Course Syllabus for Spring 2010

Course:           CSC 410/510 Parallel Computing (3 credits)
Prerequisites:    CSC 456 or Permission of Instructor.
Room:             McLaury 310
Time:             MWF 1:00-1:50 a.m.
Website:          http://www.mcs.sdsmt.edu/zzong/csc410

Instructor:       Dr. Zong
Office:           McLaury 203A
Phone:            605-394-6034
Email:            Ziliang.Zong@sdsmt.edu
Office Hours:     Monday: 2:00 – 4:00 p.m.
                  Wednesday: 2:00 – 4:00 p.m.

Course Description

This course will introduce the fundamental ideas and issues involved in programming and using parallel computers. Topics include multi-core programming using Visual Studio 2010 and Linux gcc, threads, rethinking of traditional algorithms (searching, sorting, and graphs) in parallel, OpenMP, MPI, Map-Reduce, Hadoop clusters, parallel file systems and parallel storage systems. This course is software oriented with applications in high performance computing and scientific computing. The text is comprehensive and contains more material than can be covered in one semester. Lectures will focus on selected topics from the text to emphasize or clarify topics. Students are expected to read the relevant sections in the text and, on occasion, look at reference materials; and will be responsible for material contained in the required reading and lecture.

Textbook:         Principles of Parallel Programming by Lin & Snyder, Addison-Wesley
                  ISBN-10: 0321487907

Course Evaluation:

Grades for this course will be based on:

1) Project 1       10%
2) Project 2       15%
3) Project 3       20%
4) Project 4       20%
5) Midterm Exam    15%
6) Final Exam      20%

and will be assigned as follows:
Note: For group projects, evaluations from your team about your performance will play an important role when I assign your grade.

Topics

The following list shows an approximate guide to the topics we will cover. Note that the topics may vary slightly to accommodate the class.

1. Course Introduction and Goals
2. Preview, Definitions and Jargon
3. Understanding Parallel Computers
4. Review – Process/Threads
5. Semaphores & Synchronization problems, Deadlocks, Priority and Starvation
6. Principles of Parallel Algorithm Design
7. Analytical Modeling of Parallel Programs: Proving Correctness and Measuring Performance, speedup, efficiency, cost, Amdahl's law, overhead
8. Introduction to Parallel Features in VisualStudio 2010 (Beta 2)
9. Introduction to OpenMP
10. Introduction to Massage Passing Interface (MPI)
11. Data Dependencies and Task Dependencies
12. Data Decomposition & Task Decomposition
13. Parallel Task Scheduling
14. Parallel Searching
15. Parallel Sorting
16. Parallel Graph Algorithms
17. Analysis of parallel algorithms
18. Load Balancing
19. Special Topics: Apache Hadoop
20. Special Topics: Parallel File systems
21. Special Topics: Map-Reduce
22. Special Topics: GPU and Cell

Objectives

The student will be introduced to the primary parallel computing architectures and gain an understanding of the main parallel programming models. A thorough understanding of shared memory and message passing models will follow. The student will gain experience in programming tools and solution methods in the main parallel programming models. Additionally, the student will become experienced in parallel algorithm development and performance analysis.

Student Learning Outcomes

Upon completion of this course, students should, at a minimum be able to:

1. Define parallel computing and grand challenge problems;
2. Describe the current parallel computing platforms;
3. Describe the parallel models and network topologies;
4. Determine communication costs for standard interconnection networks;
5. Gather data and statistics to access program performance;
6. Compute speedup and efficiency from runtime data;
7. Estimate serial and parallel time fractions from runtime data;
8. Write OpenMP based parallel programs using standard OpenMP functions;
9. Write MPI based parallel programs using standard MPI functions;
10. Perform function, domain and task decompositions;
11. Define parallel algorithm design approaches;
12. Write load balanced programs using static and dynamic parallelism;
13. Write programs which use shared memory;
14. Describe distributed shared memory implementations;
15. Work in teams to design, code and present parallel algorithms;
16. Describe standard parallel algorithms for searching, sorting and graph algorithms;

**Attendance**

If you do not attend class, it gives the instructor no opportunity to teach you the course material. Good attendance is particularly important in this class. Attendance is required for all class sessions.

**Make-up Policy and Late Policy**

There is NO “late policy” for projects. All projects must be turned in by the due date, otherwise they will not be accepted. Make-up examinations will be given only if you contact me ONE WEEK before the examination and you must provide a legitimate reason. There will be NO make-up exams without a verified excuse.

**Academic Integrity**

Although you may exchange ideas with your classmates, you must complete these assignments by yourself (or with members of your team, in the case of group projects). In particular, it is forbidden under any circumstances whatsoever to exchange source code with your classmates. COPYING CODE IS A SERIOUS INFRINGEMENT UPON THE SDSM&T ACADEMIC INTEGRITY POLICY, AND WILL BE TREATED SERIOUSLY. Academic integrity is further discussed in the Academic Integrity policy statement on the course Website.

**ADA Statement**

Students with special needs or requiring special accommodations should contact the instructor and/or the campus ADA coordinator, Jolie McCoy, at 394-1924 at the earliest opportunity.

**Freedom in Learning Statement**

Under Board of Regents and University policy student academic performance may be evaluated solely on an academic basis, not on opinions or conduct in matters unrelated to academic standards. Students should be free to take reasoned exception to the data or views offered in any course of study and to reserve judgment about matters of opinion, but they are responsible for learning the content of any course of study for which they are enrolled. Students who believe that an academic evaluation reflects prejudiced or capricious consideration of student opinions or conduct unrelated to academic standards should contact the dean of the college which offers the class to initiate a review of the evaluation.
Electronic Devices Policy

Please turn off your cell phone before class starts. No text messaging in class. No headphones. It is allowed to use a laptop in class only for purposes of note taking and in-class programming practice. No other use of any other electronic/computer media is allowed during class time.