High Performance Computing Center
Hanoi University of Science & Technology

Introduction to GP-GPU and CUDA

Duong Nhat Tan (dn.nhattan@gmail.com)

2012

Outline

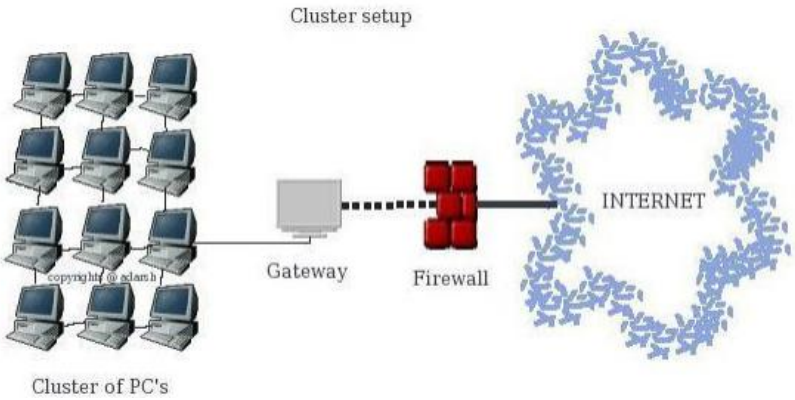
- Overview
- What is GPGPU?
- GPU Computing with CUDA
  - Hardware Model
  - Execution Model
  - Thread Hierarchy
  - Memory Model
- GPU Computing Application Areas
- Summary

Overview

- Scientific computing has the following characteristics:
  - The problems are not interested.
  - Use computer to calculate the arithmetic.
  - Always want the programs run faster
- For examples: weather forecasting, climate change, modeling, simulation, gene prediction, docking…

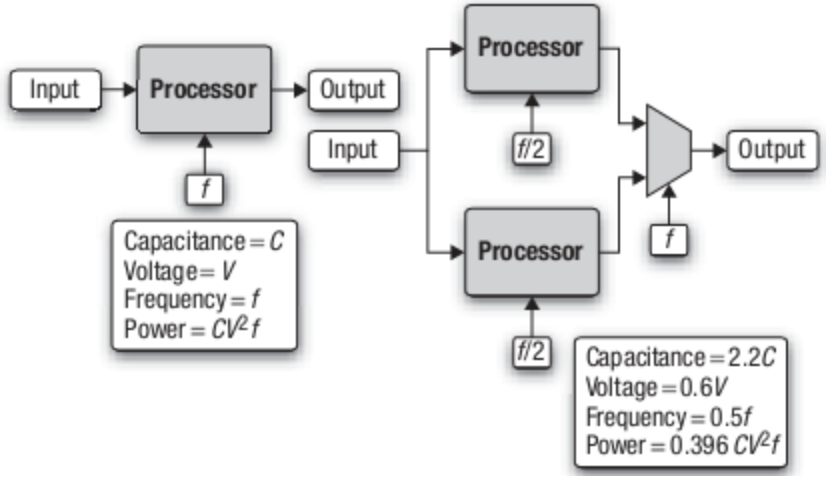
Several Approaches

- Supercomputers
- Mainframe
- Cluster
- Multi/many cores systems



Microprocessor trends

- Many cores running at lower frequencies are fundamentally more power-efficient



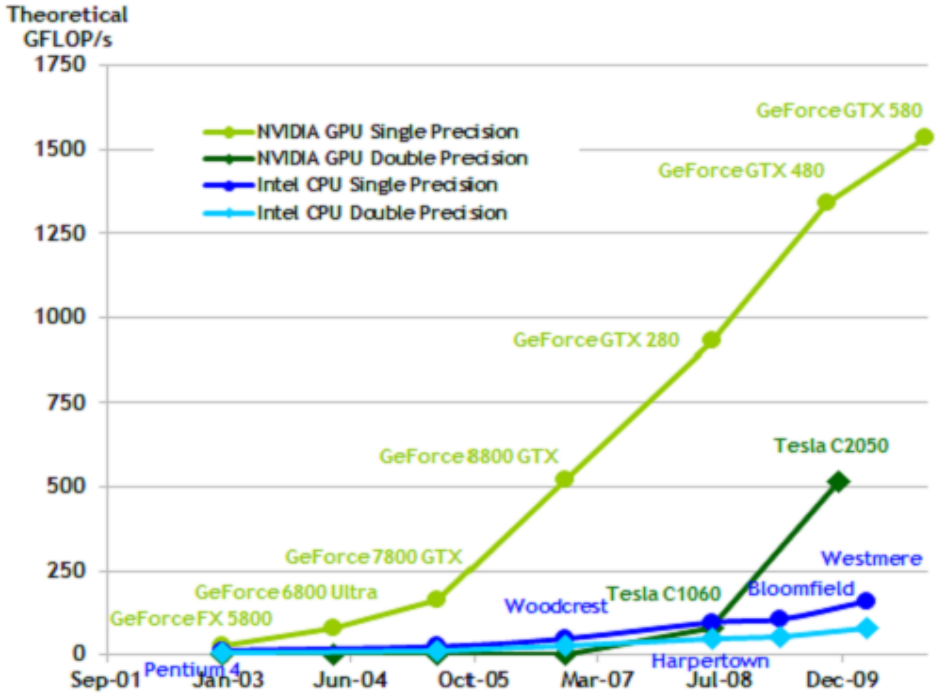
- Multi- cores (2-8 cores)
  - CPU Intel pentium D/core duo/ core 2 duo/ quad cores, core i3,i5, i7
- Many-cores (> 8 cores)
  - GPU - Graphics Processing unit

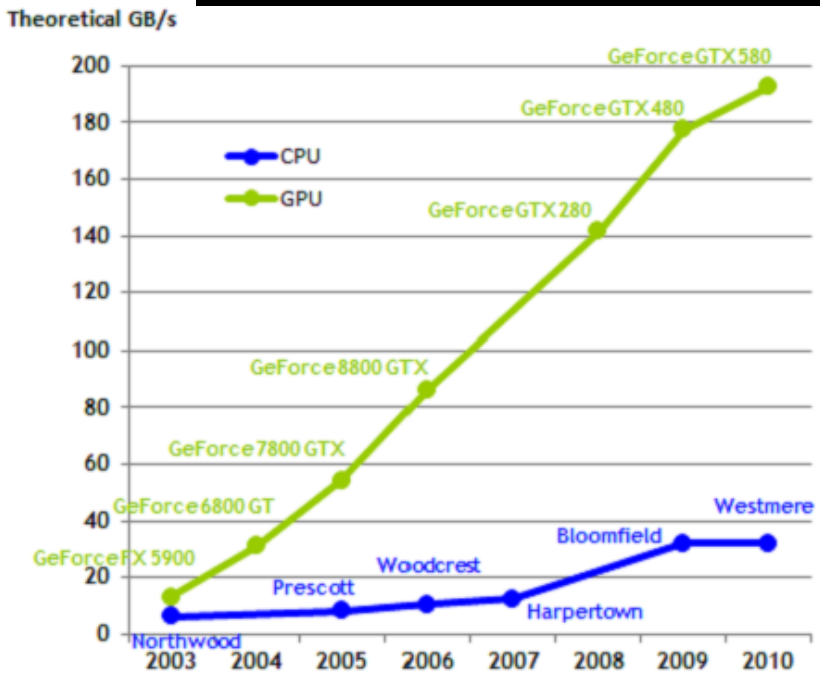
A. P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Brodersen,
"Optimizing Power Using Transformations," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

The development of modern GPUs

- GPU - NVIDIA GeFore GTX 295

CUDA Cores	480 (240 per GPU)
Graphics Clock (MHz)	576
Processor Clock (MHz)	1242
Memory Clock (MHz)	999
Memory Bandwidth (GB/sec)	223.8
Benchmark (GFLPOS)	1788.48





CPU vs GPU

- CPUs are optimized for high performance on sequential code: transistors dedicated to data caching and flow control
- GPUs use additional transistors directly for data processing

Books: "Program ming Massively Parallel Processors: A Hands-on Approach"

GPU Solutions

- NVIDIA
  - GeForce (gaming/movie playback)
  - Quadro (professional graphics)
  - Tesla (HPC)

NVIDIA Tesla C1060

- AMD/ATI
  - Radeon (gaming/movie playback)
  - FireStream (HPC)

