UNIVERSITY OF NEBRASKA AT OMAHA
COURSE SYLLABUS/DESCRIPTION

Department and Course Number	CSCI 4440
Course Title	Introduction to Parallel Computing
Course Coordinator	Sanjukta Bhowmick
Total Credits	3
Date of Last Revision	June 27 2015

1.0 Course Description

1.1 This course is an introduction to parallel computing, that is using multiple processors to execute algorithms. The course focuses on the design of parallel algorithms for a variety of problems and their implementation in shared memory (OpenMP) and distributed memory (MPI) systems. The course also discusses the latest HPC trends including computing on GPUs, massively multithreaded machines and grid computing.

1.2 For whom course is intended.
This course is intended for seniors, and graduate students in computer science. The course may also be of value to practicing computer professionals with suitable backgrounds who are seeking to employ high performance computing.

1.3 Prerequisites of the course (Courses).

CSCI 3660 Data Structures

1.4 Prerequisites of the course (Topics).
None

1.1 Unusual circumstances of the course
None

2.0 Objectives

2.1 Upon completion of this course, a student should be able to:
- 2.1.1 define common terms used in parallel and distributed computing;
- 2.1.2 explain the advantages and disadvantages of systems using shared memory and distributed memory systems;
- 2.1.3 be knowledgeable about different metrics to experimentally and analytically evaluate parallel algorithms;
- 2.1.4 design parallel algorithms for a given problem, implement them in MPI and/or OpenMP and analyze their performance;
- 2.1.5 know about latest developments in HPC;
- 2.1.6 read and critically review papers on parallel algorithms;

3.0 Content and Organization

3.1 Introduction
- 3.1.1 Why is high-performance computing important?

3.1.2 Short history of high performance computing. Moore's Law. Top500 supercomputer list
3.1.3 Use of HPC in different applications; such as life sciences, fluid dynamics, weather simulation, etc.
3.1.4 Overview of shared and distributed memory systems
3.1.5 Basic principles of parallel computing; pipelining, use of reduction, load balancing and synchronization techniques. Simple problems such as finding maximum value, HCF will be discussed
3.1.6 Related Paper: *A view of the parallel computing landscape*. K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek, D. Wessel, and K. Yelick. *Commun. ACM* 52, 10 (October 2009).

3.2 Analyzing Parallel Performance 2.0
3.2.1 Tradeoff between communication and computation
3.2.2 Amdahl's Law
3.2.3 Gustafson-Barsis's Law
3.2.4 Karp-Flatt Metric
3.2.5 IsoEfficiency
3.2.6 Related Paper: *Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers* David H. Bailey June 11, 1991

3.3 Programming with MPI and OpenMP
3.3.1 Introduction to MPI and OpenMP constructs
3.3.2 In class exercises with simple problems (addition, finding maximum)
3.3.3 Assignments with more complex problems (find HCF, LCM, mod)
3.3.4 Related Papers: Links to MPI and OpenMP tutorials

3.4 Parallelization Techniques on 1-D Array
3.4.1 Finding Prime Numbers using Sieve of Eratosthenes
3.4.2 Sorting Numbers using quicksort like methods
3.4.3 Related Papers: Recent papers on sorting

3.5 Parallelization Techniques on 2-D Array
3.5.1 Matrix Vector Multiplication
3.5.2 Matrix Matrix Multiplication
3.5.3 Solution of Linear systems using Guassian Elimination
3.5.4 Extra Credit: Implementation of Matrix Multiplication on GPUs
3.5.5 Related Papers: Papers on improved Matrix Matrix Multiplication such as Cannon's Algorithm; Introduction to Numerical Libraries (PETSc, Zoltan)

3.6 Parallelization on Graphs
3.6.1 All pair shortest paths (Floyd Warshall algorithms)
3.6.2 Breadth First Search
3.6.3 Extra Credit: Parallel Implementation of Betweeness Centrality, Clustering algorithms
3.6.4 Related Papers: Recent papers from Graph500 list

3.7 Recent Trends in HPC (1-2 Topics will be selected based on interest of the class)
3.7.1 Paper on power management in HPC systems
3.7.2 Architecture and algorithm design on GPUs
3.7.3 Architecture and algorithm design on massively multithreaded machines
3.7.4 MapReduce
3.7.5 Special purpose machines; Anton, Micron AP, etc.

4.0 Teaching Methodology

4.1 Methods to be used
Teaching methods will include in-class lectures and problem solving, homework assignments, presentation of papers, and in-class exams.

4.2 Student role in the course
Students are expected to attend all lectures, participate in class discussions, and complete assigned homework and examinations.

4.3 Contact hours
Two and half hours per week

5.0 Evaluation

5.1 Type of student projects that will be the basis for evaluating student performance, specifying distinction between undergraduate and graduate, if applicable. For laboratory projects, specify the number of weeks spent on each project).

The students will be expected to complete homework assignments (written and programming), examinations, and participate in class discussions. Programming assignments will use MPI and/or OpenMP (www.openmp.org). Interested students can also explore GPU programming.

Basis for determining the final grade (Course requirements and grading standards) specifying distinction between undergraduate and graduate, if applicable

Class participation	5%
Class presentations (1)and review of paper (5)	5% (presentation) 10% (papers)
Small programming projects (3)	3*10%=30%
Large programming project (1)	20%
Examinations	2*15=30%

5.2 Grading scale and criteria

Percent	Grade	Percent	Grade
97 – 100	A+	77 – 79	C+
94 – 96	A	70 – 76	C
90 – 93	A–	70 – 73	C–
87 – 89	B+	67 – 69	D+
84 – 86	B	64 – 66	D
80 – 83	B–	60 – 63	D–

6.0 Resource Material

6.1 Textbooks and/or other required readings used in course
Michael J. Quinn, *Parallel Programming in C with MPI and OpenMP*. McGrawhill 2004

6.2 Other suggested reading materials, if any

Papers and Links on related topics as assigned

6.3

6.4 Other sources of information
Relevant Internet websites, which discuss various state-of-the-art topics

6.5 Current bibliography of resources for student's information
None

7.0 Computer Science Accreditation Board (CSAB) Category Content (class time in hours)

CSAB Category	*Core*	*Advanced*
Data structures		
Computer organization and architecture		
Algorithms and software design		
Concepts of programming languages		

8.0 Oral and Written Communications

Every student is required to submit at least ___5__ written reports (not including exams, tests, quizzes, or commented programs) to typically __2___ pages and to make 1 oral presentations of typically 15 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

9.0 Social and Ethical Issues

No coverage

10.0 Theoretical content

Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

	Contact hours
Linear algebra, graph theory, load balancing,	27
Computer performance parameters and parallel algorithm analysis	15

11.0 Problem analysis

The course involves learning different design issues and trade-offs. The students are challenged to apply the design techniques learned through the course to analyze and solve specified problems in computer architecture.

12.0 Solution design

Students taking the course are expected to develop and/or analyze the design of all the problems given as part of the homework assignments, in class problems, and examinations.