ICS 556 — Parallel Algorithms

Term: 062 Section: 1

Time & Place: SM 5:00 – 6:15, Bldg 24-162

INSTRUCTOR: Ebrahim Malalla
OFFICE: Bldg 22 (124-8)
PHONE: 860-3819
E-MAIL: malalla@kfupm.edu.sa

COURSE SITE: http://www.ccse.kfupm.edu.sa/~malalla/ICS556/index.htm
OFFICE HOURS: SMW 12:30 – 2, and whenever you catch me.

DESCRIPTION

After introducing the parallel computational models (PRAM, Meshes, Trees, Hypercubes, Shuffle-Exchange, Mesh-of-Trees) and the complexity measures, the course covers various parallel algorithms design techniques such as divide-and-conquer, parallel prefix, pointer jumping, list ranking, and Euler's path technique. Applications are ranging from classical problems like selection, merging, sorting, searching, graph problems and computational geometry to more advance problems in scientific computations, signal processing, and simulations. Students will study also some parallel computational complexity classes: equivalence of Boolean circuits and the PRAM models, the NC class, and P-complete problems.

PREREQUISITES ICS 553

COURSE OBJECTIVES

1. To know the fundamental concepts and techniques of parallel computing.
2. To learn how to design and analyze parallel algorithms to solve given problems in specific parallel computation models.

COURSE LEARNING OUTCOMES

After completion of this course, the student should be able to:

1. define and compare between the structures of classical parallel computation models.
2. use the metrics of cost, speed-up and efficiency to analyze the performance of given parallel algorithms and compare between them and their sequential counterparts.
3. design a parallel algorithm to solve a given new problem in a specific parallel computation model.
4. compare between the parallel computational complexity classes.

TEXTBOOK

In addition to the lecture notes we will use the following textbook is

Selim G. Akl, ***Parallel Computation: Models and Methods,*** Prentice Hall, 2nd Ed., 1997.

Students are also encouraged to refer to other books on parallel computing available in the library. Some of the highly recommended books are:

1. J. Jaja, *Introduction to Parallel Algorithms*, Addison Wesley, 1992.
2. C. Xavier and S. S. Iyengar, *Introduction to Parallel Algorithms*, John Wiley & Sons, Inc, 1998.
3. B. Wilkinson & M. Allen, *Parallel Programming Techniques & Applications Using Networked Workstations & Parallel Computers*, 2^{nd} Ed., Pearson Education Inc, 2004.
4. F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays . Trees . Hypercubes*, Morgan Kaufmann Publishers, 1992.
5. A. Grama, A. Gupta, G. Karypis, V. Kumar, *Introduction to Parallel Computing: Design and Analysis of Algorithms*, 2^{nd} Ed., Addison-Wesley, 2003.
6. K. Berman and J. Paul, *Algorithms: Sequential, Parallel and Distributed*, Thomson-Course Technology, 2005.
7. R. Miller and L. Boxer, *Algorithms Sequential and Parallel-A Unified Approach*, Prentice Hall Inc, 2000.

EVALUATION

Assignments	20%
Major Exam I Mon. Apr. 2^{nd}	25%
Major Exam II May Mon 21^{st}	30%
Presentation & Report*	25%

***** Each student is to write a report on state-of-the-art research papers pertaining to the area of parallel algorithms and give a brief presentation about the report. It is his responsibility to search for a suitable and recent problem. Presentations will be scheduled at the end of the semester, and hence you are expected to start searching by the end of the 7th week.

TENTATIVE COURSE CONTENTS

1. **Introduction to parallel computational models**. Examples of parallel algorithms on the PRAM, interconnections networks, and combinatorial circuits parallel computational models. Parallel algorithms analysis measures: time, space, cost, speedup, efficiency.
1. **Parallel algorithms design techniques:** divide-and-conquer, parallel prefix, pointer jumping, list ranking, Euler's path technique.
2. **Parallel algorithms** for searching, merging, selection, sorting and Fast Fourier Transform.
3. **Graph algorithms**: connected components, minimum spanning trees, shortest paths, ear decomposition.
4. **Computational geometry**: convex hull, the closest pair problem.
5. **Parallel randomized algorithms**: routing in hypercube, maximal independent set, maximum matching.
6. **Parallel computational complexity**: equivalence of Boolean circuits and the PRAM models, the NC class, P-complete problems.

REMINDERS

1. The course website/WebCT is an important source of information. It will be updated regularly to contain up-to-date announcements, assignments, handouts, etc.
2. By the university rules, 9 absences yield a DN grade.
3. No assignments would be accepted without penalty after the due date.