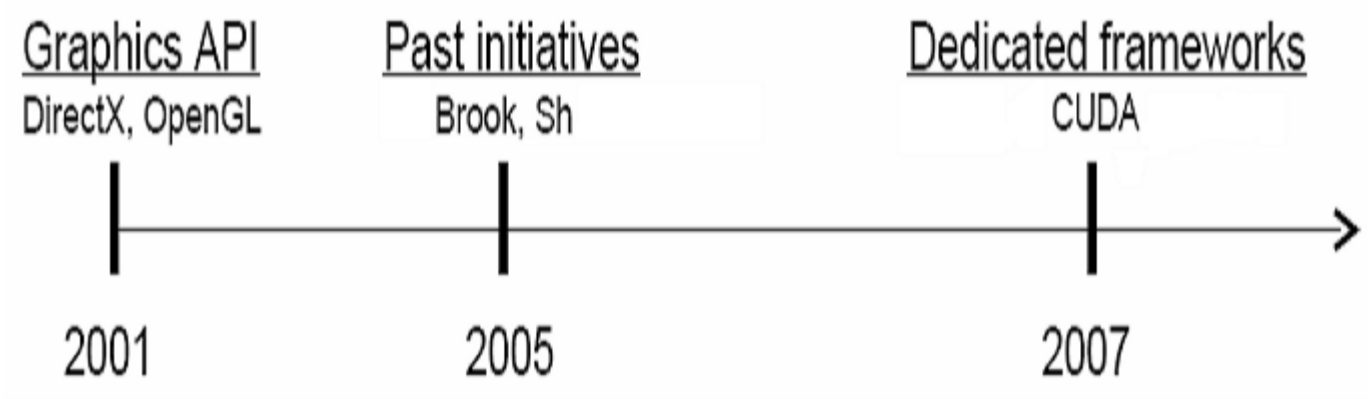
AMD FireStream 9170

Motivation

- Costs/performance ratio
- Costs for power supply
- Costs for maintain, operation

GPGPU

- GP-GPU stands for General Purpose Computation on GPU
  - A technique/technology/approach that consists in using the GPU chip on the video card as a coprocessor that accelerates operations that are normally executed on the CPU
- GPGPU is different from general graphics operations?
  - GPGPU – running various kinds of algorithms on a GPU, not necessarily image processing.
  - For example: FFT, Monte-Carlo, Data-Sorting, Data mining and the list continues
- Until 2006, developers must cast their problems to graphics field and resolve them using graphics API



Parallel Computing with GPU

NVIDIA's GeForce 8800 with CUDA

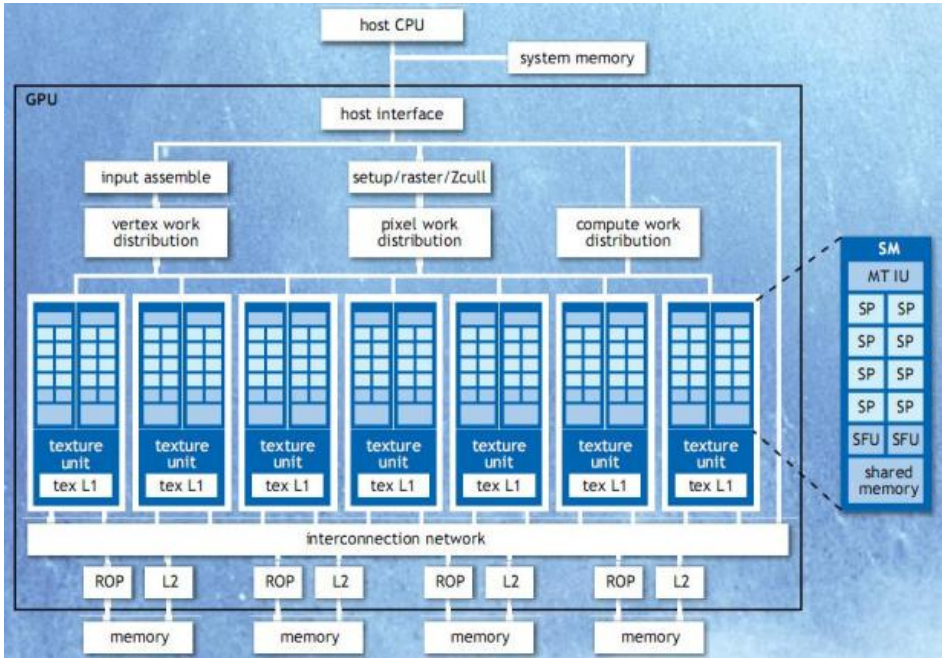
AMD's Stream Processor with CTM

AMD FireStream 9170 – 100w, 2GB, 500Gflops peak

- Now can be programmed not only through the usual graphics APIs
- Simple extensions of C: CUDA, Brook+

NVIDIA GPU

- 11/2006: NVIDIA released G80 architecture with an environment application development - CUDA
  - Allow developers to develop GPGP applications on high level programming languages



G80 Architecture

- Built from a scalable array of Streaming Processors (SM)
- Each SM contains 8 SP (Scalar Processor)
- Each SM can initialize, manage, execute up to 768 threads

NVIDIA GPU

- G80-based GPU
  - Geforce 8800 GT
    - 14 SMs equivalent 112 cores
    - DRAM 512MB

06/2008

- Geforce GT 200 series
  - 30 SMs (240 cores)
  - DRAM 1GB
- Tesla
  - 30 SMs (240 cores)
  - DRAM 4GB

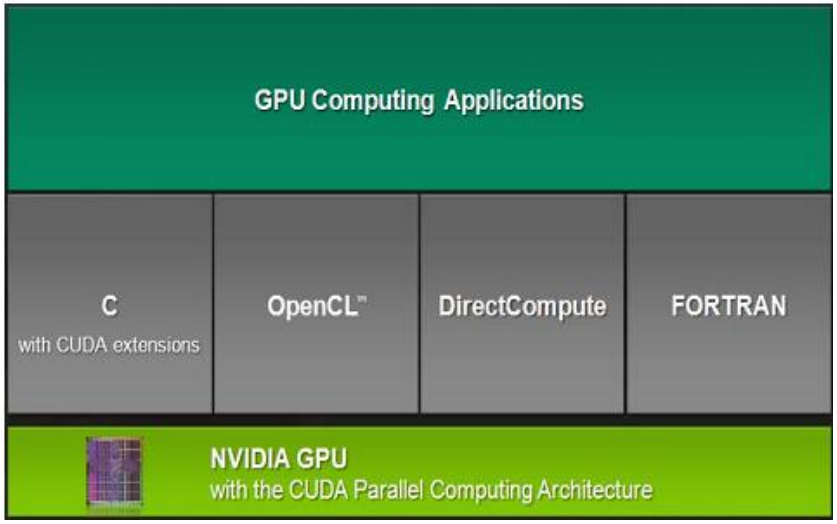
Tesla Specification

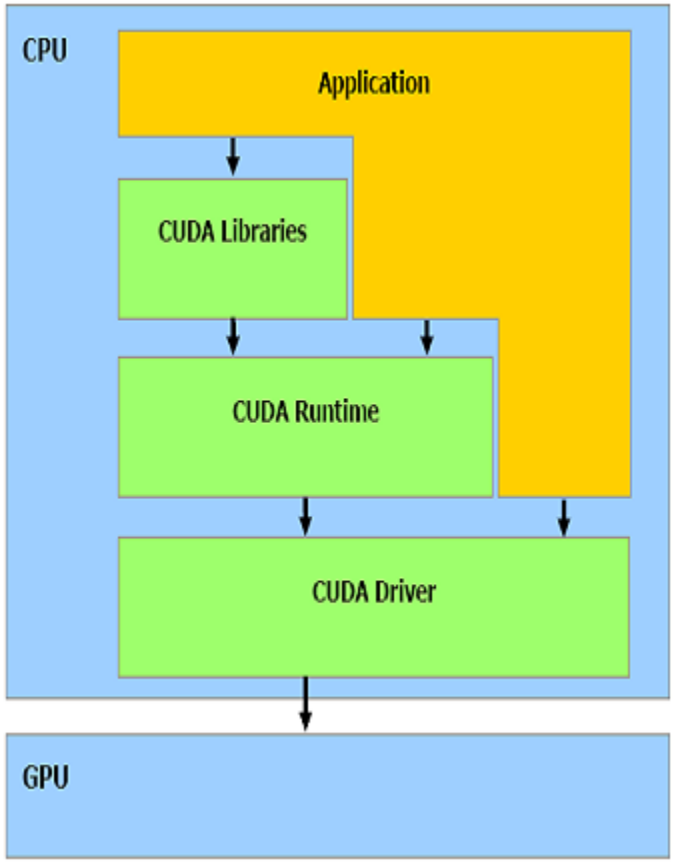
	C870	D870	S870
GPU#	1	2	4
Cores	128	256	512
Memory	1.5 GB	3 GB	6 GB
Performance	0.5 TFlops	1 TFlops	2 TFlops
Bandwidth	76.8 GB/s	153.6 GB/s	307.2 GB/s

- Power consumption: 187 W!

