Course Title: COSC 4311: Parallel Computing

Semester Credit Hours: 3 (3,0)

I. Course Overview

This course provides a basic, in-depth look at techniques for the design and analysis of parallel algorithms and for programming them on commercially available parallel platforms. Principles of parallel algorithms design and different parallel programming models are both discussed. MPI, POSIX threads, and Open MP all are discussed. This course is for anyone wanting to gain proficiency in all aspects of parallel and distributed programming.

II. PMU Competencies and Learning Outcomes

Students of COSC 4311: Parallel Computing develop skills necessary for understanding the design of parallel computing applications so as to appreciate the strengths and limitations of parallel computing approaches to problem solving. This course makes extensive use of the PMU technology infrastructure to provide communication between faculty and students. The course is primarily a lecture-based course with the student required to complete significant projects outside of class time. The course includes individual as well as group projects and provide opportunities for the presentation and defense of designed solutions. The course encourages the development of professional communication skills and provides opportunities for collaborative project development.

III. Detailed Course Description

This course provides a basic, in-depth look at techniques for the design and analysis of parallel algorithms and for programming them on commercially available parallel platforms. Principles of parallel algorithms design and different parallel programming models are both discussed. MPI, POSIX threads, and Open MP all are discussed. This course is for anyone wanting to gain proficiency in all aspects of parallel and distributed programming. Students develop skills necessary for understanding the design and analysis of parallel and distributed algorithms, and to appreciate not only the advantages, but also the difficulties of adapting algorithms to a parallel or distributed paradigm. The course is primarily a lecture-based course with the student required to complete significant projects outside of class time. The course includes individual as well as group projects and provide opportunities for the presentation and defense of designed solutions. The course encourages the development of professional communication skills and provides opportunities for collaborative project development.

Parallel computing is a critical component of the computing technology of the 21st century, and is likely to grow in importance with the proliferation of multiprocessor PC desktops and servers and scalable clusters of commodity workstations. This course examines the organizing principles behind parallel computing both from an architectural and a programming perspective. The course consists of two parts, organized around a common set of issues relevant to all parallel systems: naming, synchronization, latency, and bandwidth. The first part discusses how modern parallel computer architectures deal with these issues, both at the small (shared memory multiprocessors) and large (scalable multiprocessors) scales. The second part of the course discusses how the issues are dealt with in several common programming paradigms including message-passing, shared-memory, data-parallel, as well as higher-level approaches. The focus in this part of the course is on both programming techniques and programming for performance.

IV. Requirements Fulfilled

COSC 4311: Parallel Computing satisfies three hours of the requirements for the degree in computer science. It is required of all students pursuing a degree program in computer science within the College of Information Technology. This course may be used as an elective in the Information Technology and Computer engineering degree programs. This course should be taken in the senior year.

V. Required Prerequisites

COSC 3351: Algorithms

VI. Learning Outcomes

In this course, students learn:

- To develop an understanding of various basic concepts associated with parallel computing environments.
- To understand the effects that issues of synchronization, latency and bandwidth have on the efficiency and effectiveness of parallel computing applications.
- To gain experience in a number of different parallel computing paradigms including memory passing, memory sharing, data-parallel and other approaches.
- To earn experience in designing and testing parallel computing solutions to programming problems.
- To develop improved communication and collaborative skills.

VII. Assessment Strategy

The course grade involves an assessment of student performance on examinations that focus on the understanding of various concepts and constructs underlying Artificial Intelligence, and the communication of those concepts and the characteristics of designed solutions to AI problems to an audience. Course grades are based on:

- Weekly assigned homework to motivate students to do the work and earn credit accordingly.
- Weekly, in-class presentations by students related to independent literature research on aspects of the course material and classroom discussion and critique of the presentation.
- Two in-class examinations to assess the student's accumulative mastery of content covered prior to the time of the examination.
- Four programming assignments testing students understanding of the major concepts introduced during the course.
- A comprehensive final examination to assess the student's accumulative mastery of course material.

The final grade is based on 10% credit for the homework, 10% for the presentations and participation in classroom discussion, 20% on in-class examinations, 40% on programming assignments and 20% for the final examination.

Students are required to maintain a journal of thoughts and commentaries during the course. The journal contains daily entries including the identification of areas of interest and concern, notes on the preparation of presentation and comments and analysis of classmate's presentations. The journal is reviewed weekly by the instructor to provide feedback to the students.

Final grades and the student and instructor observations from reflective notebooks are included in the student's portfolio for use in the final assessment capstone course. The intent is to document the student's maturation as he proceeds through the curriculum.

VIII. Course Format

A. <u>Instruction</u>

This course is primarily a lecture/discussion course. Students are expected to attend three hours of lecture/discussion per week. At least once per week students should be prepared to make presentation summarizing an aspect of the AI literature selected by the instructor and to take part in a discussion based on that presentation. Once a week students should have at least 30 minutes of collaborative problem solving activity. Students should expect make significant use of the university's parallel computing facility outside of lecture class hours.

B. Web supplement

Course home page (the university's Web tool, WebCT or Blackboard) should contain the following:

- Course syllabus
- Course assignments
- Sample solutions to examinations (after being graded and returned)
- Sample solutions to programming assignments (after being graded and returned)
- Course calendar (an active utility)
- Course e-mail (an active utility)
- Course discussion list (an active utility)
- Student course performance (an active utility)

Classroom Hours (3 hours per week)	**Class:**	3
	Lab:	0

IX. Topics to Be Covered

A. Introduction, theoretical basis, system organization, and performance
B. Introduction to message passing
C. Programming with MPI
D. Mesh based computations, performance modeling
E. Performance measurement, scalability, projects
F. MPI datatypes; interconnection networks
G. Matrix multiplication; MPI communicators
H. High performance collective communication
I. Parallel programming languages , OpenMP, POSIX threads
J. Parallel sorting
K. Shared memory architecture and programming
L. Message passing implementation
M. Alternative programming models
N. Directions in high performance computer architecture

X. Laboratory Exercises

This course does not require a separate lab.

XI. Technology Component

This course makes use of the university's wireless access infrastructure. The course relies on the university and the students having access to professional grade application development environments for the students to use.

XII. Special Projects / Activities

Students are required to keep a "reflective notebook" in which, after each class, they enter their own assessments of what they learned, and what questions remain from the class. From each exercise set, each student selects one problem, which the student thinks best reflects the way the topic is used in a technical context. A detailed solution to the problem is included in the student's reflective notebook.

XIII. Textbooks and Teaching Aids

A. Required Textbook

Grama, A., Gupta, A., Karypis, G., and Kuman, V. *Introduction to Parallel Computing*, 2nd Edition, ______: Addison-Wesley, 2003.
ISBN:

B. Alternative Textbooks

None.

C. Supplemental Print Materials

1. Wilkinson, B., and M. Allen. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. _____: Prentice-Hall, 2000.
   ISBN:
2. Pacheco, Peter. *Parallel Programming with MPI*. _____: Morgan-Kaufmann, 1996.
   ISBN:

D. Supplemental Online Materials

None.