LBM BASED FLOW SIMULATION USING GPU COMPUTING PROCESSOR

Frédéric Kuznik, **frederic.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]

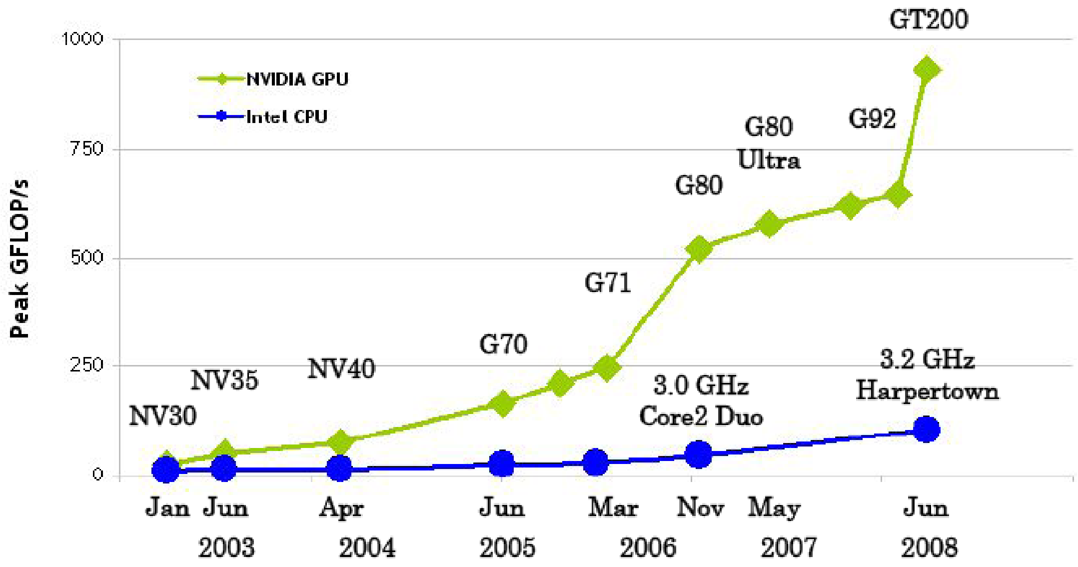
Objective: coupling efficiently the two approaches

1 A double population lattice Boltzman method with non-uniform mesh for the simulation of natural convection in a square cavity, Int. J. Heat and Fluid Flow, vol. 28(5), pp. 862-870, 2007

2 Numerical prediction of natural convection occuring in building components: a double population lattice Boltzmann method, Numerical Heat Transfer A, vol. 52(4), pp. 315-335, 2007

