COMPSCI 430: Parallel Computing
3-credit course with 3 lecture hours per week
Course Coordinator: Amit Jain

Textbook(s) and Supplemental Material


Catalog Description


PREREQ: COMPSCI 253 and COMPSCI 342.

Elective

Goals for the Course

Successful students will be expected to:

- describe parallel models of computation and parallel architectures
- explain the fundamental concepts in parallel computing
- solve problems faster by using multiple processors, whether “in a box” or spread across a cluster
- design parallel programs for problems
- convert sequential programs to parallel programs
- write parallel programs for at least one parallel system
- know the issues in the design of a parallel cluster

Outcomes Addressed

a. an ability to apply knowledge of computing and mathematics appropriate to the discipline
b. an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
c. an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
h. recognition of the need for and an ability to engage in continuing professional development
i. an ability to use current techniques, skills, and tools necessary for computing practice
j. an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs involved in design choices
k. an ability to apply design and development principles in the construction of software systems of varying complexity

Outcomes Assessed: None

Topics Covered
Introduction
Message Passing Model
Introduction to MPI
Introduction to the Lab Cluster Environment
Parallel Programming Paradigms: Embarrassingly Parallel Computations
Parallel Programming Paradigms: Partitioning
Parallel Programming Paradigms: Divide-and-Conquer
Parallel Programming Paradigms: Software Pipelining
Parallel Programming Paradigms: Synchronous Computations
Parallel Programming Paradigms: Load Balancing and Termination Detection
Sorting, Numerical Algorithms, Searching and Optimization
Map-Reduce Concepts
Advanced topics in MPI
Introduction to OpenMP
Map-Reduce Concepts
Hadoop Distributed System
Developing a Map-Reduce Application
Advanced Map-Reduce Features
Introduction to Parallel Computing on the GPUs
Parallel Systems Hardware and Software

Grading
Letter grades are assigned to students based on numerical scores for the following activities:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>10%</td>
</tr>
<tr>
<td>Programs</td>
<td>70%</td>
</tr>
<tr>
<td>Midterm</td>
<td>10%</td>
</tr>
<tr>
<td>Final</td>
<td>10%</td>
</tr>
</tbody>
</table>

Curriculum Category Content (Credits)

<table>
<thead>
<tr>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Software Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Architecture</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Data Structures</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Programming Languages</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>