

CS231, DLDCA Lab, Lab 09

Goals

1. Learning to use a cache simulator
2. Understanding and analysing cache performance using program traces on the cache simulator

Instructions

1. These exercises are to be done individually.
2. While you are encouraged to discuss with your colleagues, do not cross the fine line between discussion *to understand* versus discussion as a *short-cut* to complete your lab without really understanding.
3. Create a directory called <rollno>-<labno>. Store all relevant files to this lab in that directory.
 - a. In the exercises, you will be asked various questions. Note down the answers to these in a file called “answers.txt”.
 - b. In some parts of the exercises, you will have to show a demo to a TA; these are marked as such. The evaluation for each lab will be in the subsequent lab, or during a time-slot agreed upon with the TAs. For this evaluation, you need to upload your code as well.
4. Before leaving the lab, ensure the following:
 - a. You have marked attendance on SAFE
 - b. You have uploaded your submission on BodhiTree, and downloaded and checked if the submissions is right
5. Things to ensure during TA evaluation of a particular lab submission:
 - a. The TA has looked at your text file with the answers to various questions
 - b. The TA has given you marks out of 10, and has entered it in the marks sheet
6. You have to use the MIPS conventions, unless mentioned otherwise.
7. **House points:** Some questions carry house points. For these, you are allowed to work along with others, but only if *all* of those working together have completed the lab. (You can work on it individually, even before finishing the lab completely, but only if you care about house points more than your own marks!). All questions related to house points have to be shown to the instructor (Bhaskar).

Understanding the dineroIV cache simulator

- DineroIV is a cache simulator which takes as input a program's memory access trace. You first have to download and compile the simulator.

- Learn to use the dineroIV simulator. You can use the man page as `man -l /opt/dineroIV/d4.1` or use the `-help` command line option. Learn the various cache configuration options; there may be some which you do not understand, that's alright.
- Also learn the input trace format. There are some memory access traces from real programs, given as part of the lab. Can you interpret the format? The man page above tells you the data format.

Warmup exercise

- Use the following cache configuration: unified cache size of 4KB, 4-word blocks, direct-mapped, write-back+write-allocate.
- **Question [1 mark]:** If the N^{th} memory reference is the first non-compulsory miss, find the value of N. Find N for each of the 3 trace files given. *Hint: the simulator can be told to run for only a certain number of memory references.*

Impact of cache size

- Use the following cache configuration: unified cache, 2-way set associative, 4-word blocks, write-back+write-allocate.
- **Question [3 marks]:** Vary the cache size from 4KB to 512KB (8 different sizes). Plot the miss-rate as a function of the cache size. Your graph should have 3 plots: one for each of the 3 trace files. Explain the shape of the graph. **Files: cache-plots.py cache-size-plot.png** (if you plot using some other software, you can leave the py file empty, but be sure to include the relevant data in answers.txt in that case).

Impact of block size

- Use the following cache configuration: unified cache of 8KB size, 2-way set associative, write-back+write-allocate.
- **Question [3 marks]:** Vary the block size from 1-word to 256-words (9 different sizes). Plot the miss-rate as a function of the block size. The graph should have one plot each for the 3 trace files. Explain the shape of the graph. **Files: cache-plots.py block-size-plot.png** (if you plot using some other software, you can leave the py file empty, but be sure to include the relevant data in answers.txt in that case).

Impact of associativity

- Use the following cache configuration: unified cache of 8KB cache size, 4-word blocks, write-back+write-allocate.
- **Question [3 marks]:** Vary the associativity from the minimum possible to the maximum possible. Plot the miss-rate as a function of the associativity. As earlier, there should be a plot each for the 3 trace files. Explain the shape of the graph. **Files: cache-plots.py assoc-plot.png** (if you plot using some other software, you can leave the py file empty, but be sure to include the relevant data in answers.txt in that case).

House points

- Use the following cache configuration: unified cache of 8KB cache size, 4-word blocks, write-back+write-allocate.

- **10 house points:** This is to be done individually, and 10 points per student is possible. Use appropriate scripts (python or bash) to automate the entire exercise above, including plotting the graphs (using gnuplot or pyplot or such).