HIGH-PERFORMANCE COMPUTING WITH NVIDIA TESLA GPUS

Timothy Lanfear, NVIDIA
WHY GPU COMPUTING?
Science is Desperate for Throughput

Gigaflops

1 Exaflop

1 Petaflop

1,000,000,000

1,000,000

1,000

1


BPTI 3K atoms

Estrogen Receptor 36K atoms

F1-ATPase 327K atoms

Ribosome 2.7M atoms

Chromatophore 50M atoms

Bacteria 100s of Chromatophores

Ran for 8 months to simulate 2 nanoseconds
Power Crisis in Supercomputing

- Exaflop: 25,000,000 Watts
- Petaflop: 7,000,000 Watts
- Teraflop: 850,000 Watts
- Gigaflop: 60,000 Watts

Devices:
- Jaguar, Los Alamos
Double Performance per Watt

- Jaguar x86 CPU
- Tesla GPU
- Nebulae
- JUGENE
- BlueGene
- Roadrunner Cell
- IPE, CAS
- Tesla GPU

Linpack Performance (Teraflops) vs. Power (MWatts)
What is GPU Computing?

Computing with CPU + GPU

Heterogeneous Computing
Low Latency or High Throughput?

**CPU**
- Optimised for low-latency access to cached data sets
- Control logic for out-of-order and speculative execution

**GPU**
- Optimised for data-parallel, throughput computation
- Architecture tolerant of memory latency
- More transistors dedicated to computation
Why Didn’t GPU Computing Take Off Sooner?

- **GPU Architecture**
  - Gaming oriented, process pixel for display
  - Single threaded operations
  - No shared memory

- **Development Tools**
  - Graphics oriented (OpenGL, GLSL)
  - University research (Brook)
  - Assembly language

- **Deployment**
  - Gaming solutions with limited lifetime
  - Expensive OpenGL professional graphics boards
  - No HPC compatible products
NVIDIA Invested in GPU Computing in 2004

- **Strategic move for the company**
  - Expand GPU architecture beyond pixel processing
  - Future platforms will be hybrid, multi/many cores based

- **Hired key industry experts**
  - x86 architecture
  - x86 compiler
  - HPC hardware specialist

Create a GPU based Compute Ecosystem by 2008
NVIDIA GPU Product Families

GeForce®
Entertainment

Tesla™
High-Performance Computing

Quadro®
Design & Creation
Fermi: The Computational GPU

**Performance**
- More than ½ Teraflop 64-bit performance
- IEEE 754-2008 SP and DP Floating Point

**Flexibility**
- Increased Shared Memory from 16 KB to 64 KB
- Added L1 and L2 Caches
- ECC on all Internal and External Memories
- Enable up to 1 TeraByte of GPU Memories
- High Speed GDDR5 Memory Interface

**Usability**
- Multiple Simultaneous Tasks on GPU
- 10× Faster Atomic Operations
- C++ Support
- System Calls, printf support
NVIDIA Tesla GPU Computing Products

Data Center Products

<table>
<thead>
<tr>
<th>GPUs</th>
<th>Tesla M1060</th>
<th>Tesla M2050</th>
<th>Tesla S2070</th>
<th>Tesla 2050</th>
<th>Tesla S1070</th>
<th>Tesla C2070</th>
<th>Tesla C2050</th>
<th>Tesla C1060</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 T10 GPU</td>
<td>1 T20 GPU</td>
<td>4 T20 GPUs</td>
<td>4 T10 GPUs</td>
<td></td>
<td>1 T20 GPU</td>
<td>1 T10 GPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single Precision</td>
<td>933 GFlops</td>
<td>1030 GFlops</td>
<td>4120 GFlops</td>
<td>4140 GFlops</td>
<td>1030 Gflops</td>
<td>933 GFlops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Double Precision</td>
<td>78 GFlops</td>
<td>515 GFlops</td>
<td>2060 GFlops</td>
<td>346 GFlops</td>
<td>515 Gflops</td>
<td>78 GFlops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB</td>
<td>3 GB</td>
<td>12 GB</td>
<td>24 GB</td>
<td>16 GB</td>
<td>6 GB</td>
<td>3 GB</td>
<td>4 GB</td>
</tr>
<tr>
<td>Mem BW</td>
<td>102 GB/s</td>
<td>148.4 GB/s</td>
<td>148.4 GB/s</td>
<td>102 GB/s</td>
<td></td>
<td>144 GB/s</td>
<td>102 GB/s</td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td>No display IO</td>
<td>No display IO</td>
<td>No display IO</td>
<td>No display IO</td>
<td>Single dual-link DVI</td>
<td>No display IO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Workstation

