NVIDIA GPU Architecture
for General Purpose Computing

Anthony Lippert
4/27/09
Outline

- Introduction
- GPU Hardware
- Programming Model
- Performance Results
- Supercomputing Products
- Conclusion
Introduction

GPU: Graphics Processing Unit

- Hundreds of Cores
- Programmable
- Can be easily installed in most desktops
- Similar price to CPU
- GPU follows Moore's Law better than CPU
Introduction

Motivation:

![Graph showing the performance of NVIDIA GPUs and Intel CPUs over time. The graph includes key points such as NV35, NV40, G70, G71, G80 Ultra, G92, GT200, and Harpertown. The peak GFLOP/s values are labeled for each point.]
GPU Hardware

Multiprocessor Structure:
GPU Hardware

Multiprocessor Structure:

- N multiprocessors with M cores each
- SIMD – Cores share an Instruction Unit with other cores in a multiprocessor.
- Diverging threads may not execute in parallel.
GPU Hardware

Memory Hierarchy:

- Processors have 32-bit registers
- Multiprocessors have shared memory, constant cache, and texture cache
- Constant/texture cache are read-only and have faster access than shared memory.
GPU Hardware
NVIDIA GTX280 Specifications:

- 933 GFLOPS peak performance
- 10 thread processing clusters (TPC)
- 3 multiprocessors per TPC
- 8 cores per multiprocessor
- 16384 registers per multiprocessor
- 16 KB shared memory per multiprocessor
- 64 KB constant cache per multiprocessor
- 6 KB < texture cache < 8 KB per multiprocessor
- 1.3 GHz clock rate
- Single and double-precision floating-point calculation
- 1 GB DDR3 dedicated memory
GPU Hardware

GeForce GTX 280 Parallel Computing Architecture

- Thread Scheduler
- Thread Processing
- Clusters
- Atomic/Tex L2
- Memory
GPU Hardware

Thread Scheduler:

- Hardware-based
- Manages scheduling threads across thread processing clusters
- Nearly 100% utilization: If a thread is waiting for memory access, the scheduler can perform a zero-cost, immediate context switch to another thread
- Up to 30,720 threads on the chip
GPU Hardware

Thread Processing Cluster:

IU - instruction unit
TF - texture filtering

GeForce GTX 280 Parallel Computing Architecture
GPU Hardware

Atomic/Tex L2:

- Level 2 Cache
- Shared by all thread processing clusters
- Atomic
  - Ability to perform read-modify-write operations to memory
  - Allows granular access to memory locations
  - Provides parallel reductions and parallel data structure management
# GPU Hardware

<table>
<thead>
<tr>
<th>Features</th>
<th>8800 GTX</th>
<th>GTX 280</th>
<th>% Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores</td>
<td>128</td>
<td>240</td>
<td>87.5 %</td>
</tr>
<tr>
<td>TEX</td>
<td>64t/clk</td>
<td>80t/clk</td>
<td>25 %</td>
</tr>
<tr>
<td>ROP Blend</td>
<td>12p/clk</td>
<td>32p/clk</td>
<td>167 %</td>
</tr>
<tr>
<td>Precision</td>
<td>fp32</td>
<td>fp64</td>
<td>--</td>
</tr>
<tr>
<td>GFLOPs</td>
<td>518</td>
<td>933</td>
<td>80 %</td>
</tr>
<tr>
<td>FB Bandwidth</td>
<td>86 GB</td>
<td>142 GB</td>
<td>65 %</td>
</tr>
<tr>
<td>Texture Fill</td>
<td>37 GT/s</td>
<td>48 GT/s</td>
<td>29.7 %</td>
</tr>
<tr>
<td>ROP Blend</td>
<td>7 GBL/s</td>
<td>19 GBL/s</td>
<td>171 %</td>
</tr>
<tr>
<td>PCI Express</td>
<td>6.4 GB</td>
<td>12.8 GB</td>
<td>100 %</td>
</tr>
<tr>
<td>Video</td>
<td>VP1</td>
<td>VP2</td>
<td>--</td>
</tr>
</tbody>
</table>
GPU Hardware

