Introduction to CUDA Programming in C and Fortran

Abstract:

This tutorial is a beginning/intermediate course on programming NVIDIA GPUs with CUDA. After a short segment on why we are using accelerators in high performance computing and on accelerator hardware, we will describe all of the pieces necessary to write a CUDA program in C and Fortran. The example will be a stencil update, which is simple enough to be written in a few lines of code. The code design will be guided by the hardware; we will put emphasis on motivating common design principles by the desire to write fast code for GPU accelerators. In the second part of the presentation, we will focus on two common optimization strategies: using shared memory and overlapping computation with data transfer using CUDA streams. Experience with writing serial code in C or Fortran will be helpful to follow the examples.

Instructors:

Lars Koesterke, Ph.D. TACC Research Staff The University of Texas at Austin

Jeffrey L. Overbey, Ph.D. Assistant Professor Department of Computer Science & Software Engineering Auburn University

Software Requirements:

Participants should bring a laptop with an SSH client. The instructors will arrange for students to have access to training accounts on an XSEDE resource to complete the hands-on exercises.

Tentative Agenda:

8:00–8:45 Introduction and Background

8:45–9:15 CUDA Thread Organization

9:15–9:30 Hands-on: Compiling and running CUDA on an XSEDE resource

9:30–10:00 Break

10:00–10:45 CUDA Execution Model & Memories

10:45–11:15 Hands-on: Parallelization using CUDA C/Fortran

11:15–12:00 Best Practices, Tools, and Wrap-Up
Module Summaries:

**Module 1: Introduction.** Moore’s Law and the rise of GPGPUs. XSEDE resources with GPUs. Task vs. data parallelism. CUDA C and CUDA Fortran. Example: vector addition.

**Module 2: CUDA Thread Organization.** CPUs vs. GPUs; CUDA threading basics. Threads in the Vector Addition example. Multidimensional blocks and grids.


**Hands-On Exercises.** Compiling a simple CUDA C or CUDA Fortran program (stencil update) and submitting a job on an XSEDE resource. Parallelizing a simple program using CUDA C or CUDA Fortran.

**Draft Slides:** *(Please notice the inclusion of draft slides)*

Draft slides are posted http://jeff.over.bz/tutorials/cuda/2015/

**Network Outage Contingency Plans:**

Both of the instructors have taught semester-long courses on CUDA. In the event of a network outage, the hands-on time will be spent on more thorough coverage of advanced topics, or the instructors will “live code” the solution on a laptop with CUDA.