CS 250 Fall 2017 Homework 10

Due 11:58pm Thursday, Nov.16, 2017

Submit your typewritten file in PDF format to Blackboard

1. Assume an instruction cache with capacity sufficient to hold 10% of the instructions of a program of interest and that this program behaves according to the 90/10 Rule.
   1. Assume the cache is 10 times faster than main memory, which is also the next level below this cache in the memory hierarchy. What does Amdahl’s Law predict for the overall speedup from using this cache?

Fractionenhanced =0.9

SpeedupEnhanced = 10

Speedup Overall = 1/((1-0.9+(0.9/10)) = 5.26

* 1. Assume that the CPI for this computer and program is 1.5 when LOAD and STORE instructions are excluded from the measurement. LOAD and STORE instructions comprise 10% of the total number of executed instructions, hit in the data cache 0% of the time, and take 2 clock cycles when they do hit and 20 clock cycles when they do not hit. What is the overall CPI for this program on this computer?

(0.9)\*1.5+(0.1\*20) = 3.35

* 1. The program has been re-written, interchanging two nested loops so that data accesses to a two-dimensional integer array are sequential (row major order), to words, in a data cache with 4-word blocks, so that the hit ratio for data cache now is 75% (cf. HW08, Question 8). What is the new overall CPI and what is the speedup due to loop interchange?

Cpi for load & store = ((0.75)\*2)+(0.25)\*20 = 6.5

Overall cpi = (0.9)\*1.5+(0.1)\*6.5= 2

Speed up = (3.35/2)\*100 = 167.5%

1. How many direct-mapped cache organizations are possible given 32-bit byte addressing for programmers, word-addressed physical memory, 4-byte words, and a 16-bit tag?

Physical address = tag + line number + block offset

Line number = 32-(16+2)

= 14bit

cache size / block size = number of lines

cache size = 2^14 \* 2^2

= 2^16B

1. A model for virtual memory performance is the average main memory access time, given by  
    Average\_main\_memory\_access\_time = Hit\_Ratio x Hit\_time + Miss\_ratio x Miss\_penalty,  
    which is Equation 12.5 in the textbook.  
   Explain why a faster replacement algorithm improves, has no effect, or worsens each of the factors of the average main memory access time equation.

Generally, algorithm is a step by step procedure to solve a program. The faster replacement algorithm improves the main memory.

It has no effects.

1. Assuming a page size of 8 Kbytes what is the page number in hexadecimal and offset in hexadecimal for addresses 0x00000FFF, 0x00001FFF, 0x00002002, and 0xFFFFDFFD? Pad as needed with leading zeros to obtain hexadecimal representations for your answers.

|  |  |  |  |
| --- | --- | --- | --- |
| Given address | Binary address showing split | Page number | Offset |
| 0x00000FFF | 0000 0000 0000 0000 0000 1111 1111 1111 | 0x00000 | 0xFFFE |
| 0x00001FFF | 0000 0000 0000 0000 0001 1111 1111 1111 | 0x00000 | 0x1FFF |
| 0x00002002 | 0000 0000 0000 0000 0010 0000 0000 0010 | 0x00001 | 0x0002 |
| 0xFFFFDFFD | 1111 1111 1111 1111 1101 1111 1111 1101 | 0x7FFFE | 0x1FFD |

1. A computer has 64-bit virtual addresses, byte-addressed memory, 1 Kbyte page size, 16 M page frames in main memory, and has 8 bits of metadata about each page.
   1. How many physical address bits are there in the address bus to main memory for this computer?

2^34

* 1. A program for this computer uses a total of 8 consecutive pages of virtual memory to hold all of its instructions and data. How many bytes of main memory will be consumed for the page table of this program if the page table is single level?

(2^64/2^10)\*2^2 = 2^56

* 1. Same question as part (b) but now the page table is a two-level design, so give an upper bound. The second level table holds the page metadata, not the first-level table. However, all page table entries are aligned in memory as 32-bit integers.

2\*2^27\*2^5+1\*2^27\*24

* 1. Same question as part (b) but for a three-level table.

1\*2^18\*24 + 2\*2^18\*2^5 + 3\*2^18 \* 40