

## **ECE7995**

# **Caching and Prefetching Techniques in Computer Systems**

Instructor: Dr. Song Jiang (sjiang@eng.wayne.edu)

The ECE Department

Lecture: Tuesday/Thursday 10:40AM - 12:30PM

Location: 54 MANO

Office hours: Tuesday 2:30am---4:00am

Engineering Building, Room 3150

# What is the course about?

- Learn how the two most effective and widely used techniques in CSE  
--- caching and prefetching
  - Caching is to temporarily buffer data in a **faster** storage device for efficient **reuse**.
  - Prefetching is to **speculatively** fetch data into a faster storage device **before** an expected request for them is generated to a slower storage device.
  - By servicing a seemingly **expensive** request with a much reduced cost, these techniques are effective.
  - While **speed differences** among storage devices deployed at different levels of a **hierarchical structure** are ubiquitous, these two techniques are widely applied: from processor cache, main memory, hard disk, to proxy server.
  - The large impact of these techniques in CSE warrants a **rewarding** learning effort with a dedicated course

# What is the course about? (cont'd)

- Use a concentrated topic as example to understand how various design choices are made in CSE.
  - While many alternatives may exist even for one component design, a judicious decision-making process is well exemplified in these techniques
- Cover both fundamental principles and state-of-the-art techniques
  - learn not only **how**, but also understand **why**.
- Have a taste of what the research in the CS/CE area looks like
  - Foster your curiosity in computer system research
- Learn by doing
  - You are guided to obtain **hands-on experience** though Linux kernel hacking, though this is not a programming course.

# Why you should take the course

- Caching and prefetching are two of the most techniques leading to high performance
  - From computer system design to application coding, the areas where these techniques are essential are numerous.
- Linux is increasingly popular, especially in enterprise-level computing.
  - Linux rules the Top 500. (Linux machines account for 372 out of the 500 fastest machines, and various UNIXes account for other 109 machines, (2005 data))
  - Most people don't understand Linux, especially how it works – including sysadmins and computer scientists!
- Research experience will go a long way for your career development.
  - The course will provide a path for independent scholarship beyond the fall semester

# The Organization of Course (tentative)

Part I: Lectures by instructor about caching and prefetching

- Motivation and principles
- Processor cache and row-buffer in DRAM
- Memory buffer management
- Web Proxy caching and prefetching
- Tutorials on lab projects

# The Organization of Course

Part II: Paper reading and presentation by students.

- Papers on various topics will be posted.
- Each student selects a paper on a particular topic, read the paper.
- Each student gives one presentation on his/her selected paper.

# The Organization of Course

## Part III: Research projects and presentation

- Select one topic covered in the course;
- Find the issues in the topic;
- Propose your solution;
- Design your experiment to evaluate the solution;
- Present your work;
- Write a report on the work.
- (Optionally) Submit a paper for publication!

# Workload on Students

- Attend classes and be actively involved.
- Two homework assignments on understanding lectures
- Three Linux kernel labs (step-by-step instructions will be provided, and modest C programming skills are required)
- Paper reading / presentation
- Midterm (open book)
- Research project.

## Am I ready to deal with them?

- Modest knowledge on computer architecture, OS, and C programming skills is required.
- The actual workload differs for students with different background.
- Attempt to start your work, such as labs and preparation of exams, at the last minute would warrant a failure.

## Course Facts

- Meet Monday/Tuesday 10:40pm --- 12:30pm
- Check website regularly  
<http://www.ece.eng.wayne.edu/~sjiang/ECE7995-winter-09/ECE7995.htm>
- Send course-related email to  
[sjiang@eng.wayne.edu](mailto:sjiang@eng.wayne.edu)
- Office hours: Tuesday 2:30am---4:00am or by appointment at 3150 Engineering building

# Reading Materials

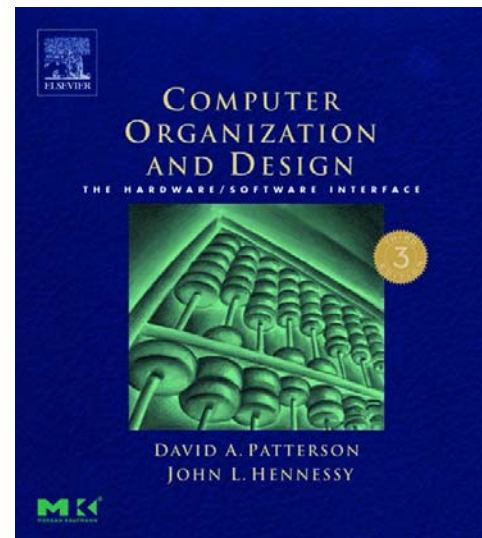
- Suggested textbook that covers basic design of processor cache.

David A. Patterson, John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Third Edition (The Morgan Kaufmann Series in Computer Architecture and Design)

(or any book about operating system that covers buffer cache design. )

The above books are not required. You can find almost all the contents on the internet.

- Will post lecture slides, reading materials, and assignments on the course webpage.



# Late Policy

- In general, no late submissions will be accepted.
- However, you have three (3) late days:
  - Self-granted extensions, no need to ask for permission.
  - You use one late day even if you pass the due time by one minute
  - When late days are used up, your grade is deducted by one level for each day you pass deadline (e.g., from A- to B+ or from B+ to B)
  - This policy will be strictly enforced.
- Contact instructor in extraordinary circumstances only.
  - Job interviews do not count.

# Grading

- 50%: homework assignments
  - 20%: two writing homework assignments
  - 30%: three Linux kernel programming labs
- 15%: paper reading and presentation
- 15%: mid-term (open book)
- 20%: research project and final report on self-selected topic (No Final Exam)
  
- Final grade will be distributed as
  - A (100-90), A- (89-85), B+(84-80) B (79-75), B-(74-70), C+(69-65), C(64-60), C-(59-55) F (below 55).

# Honor Code

- Will be enforced in this class;
- Avoid plagiarism, copyright infringement, and other types of academic dishonesty.
- No collaboration to specific questions in assignments, lab, and exams is allowed, while general discussions on understanding course materials and setup of coding environment are encouraged.
- As a rule of thumb, cite the source of both written and verbal contributions to your ideas and products.