GPU Computing with CUDA

- CUDA: Compute Unified Device Architect
- Application Development Environment for NVIDIA GPU
  - Compiler, debugger, profiler, high-level programming languages
  - Libraries (CUBLAS, CUFFT, ..) and Code Samples





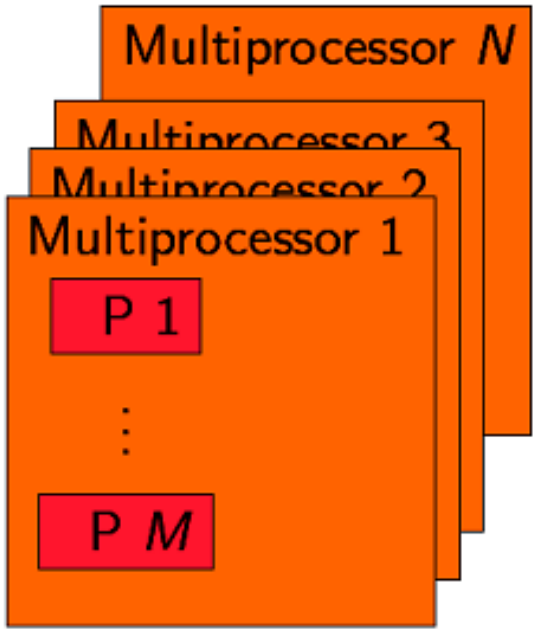
GPU Computing with CUDA

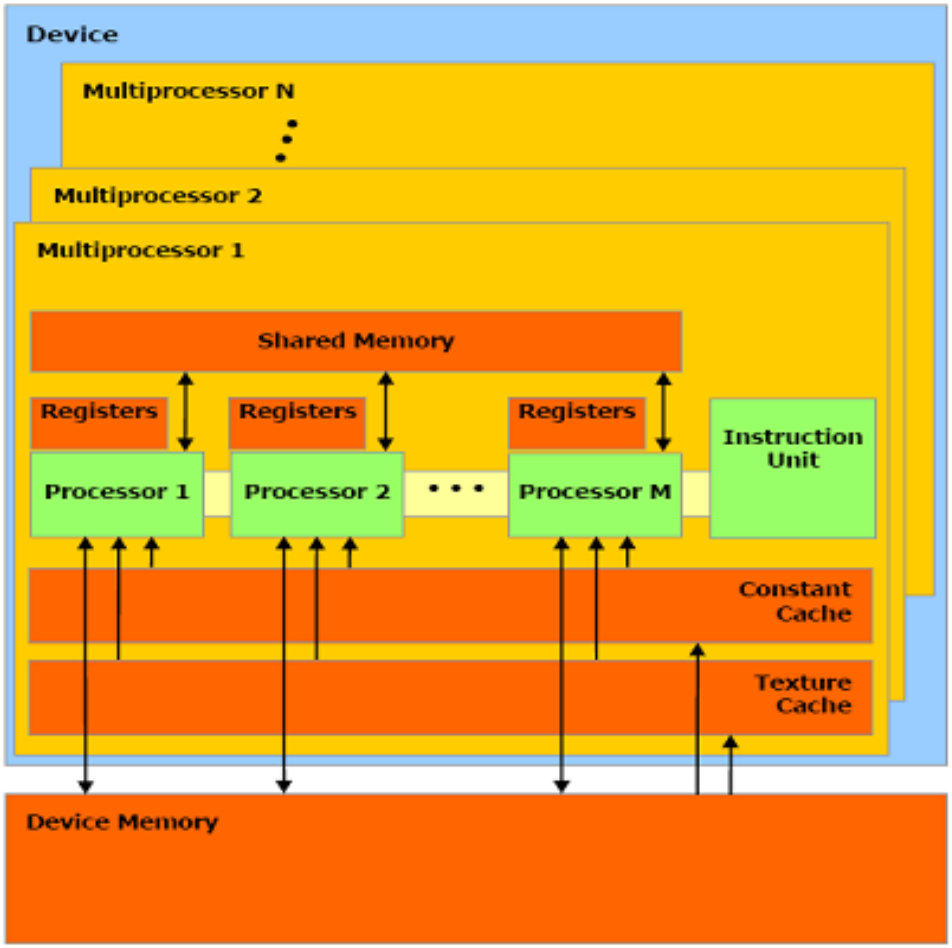
- The GPU is viewed as a compute device that:
  - Is a coprocessor to the CPU or host
  - Has its own DRAM (device memory)

- CUDA C is an extension of C/C++ language
- Data parallel programming model
- Executing thousands of processes in parallel on GPUs
- Cost of synchronization is not expensive

