Study Guide

Week 11 March 28th, 2016 – April 3rd, 2015

Author B. Wilkinson Modification date March 26, 2016

Study Materials on Moodle

- *PowerPoint Slides*
  - Data parallel pattern
  - GPU Introduction
  - CUDA Programming model
  - CUDA Multidimensional block structure
  - CUDA Performance measurements
  - CUDA Device routines

- *Videos*
  - Lecture 19 video: 75-minute video of Lecture 19 in Fall 2014 on *Data Parallel Pattern, Introduction to GPUs and CUDA,* and *CUDA Programming Model.*
  - Lecture 20 video: 75-minute video of Lecture 20 in Fall 2014 on *Multidimensional thread structure, Performance measurements,* and *Device routines.*

- *Sample Quiz Questions*
  - CUDA quiz questions

Tasks

- **Mini-Quiz:** Answer the short posted quiz before 11:55 pm Sunday April 3rd, 2016.
- **Complete Assignment 5** Using Suzaku to Create MPI Programs
  - Assignment 5 Due: *Sunday* Sunday April 3rd, 2016 *(Week 11)*
- **Assignment 6** Short assignment on using Suzaku workpool version 2
  - Assignment 6 Instructions
  - Assignment 6 Due: *Sunday April 10th 2016 (Week 12)*

Moodle Saba meeting – 5 pm Friday April 1st, 2016

Data Parallel Pattern introduces a pattern whereby the same operation is performed on different data items in synchronism. This so-called data parallel pattern was used in vector supercomputers in the 1970s and has reappeared has vector SSE instructions in Intel and related processors, and in GPU processors. A data parallel algorithm for the prefix sum calculation is described. This algorithm will be seen in Assignment 7. The sequential pseudo code and parallel pseudo code given in the video/slides for the prefix sum calculation is not actually working code because of data dependences as noted in the slides, but they can be make working code easily by using a temporary array for the results of each step. As shown, matrix multiplication can also be made into a data parallel calculation. We will see this in Assignment 7.

GPU Introduction describes the emergence of GPUs for high performance computing, ending on the recent NVIDIA GPU's and their programming model called CUDA.

CUDA Programming Model then describes this model using 1-D vector addition as an illustration. Notice a single program is written and compiled that includes the code executing on the CPU "host" and the code executing of the GPU "device."

CUDA Multidimensional Block Structure continues on CUDA and also develops 2- and 3-dimensional organizations using matrix multiplication as the illustration for 2-dimensional structures. You are expected to understand how to create single (flattened) thread ID from 2-dimensional thread indices.

CUDA Performance Measurements describes how to measure execution time and some important aspects on CUDA kernels and routines. Note in particular that kernels are asynchronous and return before completing and some CUDA routines are similar.

CUDA Device Routines outlines how to declare routines that are to be called or executed by the device or host (several combinations) and also how to declare and access device (kernel) variables.

In Week 12 we will complete the CUDA materials and set the final assignment (CUDA).

Assignment 6 Instructions: This is a new short assignment for Spring 2016. The assignment asks you use the Suzaku workpool version 2. Some code is given including matrix multiplication. The most time will be spent modifying the matrix multiplication code for block matrix multiplication, which theoretically is more efficient than the original code. You are given only one week to do this particular assignment. The given code itself should simply work without change. The modifications to the matrix multiplication could take an hour to figure out and get working (I have done it) but please do not leave this assignment to the last day. It is extremely important your code actually performs block matrix multiplication and not simply the original code with the specified matrix sizes.