<table>
<thead>
<tr>
<th>GPUs</th>
<th>Tesla M1060</th>
<th>Tesla M2050</th>
<th>Tesla S2070</th>
<th>Tesla 2050</th>
<th>Tesla S1070</th>
<th>Tesla C2070</th>
<th>Tesla C2050</th>
<th>Tesla C1060</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 T10 GPU</td>
<td>1 T20 GPU</td>
<td>4 T20 GPUs</td>
<td>4 T10 GPUs</td>
<td></td>
<td>1 T20 GPU</td>
<td>1 T10 GPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single Precision</td>
<td>933 GFlops</td>
<td>1030 GFlops</td>
<td>4120 GFlops</td>
<td>4140 GFlops</td>
<td>1030 Gflops</td>
<td>933 GFlops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Double Precision</td>
<td>78 GFlops</td>
<td>515 GFlops</td>
<td>2060 GFlops</td>
<td>346 GFlops</td>
<td>515 Gflops</td>
<td>78 GFlops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB</td>
<td>3 GB</td>
<td>12 GB</td>
<td>24 GB</td>
<td>16 GB</td>
<td>6 GB</td>
<td>3 GB</td>
<td>4 GB</td>
</tr>
<tr>
<td>Mem BW</td>
<td>102 GB/s</td>
<td>148.4 GB/s</td>
<td>148.4 GB/s</td>
<td>102 GB/s</td>
<td></td>
<td>144 GB/s</td>
<td>102 GB/s</td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td>No display IO</td>
<td>No display IO</td>
<td>No display IO</td>
<td>No display IO</td>
<td>Single dual-link DVI</td>
<td>No display IO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
OEM Servers with Tesla M2050 GPUs
Announced on May 4th, 2010

2 Tesla M2050 GPUs

SuperServer 6016GT-TF
2 CPUs + 2 GPUs in 1U

4 Tesla M2050 GPUs

Appro Tetra
2 CPUs + 4 GPUs in 1U

10 Tesla M2050 GPUs

Appro GreenBlade
10 CPUs + 10 GPUs in 5U

8 Tesla M2050 GPUs

Tyan B7015
2 CPUs + 8 GPUs in 4U

….. many more coming soon …..
OEM Servers with Tesla M1060 GPUs

- **SuperServer 6016GT-TF**: 2 CPUs + 2 GPUs in 1U
- **Cray CX1000 and Bull Bullx**: 36 CPUs + 18 GPUs in 7U
- **Tyan B7015**: 2 CPUs + 8 GPUs in 4U

© NVIDIA Corporation 2009
CUDA Parallel Computing Architecture

- Parallel computing architecture and programming model
- Includes a CUDA C compiler, support for OpenCL and DirectCompute
- Architected to natively support multiple computational interfaces (standard languages and APIs)

GPU Computing Application

| C       | C++      | Fortran  | Java   | C#     | ...
|---------|----------|----------|--------|--------|--------
| CUDA C  | OpenCL™  | DirectCompute | CUDA Fortran

NVIDIA GPU with the CUDA parallel computing architecture
Application Software (written in C)

<table>
<thead>
<tr>
<th>CUDA Libraries</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cuFFT</td>
<td>cuBLAS</td>
<td>cuDPP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU Hardware</th>
<th>CUDA Compiler</th>
<th>CUDA Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>1U PCI-E Switch</td>
<td>C Fortran</td>
<td>Debugger</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Profiler</td>
</tr>
</tbody>
</table>

4 cores 240 cores
NVIDIA Parallel Nsight™

The first development environment for massively parallel applications.

- **Hardware** GPU Source Debugging
- **Platform-wide** Analysis
- **Complete** Visual Studio integration

Register for the Beta
http://developer.nvidia.com/nsight
Allinea DDT — CUDA Enabled

GPU Debugging
Making it easy
Allinea DDT — CUDA Enabled
TotalView for CUDA
<table>
<thead>
<tr>
<th>146X</th>
<th>36X</th>
<th>19X</th>
<th>17X</th>
<th>100X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive visualization of volumetric white matter connectivity</td>
<td>Ion placement for molecular dynamics simulation</td>
<td>Transcoding HD video stream to H.264</td>
<td>Simulation in Matlab using mex file CUDA function</td>
<td>Astrophysics N-body simulation</td>
</tr>
<tr>
<td>149X</td>
<td>47X</td>
<td>20X</td>
<td>24X</td>
<td>30X</td>
</tr>
<tr>
<td>Financial simulation of LIBOR model with swaptions</td>
<td>GLAMERlab: An M-script API for linear Algebra operations on GPU</td>
<td>Ultrasound medical imaging for cancer diagnostics</td>
<td>Highly optimized object oriented molecular dynamics</td>
<td>Cmatch exact string matching to find similar proteins and gene sequences</td>
</tr>
</tbody>
</table>
What We Did in the Past Three Years

