CSE 471: Computer Design and Organization
Spring 2012
Lectures: TTh 10:30-11:50 AC 203
Section: Th 1:30-2:20 AC 203

Instructor

Susan Eggers, AC 458, 543-2118, eggers@cs.washington.edu
Office Hours: Tuesdays, 1:30 - 2:20, or by appointment

TA

Brandon Myers, AC 490, bdmyers@cs
Office Hours: Mondays, 2:00-3:00

Course Material

The purpose of this course is to give you a broad understanding of the concepts behind several advanced microarchitectural features in today's microprocessors and to illustrate those concepts with appropriate (usually modern) machine examples. Since parallel programming is currently the biggest unsolved issue in computer architecture, we'll focus primarily on parallel designs and issues that relate to them. We will also cover designs that are based on two other important concepts in computer architecture: locality and speculation. In particular, we'll cover the rationale for and the designs of strategies for dynamic branch prediction (speculation), multiple-instruction issue (instruction-level parallelism), dynamic (out-of-order) instruction scheduling (instruction-level parallelism), caching techniques (locality), chip multiprocessors thread-level (parallelism), multithreaded processors (thread-level parallelism), vector and graphics processors (data parallelism), and dataflow machines (all sorts of parallelism). Some of these topics depend on material you covered in 351; for these, we'll quickly review that material, and then go on from there.

You'll augment your knowledge of the architectural concepts and designs by doing experimental studies that examine and compare the performance of several alternative implementations for a particular feature. Here you'll learn: (1) how to pose an architectural research problem, (2) how to determine the severity of the problem in practice, (3) how to solve the problem, (4) how to design architectural experiments that reveal the severity of and solution to the problem, (5) how to choose metrics that best illustrate a design's performance, (6) how to analyze performance data, and (7) how to write up your experiment and results − all skills you'll need if you plan to do computer evaluation either in development or research, and actually in any applied subfield of computer science, not just computer architecture.

You must have already taken 370/378 or 311/351/352 in order to take this class.

Reading

Most reading assignments will be taken from *Computer Architecture: A Quantitative Approach* by John L. Hennessy & David A. Patterson, Morgan Kaufmann, 5th edition, 2011. To get the most out of the lectures, **read the material *before* topics are discussed in class**. There will also be some supplementary reading that you will be able to access from the course web pages.

Schedule

There is a daily schedule in the course web area. The schedule will tell you what topics we will cover and when, what reading should be done before you come to a particular lecture, and when homework is due and exams will be held. This document is a work in progress. I'll be updating it continuously, as I plan each set of lectures. You should check it frequently, so that you can anticipate what material we'll be covering.

Class Discussion

Before each class, think about what you've read for the upcoming lecture and about the material in the previous lecture, and come prepared to pose and answer questions, present your views of the architectural schemes we discuss, and offer alternatives. Although I'll be lecturing, this class is part seminar. And the more seminar-like it is in reality, the more fun we'll have!

Exams

There will be two midterms, one in the middle of the quarter and one at the end. The one at the end will cover material from the second part of the course only.

Homework

There will be three or four homework assignments, each progressively more challenging. The first one is an empirical study that will give you experience in designing architectural experiments and evaluating architecture features, and will hone your intuitions about the performance ramifications of changing certain aspects of a hardware implementation – in this case branch prediction hardware. You'll also make your first stab at writing up your results in research-paper style. The second and third will involve designing and empirically evaluating a multiprocessor cache coherency scheme. If there is a fourth, it will either explore a variety of issues in concurrency or require you to empirically exploit different forms of thread- and data-level parallelism. In the discussion sections Brandon will explain more about the different assignments and how to use the simulators.

You should work in teams of two students for each project. Try to work with a different partner for the first and second/third assignments.

All project reports are due at the beginning of class; no late assignments will be accepted.

Machines

We'll be using attu, a cluster of four Dual Pentium IV 2.8GHz Xeon servers with 4GB of memory and running Linux 2.6.12. You may use the workstations in AC 002, AC 006 or AC 022 to log into attu. Alternatively you may use other computers to log in.

Grading

Grades will be computed using the following **approximate** weighting: the first midterm = 15%, the second midterm = 30%, and all the homework = 55% for all. This may change somewhat, depending on the size and difficulty of the last two homeworks.

Collaboration

Discussing the course content with fellow students is an effective way to learn the material, and is encouraged – this is why you'll be working in teams on the homeworks. However, exams must represent your own mastery of the material.

Communicating

We will communicate a lot through e-mail. Brandon and I will be emailing out assignments and clarifications of the assignments, if needed. And you should use e-mail for asking and answering each others' questions. (But if you have questions that need a detailed or long explanation, it would be much easier to come to our office hours, Brandon for the assignments, Susan for the lectures.) So register on the class mailing list **immediately**. To add yourself to the class email list, follow the instructions on the 471 web pages.