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

Introduction

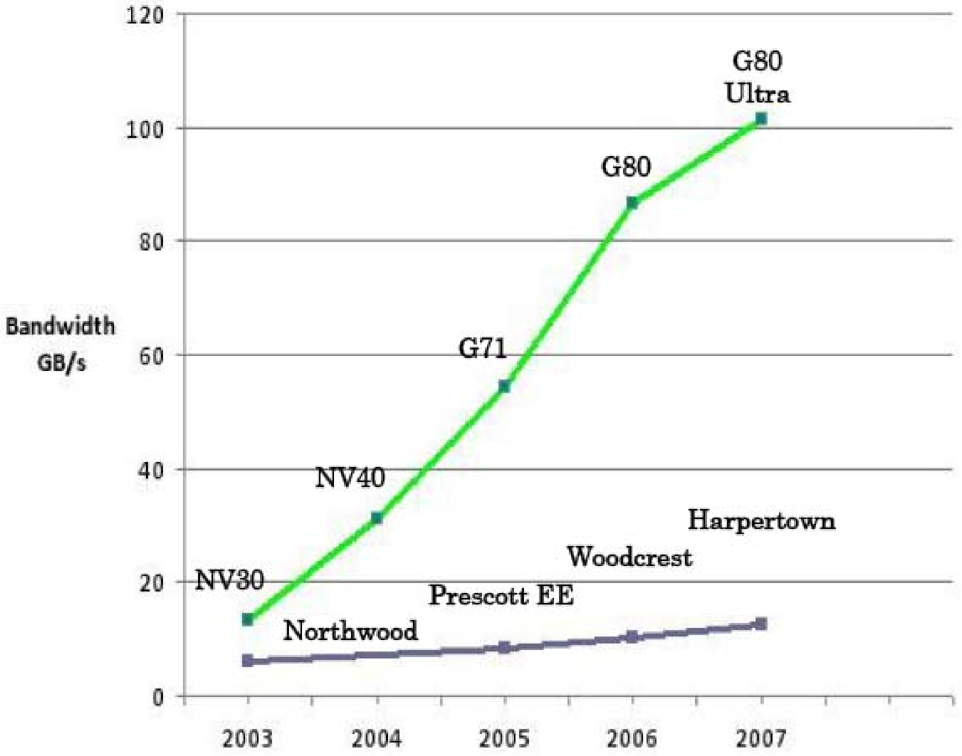


GT200 = GeForce GTX 280	G71 = GeForce 7900 GTX	NV35 = GeForce FX 5950 Ultra
G92 = GeForce 9800 GTX	G70 = GeForce 7800 GTX	NV30 = GeForce FX 5800
G80 = GeForce 8800 GTX	NV40 = GeForce 6800 Ultra	

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

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

Introduction



(from *CUDA Programming Guide 06/07/2008*)

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

Previous works using GPU & LBM

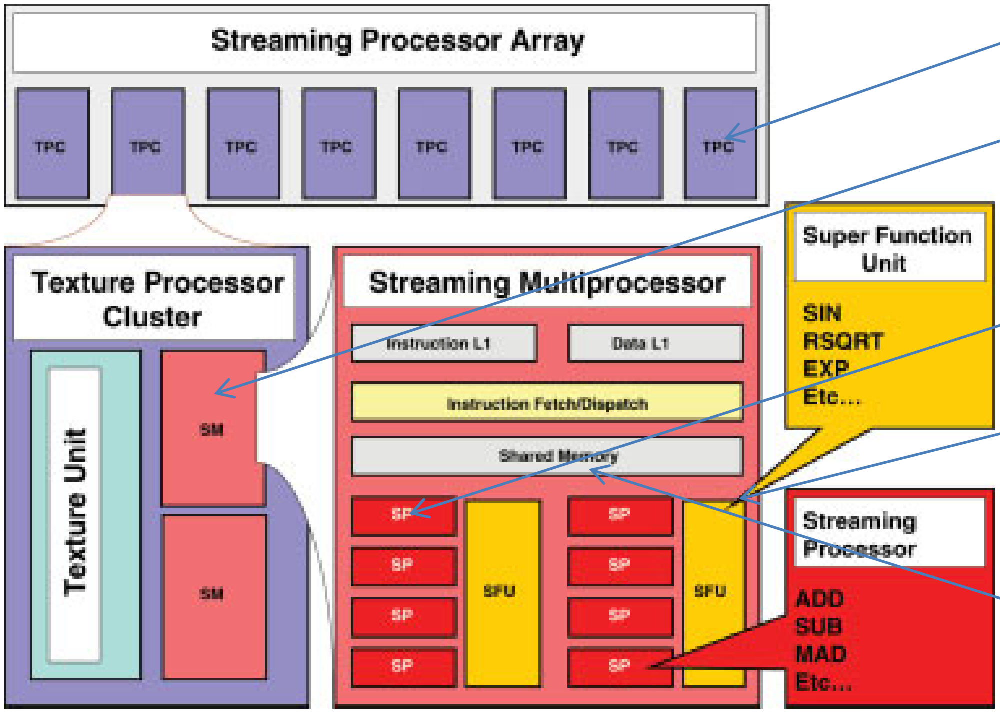
- GPU Cluster for high performance computing (*Fan et al. 2004*): 32 nodes cluster of GeForce 5800 ultra
- LB-Stream Computing or over 1 Billion Lattice Updates per second on a single PC (J. *Tölke ICMMES 2007*): GeForce 8800 ultra

