CS 411/511 Parallel Programming  
Semester: Fall 2011  
Moscow Time: MW 4:30pm - 5:45pm  
Room: JEB 26  
Idaho Falls Time: MW 5:30pm - 6:45pm  
Room: CHE 301

Instructor: Robert Hiromoto  
Office: CAES 258  
Office Hours: To be arranged  
Phone: 208 533-8119  
E-mail: hiromoto@uidaho.edu

Textbooks: An Introduction to Parallel Programming by Peter Pacheco, Morgan Kaufmann, ISBN-10: 0123742609

Recommended Textbooks:

- Parallel Programming with MPI by Peter Pacheco, Morgan Kaufmann, ISBN: 1558603395
- The Art of Multiprocessor Programming by Maurice Herlihy, and Nir Shavit, Morgan Kaufmann, ISBN-10: 0123705916

Course Description:

This course teaches the principles of parallel programming to upper division and graduate level computer science students. Topics to be covered include programming for symmetric multi-core processors using Pthreads and OpenMP, and distributed memory workstation clusters using the message-passing MPI communication library. Programming tools such as gdb and gprof, parallel programming semantics, and parallel program performance issues will also be covered. Several programming assignments will be used to reinforce the concepts learned in the class room, and a final parallel programming project will be required.

Prerequisites: Analysis of Algorithms, Operating Systems, Concurrent Systems and Computer Architectures, or instructor permission.
Outline

- Introduction to spatial- and temporal-locality
- Parallel programming concepts
  - Task and Data decomposition
  - Race-conditions
  - Critical-section (atomic updates)
  - Process starvation (live lock)
  - Dead-lock
  - Performance evaluation
- Primitives to control shared resources
  - Busy-wait
  - Mutual exclusion (mutexes)
  - Semaphores
  - Thread-safe system libraries
- Shared memory programming using threads (Pthreads and OpenMP)
  - Shared memory architectural model
  - Asynchronous computations
  - Parallel programming advantages/disadvantages
- Message passing programming using MPI
  - Block algorithms
  - Single program multiple data (SPMD)
- Performance issues
  - Bottlenecks and speedups
  - Impact of communication
  - Tuning programs

Grading: The course grade will be based on homework and programming assignments, and one in-class or take-home exam. A final parallel programming project will be required. Graduate students will be required to complete a project of greater difficulty.

- Homework/Programming Assignments (40%).
- One examination (20%).
- Programming Project (40%).