2006
- G80, first GPU with built-in compute features, 128 core multi-threaded, scalable architecture
- CUDA SDK Beta

2007
- Tesla HPC product line
- CUDA SDK 1.0, 1.1

2008
- GT200, second GPU generation, 240 core, 64-bit
- Tesla HPC second generation
- CUDA SDK 2.0

2009 ...
NEXT-GENERATION GPU ARCHITECTURE — ‘FERMI’
Introducing the ‘Fermi’ Architecture

The Soul of a Supercomputer in the body of a GPU

- 3 billion transistors
- Over 2× the cores (512 total)
- 8× the peak DP performance
- ECC
- L1 and L2 caches
- ~2× memory bandwidth (GDDR5)
- Up to 1 Terabyte of GPU memory
- Concurrent kernels
- Hardware support for C++
Design Goal of Fermi

Expand performance sweet spot of the GPU

Bring more users, more applications to the GPU

Data Parallel

Instruction Parallel

Many Decisions

Large Data Sets
Streaming Multiprocessor Architecture

- 32 CUDA cores per SM (512 total)
- 2:1 ratio SP:DP floating-point performance
- Dual Thread Scheduler
- 64 KB of RAM for shared memory and L1 cache (configurable)
CUDA Core Architecture

- New IEEE 754-2008 floating-point standard, surpassing even the most advanced CPUs
- Fused multiply-add (FMA) instruction for both single and double precision
- Newly designed integer ALU optimized for 64-bit and extended precision operations
Cached Memory Hierarchy

- First GPU architecture to support a true cache hierarchy in combination with on-chip shared memory

- L1 Cache per SM (32 cores)
  - Improves bandwidth and reduces latency

- Unified L2 Cache (768 KB)
  - Fast, coherent data sharing across all cores in the GPU

Parallel DataCache™ Memory Hierarchy
Larger, Faster Memory Interface

- GDDR5 memory interface
  - 2× speed of GDDR3
- Up to 1 Terabyte of memory attached to GPU
  - Operate on large data sets
Error Correcting Code

- ECC protection for
  - DRAM
    - ECC supported for GDDR5 memory
- All major internal memories are ECC protected
  - Register file, L1 cache, L2 cache
GigaThread™ Hardware Thread Scheduler

- Hierarchically manages thousands of simultaneously active threads
- 10x faster application context switching
- Concurrent kernel execution
GigaThread Hardware Thread Scheduler

Concurrent Kernel Execution + Faster Context Switch

Serial Kernel Execution

Parallel Kernel Execution
GigaThread Streaming Data Transfer Engine

- **Dual DMA engines**
  - Simultaneous CPU→GPU and GPU→CPU data transfer
  - Fully overlapped with CPU and GPU processing time

- **Activity Snapshot:**
  - Kernel 0: CPU, SDT0, GPU, SDT1
  - Kernel 1: CPU, SDT0, GPU, SDT1
  - Kernel 2: CPU, SDT0, GPU, SDT1
  - Kernel 3: CPU, SDT0, GPU, SDT1
Enhanced Software Support

- Full C++ Support
  - Virtual functions
  - Try/Catch hardware support

- System call support
  - Support for pipes, semaphores, printf, etc

- Unified 64-bit memory addressing
I believe history will record Fermi as a significant milestone.

Dave Patterson
Director Parallel Computing Research Laboratory, U.C. Berkeley
Co-Author of Computer Architecture: A Quantitative Approach

Fermi surpasses anything announced by NVIDIA's leading GPU competitor (AMD).

Tom Halfhill
Senior Editor
Microprocessor Report
Fermi is the world’s first complete GPU computing architecture.

Peter Glaskowsky
Technology Analyst
The Envisioneering Group

The convergence of new, fast GPUs optimized for computation as well as 3-D graphics acceleration and industry-standard software development tools marks the real beginning of the GPU computing era. Gentlemen, start your GPU computing engines.

Nathan Brookwood
Principle Analyst & Founder
Insight 64
A 2015 GPU *
- ~20× the performance of today’s GPU
- ~5,000 cores at ~3 GHz (50 mW each)
- ~20 TFLOPS
- ~1.2 TB/s of memory bandwidth

* This is a sketch of what a GPU in 2015 might look like; it does not reflect any actual product plans.