Course Information

Instructor: Dr. Amit Jain

Office: MEC 302E  Phone: 426-3821  FAX: 426-2470
email: ajain@boisestate.edu  URL: http://cs.boisestate.edu/~amit

Home Page for the Class: http://cs.boisestate.edu/~amit/teaching/430/cs430.html

Office Hours: Check the class home page.

Class Time: M W 12:00pm–1:15pm  Location: MEC 307

Catalog Description

CS 430/530 PARALLEL COMPUTING (3-0-3)(F)(Even Years). Models of parallel computation. Fundamental design patterns used in parallel algorithms: embarrassingly parallel, partitioning, divide and conquer, software pipelining, synchronous computations and load balancing. Implementation of parallel programs using MPI, GPUs and Map-Reduce on parallel clusters. PREREQ: CS 253 and CS 321 or PERM/INST.

Prerequisites

The student should be familiar with the fundamental data structures and algorithms used in sequential computing. The course on Data Structures (CS 321) should provide the requisite background. Also the student should have prior experience with C programming, which the course CS 253 provides.

Goals

Parallel computing covers a wide area in computer science and an introductory course cannot span all the topics. The emphasis in this course is to teach students how to “think in parallel.” Fundamental ideas that are not likely to change with time are stressed. By taking this course, the student will be able to:
• describe parallel models of computation and parallel architectures,
• explain the fundamental concepts in parallel computing,
• solve problems faster by using multiple processors, whether “in a box” or spread across a cluster,
• design parallel programs for problems,
• convert sequential programs to parallel programs,
• and write parallel programs for at least one parallel system.

Text
• Lecture Notes (available on class website)

References
• Principles of Parallel Programming, Calvin Lin and Lawrence Snyder, Addison Wesley, 2009.
• Introduction to Parallel Computing by Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis, 1994, Benjamin Cummings.

Topics
• Introduction: Parallel Computing, Parallel Architectures, Parallel Programming Models
• Message-passing model
• Introduction to MPI library
• Introduction to the lab cluster environment
• Paradigms for parallel programs
  – Embarrassingly parallel computations
  – Partitioning
  – Divide-and-Conquer
  – Software Pipelining
  – Synchronous Computations
  – Load balancing and termination detection

• More examples.
  – Sorting Algorithms
  – Numerical Algorithms
  – Searching and Optimization

• Introduction to parallel computing on GPUs

• Map-reduce
  – Map-reduce concepts
  – Hadoop Distributed Filesystem
  – Developing a map-reduce application
  – Setting up and administering a Hadoop cluster
  – Advanced map-reduce features

• Parallel systems
  – Beowulf cluster hardware design case studies
  – Beowulf cluster setup and installation
  – Cluster scheduling software
  – Parallel file systems

Laboratory Projects

• Laboratory Experience. Familiarization with using the Linux Cluster Lab. (1 week)

• A simple parallel program to get students familiar with parallel programming. (2 weeks)

• A larger parallel program where students try to optimize the performance. (3 weeks)

• A map-reduce program of moderate complexity. (3 weeks)

• A larger team project. (4 weeks)
Format, Student Activities, and Grades

The instructor chooses the exact grading scheme, but a typical distribution would be:

- In-class quizzes: 200 points (20%)
- Programs: 700 points (70%)
- Final Examination: (12:00pm–2:00pm, 15th December, Monday): 100 points (10%)

Letter grades are based on a plus/minus scale as shown next: A+ (4.0), A(4.0), A- (3.7), B+(3.3), B (3.0), B- (2.7), C+ (2.3), C (2.0), C- (1.7), D+ (1.3), D (1.0), D- (0.7) and F (0.0).

Notes for Graduate Students

Graduates students will be assigned more complex projects/homework as opposed to undergraduate students.