General Purpose computing on Graphical Processing Units (GPGPU)

Adapted from Supercomputing Workshop

Terms

> What is GPGPU?
  * General-Purpose computing on a Graphics Processing Unit
  * Idea: use graphics hardware for non-graphical applications (computation peripheral)

> What is CUDA?
  * Compute Unified Device Architecture
  * Software architecture/framework for managing data-parallel programming
Motivation

CPU vs. GPU

- **CPU**
  - On-chip caches
  - Branching adaptability
  - Extensive instruction set

- **GPU**
  - Multiple ALUs
  - Fast onboard memory
  - High throughput on parallel tasks
    - Executes "program" on each fragment/vertex

- CPUs are great for **task** parallelism
- GPUs are great for **data** parallelism
CPU vs. GPU - Hardware

- GPU: more transistors devoted to data processing

Traditional Graphics Pipeline

- Vertex processing
  - Rasterizer
  - Fragment processing
  - Renderer (textures)
Unified (Programmable) Architecture

Unified Design

Discrete Design
- Shader A
- Shader B
- Shader C
- Shader D

Unified Design

Pixel ↔ Thread Processing

Input Registers
Fragment Program
- Texture
- Constants
- Registers
Output Registers

Thread Number
Thread Program
- Parallel Data Cache
- Texture
- Constants
- Registers
Global Memory
GPU Architectural Model

Processing Element

- Processing element = thread processor = ALU
Memory Architecture

- Device Memory
- Shared Memory
- Constant Memory
  - Read-only
- Texture Memory
  - Read-only
  - Spatial locality

Data-parallel Programming

- Think of the GPU as a massively-threaded co-processor (a compute engine)
- Write “kernel” functions that execute on the device -- processing multiple data elements in parallel

Rules of thumb
- Keep GPU busy ⇔ massive threading
- Keep your data close ⇔ local memory
Hardware Requirements

- CUDA/OpenCL-capable video card(s)
- Ample power supply (~1000 W)
- Sufficient cooling
- High-speed bus (PCI-Express 3.0 x16)

GPU Specifications

- Installation directory (Seawolf)
  /usr/local/cuda/8.0

- GPU specs utility
  samples/1_Utilities/deviceQuery/
  ./deviceQuery
Acknowledgements

- Andrew Bellanir: S³ Project
  GVSU Scholarworks Repository (on-campus access)
  http://scholarworks.gvsu.edu/sss/9/

- NVidia Corporation
  developer.nvidia.com/cuda-zone
  • CUDA Tools / Resources / Documentation