GPU Teaching Kit

Accelerated Computing

Lecture 1.1 – Course Introduction

Course Introduction and Overview

Course Goals

- Learn how to program heterogeneous parallel computing systems and achieve
  - High performance and energy-efficiency
  - Functionality and maintainability
  - Scalability across future generations
  - Portability across vendor devices
- Technical subjects
  - Parallel programming API, tools and techniques
  - Principles and patterns of parallel algorithms
  - Processor architecture features and constraints

People

- Wen-mei Hwu (University of Illinois)
- David Kirk (NVIDIA)
- Joe Bungo (NVIDIA)
- Mark Ebersole (NVIDIA)
- Abdul Dakkak (University of Illinois)
- Izzat El Hajj (University of Illinois)
- Andy Schuh (University of Illinois)
- John Stratton (Colgate College)
- Isaac Gelado (NVIDIA)
- John Stone (University of Illinois)
- Javier Cabezas (NVIDIA)
- Michael Garland (NVIDIA)

Course Content

Module 1 Course Introduction	• Course Introduction and Overview • Introduction to Heterogeneous Parallel Computing • Portability and Scalability in Heterogeneous Parallel Computing
Module 2 Introduction to CUDA C	• CUDA C vs. CUDA Libs vs. OpenACC • Memory Allocation and Data Movement API Functions • Data Parallelism and Threads • Introduction to CUDA Toolkit
Module 3 CUDA Parallelism Model	• Kernel-Based SPMD Parallel Programming • Multidimensional Kernel Configuration • Color-to-Greyscale Image Processing Example • Blur Image Processing Example
Module 4 Memory Model and Locality	• CUDA Memories • Tiled Matrix Multiplication • Tiled Matrix Multiplication Kernel • Handling Boundary Conditions in Tiling • Tiled Kernel for Arbitrary Matrix Dimensions
Module 5 Kernel-based Parallel Programming	• Histogram (Sort) Example • Basic Matrix-Matrix Multiplication Example • Thread Scheduling • Control Divergence

Course Content

Module 6 Performance Considerations: Memory	• DRAM Bandwidth • Memory Coalescing in CUDA
Module 7 Atomic Operations	• Atomic Operations
Module 8 Parallel Computation Patterns (Part 1)	• Convolution • Tiled Convolution • 2D Tiled Convolution Kernel
Module 9 Parallel Computation Patterns (Part 2)	• Tiled Convolution Analysis • Data Reuse in Tiled Convolution
Module 10 Performance Considerations: Parallel Computation Patterns	• Reduction • Basic Reduction Kernel • Improved Reduction Kernel
Module 11 Parallel Computation Patterns (Part 3)	• Scan (Parallel Prefix Sum) • Work-Inefficient Parallel Scan Kernel • Work-Efficient Parallel Scan Kernel • More on Parallel Scan

Course Content

Module 12 Performance Considerations: Scan Applications	• Scan Applications: Per-thread Output Variable Allocation • Scan Applications: Radix Sort • Performance Considerations (Histogram (Atomics) Example) • Performance Considerations (Histogram (Scan) Example)
Module 13 Advanced CUDA Memory Model	• Advanced CUDA Memory Model • Constant Memory • Texture Memory
Module 14 Floating Point Considerations	• Floating Point Precision Considerations • Numerical Stability
Module 15 GPU as part of the PC Architecture	• GPU as part of the PC Architecture
Module 16 Efficient Host-Device Data Transfer	• Data Movement API vs. Unified Memory • Pinned Host Memory • Task Parallelism/CUDA Streams • Overlapping Transfer with Computation
Module 17 Application Case Study: Advanced MRI Reconstruction	• Advanced MRI Reconstruction
Module 18 Application Case Study: Electrostatic Potential Calculation	• Electrostatic Potential Calculation (Part 1) • Electrostatic Potential Calculation (part 2)

Course Content

Module 19 Computational Thinking For Parallel Programming	• Computational Thinking for Parallel Programming
Module 20 Related Programming Models: MPI	• Joint MPI-CUDA Programming • Joint MPI-CUDA Programming (Vector Addition - Main Function) • Joint MPI-CUDA Programming (Message Passing and Barrier) (Data Server and Compute Processes) • Joint MPI-CUDA Programming (Adding CUDA) • Joint MPI-CUDA Programming (Halo Data Exchange)
Module 21 CUDA Python Using Numba	• CUDA Python using Numba
Module 22 Related Programming Models: OpenCL	• OpenCL Data Parallelism Model • OpenCL Device Architecture • OpenCL Host Code (Part 1) • OpenCL Host Code (Part 2)
Module 23 Related Programming Models: OpenACC	• Introduction to OpenACC • OpenACC Subtleties
Module 24 Related Programming Models: OpenGL	• OpenGL and CUDA Interoperability

Course Content

Module 25 Dynamic Parallelism	• Effective use of Dynamic Parallelism • Advanced Architectural Features: Hyper-Q
Module 26 Multi-GPU	• Multi-GPU
Module 27 Using CUDA Libraries	• Example Applications Using Libraries: CUBLAS • Example Applications Using Libraries: CUFFT • Example Applications Using Libraries: CUSOLVER
Module 28 Advanced Thrust	• Advanced Thrust
Module 29 Other GPU Development Platforms: QwickLABS	• Other GPU Development Platforms: QwickLABS
Where to Find Support	

GPU Teaching Kit

Accelerated Computing