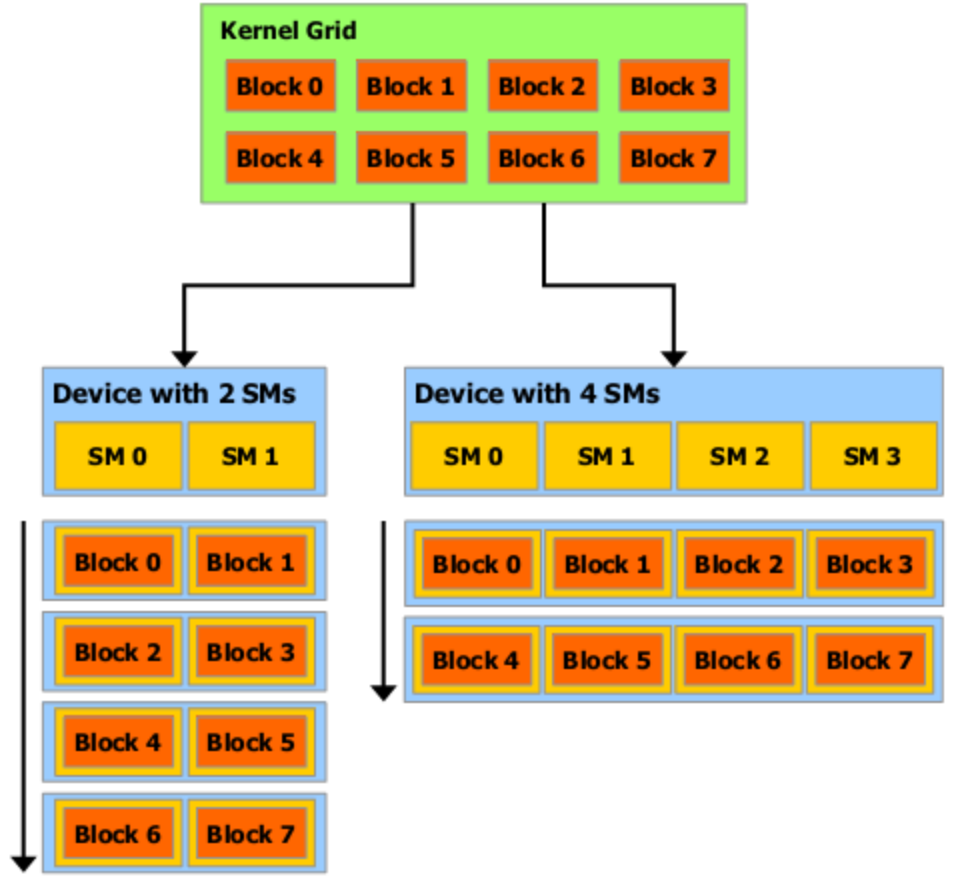
Hardware implementation

A set of SIMD Multiprocessors with On- Chip shared memory





Scalable Programming Models

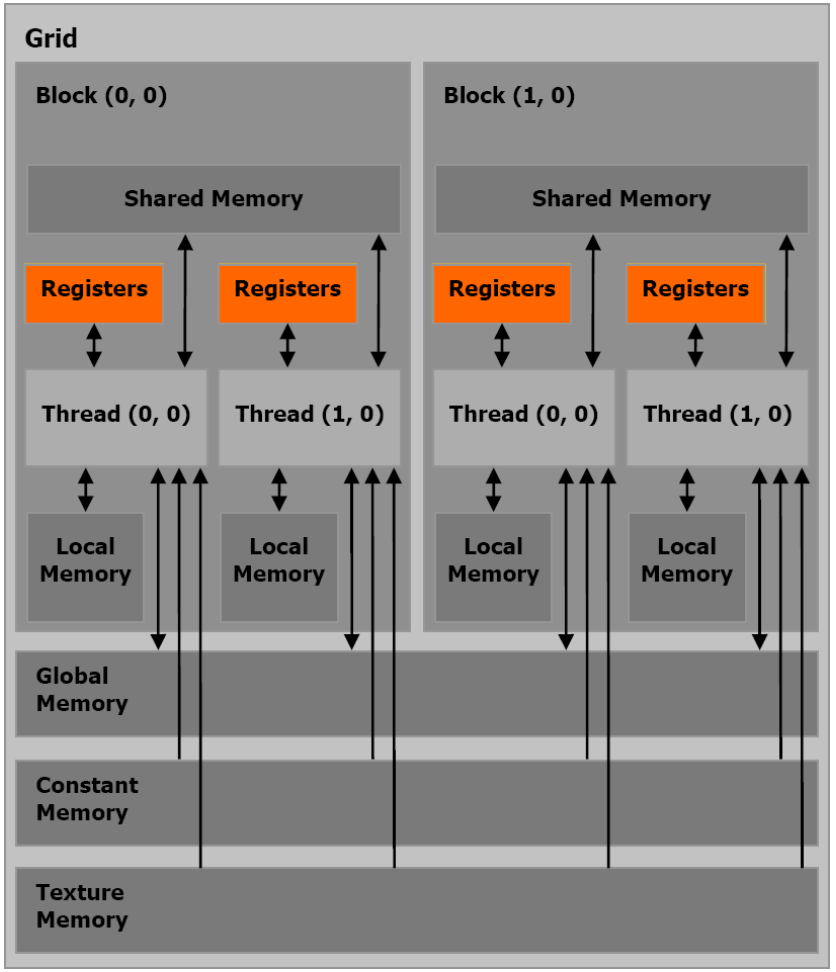


A device with more multiprocessors will automatically execute a kernel grid in less time than a device with fewer multiprocessors.

Memory Model

There are 6 Memory Types :

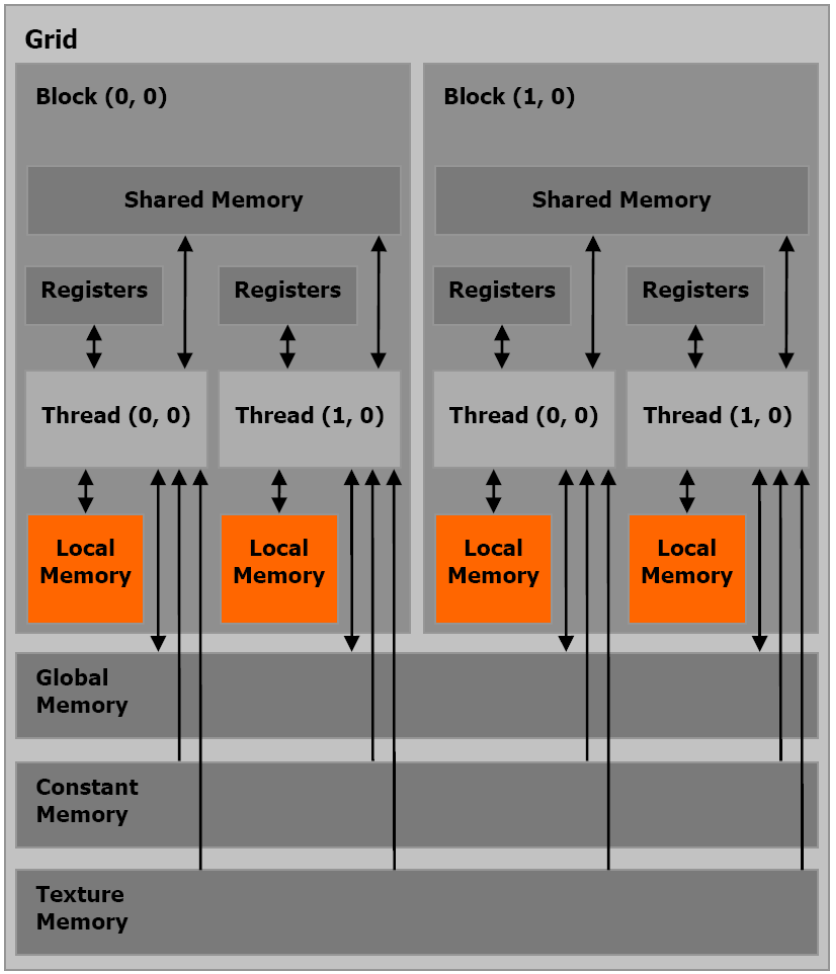
- **Registers**
  - on chip
  - fast access
  - per thread
  - limited amount



Memory Model

There are 6 Memory Types :

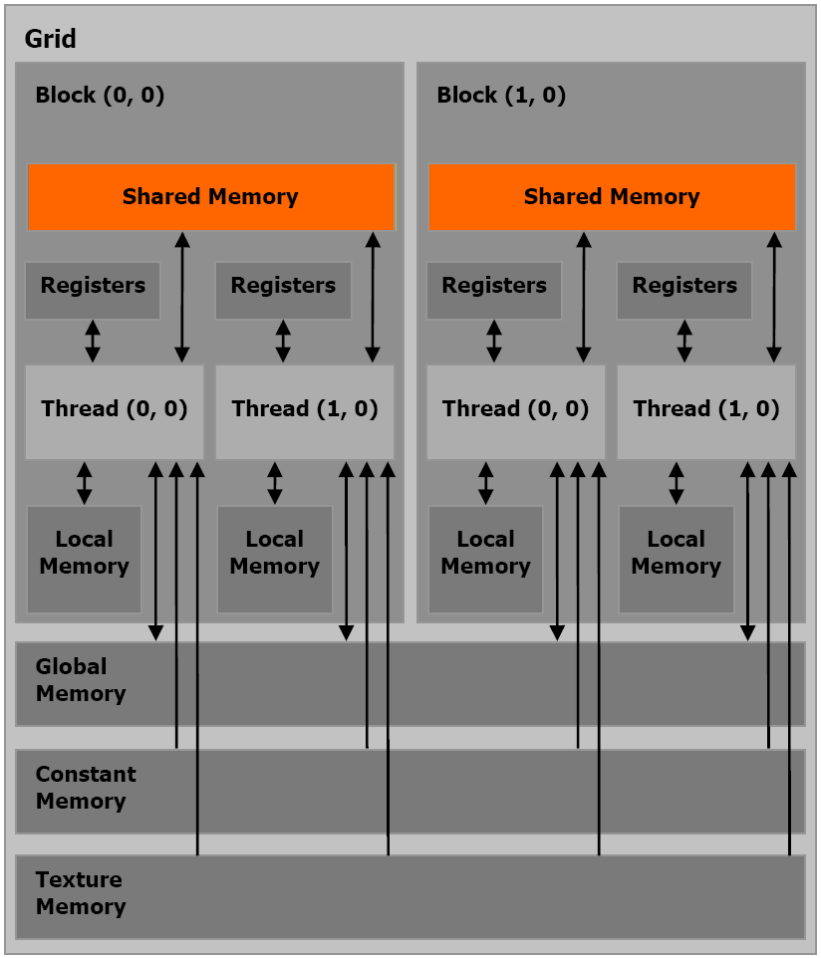
- Registers
- **Local Memory**
  - in DRAM
  - slow
  - non-cached
  - per thread
  - relative large



Memory Model

There are 6 Memory Types :

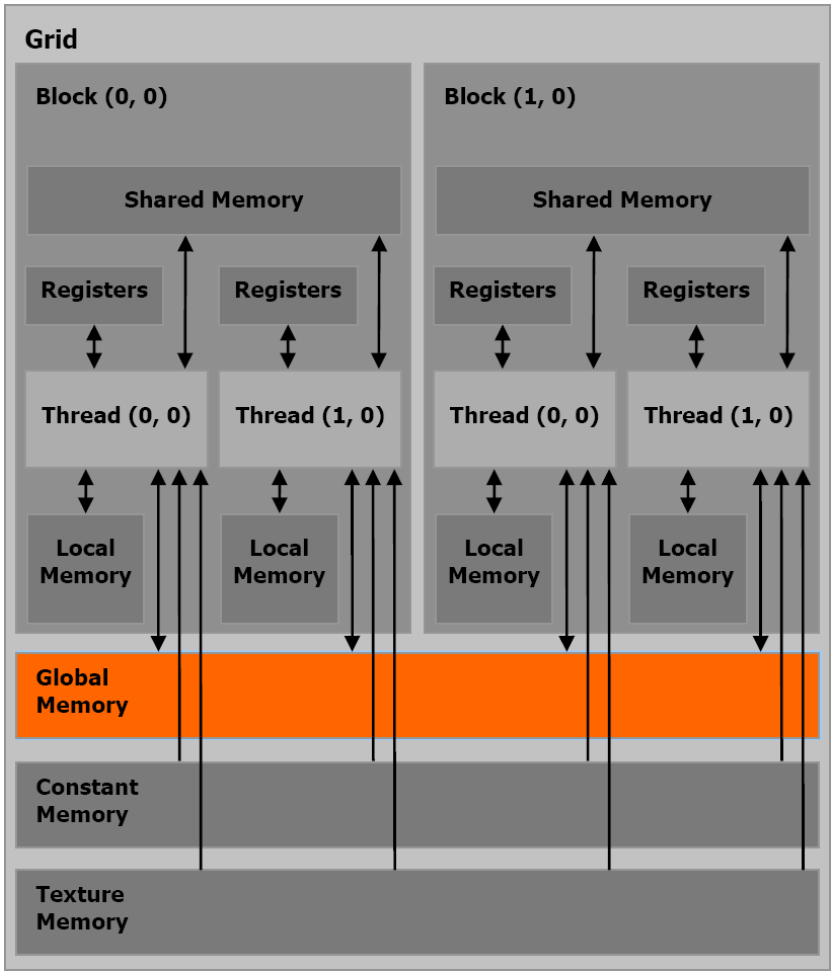
- Registers
- Local Memory
- **Shared Memory**
  - on chip
  - fast access
  - per block
  - 16 KByte
  - synchronize between threads



