CS 5970: Introduction to Parallel Programming

Class meets Mondays 4:25 to 7:10 pm

Short Description: Parallel and distributed architectures, algorithms, and programming paradigms. Topics include asynchronous/synchronous computation, GPU architectures, heterogeneous computing clusters, load balancing, memory hierarchies, message passing interface (MPI), SIMD/SPMD/MIMD, and multithreaded programming. Students will learn the ability to program on a variety of current languages, systems, and tools.

Course Description: This course covers a broad range of topics related to parallel and distributed programming, including the study of parallel and distributed architectures and systems, systems that are currently available, and parallel and distributed programming paradigms such as CUDA, MPI, OpenMP, OpenCL, and MapReduce. We will also explore various parallel and distributed algorithms for fundamental problems such as sorting, searching, matrix manipulations, and others. Parallel and distributed algorithms for processing “Big-Data” will be discussed along with student projects for their implementation on the current systems. The class will be a combination for lecture and seminar-style discussion. The instructor will introduce the students with the important elements of the course material and provide them with sufficient background to carry out a semester long project related to parallel and distributed programming. The University of Oklahoma through its Supercomputing Center has several systems that are available for use. In addition the School of Computer Science has a GPU computing node that the students can use for programming. Additional computing resources will be acquired through University-Industry cooperative agreements.

Course Prerequisite: CS 4413 – Algorithm Analysis and CS 3113 – Operating Systems

Required Textbook: None

References Textbooks:
   a) CUDA by Example: An Introduction to General Purpose GPU Programming, Jason Sanders, Edward Kandrot, Nivida, Addison-Wesley, 2010
   b) Introduction to Parallel Programming, by Peter Pacheco, Morgan Kaufmann, 2011
   c) Heterogenous Computing with OpenCL, Benedict Gaster, Lee Howes, David R. Kaeli, Perhaad Mistry, Dana Schaa, Morgan Kaufmann, 2012

Course Outline:
1) Parallel Computing, Hardware, and Software
2) Distributed-Memory Programming with MPI
3) Shared-Memory Programming with OpenMP
4) Parallel Algorithms and Applications
5) CUDA and OpenCL programming for the GPUs and multicore architectures
6) Distributed Programming Issues and Algorithms
7) Distributed Computing Tools and Technologies: MapReduce, Hadoop, and others
8) Grid and Peer-to-Peer Computing

Grading Policies: >= 90 – A; >= 80 and < 90 – B; >= 70 and < 80 – C; >=60 and < 70 – D; < 60 – F

Programming Projects: 30%
Midterm: 20%
Final Project: 25%
Class Presentation: 5%
Final Examination: 20%