Course Description

Nowadays, leveraging parallelism to speedup computation is ubiquitous, from dual-core A7 processor in iPhone 5s, 4-core Intel i7 in your laptop, to IBM BlueGene/L with 131,072 processors. All system manufacturers offer multiprocessors on their product line. Recent advances in high bandwidth, low latency networks are enabling parallel computing on clusters of workstations in massive scale.

This course is designed to provide seniors and graduate students majored in computer science and engineering with an understanding of fundamentals of parallel and distributed systems. Specific topics to be discussed include: parallel computer architectures including shared memory multiprocessors and scalable multicomputers, common programming paradigms including shared address space and message-passing approaches, and parallel algorithm design including principles and techniques of designing numerical and non-numerical algorithms. The focus in parallel programming will be on both programming expression and programming for performance. Upon completion of this course, students are expected to become familiar with the fundamentals of parallel and distributed computing and obtain skills for making efficient use of the parallel and distributed systems.

Tentative Syllabus

Overview of parallel/distributed computing, models of parallel computers, parallel software basics, Message Passing Interface (MPI) programming, multithreaded programming using Pthread, design of typical numerical and non-numerical algorithm, and programming using distributed execution engines such as MapReduce and Pregel.
Course Prerequisites

It is assumed that you have background of computer architecture and are familiar with C programming. If you feel that your background is lacking in some of these areas, and are not sure whether this course is suitable to you, please come to talk with your instructor.

Course Textbook:

(Required)
A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing*, (2nd Ed), Addison Wesley, 2003 or later versions

(Recommended)

The materials to be presented in the course will be complemented by the other reference books, programming manuals and tutorials. Many of them are available on-line, and will be available on the course web page. In addition, other course materials, including lecture notes, problem sets and reading papers, will also be posted.

Each student needs an account on workstations in Computer Laboratory of COE for MPI and Pthread programming.

Class Attendance

Class Attendance is required. Please arrive on time. Ask questions if there is something you don't understand or want me to clarify. I can always stop and explain it.

Required Work and Grading Policy

The course will be fast-moving, so it is important to keep up with the work. You have two weeks for a programming lab, and one week for a homework assignment. There will be two in-class midterm exams, and one final exam. Both final and midterms are closed-book and closed-notes. If you are ill on the date of an exam, you should provide the instructor a documentation of the illness (doctor's note) and arrange for a make-up exam date as soon as possible.

End-of-semester numeric scores will be weighted as follows:

- 35% programming lab and homework assignments
  - 25% programming assignments
  - 10% writing assignments
- 60% Exams
  - 15% midterm X 2
  - 30% final
5% Class participation

The final will be distributed as A (100-90), A- (89-85), B+ (84-80) B (79-75), B- (74-70), C+ (69-65), C (64-60), C- (59-55) F (below 55).

Late Policy

NO LATE ASSIGNMENTS WILL BE ACCEPTED, but with the following exception: every student of the course is allowed to have a total of TWO free late days. Consider them as non-transferable, non-replaceable credits and use them wisely. If you wish to use one or more of these, indicate it clearly in your submission. No partial late day is allowed. You need one whole late day to cover a one-hour late submission. Late submissions will not be graded if you have used all three late days. Partial credits are available as long as you document well which parts have been implemented.

Extra Credit

There will be opportunities for extra credit on some of the projects (programming assignments). To receive any extra credit, all base functionality of a project must be implemented; extra credit portions of a project cannot count in place of required features. Extensions will not be given for completion of extra credit.

Academic Honesty

All course work must be individual efforts. Also, it is your responsibility to protect your work from unauthorized copying. Copying and altering other’s code or solution sets are strictly prohibited. Students who are caught cheating will receive a grade “F” and face possible expulsion from the University. I will probably use similarity detection software to assist me to find any cheatings, should they occur. Note that cooperation is not considered as cheating. You can ask someone about concepts and ideas that are not directly related to the homework or programming labs before you start on them.

The instructor welcomes comments, in person or otherwise, on the teaching and grading.