

Introduction to Research in Computer Architecture

Overview

This is a 10hp course in learning basic tools and techniques for architecture research. The course is expected to take about 20h/week for 6 weeks. The goal is to learn a variety of tools and techniques, gain experience analyzing data and benchmarks, practice writing up reports and giving presentations, and start reading up on current research results.

Format

6 week course. You should work together with one other partner. Each person must do at least half the work. (E.g., it is okay if you both do your own implementations.) You should work actively together with the other students in the class, but group of needs to do its own implementation, data collection, analysis, report, and presentation.

There are two main parts to this course:

1. Paper reading: reading 2 papers each week and reporting on them
2. Cache analysis: exploring 5 different ways to analyze cache effects

Paper Reading

Each week each student will select a different paper from a recent ISCA, MICRO, HPCA, CGO, or ASPLOS conference to read. (Coordinate among yourselves to avoid overlap.) Each week all students will be assigned an additional paper to read and each student will give a 5-minute overview of the paper they read and must identify one great part of the paper and one bad part, and be prepared to answer questions from the rest of the group.

Group papers for each week:

1. D. Jevdjic, S. Volos, and B. Falsafi. "Die-stacked DRAM caches for servers: hit ratio, latency, or bandwidth? Have it all with footprint cache." ISCA 2013.
2. R. Hameed, et. al. "Understanding sources of inefficiency in general-purpose chips." ISCA 2010.
3. M. Rhu, M. Sullivan, J. Leng, and M. Erez. "A Locality-Aware Memory Hierarchy for Energy-Efficient GPU Architectures." MICRO 2013.
4. J. Zebchuk, B. Falsafi, and A. Moshovos. "Multi-Grain Coherence Directory." MICRO 2013.
5. A. Putnam, et. al. "A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services." ISCA 2014. (Read this before going to the SICS multicore day.)
6. A. Arelakis and P. Stenström. "SC2: A Statistical Compression Cache Scheme." ISCA 2014.

Project: Cache Analysis

Analyze the cache behavior of two SPEC 2006 benchmarks (lbn and sphinx3) using several different techniques to learn the pros/cons of each. The techniques are:

1. Simulating a cache from an address trace
 - a. Write a pin tool to capture all memory addresses
 - b. Write a simple cache simulator for associativity/LRU/size
2. Modeling with memory access samples
 - a. Read StatStack and StatCache papers
 - b. Use the group pin tool to sample or sample from your address trace
 - c. Write your own StatCache model
3. Performance counter measurements with a Pirate
 - a. Read Pirate and Bandit papers
 - b. Write a pirate that steals cache
4. Full system simulation in gem5
 - a. Read the Full Speed Ahead paper
 - b. Setup gem5
 - c. Create benchmark checkpoints
5. LLVM trace generation
 - a. Create an LLVM pass that instruments memory instructions to generate an address trace
 - b. Run it through your cache simulator

For each method and benchmark, determine the effect of varying the cache size from 16kB to 16MB by powers of 2 (11 steps). You should try to capture data about performance, cache behavior (# accesses, miss/hit ratio, miss/hit rate), and prefetcher behavior.

Your goal is to compare the different methods to see how they are similar/different and analyze the pros/cons of each for research and analysis: accuracy, flexibility, performance, and the kind of data you can collect. Be sure to keep data on how long your experiments take to run!

Be sure to figure out whether the applications have rapidly changing behavior or not so you know what you need to analyze.

Not all of the methods listed above can capture all of the data requested, but all can get something for some cache sizes.

If you run into any interesting questions, be sure to bring them up in the discussions and the report. You should come up with interesting questions/issues!

You will not be able to capture data for the full application runs for all of the methods due to storage space and/or speed. To get around this you will probably need to fast-forward into the application and then gather data for a limited window, sample multiple windows within the application run, or use phase detection to determine where in the application to sample and/or warmup the caches before collecting data. Make sure you figure out which you need to do before you get started, based on the application's behavior! Be sure to use the same approach for all of your measurements and comment on its impact on the final results.

Report

You have 10 pages to write up a report from your results in standard conference format. (E.g., LaTeX with bibtex citations.)

- Abstract: why this is important, what you did, what you found (3 paragraphs)
- Introduction: why looking at the effect of cache size is important and why this range is interesting.
- Related work: what others have done to look at cache effects. You need to cite at least two works for each approach, and at most 4 works from Uppsala.
- One section for each technique, explain the setup, the results, and the issues
- Analysis: how do they compare, what are the pros/cons/issues
- Conclusion: what have you learned, what should others take away from your work?

Presentation

You will give a 15 minute presentation to the group on what you've done and how it worked out. You will have to give a practice presentation before your final one. Expect tough questions given the group. See the Presentation checklist sheet. Schedule a practice presentation with other students and have them record you so you can see your own performance. Use the presentation reflection and feedback forms.

Schedule

Each week the course will have a meeting to go through papers and progress.

Week 0	First meeting
Week 1	Address trace simulation results
Week 2	Modeling result
Week 3	Pirate due result
Week 4	gem5 simulation results
Week 6	llvm results
	Attend SICS Multicore Day on Oct 20th!
Week 7	Final report and presentations due Friday

Bonus

Show the impact on power as the applications get more or less of the cache. (No help on this one: figure out how to do it and what is important on your own, but feel free to ask questions. This won't affect your grade in the course, but will impress us.)