Special Topics in Computer Architecture

Course Overview

Hitoshi Oi

The University of Aizu

April 12, 2017

AY2017, 1st Quarter

Essential Course Information

Course Number and Title: SYA03 Special Topics in Computer Architecture

Class Schedule Wed, 4th and 5th Periods at RQ 364 (S12)

Instructor Hitoshi Oi, `hitoshi©u-aizu.ac.jp`, ext 2580

Office Hours By appointment only

Grading Scheme Project report (60%) + Presentation (20%) + Class Participation (20%) (tentative).

Website `http://www.u-aizu.ac.jp/~hitoshi/COURSES/SCA/`

Schedule Outline

- Today:
  - Introduction of the course
  - Sample topics for the projects (with the cases of past students)
  - Discussion for helping students to choose project topics
- Next Week:
  - Literature study (material provided today)
  - More discussion on project topics (students should have (at least) some ideas on what he/she wants to do by the end of next week)
- Last Week: Project Report Presentation

Objectives and Course Organization

- This is a project-oriented course to learn and experiments in design and implementation techniques of modern computer systems.
- By "project-oriented," you need to work proactively to choose your project topic and to perform necessary tasks.
- It is assumed that you understand the prerequisite courses (what are they ???)
- It is also assumed that you are literate in using computer systems (and capable of finding ways to do something you need to do).
- If you are not, literature study is also possible (though...)

Topics Covered

Through the prerequisite courses, students must have understood basic (mostly processor) architecture. Therefore, the topics covered by this course should include

- Exploitation of parallelism in various levels (superscalar, out-of-order execution, multi-threading, multi-core)
- Power management and energy efficiency
- Memory hierarchy (cache, main memory, TLB)
- I/O devices (disks, network interface)
- Applications and benchmark programs
- Performance evaluation methodologies (instrumentation, simulation)

Sample Topic (1)

Effectiveness of Microarchitecture (simulation)

- Modern CPUs are implemented with various microarchitectural techniques
- However, their effectiveness varies with the characteristics of the workload. For example a bigger cache may accelerate program A but not program B
- A ’cycle-accurate’ simulator is needed for this type of study. It simulates the operations and events within a CPU, but could be quite slow (can take $\times 10^4$ or longer than real CPU).
- Simplescalar might be a good starting point (although could be too simple)

Sample Topic (2)

Effectiveness of Microarchitecture (profiling)

- With a simulator, you can change CPU parameters as you wish (e.g. changing the cache size is not possible in a real hardware)
- However, a simulator is a simulator. Lots of details would be omitted.
- Also, due to the resource (e.g. memory) and slow-down mentioned previously, only the scaled-down programs can be executed.
- Rather you may want to run real applications and instrument what are happening inside the CPU.

Continued to the next slide...

Sample Topic (2) Continued

- Modern CPUs have a mechanism called ’performance counters’ which count the occurrences of the events, such as cache misses, taken branches.
- Tools for profiling workloads using performance counters are also available (e.g. Linux perf).
- A good news: we have a free access to SPEC benchmark programs since we are an academic member of SPEC (otherwise, they would cost $$$).
- You need to know the internal of the CPU for the profiling (most events are model-specific).
- You may also use other profiling tools, such as `mpstat`.

Sample Topic (3)

Trace Analysis

- Instruction execution is initiated with an instruction fetch, followed by (optional) reading and writing of operands; all stored in the main memory.
- There are some repositories of memory traces of (typical) program executions.
- You write a memory hierarchy simulator (which should be much simpler than cycle accurate CPU simulators), and feed such traces to your simulator.
- Disk access trace analysis with a simulator (e.g. `disksim` from Carnegie Mellon) is also possible

Sample Topic (4)

Power Consumption Management

- Power consumption is a major (sometimes, most significant) concern for computer systems.
- Modern CPUs have various techniques to manage the power consumption and dynamics voltage and frequency scaling (DVFS) is one of them. You may tune its behavior using the `/sys` file system.
- Recent Intel CPUs are equipped with a mechanism called "running average power limit (RAPL)" which works similarly to the performance counters but profiles and limits the power consumption. Playing around with RAPL can be an option.

Sample Topic (5)

Studying Standard Benchmark Programs

- CPUs (or computer systems) are designed with various techniques but their effectiveness varies among workload
- Standard benchmark programs (e.g. those from SPEC or TPC) are chosen to be fair, realistic and useful.
- Study, how and why each benchmark was chosen, the area of the application, the workload behavior, or data structures used in the program.
- This topic CANNOT be just a summary of the document included in the benchmark kit.

Reading Assignment by Next Week

- Will provided a paper by the end of today (either by email, placing it somewhere on the (internal) network (e.g. moodle, or Google Docs)
- Divide the paper into three parts. Talk with other students in the class and decide who takes which part.
- Next week, you need to talk (explain) the part you have taken. Don't need to make slides (no need to be a ppt-meister). Just browsing the pdf file and zoom in where you are explain (with highlighting important part), can be an option.

Now let's discuss to find your project topic

- Any of the sample topics seemed attractive ?
- Or, already have something in your mind ? (OK if it's vague at this moment)
- What is your technical expertise ? (e.g. programming language, OS, utility tools)
- Or, good in math ?
- What was your thesis topic ? Or, any project in the past ?