GPU Computing

Numerical Simulation - from Models to Software

Andreas Barthels

JASS 2009, Course 2, St. Petersburg, Russia

Prof. Dr. Sergey Y. Slavyanov
St. Petersburg State University

Prof. Dr. Thomas Huckle
Technische Universität München

March 29 – April 7, 2009

Outline

Introduction

GPU Hardware

- highly parallel
- 1 Tflops for $<$120 € ($\triangleq$ 5,500 RUB)
- increasingly easy to program

This talk:

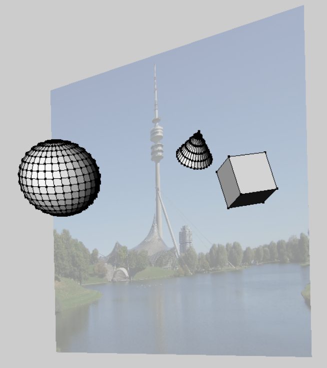
Applicability to numerical simulation and immediate visualization?

Outline

GPU Functionality I

Standard Tasks

- display of graphical models and scenes on a computer
- works with polygons, vertices, textures



GPU Functionality II

Standard Task: Shading of Pixels

- The same code fragment for every pixel ($\Rightarrow$*SPMD*)
- Typically a pixel's value is independent on the others'
- $\Rightarrow$ massive gain through parallelization

Recent Extensions

- increased programmability for more realistic visual effects
- simulation of rigid, deformable or fluid objects in game physics
- $\Rightarrow$ general purpose programmability

GPU Functionality III

Programming Interfaces & Terminology

DirectX

- Vertex Shader
- Pixel Shader
- Geometry Shader

OpenGL

- Vertex Shader
- Fragment Shader
- (Geometry Shader)

GPU Functionality IV

Demo of AMD RenderMonkey

Outline

GPU–CPU Comparison

x86 CPU

- Multi Purpose
- OS Instructions
- One ALU per Core
- Mild SIMD per Core
- Slow Memory
- Big Cache

GPU

- Specialized to Graphics
- No OS Concepts
- +100 Parallel ALUs
- Extensive MIMD
- Fast Memory
- Small Cache

Peak Performance

AMD/ATI Radeon HD 4870 – ≈180 €

- 1,200 MADD Gflops (single precision) – 240 Gflops
- 115.2 GBytes/s Memory Bandwidth

NVIDIA Geforce GTX 285 – ≈330 €

- 1,063 MADD Gflops (single precision) – 78 Gflops
- 159.0 GBytes/s Memory Bandwidth

Intel Core i7 965 XE – ≈1000 €

- ≈70 M+ADD Gflops (single precision) – ≈35 Gflops
- ≈25.6 GBytes/s Memory Bandwidth

($\Longrightarrow$ What about double precision?)

Peak Performance

AMD/ATI Radeon HD 4870 – ≈180 €

- 1,200 MADD Gflops (single precision) – 240 Gflops
- 115.2 GBytes/s Memory Bandwidth

NVIDIA Geforce GTX 285 – ≈330 €

- 1,063 MADD Gflops (single precision) – 78 Gflops
- 159.0 GBytes/s Memory Bandwidth

Intel Core i7 965 XE – ≈1000 €

- ≈70 M+ADD Gflops (single precision) – ≈35 Gflops
- ≈25.6 GBytes/s Memory Bandwidth

(⟹ What about double precision?)

Peak Performance

AMD/ATI Radeon HD 4870 – ≈180 €

- 1,200 MADD Gflops (single precision) – 240 Gflops
- 115.2 GBytes/s Memory Bandwidth

NVIDIA Geforce GTX 285 – ≈330 €

- 1,063 MADD Gflops (single precision) – 78 Gflops
- 159.0 GBytes/s Memory Bandwidth

Intel Core i7 965 XE – ≈1000 €

- ≈70 M+ADD Gflops (single precision) – ≈35 Gflops
- ≈25.6 GBytes/s Memory Bandwidth

($\Longrightarrow$ What about double precision?)

