CSc 387 - Parallel Programming with MPI

Instructor : Fikret Ercal - Office: CS 314, Phone: 341-4857
E-mail & URL : *ercal@mst.edu* http://web.mst.edu/~ercal/387/387.html
Meeting Times : 2-3:15pm Office Hours: Provided on the class website
**If the instructor is late for the class, students are expected to wait ~5 minutes before they leave the classroom.
Textbook (Recommended): Parallel Programming; B. Wilkinson and M. Allen, Prentice Hall, 2005
Required background: Algorithms, complexity theory, operating sys, Linux, and C/C++

Objectives:
- Learn how to write parallel programs using Message Passing Interface (MPI)
- Study parallel algorithms and their complexities related to broadcasting, routing, sorting, image processing, graphs, and numerical computation.
- Study the issues that influence the speedup and efficiency of parallel programs
- Study popular parallel architectures and network topologies
- Study pipelining, message passing, process communication/synchronization, process level parallelism, data parallelism, task partitioning and load balancing.

Project Presentations:
Students will work in groups of two to complete a semester project. The project will consist of:
(i) selecting a research topic and collecting and reading papers related to the project (a list of potential projects will be provided) and, (ii) giving a 20-25 min. presentation in class.

Assignments, attendance, and grading policy:
Throughout the semester, several homeworks and programming assignments will be given related to the material covered in class. Students will write parallel programs to run on a network of workstations using Message Passing Interface (MPI). All assignments will be graded. Project descriptions will be made available on the class website. Make sure that you regularly check the class website for announcements, assignments, and other course related materials.

Late assignments: Unless there is a reasonable excuse, late assignments will be penalized as follows: 1-2 school days late $\Rightarrow$ 25% off, 3-4 school days late $\Rightarrow$ 50% off. More than 4 days late $\Rightarrow$ 75% off.

Since this is a senior/graduate level class, participation is very important. Sometimes, the decision to adjust a grade is made partially on the basis of attendance and classroom participation. You are encouraged to exchange information regarding the term projects, and the assignments. However, the final work, programs, and report must be your own work. Cheating and plagiarism will not be tolerated. Page 30 of the UMR Student Academic Regulations handbook (http://registrar.mst.edu/academicregs/index.html) describes the student standard of conduct relative to the System's Collected Rules and Regulations section 200.010, and offers descriptions of academic dishonesty including cheating, plagiarism or sabotage.

Any student inquiring about academic accommodations because of a disability will be referred to Disability Support Services (http://dss.mst.edu/) so that appropriate and reasonable accommodative services can be determined and recommended.

Grading:
Midterm (225), Final (225), programming projects (4 * 100), Term Project (150). (TOTAL: 1000 points)
Bonus Credits: *Fastest* code in each assignment will receive 25 extra points and significant parallel code for the semester project will receive upto 100 points.

COURSE SCHEDULE

CHAPTER 1: PARALLEL COMPUTERS — 1.5 week
parallel computers/programming, Shared/distributed memory, message-passing, MIMD/SIMD paradigms,Embedding, Granularity, Speedup/Efficiency, Amdahl's/Gustafson's law, processor-time product (cost), Scalability

CHAPTER 2: MESSAGE PASSING & MPI PROGRAMMING — 1.5 weeks
MPI for Parallel Programming on a network of workstations, Modeling communication, Comm. latency, Time complexity of parallel alg.s, Broadcasting/Routing on networks

***** Programming Project-1: Entropy Calculation: DPCM Example *****

CHAPTERS 3 & 4: — 1 week
Embarrassingly parallel applications, Divide-and-conquer, partitioning data, work pool approach, Monte Carlo methods bucket sort, and numerical integration

CHAPTER 5: PIPELINED COMPUTATIONS — 1 week
The pipeline concept, application areas and analysis Examples: Insertion sort, prime number generation, and solving an upper triangular system of linear equations.

***** Programming Project-2: Parallel prime number generation *****

CHAPTERS 6 & 7 (briefly): — 1 week
Global and local barriers, Data parallel computations, Synchronous iterations, Examples: prefix sum, iterative solution of equations, cellular automata, parallel MST (Prim's) and SSSP (Dijstra's), Distributed termination detection

***** REVIEW - Question/Answer Pool for TEST I (in class) ***** — 1.5 week
***** MIDTERM EXAM (in class) and Solutions ***** — 1 week

CHAPTER 10: SORTING ALGORITHMS — 1.5 weeks
Bubble sort, odd-even transposition sort, quicksort on a hypercube, bitonic sort

***** Programming Project-3: Parallel Bitonic Sort *****

CHAPTER 11: NUMERICAL ALGORITHMS — 1 week
Parallel matrix multiplication (direct, recursive, mesh). parallel Gaussian elimination Solving partial differential equations using Jacobi and Gauss Seidel methods

***** Programming Project-4: Computing the Determinant of a Matrix *****

GPUs and CUDA Programming — 1 week

CHAPTER 12: IMAGE PROCESSING — 0.5 week
simple image operations (e.g. noise reduction, edge detection)

PROJECT PRESENTATIONS — 1 week
***** REVIEW and TEST II ***** — 1.5 week