Philip Lenox, Devlin Junker
November 11, 2013
CS 472 HW 3

(9.2) Why do computers use cache memory?

Computers use cache memory because in an ideal world, we want large, fast, cheap memory. This memory does not exist, so we use the cached memory heirarchy to approximate this large, fast, cheap memory.

(9.3) What is the meaning of the following terms?

(a) temoral locality : data that was recently used will likely be used again soon.

(b) spacial locality : data physically near data that was recently used will likely be used soon.

(9.4) ratio = $ratio = \frac{1}{kh-h+1}$

(9.5) (a) s = 5.263

(b) s = 6.896

(c) s = 4.166

(d) s = 12.739

(9.6) (a) h = 0.096

(b) h = 0.526

(c) h = 0.842

(d) h = 0.982

(9.8) ratio = $S = \frac{m}{ch-m(h-1)}$

(a) S = 4

(b) S = 2

(c) S = 4

(9.11) word - the word at the particular line which contains the data we are interested. line - a group of words containing at least 1 word set - a block of data the size of the cache, breaks memory into "set size block".

(9.12) direct mapped: each memory location is mapped to 1 cache location Fully associative: each memory location can be stored in each cache location Associative (set): each memory location can be stored to multiple cache locations, but not all

(9.17) Cache coherency refers to the possible discrepancies between the different memory hierarchies

(9.22) The instruction cache is easier to implament because the contents of the instruction cache are not accessed because they are not modified, so no write-back is necessary.

(9.23) writeback: if cache hit, put it in. i fcache miss, pull in from memory, write to cache

$h(t_c) + t_m(1-h) + t_1(1-h)$

(9.26) $1cycle(0.9) + 4cycles(0.08) + 50cycles(0.02)$

Average cycles = 2.22 cycles

(9.28) global miss rate: misses in the cache divided by the total accesses to memory local miss rate: misses in the cache dvided by the total access to that cache.

(9.35) 64 bit precessor: 8 byte words Cache size: 4 way sset associative 32 byte lines 4 words per line: 2 bits

8 MB / ways = 2 MB 2M -¿ 23 bits

set	line	word
23 bit	2 bit	bit

(9.41) cache CPU - prevent from hitting memory cache Disk - prevents from hitting disk

(9.42) write back -only write to the lower tiers of memory when the data is being removed from the cache

write through -write back to the lower tieres of memory every time the cache is written.

write through is overall slower, but is better for coherency because the main store is update on average more quickly.

(9.43) 2^32 = 4GB 4GB / 4KB = 1 million page table entries. 55 bits per page table entry. 20bits + 32bits + 3bits = 55bits. 55bits * 1 million = 55Mbits or 6.88MBytes.

(9.45) 1 cycle(0.07) + 2 cycles(0.15) + (0.1)[0.95(2 cycles) + 0.05(10 cycles)] + (0.05)[0.95(2 cycles) + 0.05 (10 cycles) + 5] Average cycles per instruction = 1.61

(9.46) $(100 loop iterations) * (50 cycles + 2 cycles + 2 cycles) = 5400 cycles$

(9.57) (a)

set	line	word
8 bit	10 bit	6 bit

(b)

line	word	byte
18 bit	5 bit	1 bit

(c)

line	word	byte
18 bit	5 bit	1 bit