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

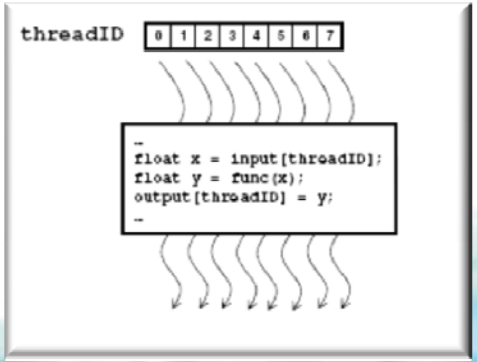
GPU Hardware Architecture



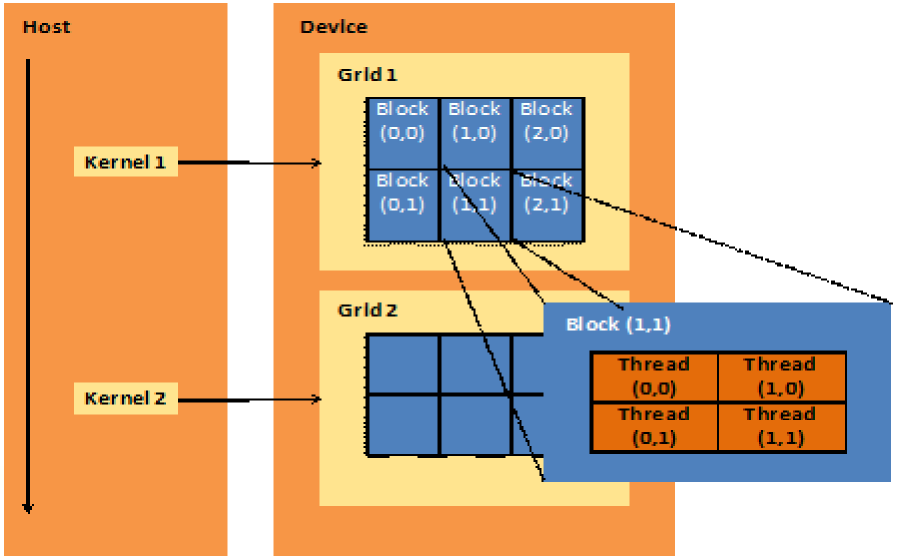
- ◊ 15 texture processors (TPC)
- ◊ 1 TPC is composed of 2 streaming multiprocessors (SM)
- ◊ 1 SM is composed of 8 streaming processors (SP) and 2 special functions units
- ◊ Each SM has 16kB of shared memory

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 exucuted by an array of threads

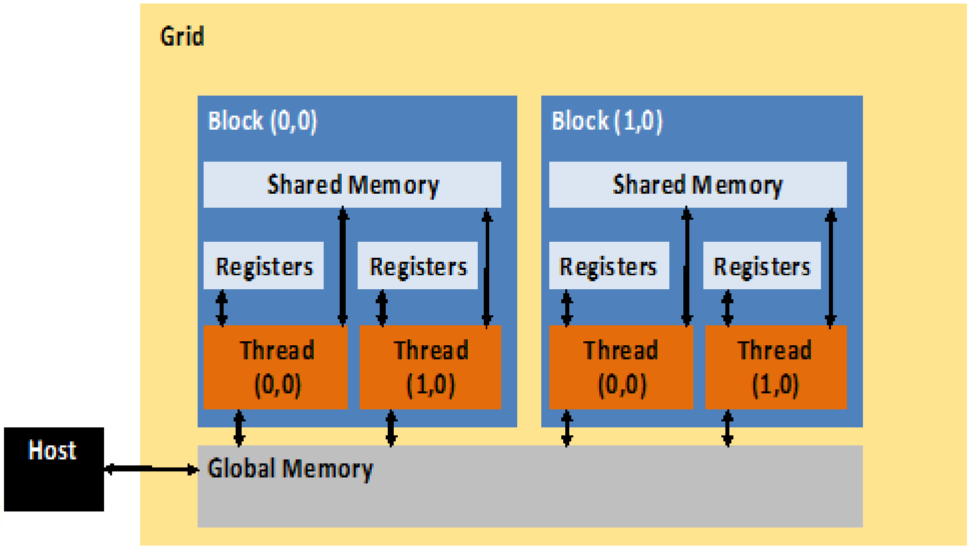


GPU Programming Interface



- 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

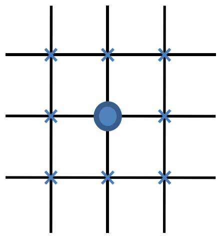
Kernel memory access



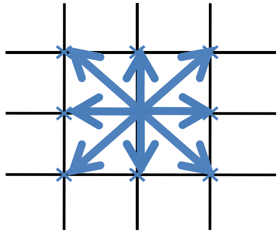
- Registers
- Shared memory: on-chip (fast), small (16kB)
- Global memory: off-chip, large
- The host can read or write global memory only.