Outline

BrookGPU I

- BrookGPU was the first GPGPU framework
- Developed at Stanford
- Extension to C
- Vector datatypes:
  - `float, float2, float3, float4`
  - `int, int2, int3, int4`
  - `...`
  - `double, double2`

BrookGPU II

- Streams

  `float s<10, 10>;` – 2 dimensional 10x10 float matrix

  Access to stream items is restricted and regularized

- Kernels

```
kernel void k(float s<>, float3 f,
    float a[10][10], out float o<>) {...
```

BrookGPU III

- Reduction Operations on Streams

```
void reduce sum (float a<>,
           reduce float result<>)
                 result = result + a;
```

- Scatter & Gather Ops on Streams

```
    streamScatterOp(s,...)
    streamGatherOp(s,...)
```

Outline

AMD/ATI: Stream SDK I

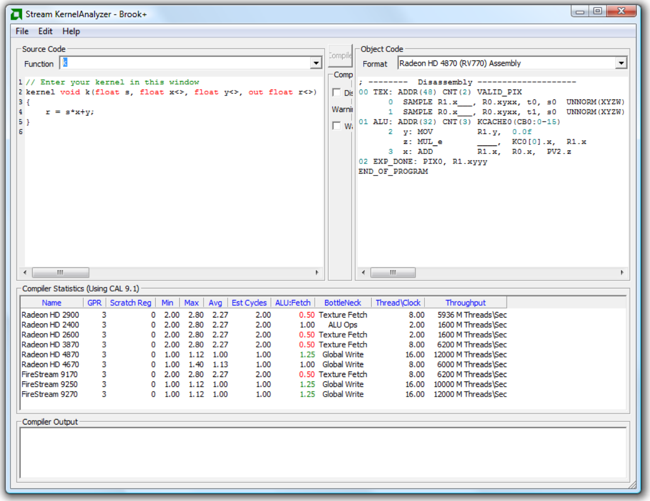
Custom Operations

AMD relies on Brook+, an extension to BrookGPU

AMD Core Math Library

- Optimized implementation of BLAS
- Can use multiple CPU cores and multiple GPUs

AMD/ATI: Stream SDK II



Demo of Brook+ and AMD Stream KernelAnalyzer

NVIDIA: CUDA SDK

Custom Operations

CUDA is very similar to Brook (streams, kernels, . . .)

Math Libraries

- CUBLAS: BLAS implemented in CUDA
- CUFFT: Fourier Transforms on GPU

Outline

Applications using GPU Computing

- Image processing
- Video de-/encoding
- Protein folding (folding@home)
- SETI@home

Outline

2D Simplified Fluid Simulation I

Denote u to be the height of the water surface at a certain point in time and space and let c be the wave speed:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

2D Simplified Fluid Simulation II

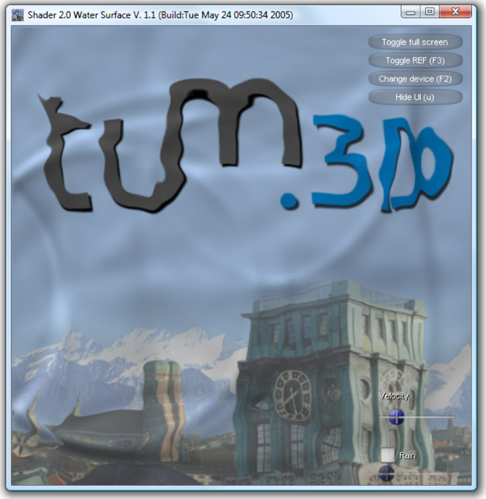
Discretizing space and time each equidistantally by Δt and h:

$$\frac{u_{i,j}^{t+1} - 2u_{i,j}^{t} + u_{i,j}^{t-1}}{\Delta t^2} = c^2 \left(\frac{u_{i+1,j}^{t} + u_{i-1,j}^{t} + u_{i,j+1}^{t} + u_{i,j-1}^{t} - 4u_{i,j}^{t}}{h^2} \right)$$

Applying $u_{i,j}^{t} := 0.5 \cdot \left(u_{i,j}^{t+1} + u_{i,j}^{t} \right)$ yields system of linear equations.

Interactive Demo

2D simplified fluid simulation using conjugate gradient solver



Outline

DirectX 11

- DirectX drives the industry standard for GPUs
- DX 11 compliant hardware is designed to support GPGPU
- Object Oriented Programming in “Compute Shader”
- Great increase in flexibility
- Increased double precision performance
- Wait until Windows 7

Outline

OpenCL

- Open standard, driven by Khronos group

- Not Object Oriented, just C, like Brook et al.
- Interoperability with OpenGL
- Abstraction layer for any type of hardware

Outline

Heterogeneous Computing

- Software support via OpenCL
- GPU and CPU combined hardware soon to come (⇒Intel Larrabee, AMD Fusion)

Outline

PDMI

Summary

- GPUs feature high performance for little money
- Capable of descent immediate visualization
- Upcoming standards in hardware & APIs are promising

Thank you