ECE 8823: GPU Architectures

Introduction

Objectives

• Distinguishing features of GPUs vs. CPUs

• Major drivers in the evolution of general purpose GPUs (GPGPUs)

• Set up the context for the course
  ❖ Programming model
  ❖ Execution model
  ❖ Architecture Model
Reading

- Chapter 1
- Chapter 2: 2.2, 2.3

What is a GPGPU?

- Graphics Processing Unit (GPU): (NVIDIA/AMD/Intel)
  - Many-core Architecture
  - Massively Data-Parallel Processor (Compared with a CPU)
  - Massively multi-threaded

- GPGPU:
  - General-Purpose GPU, High Performance Computing
  - Become popular with CUDA and OpenCL programming languages
Evolution from Graphics Pipelines

A fixed-function NVIDIA GeForce graphics pipeline.

Unified programmable processor array of the GeForce 8800 GT graphics pipeline.

Discrete GPUs in the System

© 2012 Elsevier, Inc. All rights reserved.
Fused GPUs: AMD & Intel

On-Chip and sharing the cache

Not as powerful as the discrete GPUs

Questions: memory model? Programming model?

Qualcomm Snapdragon

Road Map: NVIDIA

- All cores are not created equal
- Need to understand the programming model
NVIDIA GV100 Architecture

High Performance (domain specific) Acceleration

Why GPUs?
Post-Dennard Performance Scaling

\[ \text{Perf} = \text{Power} \times \text{Efficiency} \]

Performance limited by TDP

W. J. Dally, Keynote IITC 2012

Single-Core Era

- Enabled by:
  - Moore’s Law
  - Voltage Scaling
- Constrained by:
  - Power
  - Complexity

Multi-Core Era

- Enabled by:
  - Moore’s Law
  - SMP architecture
- Constrained by:
  - Power
  - Parallel SW
  - Scalability

Heterogeneous Systems Era

- Enabled by:
  - Abundant data parallelism
  - Power efficient GPUs
- Temporarily Constrained by:
  - Programming models
  - Comm. overhead

Phil Rogers, “Heterogeneous System Architecture Overview,” Hotchips Tutorial – August 2013
© Copyright 2012 HSA Foundation. All Rights Reserved.
Post Dennard Architecture Performance Scaling

\[
\text{Perf} \left( \frac{\text{ops}}{s} \right) = \text{Power}(W) \times \text{Efficiency} \left( \frac{\text{ops}}{\text{joule}} \right)
\]

CPU cores: \(\sim 2-4\) nJ/op
GPU cores: \(\sim 10s\) pJ/op

---

A Data Rich World

- Exponentially growing data sets
- Transform to exponential growth in performance
- → Throughput Computing
Memory Bandwidth

- Memory bandwidth growth lags throughput growth
- Hide memory latency with fine grained, massive multithreading
- Simplify and scale compute

Multithreaded Execution

Fine grained, instruction level multithreading to hide memory latency
CPUs vs. GPUs

CPU and GPU have very different design philosophy

**GPU**
Throughput Oriented Cores

- Compute Unit
- Cache/Local Mem
- Registers
- Threading
- SIMD Unit

**CPU**
Latency Oriented Cores

- Core
- Local Cache
- Registers
- SIMD Unit
- Control

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2012
ECE408/CS483, University of Illinois, Urbana-Champaign
CPUs: Latency Oriented Design

- Large caches
  - Convert long latency memory accesses to short latency cache accesses
- Sophisticated control
  - Branch prediction for reduced branch latency
  - Data forwarding for reduced data latency
- Powerful ALU
  - Reduced operation latency
- Small number of hardware threads

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2012
ECE408/CS483, University of Illinois, Urbana-Champaign

Top 500 Power Efficiency Trends

- Rapid Increase because ratio dependent on logic technology alone; more flops/s per socket
- Increasingly larger caches to recover memory bandwidth

Courtesy P. Kogge, UND
GPUs: Throughput Oriented Design

- Small caches
  - To boost memory throughput
- Simple control
  - No branch prediction
  - No data forwarding
- Energy efficient ALUs
  - Many, long latency but heavily pipelined for high throughput
- Require massive number of threads to tolerate memory latencies

Silicon Efficiency

- IBM Power 8
  - Optimized for speeding up single threads
  - Note silicon area devoted to compute!
- NVIDIA Volta - ~5K cores
  - Optimized for high thread count
Winning Applications Use Both CPU and GPU

- CPUs for sequential parts where latency matters
  - CPUs can be 10+X faster than GPUs for sequential code
- GPUs for parallel parts where throughput wins
  - GPUs can be 10+X faster than CPUs for parallel code