Memory Model

There are 6 Memory Types :

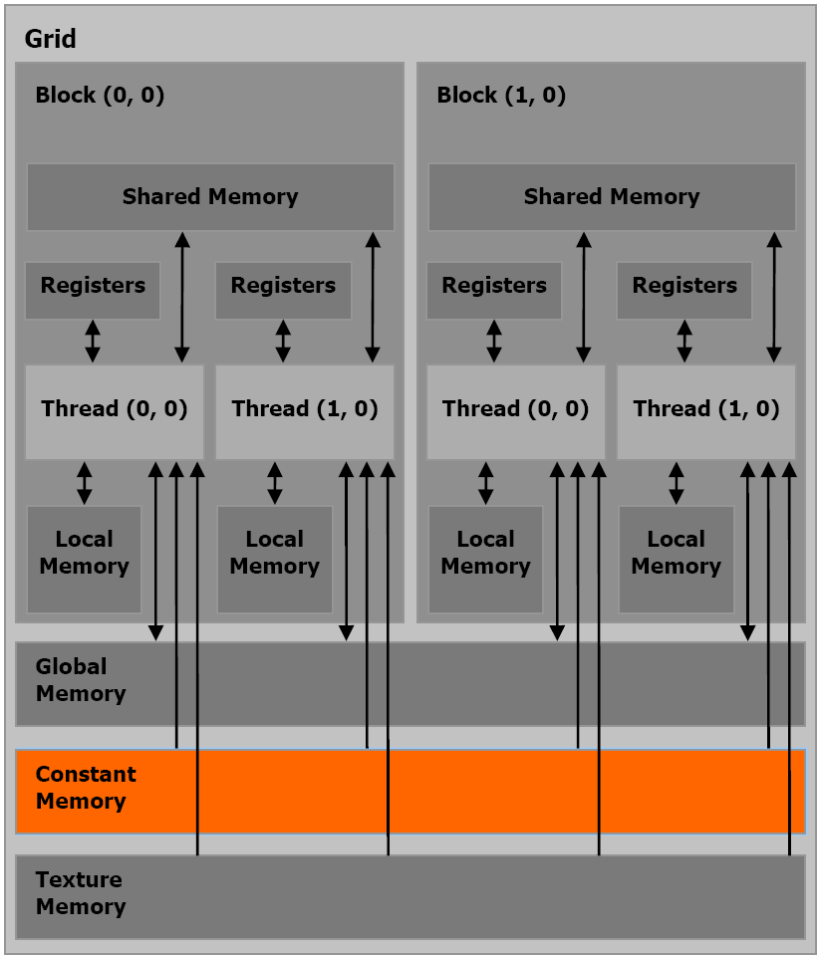
- Registers
- Local Memory
- Shared Memory
- **Global Memory**
  - in DRAM
  - slow
  - non-cached
  - per grid
  - communicate between grids



Memory Model

There are 6 Memory Types :

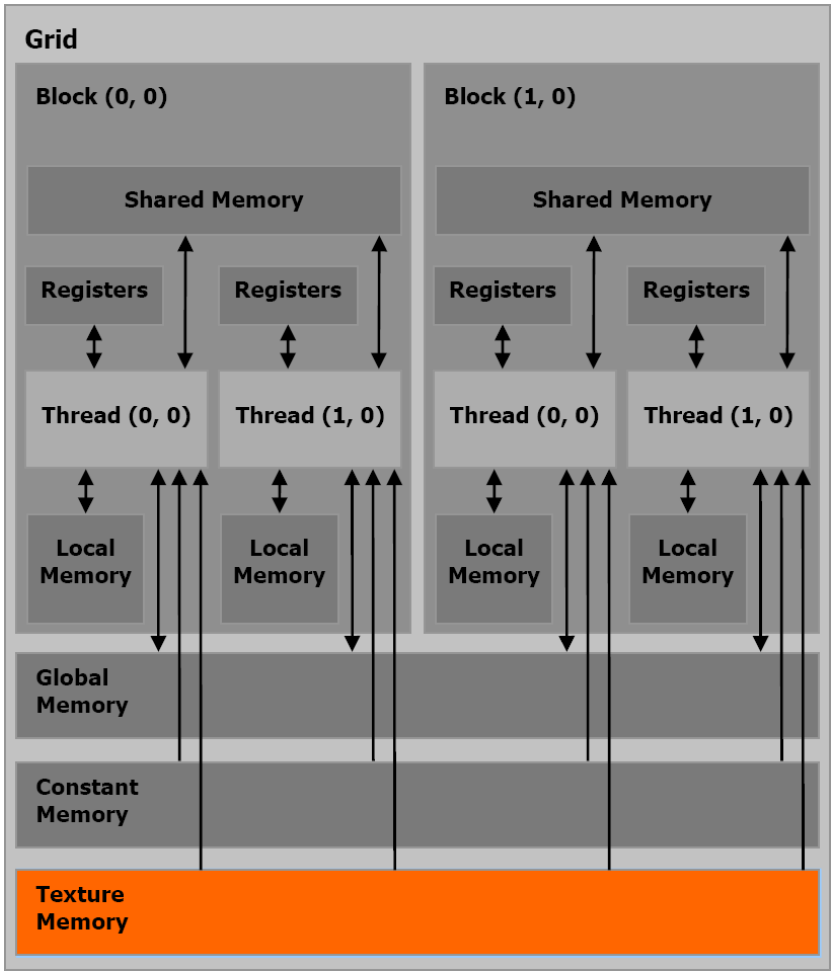
- Registers
- Local Memory
- Shared Memory
- Global Memory
- **Constant Memory**
  - in DRAM
  - cached
  - per grid
  - read-only



Memory Model

There are 6 Memory Types :

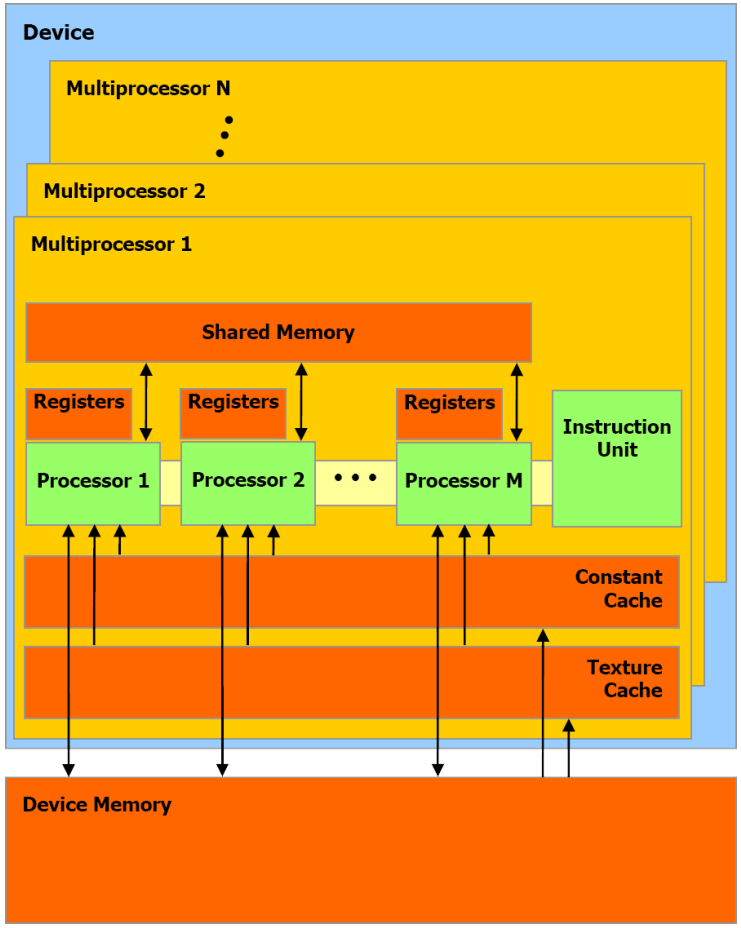
- Registers
- Local Memory
- Shared Memory
- Global Memory
- Constant Memory
- **Texture Memory**
  - in DRAM
  - cached
  - per grid
  - read-only



