Rutgers University, Department of Electrical and Computer Engineering
ABET COURSE SYLLABUS
COURSE: 14:332:451

Course Catalog Description: 14:332:451 – Intro. to Parallel & Distributed Programming (3) Parallel and distributed architectures, Fundamentals of parallel/distributed data-structures, algorithms, programming paradigms, Introduction parallel/distributed application development using current technologies.

Pre-Requisite Courses: 14:332:331, 351

Co-Requisite Courses: None

Pre-Requisite by Topic:

1. Computer architecture
2. Programming Methodology
3. Data Structures
4. Algorithms
5. Operating Systems

Textbook & Materials:

- Peter Pacheco, *An Introduction to Parallel Programming*, Morgan Kaufmann, 2011.
- Lecture Notes.

References:

- Hariri and Parashar (Ed.), *Tools and Environments for Parallel & Distributed Computing*, John Wiley, 2004.
- David Kirk, Wen-Mei W. Hwu, Wen-mei Hwu, *Programming massively parallel processors: a hands-on approach*, Morgan Kaufmann, 2010.
- Kay Hwang, Jack Dongarra and Geoffrey C. Fox (Ed.), *Distributed and Cloud Computing*, Morgan Kaufmann, 2011.

Overall Educational Objective: To introduce the fundamentals of parallel and distributed programming and application development.

Course Learning Outcomes: A student who successfully fulfills the course requirements will have demonstrated:

1. An understanding of the fundamentals of parallel and distributed computing including parallel/distributed architectures and paradigms.

2. An understanding of parallel/distributed algorithms and key technologies.

3. An ability to develop and execute basic parallel and distributed application using basic programming models and tools.

4. An understanding of performance issues in parallel/distributed computing and an ability to make appropriate design trade-offs during application

development.

5. The ability to apply parallel/distributed computing for problem solving.

How Course Outcomes are Assessed:
HW Problems & Quizzes (30 %)
Two Mid-Term Exams (40 %)
Final Project (Teams of 2) (30 %)

N = none S = Supportive H = highly related

Outcome	Level	Proficiency assessed by
(a) an ability to apply knowledge of Mathematics, science, and engineering	H	HW Problems, Project
(b) an ability to design and conduct experiments and interpret data	H	HW Problems, Project
(c) an ability to design a system, component or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	S	Project
(d) an ability to function as part of a multi-disciplinary team	H	Project
(e) an ability to identify, formulate, and solve ECE problems	H	HW Problems, Exams
(f) an understanding of professional and ethical responsibility	N	
(g) an ability to communicate in written and oral form	S	HW Problems, Project
(h) the broad education necessary to understand the impact of electrical and computer engineering solutions in a global, economic, environmental, and societal context	N	
(i) a recognition of the need for, and an ability to engage in life-long learning	S	Home-work, discussions during lectures
(j) a knowledge of contemporary issues	N	
(k) an ability to use the techniques, skills, and modern engineering tools necessary for electrical and computer engineering practice	H	HW Problems, Project, Exams
Basic disciplines in Electrical Engineering	N	
Depth in Electrical Engineering	N	
Basic disciplines in Computer Engineering	H	HW Problems, Project, Exams
Depth in Computer Engineering	H	HW Problems, Project, Exams
Laboratory equipment and software tools	S	HW Problems, Project
Variety of instruction formats	S	Lecture, office hour discussions

Topics Covered week by week:

Week 1: Introduction to Parallel and Distributed Programming (definitions, taxonomies, trends)
Week 2: Parallel Computing Architectures, Paradigms, Issues, & Technologies (architectures, topologies, organizations)
Week 3: Parallel Programming (performance, programming paradigms, applications)
Week 4: Parallel Programming Using Shared Memory I (basics of shared memory programming, memory coherence, race conditions and deadlock detection, synchronization)
Week 5: Parallel Programming Using Shared Memory II (multithreaded programming, OpenMP, pthreads, Java threads)
Week 6: Parallel Programming using Message Passing - I (basics of message passing techniques, synchronous/asynchronous messaging, partitioning and load-balancing)
Week 7: Review/Midterm I
Week 8: Parallel Programming using Message Passing - II (MPI)
Week 9: Parallel Programming – Advanced Topics (accelerators, CUDA, OpenCL, PGAS)
Week 10: Introduction to Distributed Programming (architectures, programming models)
Week 11: Distributed Programming Issues/Algorithms (fundamental issues and concepts - synchronization, mutual exclusion, termination detection, clocks, event ordering, locking)
Week 12: Distributed Computing Tools & Technologies I (CORBA, JavaRMI)
Week 13: Distributed Computing Tools & Technologies II (Web Services, shared spaces)
Week 14: Distributed Computing Tools & Technologies III (Map-Reduce, Hadoop)

Week 15: Parallel and Distributed Computing – Trends and Visions (Cloud and Grid Computing, P2P Computing, Autonomic Computing)
Week 16: Review/Midterm II

Computer Usage: Current parallel/distributed programming technologies (e.g., MPI, OpenMP, pthreads, CUDA, JavaRMI, Web Services, MapReduce, etc.) on networked systems.

Laboratory Experiences: The course will consist of multiple programming assignments and a final project that will consist of building and evaluating parallel/distributed applications using current technologies and networked-systems.

Design Experiences: "Hands-on" course with programming assignments and a final project.

Independent Learning Experiences: 1. Home-Work, 2. Term Paper, 3. Testing (Quizzes, Exams)

Contribution to the Professional Component:
(a) College-level mathematics and basic sciences: 0 credit hours
(b) Engineering Topics (Science and/or Design): 3.0 credit hours
(c) General Education: 0 credit hours
Total credits: 3
Prepared by: M. Parashar
Date: May 2011