On Optimized Implementations of the Lattice Boltzmann Method on Contemporary High Performance Architectures

Bachelor Thesis

Stefan Donath

Advisors: Dr. Gerhard Wellein, RRZE
Prof. Dr. Ulrich Rüde, LSS
Outline

- Introduction
  - Lattice Boltzmann Method
  - LBMKernel
- Standard Implementation
  - Theoretical considerations and validation
  - Effect of Blocking
- Optimized Implementation
  - Theoretical considerations and validation
  - Confirmation by Profiling
  - Comparison to standard implementation
Outline

- Introduction
Introduction

- Subject of Bachelor-Thesis:
  To find out which Optimization Techniques are really efficient, and why

- Object of Investigation:
  LBMKernel (bases on Lattice Boltzmann Method)
Examples of flows calculated using Lattice Boltzmann flow solvers

Figures by courtesy of EXA GmbH, LS CAB-Braunschweig, Thomas Zeiser

Stefan Donath
Friedrich-Alexander University of Erlangen/Nuremberg, Germany
Introduction

Lattice Boltzmann Method

Numerical Method for Simulation of Fluids

Stream-Collide (Pull-Method)

Get the distributions from the neighboring cells in the source array and store the relaxed values to one cell in the destination array.

Collide-Stream (Push-Method)

Take the distributions from one cell in the source array and store the relaxed values to the neighboring cells in the destination array.

D3Q19 model:

Two Grids:

Compressed Grid:
Introduction

**LBMKernel**

- “LBMKernel“
  - Fortran
  - Lattice Boltzmann Method
  - Push-Method (Collide-Stream)
  - D3Q19
  - Two Grids
  - Variations:
    - Compressed Grid
      (will be presented by K. Iglberger at 4:05 pm)
    - 1D-/3D-Blocking
Introduction

LBMKernel – Code Structure

double precision f(0:xMax+1,0:yMax+1,0:zMax+1,0:18,0:1)
do z=1,zMax
  do y=1,yMax
    do x=1,xMax
      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
    enddo
  enddo
enddo
Outline

- Standard Implementation
Standard Implementation

- Standard Array-Layout: $F(i, x, y, z, t)$
  - "Array-of-Structures"
- Collision optimized
- 2 cache lines per Lattice Site Update (LUP) in read process
Standard Implementation
Theoretical considerations

- with MemPHis:
  - Optimal read access: 2 cache lines per LUP
  - Bad write access: 19 stores in 19 cache lines
    But: Depending on system size some of them are already/still in cache
Standard Implementation

Theoretical considerations

- 18 write accesses on one cell from 3 different z-layers
  \[ Mem_k \approx 3 \cdot (N_i + 2) \cdot (N_j + 2) \cdot \delta \cdot 2.19 \cdot 8 \text{ Byte} \]

- 8 write accesses on one cell from 3 different rows
  \[ Mem_j \approx 3 \cdot (N_i + 2) \cdot \delta \cdot 2.19 \cdot 8 \text{ Byte} \]
Standard Implementation Validation

- Performance of LIJK-Layout (P4, 3.0 GHz)

![Performance Graph]

- Array of Structures

Stefan Donath
Friedrich-Alexander University of Erlangen/Nuremberg, Germany
Standard Implementation
Effect of Blocking

- Performance of LIJK-Layout (P4, 3.0 GHz)
Outline

- Optimized Implementation
Optimized Implementation

- Optimized Array-Layout: \( F(x,y,z,i,t) \)
  - "Structure-of-Arrays"
  - stride-1-access on \( x \) (inner loop)
  - 19 cache lines per 16 LUPs in read and write process
  - 1 cache miss each 16\(^{th}\) memory access
Optimized Implementation

Validation

- Performance on Pentium 4, 3.0 GHz

![Graph showing performance on different array sizes for Structure of Arrays and Array of Structures.]
Optimized Implementation

Confirmation by Profiling

- Theory and Reality for $300^3$, 50 time steps
- Theory:
  - 1 cache miss each 16\textsuperscript{th} memory access equals approximately $3.206 \cdot 10^9$ L3-cache misses
- Measurements (Itanium 2):
  - L3\_DATA\_READ\_MISSES: $3.503 \cdot 10^9$
  - Difference of 4.4\% (side effects on boundaries)
  - $\Rightarrow$ 1 cache miss per 15.3 memory accesses
  - Cache Hit-Ratio: 93.8 \%
Optimized Implementation

Structure of Arrays vs. Array of Structures

Pentium 4, 3.0 GHz (Theoretical Perf. Limit by Mem-BW: 14 MLUPS)

- Structure of Arrays
- Array of Structures
- Array of Structures, 3D

Stefan Donath
Friedrich-Alexander University of Erlangen/Nuremberg, Germany
Optimized Implementation

Structure of Arrays vs. Array of Structures

Nocona, 3.4 GHz  (Theoretical Perf. Limit by Mem-BW: 11.6 MLUPS)

- Structure of Arrays
- Array of Structures
- Array of Structures, 3D, CG

Stefan Donath

Friedrich-Alexander University of Erlangen/Nuremberg, Germany
Optimized Implementation
Structure of Arrays vs. Array of Structures

Itanium 2, 1.4 GHz  (Theoretical Perf. Limit by Mem-BW: 14 MLUPS)

- Structure of Arrays
- Array of Structures
- Array of Structures, 3D, CG

Friedrich-Alexander University of Erlangen/Nuremberg, Germany
Summary and Outlook

- Optimization for Lattice Boltzmann Method in 3D (D3Q19)

  **Summary:**
  - Most performance benefit by adapting layout of data in memory on underlying architecture
    - Spatial blocking techniques not necessary
  - All results can be explained by reasonable theoretical considerations

  **Outlook:**
  - Ongoing research: Fine-tuning on inner loop (app. + 1 MLUPS)
  - Optimized Implementation can be enhanced by Compressed Grid technique
  - Temporal blocking techniques (4-way blocking)
    - Both techniques will be presented in detail by Klaus Iglberger today at 4.05 pm
References / Thanks

Department of Computer Science 10, System Simulation (LSS)
http://www10.informatik.uni-erlangen.de/en/

HPC@Regional Computing Center of Erlangen (RRZE)
http://www.rrze.uni-erlangen.de/dienste/arbeiten-rechnen/hpc/

Acknowledgement:
Dr. Gerhard Wellein (RRZE)
Prof. Dr. Ulrich Rüde (LSS)

Stefan Donath:
stefan.donath@rrze.uni-erlangen.de
http://www10.informatik.uni-erlangen.de/de/Publications/Theses/Donath.pdf

Other presentations:
Ben Bergen, LSS Monday, 10.30 am
Andreas Hauck, LSE Sunday, 04.45 pm
07.50 pm
Klaus Iglberger, LSS Monday, 03.45 pm
14.05 pm
Prof. Ulrich Rüde, LSS Monday, 03.45 pm