Memory Model

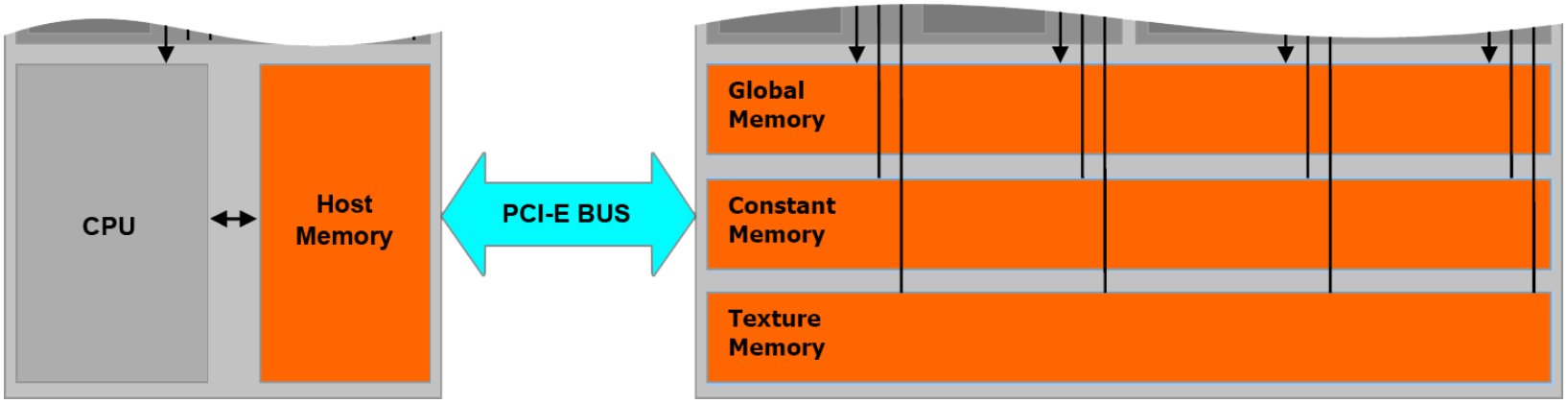
- Registers
- Shared Memory
  - on chip

- Local Memory
- Global Memory
- Constant Memory
- Texture Memory
  - in Device Memory

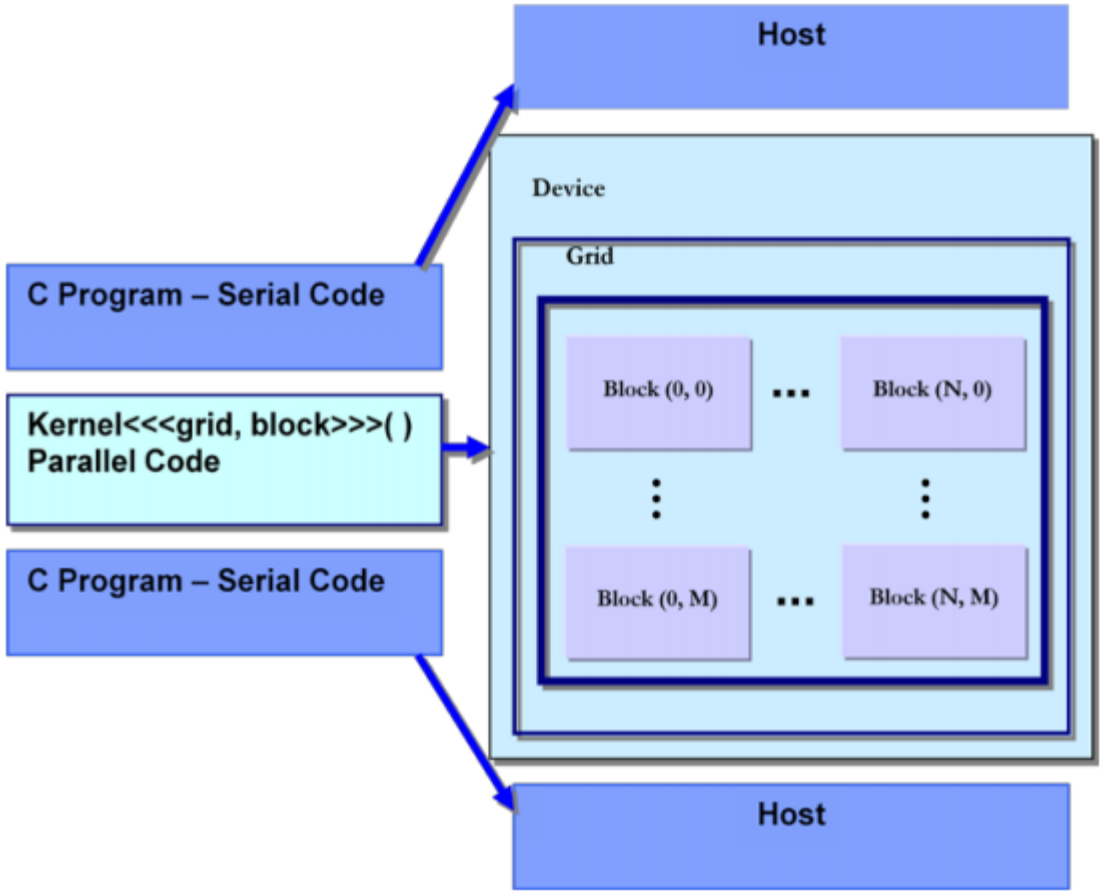


Memory Model

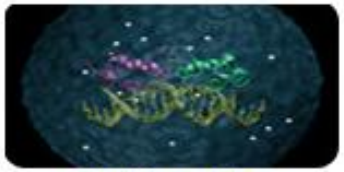
- Global Memory
- Constant Memory
- Texture Memory
  - managed by host code
  - persistent across kernels

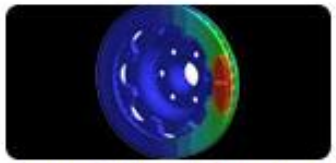


Hetegenerous Programming

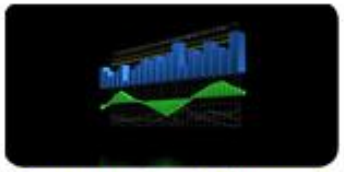


GP-GPU Applications


Bioinformatics


Computational Structural Mechanics


Computational Electromagnetics and Electrodynamics

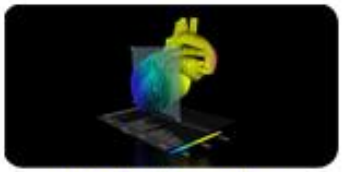

Computational Finance

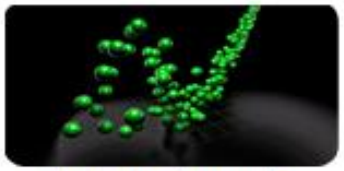

Computational Fluid Dynamics

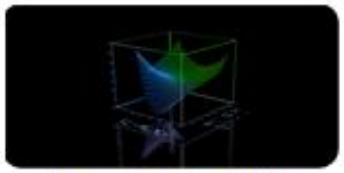

Data Mining

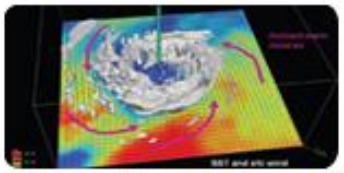

Electronic Design Automation


Imaging and Computer Vision


Medical Imaging


Molecular Dynamics


Numerical Analytics


Weather and Climate Forecasting

http://www.nvidia.com/object/tesla_computing_solutions.html

Bioinfomatics

- Sequence Alignment: to find out the most homogeneous characteristic of sequences

```
tctgcctctgccatcat---caaccc
|  |||  |||||  |||||    ||||||
tgtgcatctgcaatcatgggcaaccc
```

  - Smith-Waterman: identify the optimal local alignment of sequences by grading the similarity using the dynamic programming method
  - Search and matching a new DNA sequence in existing huge gene databases
    - BLAST http://blast.ncbi.nlm.nih.gov/Blast.cgi
    - FASTA http://www.ebi.ac.uk/Tools/sss/fasta/

