'CPU opt': optimisations concerning compiler vectorisation
  - 'CPU sse': all vectorisation handcrafted (SSE4)

Performance results

Single node performance

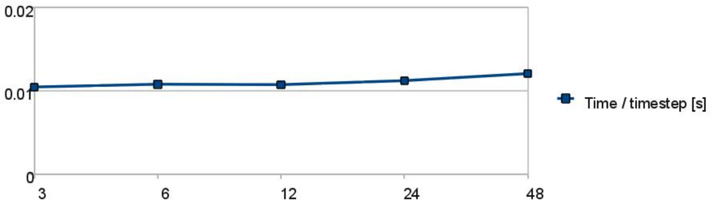


- → optimisation concerning vectorisation is crucial for CPU performance
- → in some cases: compiler unable to vectorise kernel loops at all (bed force term)
- → good serial performance only granted by organising loops / register usage by hand
- **→ hybrid only pays off, if CPU kernels are reasonably optimised**

Performance results

Scaling on a GPGPU cluster: CINECA IBM-PLX

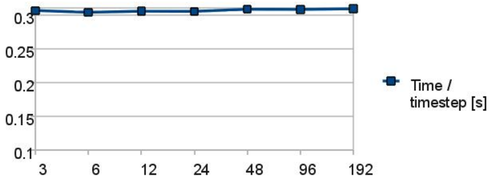
- 2 6-core Westmeres and 2 NVIDIA Tesla GPUs per node
- Infiniband
- full features (flow + pollutant)
- $2^l * (2000 \times 2000)$ lattice sites and $3 * (2^l)$ nodes on refinement level l

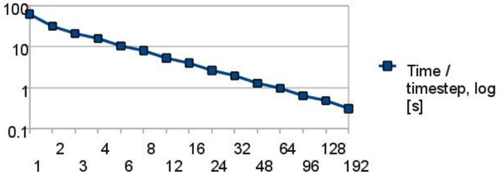


Results

Outlook: ARM-based cluster - Tibidabo (Montblanc-prototype at BSC)

- flow solver only
- 1 NVIDIA Tegra 2 SoC per node (2 × Cortex A9 at 1.2 GHz)
- 250×250 to 2000×2000 lattice sites (weak) / 2000×2000 (strong)





Conclusions

LBM extension for pollutant transport

- another good example for the extensibility of an LBM-based SWE solver
- well suited for high diffusivity coefficients
- very well suited for multi-level parallelism due to data locality / simple comm. pattern

Heterogeneity

- on node is working well (adding CPU processes still slightly beneficial)
- on chip: first step performed: ARM chips can be incorporated easily
- future work: use ULP GPUs on SoCs

Acknowledgements

Supported by BMBF, *HPC Software für skalierbare Parallelrechner: SKALB project 01IH08003D*.

Thanks to BSC for hardware access.
Thanks to all contributors to HONEI.

This work was granted access to the HPC resources of CINECA made available within the Distributed European Computing Initiative by the PRACE-2IP, receiving funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number RI-283493.