CS 6643 (Parallel Processing)

Course Instructor: A. T. Chronopoulos,
Email: anthony.chronopoulos@utsa.edu
Office: TBA, **Office Hours:** TBA
Course Meetings : MW 18-19:15pm, NPB 1.202 (starts: Jan 12, 2015)

Textbook

An Introduction to Parallel Computing, Design and Analysis of Algorithms, 2/e. Ananth Grama, Vipin Kumar, Anshul Gupta, and George Karypis. Addison-Wesley, 2003. ISBN 0-201-64865-2.

Prerequisites

Students are expected to have taken a course in undergraduate: Computer architecture, algorithms, OS, and programming experience in at least one programming language (e.g. C, C++ or Java, or other).

Grading Policy: The course work consists of:

Homeworks (30%) : 4 HWs
Mid-Term Exam (40%) : Covering up to 20 lectures
Term Project and Presentation (30%) : This Project substitutes the final exam, including a written report and a PPT presentation at the end of the course.

The programming for HWs and project can be either on a cluster (using MPI) or on a cloud system (e.g. using MPI or MapReduce or other).

Course Description:

Parallel Computing deals with emerging trends in the use of large scale computing platforms ranging from desktop multicore processors, tightly coupled SMPs, message passing platforms, and state-of-the-art virtualized cloud computing environments.
The knowledge and skills of parallel programming enables computer programmers and scientists to leverage the power of parallel computers and cloud computing infrastructures in solving big compute and big data problems. The concepts of parallel programming will provide the foundation to fully utilize popular tools e.g. MPI, Hadoop MapReduce, and others.

The course will cover material from the following topics:
Fundamentals: Introduction to parallel and distributed computing, models of parallel computers, parallel programming models.
Design methodology for parallel algorithms.
Basic communication operations,
Performance evaluation of parallel systems: Performance metrics, execution time, speedup, efficiency, cost, scalability, isoefficiency and cost effectiveness.
Dense matrix algorithms: Mapping matrices on processors, matrix transposition, matrix vector multiplication, matrix multiplication, solving systems of linear equations.