CS 341L Fall 2009 Lab 5c

Assigned: Monday, 26 October 2009

Due: 11:59pm, Tuesday 3 November 2009 by attachment as an email to labturnin.cs341l@gmail.com

You should attach your program with your last name and the lab number in the name of the file, *e.g.*, "knockel-lab5c.pdf". Be sure to include the 'c' on the end to indicate that this is the third part.

No late assignments will be accepted unless circumstances warrant an extension for the whole class (e.g., server is down all weekend, etc.). You are expected to do your own work as an individual effort, copying the code of others is considered cheating.

You must do your own work and **submit a pdf, no other format will be accepted**. OpenOffice users can export as PDF, LaTeX users can use pdflatex or ps2pdf. For MS Office users I recommend a freeware program called Bullzip that will let you print to PDF. For making the graphs, you can use GNUPlot, OpenOffice SpreadSheet, or Excel.

The purpose of this lab is to learn about cache tradeoffs.

You must have your cache simulator working properly. Because you should have learned C in 241 and you should know the workings of a basic cache we cannot provide a working simulator to you if you didn't complete lab 5b.

You are to explore various tradeoffs using your cache simulator and submit a PDF report, including embedded graphs and explanations of your results. Your report should be divided into three parts:

Part I: Cache Line Size

Assume an 8-way set associative, 4KB cache with a hit time of 4 cycles, a miss penalty of 100 cycles, and a store penalty of another 100 cycles. For block sizes ranging from 8 to 512, plot two different graphs (each graph should have all five cache traces for the five benchmarks on it): miss rate vs. block size and effective average memory access time vs. block size. Your y-axis should be % in one case and cycles in the other, respectively. Your x-axis should be block size on a logarithmic scale as is Figure 5.8 on page 465 of the book.

Based on these two graphs, write a paragraph explaining the inherent tradeoffs and making a recommendation, based on the five benchmarks, of what the best choice for a block size would be. You should also mention why you think that different benchmarks have different curves, you should be able to find information about these SPEC2000 benchmarks through Google. For comparing the benchmarks, be sure to look at the other outputs, not just the ones you're graphing.

Part II: Associativity

Assume a first-level, 2KB cache with 64-bit cache lines, where the miss penalty is 10 cycles and the store penalty is another 10 cycles. The hit time is determined by the associativity (because of the circuit delay of searching for tags):

Direct mapped => 1 cycle

2-way => 1 cycle

4-way => 2 cycles

8-way => 2 cycles

16-way => 3 cycles

Full => 4 cycles

Plot the miss rate and effective average memory access time as two separate graphs with associativity on a log scale on the x-axis (i.e., your graphs should be similar to the graphs you made for part I but now the x-axis is associativity).

What is the basic tradeoff here? Is it sometimes worth it to allow for a higher miss rate if it makes it possible to reduce the hit time?

Part III: Other considerations

Assume you have a 4-way set associative, 1 KB cache with 128-byte cache lines, a hit time of 4 cycles, a miss penalty of 20 cycles, and store penalty of another 20 cycles. What happens when you increase the associativity to 8-way? Make a table showing the effective average memory access time for 4-way vs. 8-way for all five benchmarks. Did performance get better for every benchmark? If not, can you explain? (Be sure to look at other outputs like the miss rate and so forth and include these details in your explanation).