INTRODUCTION TO GPU COMPUTING

Ciprian Mihai Mitu, Mihai Niculescu
Institute for Space Sciences
Romania

Nov 30 2010 HP-SEE Training, Sofia, Bulgaria

Contents

- Parallel Computing
- What is CUDA?
- Why CUDA?
- CUDA Programming Model
- CUDA Execution Model
- Conclusions

Parallel computing

- Definition: solving complex problems using more processors.

- Michael J.Flynn – 1966 classification:
  - Number of programs
  - Number of sets of data

Parallel Computing Classification

What is CUDA?

- **C**ompute **U**nified **D**evice **A**rchitecture - NVIDIA
  - General purpose parallel computing for GPU
  - Requirements:
    - ISA(Instruction Set Architecture)
    - NV driver
    - CUDA Toolkit
    - (CUDA SDK)

Cuda Toolkit

- nvcc C compiler
- CUFFT + CUBLAS libraries
- Cuda Profiler
- Nv gdb
- CUDA Runtime
- Programming manual

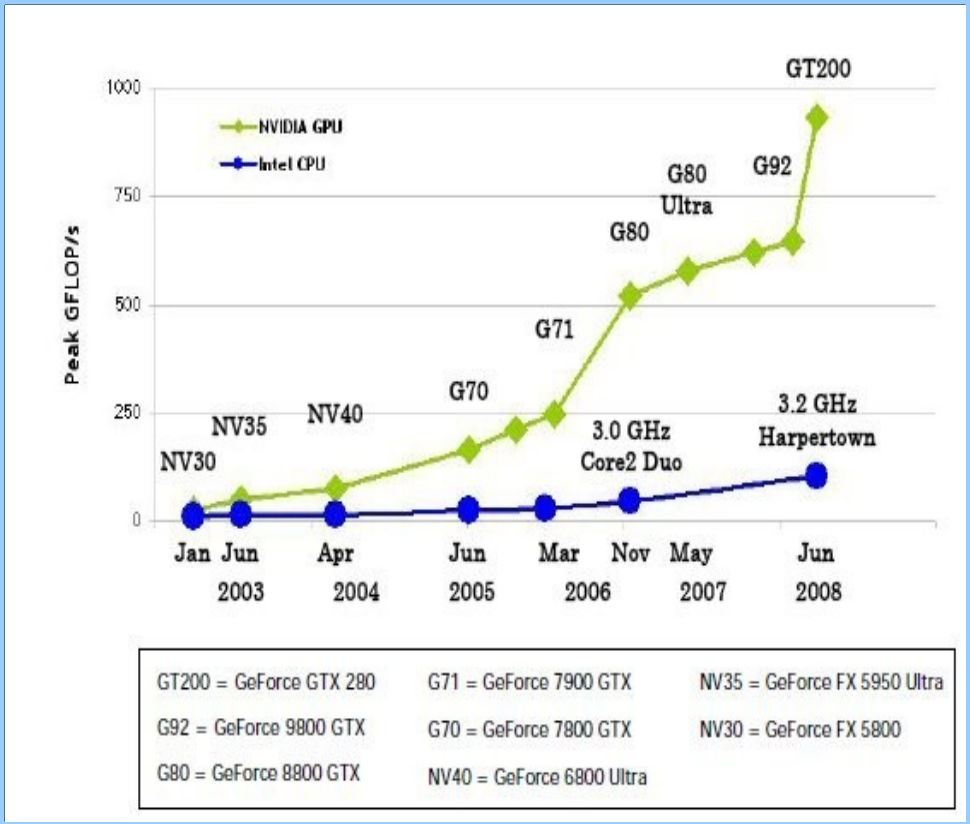
Cuda SDK

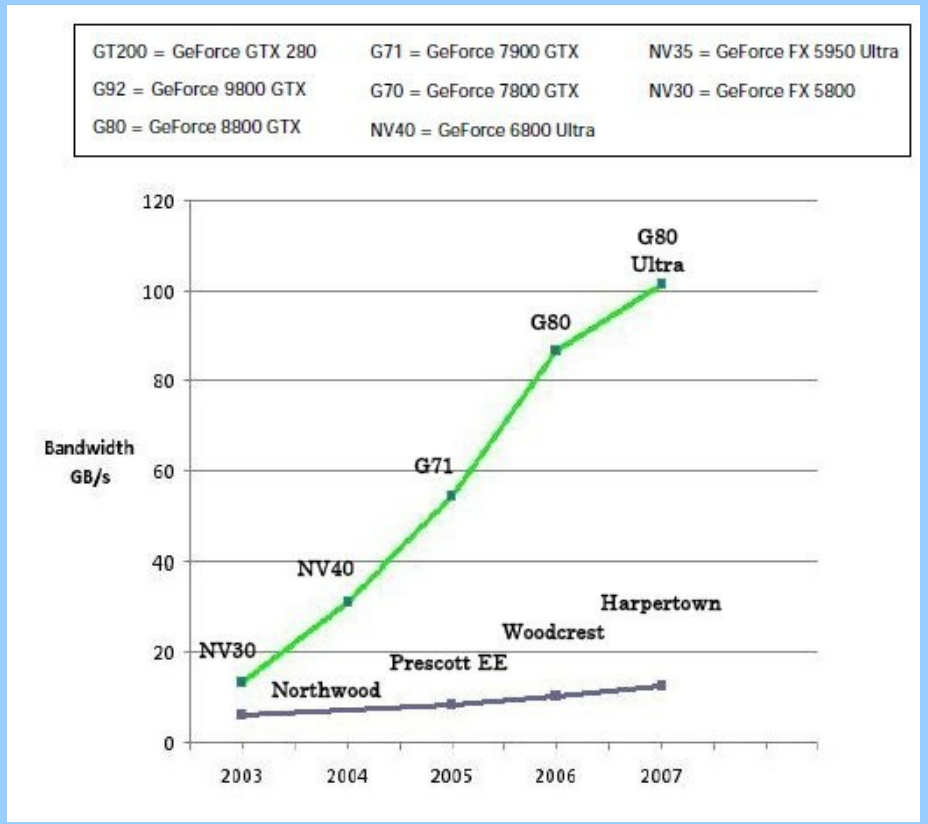
- Lots of examples
  - Mandelbrot
  - Nbody
  - Fluids
  - ...
- Dev easy

Why CUDA?

