CS 5170: INTRODUCTION TO PARALLEL COMPUTING

Course Description

Principles and practice of parallel computing. Parallel algorithm development and analysis. Design, implementation, and evaluation of parallel programs for shared memory, local memory, and vector architectures. Prerequisites: Admission to MS in CS program, or consent of department, plus CS 3270 or equivalent.

Course Syllabus

- Overview of parallel computing
- Performance measures
- Parallel architectures
- Problems amenable to parallel programming solution
- Programming languages for parallel programming
- Program portability issues
- Operating system issues
- Tools for parallel programming
- Parallel Algorithms
- Parallelizing serial programs

Knowledge Acquisition Goals

After successfully completing CS 5170 a student should have a conceptual understanding of:

- Flynn's Taxonomy
- Languages for parallel computing, including: MPI and OpenMP
- The Message passing paradigm
- Parallel Processing speedup issues, including Amdahl's and Gustafason's Laws
- Decomposition methodologies for parallel program development
- Load balancing issues for parallel programs
- Parallel Architectures and effect of architecture on design/implementation of a parallel algorithm
- Quantifying Speedup; empirical tools and theoretical models
- PRAM computational model
- Current issues in parallel processing

Skill Goals

After successfully completing this course a student should be able to

- Design, implement, test and debug a parallel application program using MPI
- Design, implement, test and debug a parallel application program using OpenMP
- Parallelize an existing application using an appropriate parallel programming paradigm
- Explain, in writing, the tradeoffs that result from using a specific programming paradigm for a given problem class
- Develop and analyze a parallel algorithm using the PRAM model.

Course Requirements
Examinations, homework assignments, programming assignments.