Bioinfomatics

- CUDA-BLASTP: *"CUDA-BLASTP is designed to accelerate NCBI BLASTP for scanning protein sequence databases on GPUs, programmed using the CUDA programming model"*
- CUDASW++: an implementation of SW algorithm on NVIDIA GPU
- GPU HMMER: "*implements methods using probabilistic models called profile hidden Markov models on GPU*"

ISV	DESCRIPTION	GPU ADVANTAGE
CUDA-BLASTP	Protein sequence database scanning using NCBI BLASTP	10x speed-up: from minutes on CPUs to seconds on GPUs
CUDASW++	Protein sequence database (Smith-Waterman) scanning	10x-50x speed-up: achieving up to 30 GCUPs on query lengths over 5000
GPU HMMER	HMMER accelerated on CUDA	60-100x speed-up: from hours on CPUs to minutes on GPUs

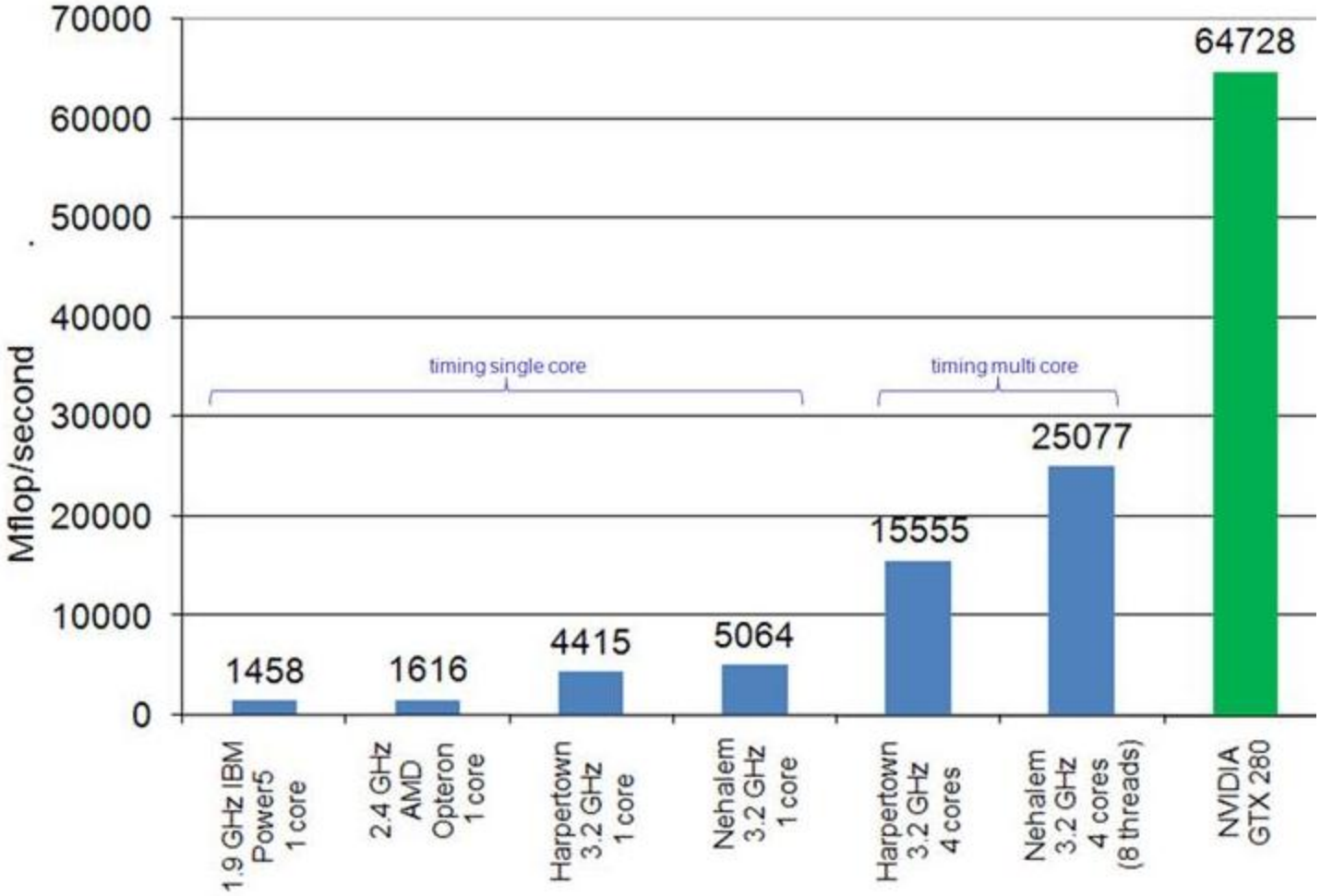
Weather Forecasting

- MM5/WRF models: numerical weather prediction system
  - Find the answers for system of equations with thousands of variables in *an acceptable time*
  - Process a huge amount of data (parameters about degree, humidity, wind speed, atmosphere, …)
  - "*characterize and model performance of the kernels in terms of computational intensity, data parallelism, memory bandwidth pressure, etc*"

http://www.mmm.ucar.edu/wrf/WG2/GPU/

WRF Single Moment 5 Cloud Microphysics

- Michalakes, J. and M. Vachharajani, "GPU Acceleration of Numerical Weather Prediction", *Parallel Processing Letters* Vol. 18 No. 4. World Scientific. Dec. 2008. pp. 531—548



HPCC - HUST

Cryptanalysis

- MD5 code breaking using GPU
- MD5 is one-way hash function

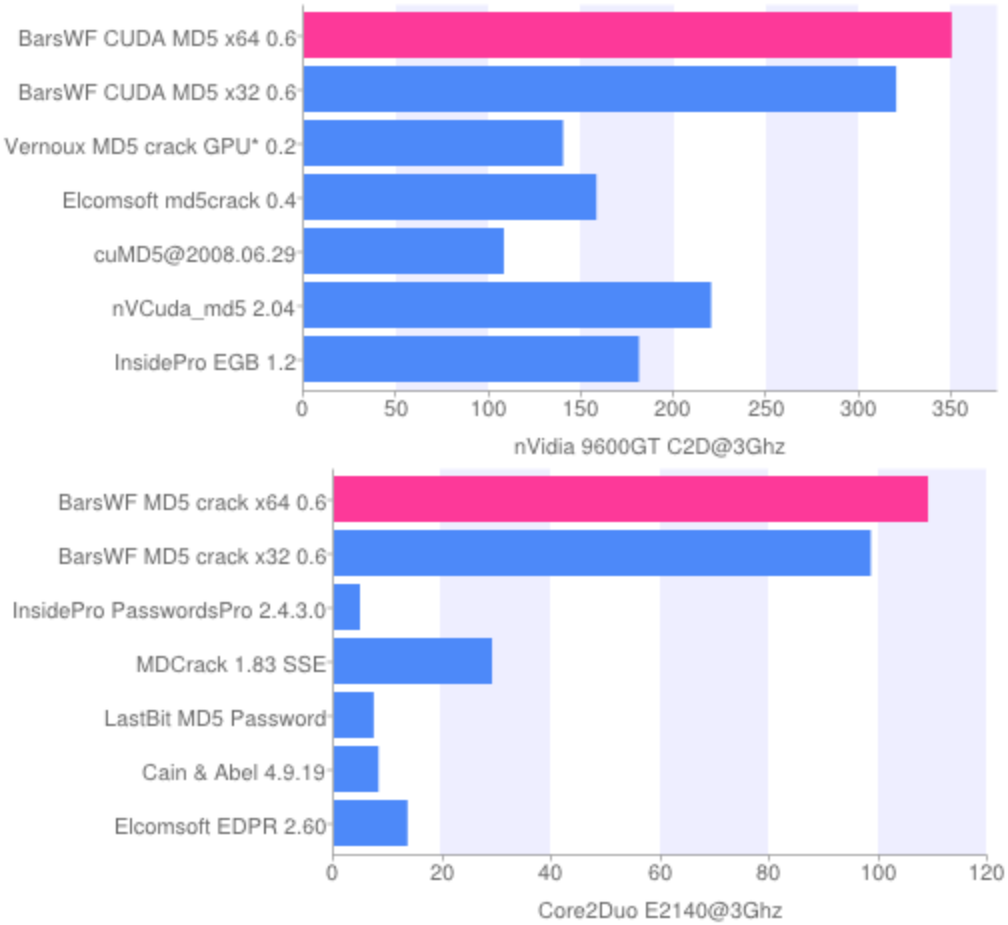
```
MD5("Test B") = CC506DE53938E1132D6CFB4746C37E13
MD5("Test C") = C78B709C472F5476546E27D88E763FA5
```

- Inverse problem
  - Input: MD5 hash
  - Ouput : the origin password

- Brute force attacks in 2 steps:
  - Step 1: Construct the password search space
  - Step 2: Implement the MD5 hash function for all passwords on GPUs