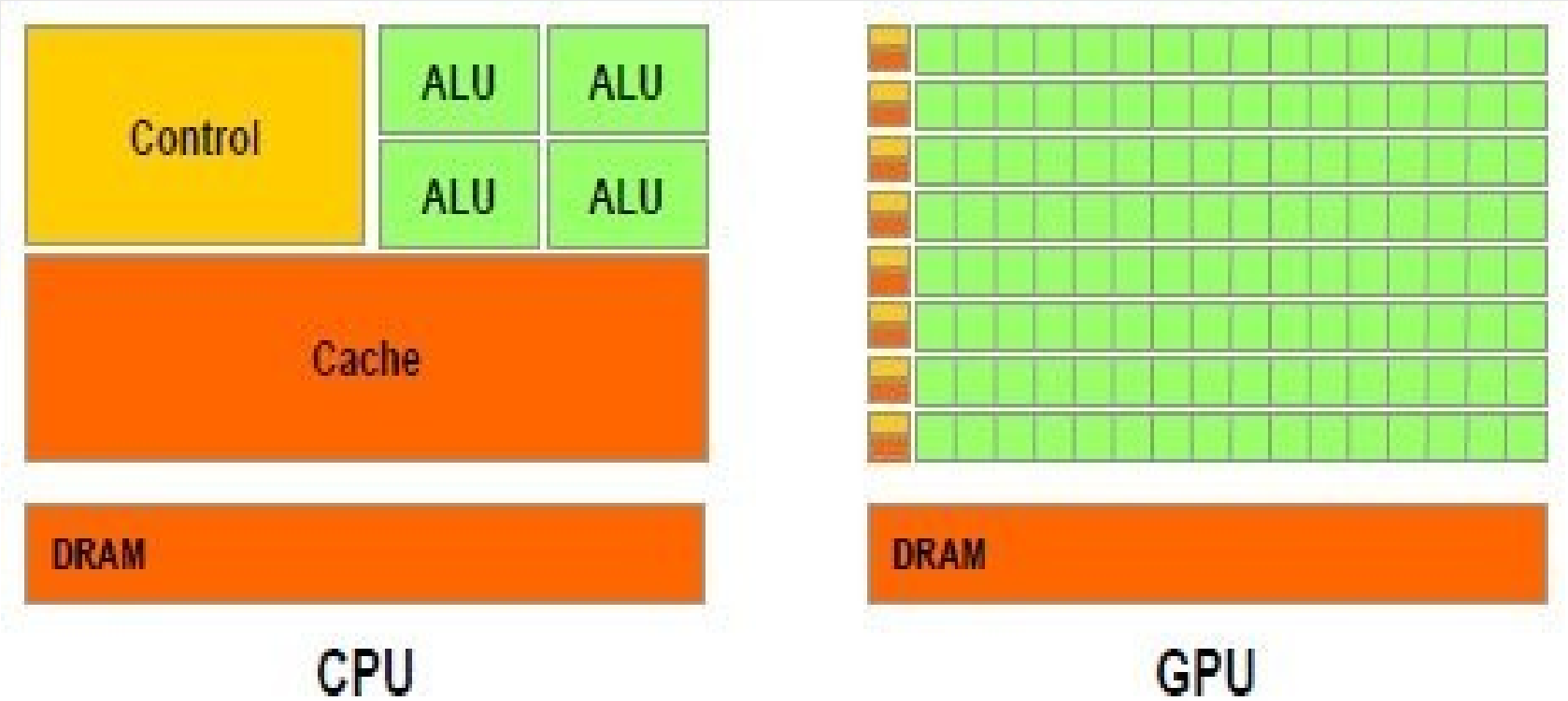
- Price ~ $ 200
- Performance - speedups > 20x
- Language Bindings C/C++ and FORTRAN
- Generality: diverse parallel algorithms
- Scalability

GPU vs CPU


Peak GFLOP/s
1000
750
500
250
0
NVIDIA GPU
Intel CPU
GT200
G92
G80 Ultra
G80
G71
G70
NV40
NV35
NV30
3.0 GHz Core2 Duo
3.2 GHz Harpertown
Jan Jun 2003
Apr 2004
Jun 2005
Mar Nov 2006
May 2007
Jun 2008
GT200 = GeForce GTX 280
G92 = GeForce 9800 GTX
G80 = GeForce 8800 GTX
G71 = GeForce 7900 GTX
G70 = GeForce 7800 GTX
NV40 = GeForce 6800 Ultra
NV35 = GeForce FX 5950 Ultra
NV30 = GeForce FX 5800


GT200 = GeForce GTX 280
G92 = GeForce 9800 GTX
G80 = GeForce 8800 GTX
G71 = GeForce 7900 GTX
G70 = GeForce 7800 GTX
NV40 = GeForce 6800 Ultra
NV35 = GeForce FX 5950 Ultra
NV30 = GeForce FX 5800
Bandwidth GB/s
120
100
80
60
40
20
0
G80 Ultra
G80
G71
NV40
NV30
Harpertown
Woodcrest
Prescott EE
Northwood
2003
2004
2005
2006
2007

