

HONR219A, Spring 2008

Towards a New Era of Desktop Supercomputing

Time and Location: MW 2:00-3:15, CHE 2116

Prof. Uzi Vishkin, Department of Electrical and Computer Engineering

Course web site: <http://www.umiacs.umd.edu/users/vishkin/TEACHING/honr219a-s08.html>

INTRODUCTION

Commodity computers are undergoing a fundamental change, unprecedented since the advent of computers in the 1940s. In the first chapter of computer history that spanned from 1946 through 2003, clock frequency had been improving at a stunning exponential rate. As of 2004, clock frequency is hardly improving and is not expected to improve much due to implementation issues, such as power consumption. However, the number of transistors on a single chip keeps increasing at an exponential rate as before and is expected to reach tens of billions by the time you graduate, up from tens of thousands 30 years earlier. The 2007 fourth edition of *Computer Architecture: a Quantitative Approach*, by J. Hennessy and D. Patterson—perhaps the single most popular computer engineering textbook--summarizes the situation as follows: "...this fork in computer architecture means that for the first time in history, no one is building a much faster serial processor. If you want your program to run significantly faster ... you're going to have to parallelize your program."

The computer of the present is already parallel and parallelism is expected to grow exponentially in future computers, as noted above. Unfortunately, it is unclear at the present time how software will take advantage of the growing hardware parallelism. Microsoft leaders take current multi-core architectures for granted, but predict that it may take 10 years to resolve this uncertainly. Others question the very viability of such architectures. This uncertainty already creates serious investment dilemmas for software vendors: making a choice among several schools-of-thought risks betting on the wrong horse. The education enterprise is facing a perhaps tougher predicament. Doing nothing means that 22-year olds who studied nothing but serial computing head for a 50-year career that in some fields will be dominated by parallelism.

The course will debate possible actionable agendas for the business, education and government sectors. Most of the class time will be spent on the articulation of an adequate level of cognition of parallelism through the study of the common denominator among the available schools-of-thoughts. We will explore the fascinating intellectual challenge that the rather common task of programming is going to entail. Parallel algorithms and parallel programming represent an "alien intellectual culture." Unlike standard serial programming that is naturally related to previously understood forms of reasoning (e.g. mathematical induction), or familiar activities in our physical world, little has prepared us for the parallel computing era that is upon us.

Students will learn parallel algorithmic thinking and acquire hands-on experience by programming a new 64-processor computer built by Dr. Vishkin's research team at UMD, as it was engineered to support the common denominator level of cognition explained above.

For a recent UMD press release on the new computer, please look up:

http://www.eng.umd.edu/media/pressreleases/pr062607_supercomputer.html

Readings will include: handouts on parallel algorithms, as well as general (New York Times, Wall Street Journal) and more focused (EE Times, Dr. Dobbs Journal) articles in popular press, presenting the timely dilemmas faced by this core area of information technology. Regarding the latter, students will be expected to demonstrate understanding of the interconnections of knowledge and its connections to past, present, and future developments; delineate and describe the importance of studying and researching this emerging challenge; articulate understanding of ways in which information and knowledge are connected to past accomplishments in the field as well as current developments; and use the programming experience learned in the course, along with information provided by the instructor, or through independent search to reason, analyze, evaluate, and assess game plans by different players in the field, their public claims, and future directions.

Prerequisite:

Students who do not major in computer science, or computer engineering are particularly encouraged to take this course. Basic experience with a programming language such as C or Java is needed. No other background is required, but the course is not recommended for students who are intimidated by basic mathematical concepts, such as mathematical induction.

Students who take ENEE459P/CMSC498V in Spring 2008 cannot get credit for HONR219A.

SYLLABUS INFORMATION

Course Goals

Introduction to the theory of parallel algorithms to general UMD Honors students. The course will highlight parallel algorithmic thinking for obtaining good speed-ups with respect to serial algorithms.

Close examination of one of the most exciting applied research questions facing computer science and engineering: Will the current “Billion-transistor-per-chip” era provide a way for building a truly parallel computer system on-chip? The examination has several parts: (1) Business: can the needed paradigm shift succeed in the business world? And, if yes, how? (2) Technical: (i) operational (i.e., how will the system work?), (ii) functional (i.e, for what application it will be useful?), and (iii) facilitating science and technology.

Prerequisite Topics

Basic computer programming.

Course readings:

Class notes and other readings will be provided by the instructor. Please download and print out the first item (104 pages) under

<http://www.umiacs.umd.edu/users/vishkin/PUBLICATIONS/papers.html>

Material presenting the unfolding debate on expectations from parallel computing, in general, and the “Billion-transistor-per-chip” era, in particular. Material provided will include position papers by experts from industry and academia, reports commissioned by Federal agencies, and articles featured in the popular press.

Reference books:

- J. JaJa, An Introduction to Parallel Algorithms, Addison Wesley, 1992.
- D.E. Culler and J.P. Singh, Parallel Computer Architecture, Morgan Kaufmann, 1999. The following quote appears under the title Potential Breakthroughs: ..."breakthrough may come from architecture" ... "that is, to truly design a machine that can look to the programmer like a PRAM".
- J.L. Hennessy and D.A. Patterson. Computer Architecture a Quantitative Approach, 4rd Edition. Morgan-Kaufmann, San Francisco, 2007.

Some Core Topics:

Introduction to Parallel Algorithms:

- Performance of Parallel Algorithms

- Optimality Notions

Basic Techniques:

- Balanced Trees
- Pointer Jumping
- Divide-and-Conquer
- Partitioning
- Pipelining
- Accelerated Cascading
- Breaking Symmetry

Trees and Lists:

- List Ranking
- The Euler Tour Techniques
- Tree Contraction, Evaluation of Arithmetic Expressions

Searching

Merging

Sorting

Graphs:

- Connected Components
- Paths Problems

Introduction to Parallel Processing:

- Principles of available multi-processors
- Parallel Models
- Critical reading and discussion of the unfolding debate on expectations from parallel computing, in general, and the "Billion-transistor-per-chip" era, in particular.

Course work

Course work will include: 5 programming assignments, and an essay (8-10 page) related to the current and emerging issue of the paradigm shift from serial to parallel computing that the field of computing is undergoing. This paper will encourage students to think seriously about this topic and develop their own independent understanding of it.

Students will be allowed to revise their papers to improve their grade.

The course will involve a variety of traditional algorithms and programming instruction methods, as well as non-traditional approaches, including class discussions.

Grading: Participation 10%; contribution to class discussion 15%; essay 25%; programming assignments 50%.

DOCUMENTED DISABILITY

If you have a documented disability and wish to discuss academic accommodations with me, please contact me as soon as possible.

ACADEMIC INTEGRITY

Academic dishonesty will not be tolerated. The University Code of Academic Integrity, which can be found at <http://www.inform.umd.edu/CampusInfo/Departments/JPO/> prohibits students from committing the following acts of academic dishonesty: cheating, fabrication, facilitating academic dishonesty, and plagiarism. Academic dishonesty in this class includes outright copying on homework; however, discussing homework problems and exchanging tips is permissible and also encouraged. If there are any take-home exams, discussing the material with anyone, inside or outside of the class, is considered academic dishonesty. Instances of academic dishonesty will be referred to the Office of Judicial Programs.