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2012
ECE408/CS483, University of Illinois, Urbana-Champaign

GPUs and High Performance Computing

- 299,088 Opteron cores
- 18,688 K20 GPUs (2496 cores/GPU)
- 710 Tbytes of memory

© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007-2012
ECE408/CS483, University of Illinois, Urbana-Champaign
Cray XK7 Compute Node

XK7 Compute Node Characteristics
- AMD Series 6200 (Interlagos)
- NVIDIA Kepler
- Host Memory: 32GB, 1600 MT/s DDR3
- NVIDIA Tesla X2090 Memory: 6GB GDDR5 capacity
- Gemini High Speed Interconnect
- Keplers in final installation

Amazon EC2 Instance

Amazon EC2 GPU Instances

<table>
<thead>
<tr>
<th>Elements</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>CentOS 5.5</td>
</tr>
<tr>
<td>CPU</td>
<td>2 x Intel Xeon X5570 (quad-core &quot;Nehalem&quot; arch, 2.93GHz)</td>
</tr>
<tr>
<td>GPU</td>
<td>2 x NVIDIA Tesla &quot;Fermi&quot; M2050 GPU Nvidia GPU driver and CUDA toolkit 3.1</td>
</tr>
<tr>
<td>Memory</td>
<td>22 GB</td>
</tr>
<tr>
<td>Storage</td>
<td>1690 GB</td>
</tr>
<tr>
<td>I/O</td>
<td>10 GbE</td>
</tr>
<tr>
<td>Price</td>
<td>$2.10/hour</td>
</tr>
</tbody>
</table>
Green 500 (2015)

<table>
<thead>
<tr>
<th>Rank</th>
<th>MFLOPS/W</th>
<th>Site*</th>
<th>Computer*</th>
<th>Total Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.031.59</td>
<td>RIKEN</td>
<td>Groun - ExaScaler-1.4 8EB max. Xeon ES-2680v3 16C 2.3GHz, Infiniband FDR, PE21-SC</td>
<td>60.32</td>
</tr>
<tr>
<td>2</td>
<td>5.042.31</td>
<td>High Energy Accelerator Research Organization (KEK)</td>
<td>Groun - ExaScaler-1.4 8EB max. Xeon ES-2680v3 16C 2.3GHz, Infiniband FDR, PE21-SC</td>
<td>28.25</td>
</tr>
<tr>
<td>3</td>
<td>2.217.04</td>
<td>High Energy Accelerator Research Organization (KEK)</td>
<td>Groun - ExaScaler-1.4 8EB max. Xeon ES-2680v3 16C 2.3GHz, Infiniband FDR, PE21-SC</td>
<td>35.39</td>
</tr>
<tr>
<td>4</td>
<td>5.271.81</td>
<td>UCI Helmholtz Center</td>
<td>Asus E5400 FDR/DFS, Intel Xeon ES-2580v3 12C 2.5GHz, Infiniband FDR, PE21-SC</td>
<td>67.16</td>
</tr>
<tr>
<td>5</td>
<td>4.207.08</td>
<td>USC Center, Tokyo Institute of Technology</td>
<td>TUBAWE-KFC - LX 9-B GPU 10440. Intel Xeon ES-2580v2 12C 2.5GHz, Infiniband FDR, NVIDIA K20</td>
<td>39.63</td>
</tr>
<tr>
<td>6</td>
<td>4.112.11</td>
<td>Stanford Research Computing Center</td>
<td>Xstream - Cerebro, Xeon ES-2580v2 12C 2.6GHz, Infiniband FDR, NVIDIA K20</td>
<td>190.00</td>
</tr>
<tr>
<td>7</td>
<td>3.952.73</td>
<td>Cray Inc.</td>
<td>Bluestar - Cray XE51-1, Xeon ES-2580v3 12C 2.3GHz, Infiniband FDR, NVIDIA K20</td>
<td>44.54</td>
</tr>
<tr>
<td>8</td>
<td>3.821.79</td>
<td>Cambridge University</td>
<td>Bills - Dell XE51-1, Xeon ES-2580v3 12C 2.3GHz, Infiniband FDR, NVIDIA K20</td>
<td>52.62</td>
</tr>
<tr>
<td>9</td>
<td>3.801.73</td>
<td>TU Dresden, ZIH</td>
<td>Turing GPU - Dell XE51-1, Xeon ES-2580v3 12C 2.3GHz, Infiniband FDR, NVIDIA K20</td>
<td>59.01</td>
</tr>
<tr>
<td>10</td>
<td>3.563.32</td>
<td>Quantitative Sciences and Computing Institute</td>
<td>DataScale - DSCC, Intel Xeon ES-2580v3 12C 2.6GHz, Infiniband FDR, NVIDIA K20</td>
<td>54.60</td>
</tr>
</tbody>
</table>