MD5 Bruteforce Benchmarks

- World Fastest MD5 cracker BarsWF



http://3.14.by/en/read/md5_benchmark

Seismic Exploration

- "*the cost of exploration and drilling deep wells can reach hundreds of millions of dollars, and there's often only one chance to do it successfully*"
- SeismicCity
  - use the most advanced depth imaging technologies
  - Using Tesla 1U System
  - Speed up 20x compared to CPU previous configuration

http://www.nvidia.com/object/seismiccity.html

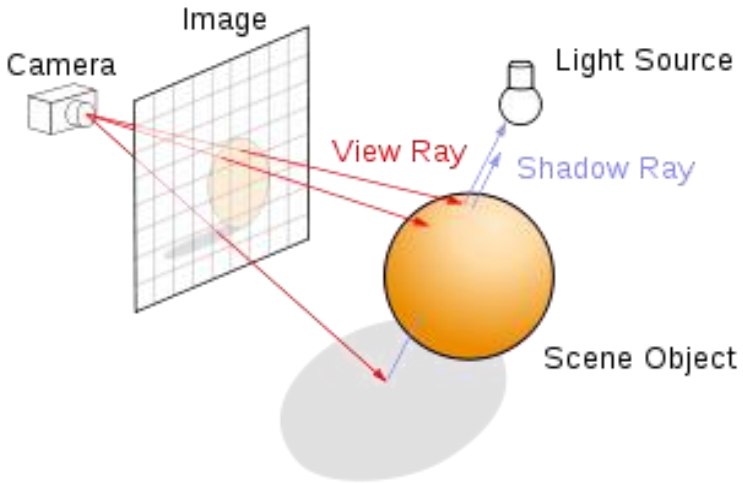
http://www.seismiccity.com/

Gamming/Entertaiment

- Two main methods in 3D rendering
  - Rasterization (supported by GPU, fast)
  - Raytracing (intensive computation but high-quality image)

a scene with 15 cars, rendered by an Apple G5 computer with two 2 GHz PowerPC processors and 2 GB memory take 15 hours! (2006)

Per H. Christensen, Julian Fong, David M. Laur and Dana Batali. *Ray Tracing for the Movie 'Cars'*. Proceedings of the IEEE Symposium on Interactive Ray Tracing 2006, p. 1-6



Solutions: NVIDIA OptiX

Other Applications

- Web Ranking on GPU
  - PageRank
  - HITS
  - TrustRank
- Search Results depend on two scores:
  - Content score: the relevance between search key word and page content
  - Popularity score: determined by analysis of the web's hyperlink structure

Web Ranking Problems

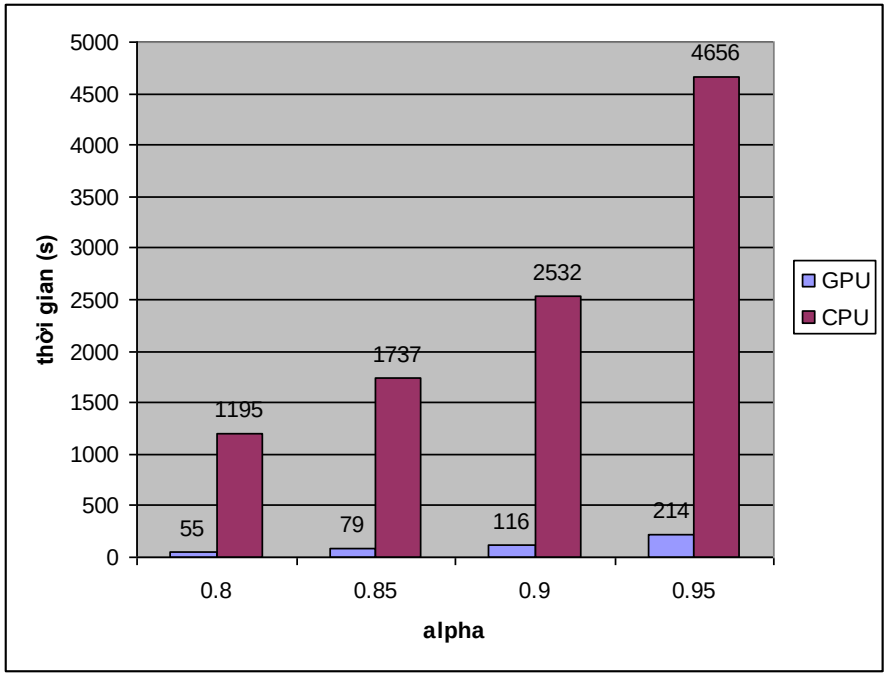
- The web is huge
  - Very large data size (millions to billions of web pages)
- The web is dynamic
  - Webpages always change (size and structure)

⇒ Require computation in a short time and continuously

⇒ Require huge computing performance

Google's PageRank on GPU

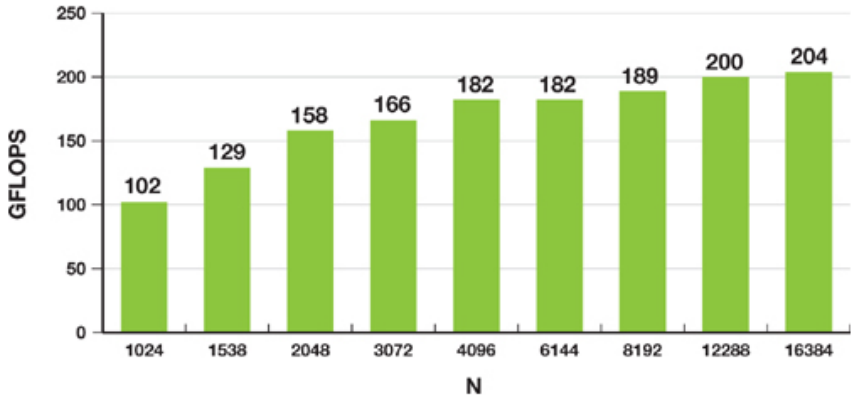
- When compared with a quad-core CPU implementation, speed up reach 21-22 x



Applying GP-GPU techonology in PageRank Computation – Msc Thesic, Pham Nguyen Quang Anh, HUST, 2010

Other Applications

- All-Pairs N-Body Simulation:
  - approximates the evolution of a system of bodies in which each body continuously interacts with every other body
  - On GeForce 8800 GTX GPU



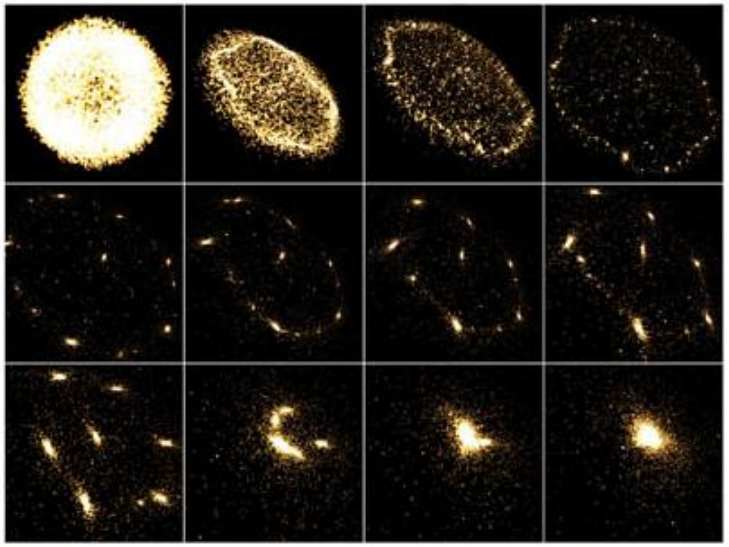


Figure 31-1 Frames from an Interactive 3D Rendering of a 16,384-Body System

http://http.developer.nvidia.com/GPUGems3/gpugems3_ch31.html

Supercomputers

- http://www.top500.org/
- The first supercomputer using GPU
  - 2009, Tsubame, Japan:
    - 170 x Tesla 1U (680 GPU), 77.48 TFLOP
    - Established in **one week !**
    - the 29th in top 500
- 2010: 11/500 supercomputers equipped GPUs
- 2011: 37/500 supercomputer in top500 use GPUs
  - Tianhe-1A, China
    - 2nd in top 500, 2.566 petaFLOPS
    - uses 7,168 Nvidia GPUs, 14,336 Intel CPUs

Summary

- GPU computing solutions is very effective
- Providing both hardware and software
- Very cost-effective solutions compared to CPU and GRID/ cluster
- Trend
  - More cores on-chip
  - Better support for float point
  - Flexiber configuration & control/data flow
  - Lower price
  - Support higher level programming language

THANK YOU