1-800-633-1440
www.mindshare.com
training@mindshare.com

CUDA Programming for NVIDIA GPUs

CUDA Description

CUDA is a parallel computing platform and programming model created by Nvidia. CUDA gives program developers access to a specific API to run general-purpose computation on Nvidia Graphic Processing Units (GPUs). It allows developers to manage data transfers between the CPU host and the GPU and distribute the operations on the GPU compute cores.

Course Overview

This CUDA course is an on-site 3-day training solution that introduces the attendees to the architecture, the development environment and programming model of Nvidia Graphic Processing Units (GPUs). In addition to GPU hardware architecture and CUDA software programming theory, this course provides hands-on programming experience in developing general-purpose applications for Nvidia GPUs. Special attention is focused on presenting the architecture characteristics.

Topics include discovery of the hardware architecture of the GPU, transferring data between the GPU and CPU host, computing on the GPU and optimization techniques for better performance of the code running on the GPU. The lab exercises are based on concrete examples of typical computational kernels and are systematically extended to demonstrate the scope of topics covered. Emphasis is put on developing practical working knowledge of the CUDA API and tools.

You Will Learn:

- CUDA technology framework for heterogeneous parallel computing,
- Nvidia GPU architecture characteristics,
- Optimization techniques of CUDA applications,
- Available tools for CUDA development.

Course Length: 3 days

Target Audience

The target audience is the advanced C and C++ developer with little or no knowledge of CUDA programming, notions of multithreading programming or parallel hardware. This course is designed for performance-oriented application developers targeting heterogeneous computing architectures that GPUs and other co-processing devices. The first parts are also ideal for project managers to discover and to understand the challenges and benefits of using CUDA in their project.

Course Contents:

- **CUDA Basics**
    - Introduction to GPU computing
    - NVIDIA GPU architectures
    - CUDA programming model
    - CUDA API
    - CUDA debugging
- **CUDA Kernel Performance**
    - Using n-dimensional CUDA grids
    - CUDA warps
    - Memory alignment and coalescing
    - Texture and Constant memories
    - Shared memory as a cache

- Bank conflicts and access serialization
- Global memory access optimization

- **CUDA Grids Optimization**
  - Maximizing occupancy
  - Interpreting profiler counters
  - CUDA tools for profiling and debugging
  - CUDA libraries
- **CUDA Streaming**
  - CUDA data transfer optimizations
  - Asynchronism and overlapping
  - CUDA streams
  - Improve transfer performance
- **CUDA with MPI/OpenMP**
  - CUDA in node-parallelism environments
  - MPI introduction
  - Mixing CUDA and MPI
  - Multi-GPU computing with CUDA and MPI
  - Introduction to CUDA and OpenMP

CUDA Hands-on Programming Experience

Hands-on examples are meant to practice CUDA programming and illustrate Nvidia GPU specificities. Through gradual and pedagogical examples, attendees will be able to optimize CUDA applications respecting coalescing issues, using GPU cache memory and CUDA streams.

Programming activities explored include:

- Compute-bound and memory-bound tasks
- Performance impact of device memory transfers and efficient/inefficient memory access patterns
- Performance impact of CUDA grid of threads specification
- Optimization techniques and code tuning to improve performance
- Computing on multiple devices (computing on a cluster using MPI or OpenMP)

Recommended Prerequisites:

Hands-on will be in CUDA on Linux OS so previous programming experience with C-like languages on Linux OS is better. Familiarity with parallel programming concepts such as task parallelism and domain decomposition is a plus.

Course Material:

Mindshare will supply electronic version of the presentation slides including the lab descriptions and source code referenced in the examples and lab exercises. The hands-on will be executed on the customer's machine (customer's in-house cluster, etc.)