LBM ALGORITHM

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



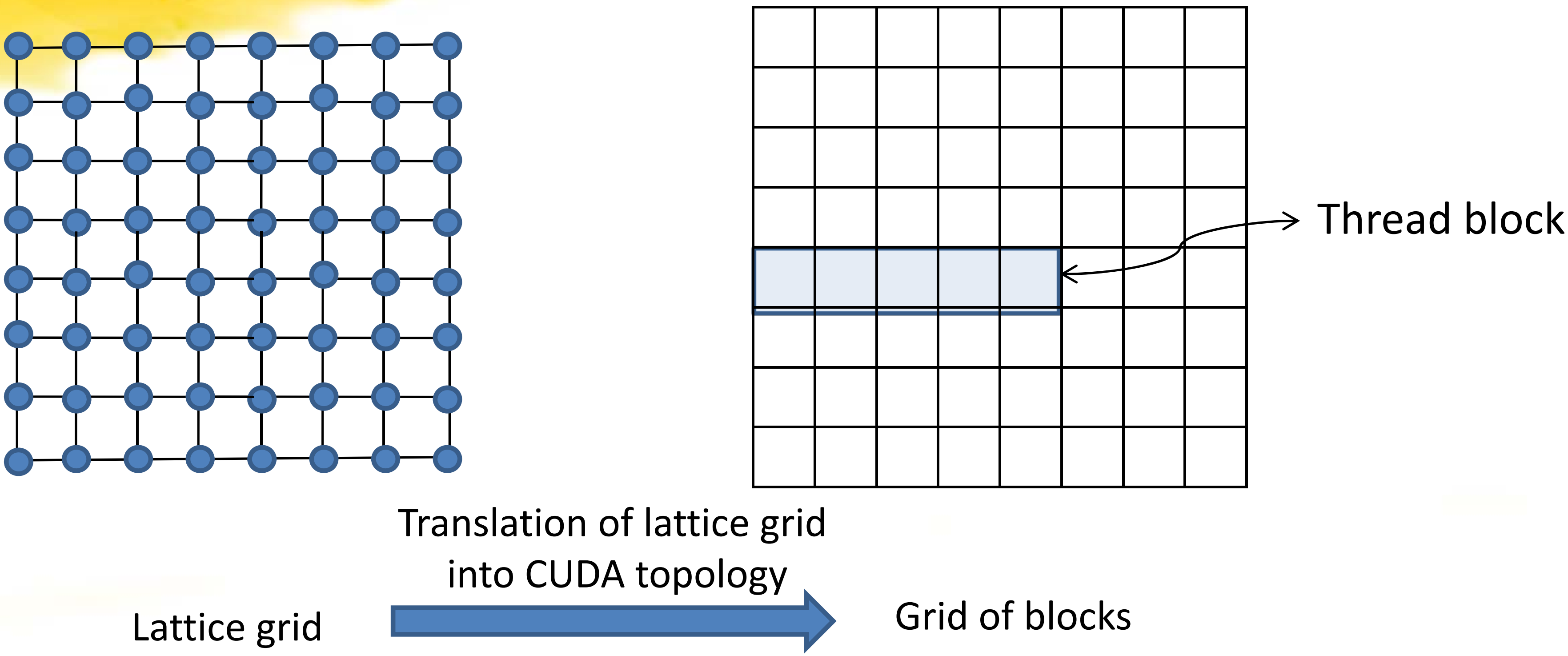
- Step 2: Propagation (Neighbors – 28% computational time)