Green 500 (June 2017)

Efficiency has doubled!
<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>System</th>
<th>Cores</th>
<th>Rmax (TFlops)</th>
<th>Reak (TFlops)</th>
<th>Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>National Supercomputing Center - West</td>
<td>Sunway TaihuLight - Sunway</td>
<td>10,647,600</td>
<td>133,044.6</td>
<td>125,405.9</td>
<td>15,271</td>
</tr>
<tr>
<td></td>
<td>China</td>
<td>MP7, Sunway SW2030 240C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.4GHz, Sunway</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>N/P/PCI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>National Super Computer Center in</td>
<td>Tianhe-2 (MilkyWay-2) - TH</td>
<td>3,120,000</td>
<td>33,082.7</td>
<td>34,952.4</td>
<td>17,808</td>
</tr>
<tr>
<td></td>
<td>Guangzhou, China</td>
<td>HET, Peking University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ES-2690 12C 2.20GHz; TH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Express-2, Intel Xeon Phi</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>P100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVIDIA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Swiss National Supercomputing Center</td>
<td>Piz Daint - Cray XC50, Swiss</td>
<td>364,748</td>
<td>19,250.8</td>
<td>21,206.0</td>
<td>2,272</td>
</tr>
<tr>
<td></td>
<td>(CSCS) - Switzerland</td>
<td>EB-2690 12C 2.20GHz Lumina</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cray Inc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVIDIA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Japan Agency for Marine-Earth Science</td>
<td>Oakpeak - ZettaciSuper-2.2 HPC</td>
<td>19,910,000</td>
<td>19,125.8</td>
<td>28,190.0</td>
<td>1,250</td>
</tr>
<tr>
<td></td>
<td>and Technology - Japan</td>
<td>system, SGI Altix Blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>EB-2690 12C 1.4GHz Heron</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interconnect</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DOE/DOE Oak Ridge National Laboratory</td>
<td>Titan - Cray XT, Oakrion 6774</td>
<td>563,648</td>
<td>17,203.8</td>
<td>27,112.5</td>
<td>8,209</td>
</tr>
<tr>
<td></td>
<td>United States</td>
<td>2.20GHzDell PowerEdge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>100% interconnect</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Programming Model, Execution, and Scalability

---

(29)

(30)
Massive Parallelism

- How do you orchestrate correct computation?
  - Bulk synchronous parallel (BSP) execution model
  - A computer kernel may have tens of thousands of threads

- Execution: Single instruction multiple thread model (SIMT)
  - Not a SIMD model or vector model!
  - How are these threads organized?
Each kernel

N-Dimensional Range

Grid of threads, each operating over a data partition

Note: Each thread executes the same kernel code!

---

1-D Data Parallelism

Map data partitions to 1D grid of thread blocks

Each thread searches each group of records in parallel

---
2-D Data Parallelism

Map data partitions to 2D grid of thread blocks

Process each square in parallel – data parallel computation

3-D Data Parallelism

Map data partitions to 3D grid of thread blocks
Execution Model

NVIDIA's GPU Programming Model

CUDA Kernel
(1D-3D grid of TBs)

Thread Block (TB)

Warp (32 threads)

Branch divergence

Reconverge

Map thread blocks to cores

(37)

(38)
Parallel Programming Work Flow

- Identify compute intensive parts of an application → encapsulate in a kernel
- Map threads to data partitions
  - Optimize data arrangements to maximize locality (CPU vs. GPU)
- **Offload** kernel/data to GPU for execution
- Onload results to CPU
- Performance Tuning

Scalability & Portability

- Execution organized as **synchronous blocks** of threads
- Portable across core counts!
Virtual ISAs

Applications

Parallel Thread Execution (PTX) ISA

Native ISA-1++

Native ISA-1+

Native ISA-1

Gen1

Gen2

Gen3

Scalability and Portability

- Performance growth with HW generations
  - Increasing number of compute units
  - Increasing number of threads
  - Increasing vector length
  - Increasing pipeline depth
  - Increasing DRAM burst size
  - Increasing number of DRAM channels
  - Increasing data movement latency

- Portability across many different HW types
  - Multi-core CPUs vs. many-core GPUs
  - VLIW vs. SIMD vs. threading
  - Shared memory vs. distributed memory
Major Themes

- Heterogeneous Architectures
  - CPU + GPU organizations

- Programming Models
  - CUDA, OpenCL, OpenACC

- Massive Parallelism and BSP Execution Model
  - Base Microarchitecture
  - Optimizations
  - Memory Hierarchy

QUESTIONS?