LBM BASED FLOW SIMULATION USING GPU COMPUTING PROCESSOR

Frédéric Kuznik, frедерic.kuznik@insa-lyon.fr
Framework

- Introduction
- Hardware architecture
- CUDA overview
- Implementation details
- A simple case: the lid driven cavity problem
Presentation of my activities

CETHIL: Thermal Sciences Center of Lyon

Current work: simulation of fluid flows with heat transfer

Use/development of Thermal LBM\(^1,2\)  
LBM computation using GPU\(^3\)

Objective: coupling efficiently the two approaches

3. LBM based flow simulation using GPU computing processor, Computers & Mathematics with Applications, under review
Introduction

Due to the market (realtime and high definition 3-D graphics), GPU has evolved into highly parallel, multithreaded, multi-core processor.

(from CUDA Programming Guide 06/07/2008)
GFLOPS=$10^9$ floating point operations per second
Introduction

The bandwidth of GPU has also evolved for faster graphics purpose.

(from CUDA Programming Guide 06/07/2008)
Previous works using GPU & LBM

- GPU Cluster for high performance computing (Fan et al. 2004): 32 nodes cluster of GeForce 5800 ultra
- LB-Stream Computing or over 1 Billion Lattice Updates per second on a single PC (J. Tölke ICMMES 2007): GeForce 8800 ultra
Hardware

- Commercial graphics card (can be included in a desktop PC)
- 240 processors running at 1.35GHz
- 1 Go DDR3 with 141.7GB/s bandwidth
- Theoretical peak at 1000GFLOPS
- Price around 500€

nVIDIA GeForce GTX 280
GPU Hardware Architecture

◊ 15 texture processors (TPC)

◊ 1 TPC is composed of 2 streaming multiprocessors (SM)

◊ 1 SM is composed of 8 streaming processors (SP) and 2 special functions units

◊ Each SM has 16kB of shared memory
GPU Programming Interface

NVIDIA® CUDA™ C language programming environment (version 2.0)

Definitions:

- **Device=GPU**
- **Host=CPU**
- **Kernel** = function that is called from the host and runs on the device

A CUDA kernel is executed by an array of threads
GPU Programming Interface

- A kernel is executed by a grid of thread block
- 1 thread block is executed by 1 multiprocessor
- A thread block contains a maximum of 1024 threads
- Threads are executed by processors within a single multiprocessor

=> Threads from different blocks cannot cooperate
Kernel memory access

- Registers
- Shared memory: on-chip (fast), small (16kB)
- Global memory: off-chip, large
- The host can read or write global memory only.
LBM ALGORITHM

- **Step 1:** Collision (Local – 70% computational time)

- **Step 2:** Propagation (Neighbors – 28% computational time)

- + Boundary conditions

(from BERNSDORF J., How to make my LB-code faster – software planning, implementation and performance tuning, ICMMES’08, Netherlands)
Implementation details

1- Decomposition of the fluid domain into a lattice grid
2- Indexing the lattice nodes using thread ID
3- Decomposition of the CUDA domain into thread blocks
4- Execution of the kernel by the thread blocks
Pseudo-code

Combine collision and propagation steps:

for each thread block
  for each thread
    load $f_i$ in shared memory
    compute collision step
    do the propagation step
  end
end

Exchange informations across boundaries
Application: 2D lid driven cavity

Problem description

u = 0, v = 0

x

y

D2Q9 MRT model of d’Humières (1992)
Results Re=1000

Reference data from Erturk et al. 2005:

Vertical velocity profil for \( x=0.5 \)

Horizontal velocity profil for \( y=0.5 \)
## Performances

<table>
<thead>
<tr>
<th>Mesh grid size</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>256²</td>
<td>71.0</td>
<td>71.0</td>
<td>71.1</td>
<td>71.1</td>
<td>70.9</td>
</tr>
<tr>
<td>512²</td>
<td>284.2</td>
<td>282.7</td>
<td>284.3</td>
<td>284.3</td>
<td>283.7</td>
</tr>
<tr>
<td>1024²</td>
<td>346.4</td>
<td>690.2</td>
<td>953.0</td>
<td>997.5</td>
<td>1007.9</td>
</tr>
<tr>
<td>2048²</td>
<td>240.0</td>
<td>583.0</td>
<td>803.3</td>
<td>961.3</td>
<td>1001.5</td>
</tr>
<tr>
<td>3072²</td>
<td>289.3</td>
<td>572.4</td>
<td>845.2</td>
<td>941.2</td>
<td>974.2</td>
</tr>
</tbody>
</table>

Performances in MLUPS
Performances: CPU comparisons

CPU = Pentium IV, 3.0 GHz
Conclusions

- The GPU allows a gain of about 180 for the calculation time

- Multiple GPU server exists: NVIDIA TESLA S1070 composed of 4 GTX 280 (theoretical gain 720)

- The evolution of GPU is not finished!

- But using double precision floating point, the gain falls to 20!
Outlooks

A workgroup concerning LBM and GPU

- 1 master student with Prof. Bernard TOURANCHEAU (ENS Lyon – INRIA) and a PhD student in 2009
- 1 master student with Prof. Eric Favier (ENISE – DIPI)

A workgroup concerning Thermal LBM ...?

Everybody is welcome to join the workgroup!
THANK YOU FOR YOUR ATTENTION

frederic.kuznik@insa-lyon.fr