GT200 Power Features:

- Dynamic power management
- Power consumption is based on utilization
  - Idle/2D power mode: 25 W
  - Blu-ray DVD playback mode: 35 W
  - Full 3D performance mode: worst case 236 W
  - HybridPower mode: 0 W

- On an nForce motherboard, when not performing, the GPU can be powered off and computation can be diverted to the motherboard GPU (mGPU)
GPU Hardware

- 10 Thread Processing Clusters (TPC)
- 3 multiprocessors per TPC
- 8 cores per multiprocessor
- ROP – raster operation processors (for graphics)
- 1024 MB frame buffer for displaying images
- Texture (L2) Cache
Programming Model

Past:

- The GPU was intended for graphics only, not general purpose computing.
- The programmer needed to rewrite the program in a graphics language, such as OpenGL
- Complicated

Present:

- NVIDIA developed CUDA, a language for general purpose GPU computing
- Simple
Programming Model

CUDA:

- Compute Unified Device Architecture
- Extension of the C language
- Used to control the device
- The programmer specifies CPU and GPU functions
  - The host code can be C++
  - Device code may only be C
- The programmer specifies thread layout
Programming Model

Thread Layout:

- Threads are organized into blocks.
- Blocks are organized into a grid.
- A multiprocessor executes one block at a time.
- A *warp* is the set of threads executed in parallel.
- 32 threads in a warp
Programming Model

- Heterogeneous Computing:
  - GPU and CPU execute different types of code.
  - CPU runs the main program, sending tasks to the GPU in the form of kernel functions.
  - Multiple kernel functions may be declared and called.
  - Only one kernel may be called at a time.
Programming Model: GPU vs. CPU Code


**CPU C program**

```c
void add_matrix_cpu(
    float *a, float *b, float *c, int N)
{
    int i, j, index;
    for (i=0;i<N;i++) {
        for (j=0;j<N;j++) {
            index = i+j*N;
            c[index] = a[index] + b[index];
        }
    }
}
void main()
{
    ....
    add_matrix(a,b,c,N);
}
```

**CUDA C program**

```c
__global__ void add_matrix_gpu(
    float *a, float *b, float *c, int N)
{
    int i=blockIdx.x*blockDim.x+threadIdx.x;
    int j=blockIdx.y*blockDim.y+threadIdx.y;
    int index = i+j*N;
    if ( i < N && j < N) c[index] = a[index] + b[index];
}
void main()
{
    ....
    add_matrix(a,b,c,N);
    add_matrix_gpu<<<dimGrid,dimBlock>>>(a,b,c,N);
}
```
Performance Results

Speedups Using GPU vs CPU

146X
Interactive visualization of volumetric white matter connectivity

36X
Ionic placement for molecular dynamics simulation on GPU

18X
Transcoding HD video stream to H.264 for portable video

17X
Simulation in Matlab using .mex file CUDA function

100X
Astrophysics N-body simulation

149X
Financial simulation of LIBOR model with interpolations

47X
GLAME@lab: M-script API for linear Algebra operations

20X
Ultrasound medical imaging for cancer

24X
Highly optimized object oriented molecular

30X
Cmatch exact string matching - find similar matching string
Supercomputing Products

Tesla C1060 GPU
933 GFLOPS

nForce Motherboard

Tesla C1070 Blade 4.14 TFLOPS
Supercomputing Products

Tesla C1060:
- Similar to GTX 280
- No video connections
- 933 GFLOPS peak performance
- 4 GB DDR3 dedicated memory
- 187.8 W max power consumption
Supercomputing Products

Tesla C1070:
- Server Blade
- 4.14 TFLOPS peak performance
- Contains 4 Tesla GPUs
- 960 Cores
- 16GB DDR3
- 408 GB/s bandwidth
- 800W max power consumption
Conclusion

- SIMD causes some problems
- GPU computing is a good choice for fine-grained data-parallel programs with limited communication
- GPU computing is not so good for coarse-grained programs with a lot of communication
- The GPU has become a co-processor to the CPU
References


nvidia.com
