Graphics Processing Unit
Introduction

What is GPU?

- It is a processor optimized for 2D/3D graphics, video, visual computing, and display.
- It is highly parallel, highly multithreaded multiprocessor optimized for visual computing.
- It provide real-time visual interaction with computed objects via graphics images, and video.
- It serves as both a programmable graphics processor and a scalable parallel computing platform.

Heterogeneous Systems: combine a GPU with a CPU, etc.
GPU Evolution

• 1980’s – No GPU. PC used VGA controller
• 1990’s – Add more function into VGA controller
• 1997 – 3D acceleration functions:
  Hardware for triangle setup and rasterization
  Texture mapping
  Shading
• 2000 – A single chip graphics processor (beginning of GPU term)
• 2005 – Massively parallel programmable processors
• 2007 – CUDA (Compute Unified Device Architecture)
GPU Graphic Trends

• OpenGL – an open standard for 3D programming
• DirectX – a series of Microsoft multimedia programming interfaces
• New GPU are being developed every 12 to 18 months
• New idea of visual computing: combines graphics processing and parallel computing
• Heterogeneous System – CPU + GPU
• GPU evolves into scalable parallel processor
• GPU Computing: GPGPU and CUDA
• GPU unifies graphics and computing
• GPU visual computing application: OpenGL, and DirectX
GPU System Architectures

- CPU-GPU system architecture
  - The Historical PC
  - contemporary PC with Intel and AMD CPUs
- Graphics Logical Pipeline
- Basic Unified GPU Architecture
  - Processor Array
FIGURE A.2.1 Historical PC. VGA controller drives graphics display from framebuffer memory. Copyright © 2009 Elsevier, Inc. All rights reserved.
FIGURE A.2.2 Contemporary PCs with Intel and AMD CPUs. See Chapter 6 for an explanation of the components and interconnects in this figure. Copyright © 2009 Elsevier
**FIGURE A.2.3 Graphics logical pipeline.** Programmable graphics shader stages are blue, and fixed-function blocks are white. Copyright © 2009 Elsevier, Inc. All rights reserved.
Basic Unified GPU Architecture

FIGURE A.2.4 Logical pipeline mapped to physical processors. The programmable shader stages execute on the array of unified processors, and the logical graphics pipeline dataflow recirculates through the processors. Copyright © 2009 Elsevier, Inc. All rights reserved.
FIGURE A.2.5 Basic unified GPU architecture. Example GPU with 112 streaming processor (SP) cores organized in 14 streaming multiprocessors (SMs); the cores are highly multithreaded. It has the basic Tesla architecture of an NVIDIA GeForce 8800. The processors connect with four 64-bit-wide DRAM partitions via an interconnection network. Each SM has eight SP cores, two special function units (SFUs), instruction and constant caches, a multithreaded instruction unit, and a shared memory. Copyright © 2009 Elsevier, Inc. All rights reserved.
Compare CPU and GPU

Nemo-3D

• Written by the CalTech Jet Propulsion Laboratory
• NEMO-3D simulates quantum phenomena.
• These models require a lot of matrix operations on very large matrices.
• We are modifying the matrix operation functions so they use CUDA instead of that slow CPU.
Nemo-3D

Simulation

NEMO-3D

Computation Module

CUDA kernel

Visualization

VoIQD

CUDA kernel
Testing - Matrices

• Test the multiplication of two matrices.
• Creates two matrices with random floating point values.
• We tested with matrices of various dimensions…
### Results:

<table>
<thead>
<tr>
<th>Dim \ Time</th>
<th>CUDA</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>64x64</td>
<td>0.417465 ms</td>
<td>18.0876 ms</td>
</tr>
<tr>
<td>128x128</td>
<td>0.41691 ms</td>
<td>18.3007 ms</td>
</tr>
<tr>
<td>256x256</td>
<td>2.146367 ms</td>
<td>145.6302 ms</td>
</tr>
<tr>
<td>512x512</td>
<td>8.093004 ms</td>
<td>1494.7275 ms</td>
</tr>
<tr>
<td>768x768</td>
<td>25.97624 ms</td>
<td>4866.3246 ms</td>
</tr>
<tr>
<td>1024x1024</td>
<td>52.42811 ms</td>
<td>66097.1688 ms</td>
</tr>
<tr>
<td>2048x2048</td>
<td>407.648 ms</td>
<td><strong>Didn’t finish</strong></td>
</tr>
<tr>
<td>4096x4096</td>
<td>3.1 seconds</td>
<td><strong>Didn’t finish</strong></td>
</tr>
</tbody>
</table>
In visible terms:

**CPU versus GPU**

- **CPU regression**
  - $y = 10.682e^{0.0085x}$
  - $R^2 = 0.9813$

- **CUDA regression**
  - $y = 0.3526e^{0.0053x}$
  - $R^2 = 0.9575$
Test results:

![Graph showing function execute time vs number of atoms. The graph includes two trendlines:
- The red trendline represents the CPU with the equation \( y = 0.0228x - 0.5522 \) and \( R^2 = 1 \).
- The green trendline represents CUDA with the equation \( y = 0.0015x - 0.3449 \) and \( R^2 = 0.9996 \).]