Study Guide

Week 10 March 21st 2016 – March 27th, 2016

Author B. Wilkinson Modification Date March 18, 2016

Study Materials on Moodle

- *PowerPoint Slides*
  - Sieve of Eratosthenes
  - Graph Algorithms
  - Hybrid Programming
  - Paraguin Hybrid Programming
- *Video*
  - Lecture 17 video: Video of Lecture 17 in Fall 2014 on the Sieve of Eratosthenes algorithm and graph algorithms.
  - Lecture 18-2 video: Second part of video of Lecture 18 in Fall 2014 continuing graph algorithms and describing hybrid programming. (The first part on the Seed framework assignment 6 is for Week 11.)
  -
- *Sample Quiz Questions*
  - Sieve of Eratosthenes quiz questions

Tasks

- **Mini-Quiz:** Answer the short posted quiz before *11:55 pm Sunday March 27th, 2016.*
- **Continue working on Assignment 5** Using Suzaku to Create MPI Programs – *N*-body problem
  - Assignment 5 Due: *Sunday April 3rd, 2016 (Week 11)*

Moodle Saba meeting – *Thursday March 24th, 2016, 5 pm (No classes Friday March 25th, 2016)*

Sieve of Eratosthenes describes ways to parallelize the Sieve of Eratosthenes algorithm of finding primes numbers. It is important to understand the basic Sieve of Eratosthenes algorithm. It is essentially a sequential algorithm - starting with all numbers from 2 to N, all multiple of each number are struck out (removed from the list of numbers), leaving just the prime numbers. It is only necessary to go to $\sqrt{N}$ as all number after that are either prime or have been struck out as not prime. (Can you prove that?) Parallelization can be done in several ways as described in the video/slides. You are expected to be able to identify in the sequential code where to parallelize it.

Graph Algorithms describes ways to parallelize various graph algorithms that you may have come across in an algorithms class, including Prim's algorithm for a minimum spanning tree, Dijkstra's algorithm for single shortest path, and Dijkstra's algorithm and Floyd's algorithm for the all pairs shortest path. You are expected to be able to identify where to parallelize the algorithms.

Coming back to using parallel software, *Hybrid Programming* discusses using both MPI message passing and thread based OpenMP in one program, which would be particularly suitable for a cluster of multicore machines (i.e. all clusters). You are expected to know how to compile and mixed MPI/OpenMP programs. According to the discussion and results in the slides, the approach does not always lead to better performance, although it does appear to be attractive approach. It can be extended to other programming paradigms, for example mixed MPI/OpenMP/CUDA programs. CUDA is the last topic of the course, to start in Week 11. Hybrid programming can also be done using Suzaku and OpenMP.

Paraguin Hybrid Programming describes using Paraguin instead of raw MPI. You will not be tested on this material.