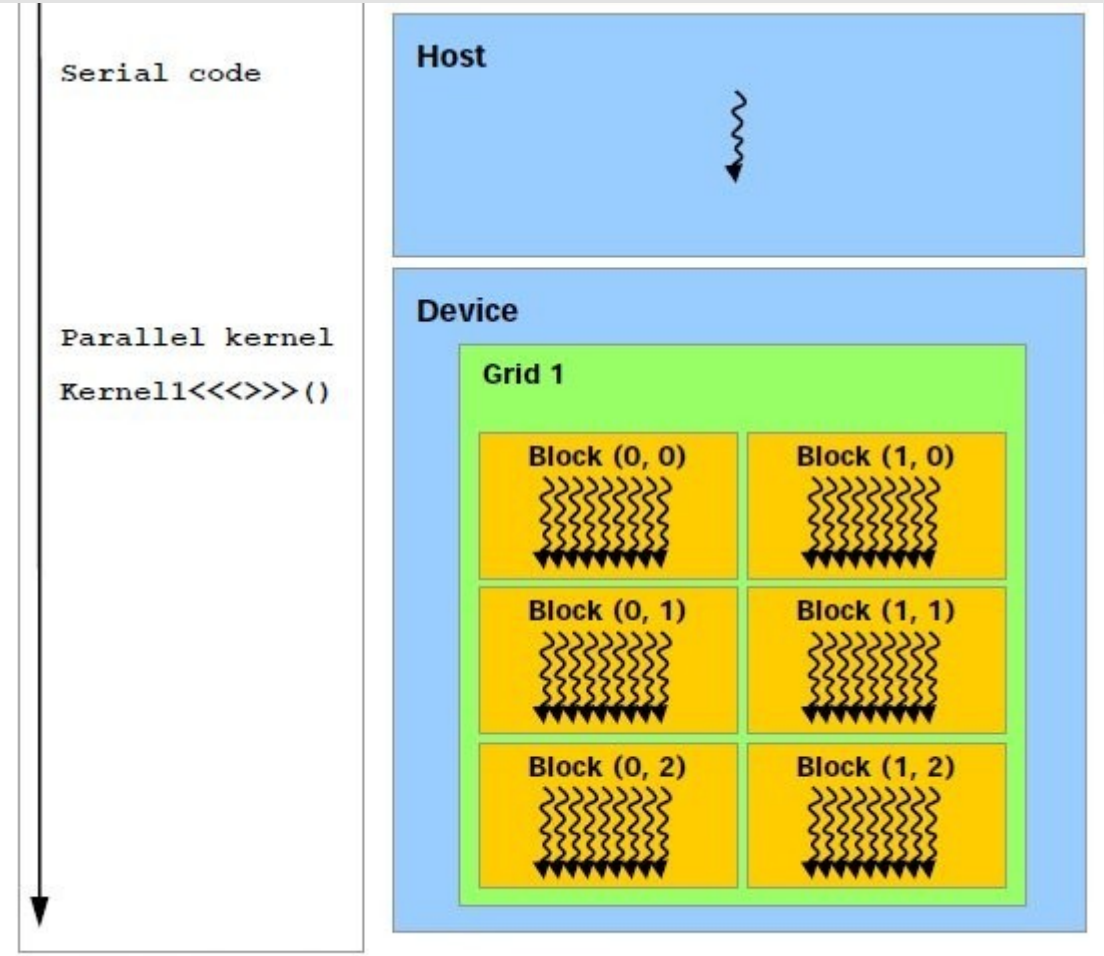
Why is GPU so fast?



Programming Model

- GPU is a computing device:
  - coprocessor for the CPU (host)
  - own memory (DRAM)
  - many threads in parallel (>1000)
  - SPMD – single program, multiple data

Execution Model



Serial code executes on the host while parallel code executes on the device.

Performance Optimization

- Expose as much parallelism as possible
- Optimize memory usage to achieve maximum memory throughput
- Optimize instruction usage to achieve maximum instruction throughput
- Maximize occupancy to hide latency

Conclusions

- CUDA is a powerful parallel programming model
  - Heterogeneous
  - Scalable
  - Accessible

References

- http://www.nvidia.com/object/cuda_home_new.html
- http://forums.nvidia.com