CS 4823 (Introduction to Parallel Programming)
FALL 2014

Instructor: A. T. Chronopoulos, **Email: Anthony.chronopoulos@utsa.edu**
Office: Peter T. Flawn Building (f.SB) 3.02.16, **Office Hours:** T,Th 10:00 AM-11:00 AM, or by appointment

Course Meetings : T, Th 8:30 AM-9:45AM, Peter T. Flawn Building (f.SB) 3.02.07

Textbook

An Introduction to Parallel Programming, 1st ed., 2011, by Peter Pacheco (ISBN: 978-0-12-374260-5).

Prerequisites

This course is suitable for (junior or senior) undergraduate students in computer science or other physical sciences or engineering. Students are expected to know discrete math or algorithms. They should also know a programming language (e.g. C, C++ , Java, Fortran 90 or other).

Grading Policy: The course work consists of:

Homeworks (30%) : 4 HWs (programming or theoretical)
Mid-Term Exam (40%) : Take home
Term Project and Presentation (30%) (No Final exam):

The course will be structured as lectures, home works, programming assignments and a final project instead of a final exam. Students will perform programming homeworks to express simple parallel computations using selected parallel programming models and measure their performance. The final project will consist of teams of 2-3 students. The programming for HWs and project can be either on a cluster (using MPI, OpenMP) or on a cloud system (e.g. using MPI or MapReduce or OpenMP).

Course Description:

This course is a comprehensive study of parallel programming paradigms. It presents core concepts and studies a widely used set of modern parallel programming models and it provides context with a few representative parallel algorithms. Recently, this area has drawn a wide interest in the scientific and business community due to a number of factors. Most significantly, the availability of multi-core microprocessors and GPUs has made parallel computing accessible to the masses. At the high- end, major vendors of large-scale clusters provide systems of tens or hundreds of thousands of processors. At the low-end, desktop multi-core processors, GPUs and embedded devices are made available at a low cost.

Modern parallel programming deals with programming of concurrent computing platforms ranging from desktop multi-core processors, GPUs, tightly coupled SMPs, message passing platforms, and state-of-the-art virtualized cloud computing environments.

Course Outline : (In total, 28 lectures plus test and final project)

1. **Chapter1** Why Parallel Computing (2 lectures)
2. **Chapter 2** Parallel Hardware and Parallel Software (7 lectures)
3. **Chapter 3** Distributed Memory Programming with MPI (6 lectures)
4. **Chapter 4** Shared Memory Programming with Pthreads (7 lectures)
5. **Chapter 5** Shared Memory Programming with OpenMP (7 lectures)