- + 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

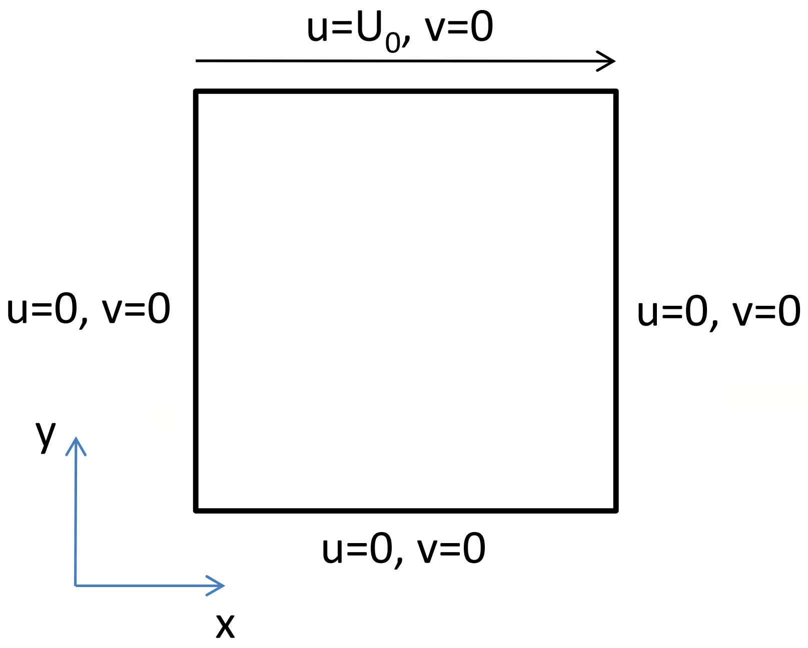
Combine collision and propagation steps:

```
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

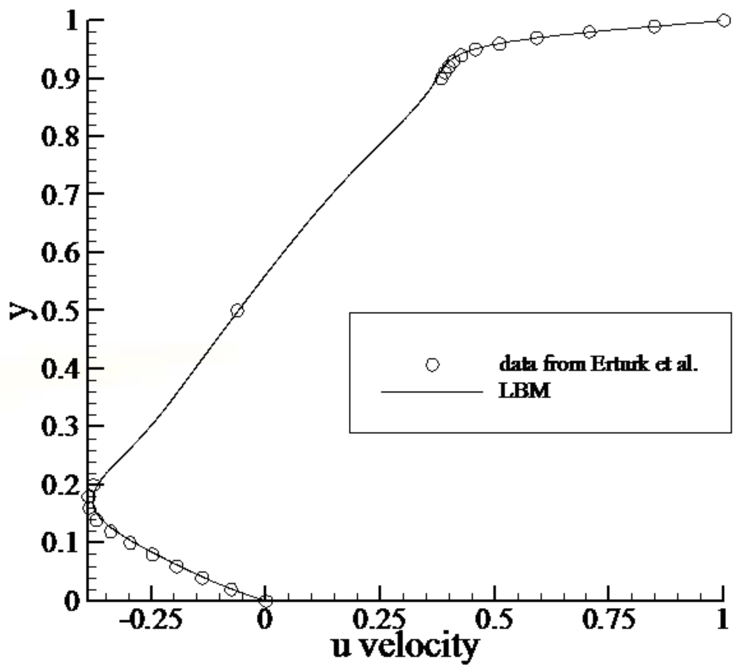
- Problem description



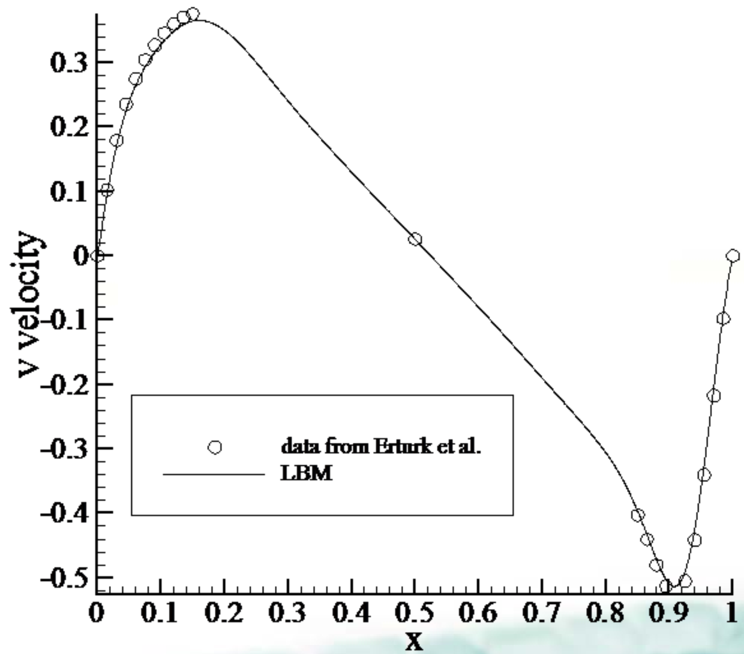
- D2Q9 MRT model of d’Humières (1992)

Results Re=1000

Reference data from *Erturk et al. 2005*:



Vertical velocity profil for x=0.5



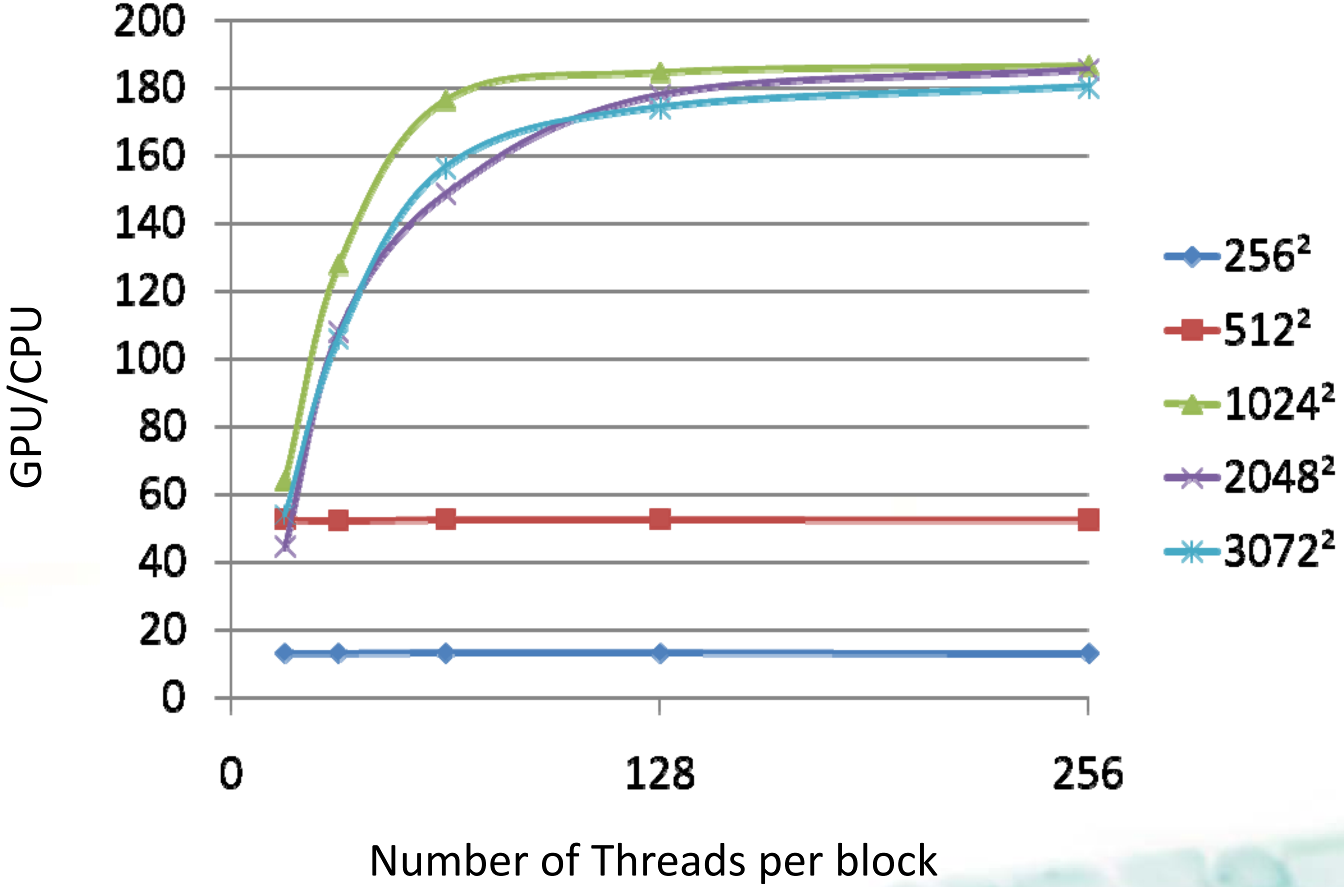
Horizontal velocity profil for y=0.5

Performances

Mesh grid size	Number of Threads				
	16	32	64	128	256
256^2	71.0	71.0	71.1	71.1	70.9
512^2	284.2	282.7	284.3	284.3	283.7
1024^2	346.4	690.2	953.0	997.5	1007.9
2048^2	240.0	583.0	803.3	961.3	1001.5
3072^2	289.3	572.4	845.2	941.2	974.2

Performances in MLUPS

Performances: CPU comparisons



CPU = Pentium IV, 3.0 GHz

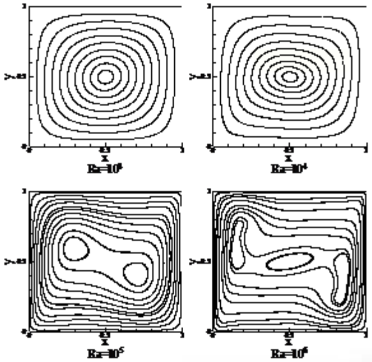
Conclusions

- 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