Homework Assignment 2

For chapter 2 of Memory hierarchy

College of Computer Science, Zhejiang University

(350 points)

提醒：列出算式可以得到过程分

建议时间：不摸鱼的话 2-5h

1. [20/20/30/30] Consider the following description of a memory hierarchy.

   Virtual address wide = 45 bits, Memory physical address wide = 38 bits, Page size = 4KB.

   Cache capacity =8KB. Block size = 32Byte. It is a write-back 2-way associative cache.

   (a) How many bits are there in the fields of tag, index and block offset of the physical memory address.

   (b) Draw a graph to show a cache line (including tag, data, and some other control bits) in the cache.

   (c) Draw a graph to show if it is implemented in the way of virtually indexed and physically tagged cache. Draw both cache and TLB.

   (d) Please describe the access procedure to the memory hierarchy in (c) that when a CPU address (virtual address) is given to access the cache.

2. [20] Assume that we have two machines A and B. The only difference between A and B lies in their cache hierarchies:

   Machine A: 64 KB level-one data cache with a *8 ns* access time and a miss rate of 8%

   Machine B: 8 KB level-one data cache with a *2 ns* access time and a miss rate of 15%, and a 1 MB level-two cache with a *20 ns* access time and a miss rate of 10%.

   Assume that both machines have an I-cache miss rate of 0%, a main memory access time of *50 ns*, and all the bus transfer time could be ignored. Which machine will have a better performance in memory access（AMAT）? Why?

3. [30] Suppose you own a computer which has the following properties:

- the pipeline can accept a new instruction every cycle
- the cache can provide data every cycle (i.e. no penalty for cache hits)
- the instruction cache miss rate is 2.5%
- the data cache miss rate is 3.5%
- 30% of instructions are memory instructions
- the cache miss penalty is 80 cycles.

Now you want to purchase a new computer. you can either

- purchase a machine with a processor and cache that is twice as fast as your current one(memory speed and CPU speed are the same as the old machine, though), or
- purchase a machine with a processor and cache that is the same speed as your oldmachine but in which the cache is twice as large.

Assume the cache miss rate will drop by 40% with this larger cache (although this is generally not true in the real world). Which computer are you best off purchasing? Explain in detail, showing the relative performance of each choice.

4. [10/10/10/10] You are building a system around a processor with in-order execution that runs at 1.0 GHz and has a CPI of 1.35 excluding memory accesses. The only instructions that read or write data from memory are loads(20% of all instructions) and stores (10% of all instructions). The memory system for this computer is composed of a split L1 cache that imposes no penalty on hits. Both the I-cache and D-cache are direct mapped and hold 32 KB each. The I-cache has a 2% miss rate and 32-byte blocks, and the D-cache is write-through with a 5% miss rate and 16-byte blocks. There is a write buffer on the D-cache that eliminates stalls for 90% of all writes. The 512 KB write-back, unified L2 cache has 64-byte blocks and an access time of 12 ns. It is connected to the L1 cache by a 128-bit data bus that runs at 266 MHz and can transfer one 128-bit word per bus cycle. Of all memory references sent to the L2 cache in this system, 85% are satisfied without going to main memory. Also, 50% of all blocks replaced are dirty. The 128-bit-wide main memory has an access latency of 80 ns, after which any number of bus words may be transferred at the rate of one per cycle on the 128-bit-wide 133 MHz main memory bus.

a. [10] <B.2> What is the average memory access time for instruction accesses?
b. [10] <B.2> What is the average memory access time for data reads?
c. [10] <B.2> What is the average memory access time for data writes?
d. [10] <B.2> What is the overall CPI, including memory accesses?
提示：L1 cache 的 miss panelty 认为是替换数据从 L2 到 L1 的传输时间，忽略响应时间。

5. [12/15/15/12]

The transpose of a matrix interchanges its rows and columns; this is illustrated below:

$$\begin{bmatrix} A11 & A12 & A13 & A14 \\ A21 & A22 & A23 & A24 \\ A31 & A32 & A33 & A34 \\ A41 & A42 & A43 & A44 \end{bmatrix} \Rightarrow \begin{bmatrix} A11 & A21 & A31 & A41 \\ A12 & A22 & A32 & A42 \\ A13 & A23 & A33 & A43 \\ A14 & A24 & A34 & A44 \end{bmatrix}$$

Here is a simple C loop to show the transpose:

```
for (i = 0; i < 3; i++) {
 for (j = 0; j < 3; j++) {
 output[j][i] = input[i][j];
 }
}
```

Assume that both the input and output matrices are stored in the row major order (row major order means that the row index changes fastest). Assume that you are executing a 256 × 256 **single-precision** transpose on a processor with a 16 KB fully associative (don't worry about cache conflicts) least recently used (LRU) replacement L1 data cache with 64 byte blocks. Assume that the L1 cache misses or prefetches require 16 cycles and always hit in the L2 cache, and that the L2 cache can process a request every two processor cycles. Assume that each iteration of the inner loop above requires four cycles if the data are present in the L1 cache. Assume that the cache has a write-allocate fetch-on-write policy for write misses. Unrealistically, assume that writing back dirty cache blocks requires 0 cycles.
For the simple implementation given above, this execution order would be nonideal for the input matrix; however, applying a loop interchange optimization would create a nonideal order for the output matrix. Because loop interchange is not sufficient to improve its performance, it must be blocked instead.
a. [12] What should be the minimum size of the cache to take advantage of blocked execution?
b. [15] How do the relative number of misses in the blocked and unblocked versions compare in the minimum sized cache above?
c. [15] Write code to perform a transpose with a block size parameter B which uses B × B blocks.
d. [12] What is the minimum associativity required of the L1 cache for consistent performance independent of both arrays' position in memory?
提示：a. 此时矩阵的一行可以充分利用一个 cache line

6. [12/15] Consider the usage of critical word first and early restart on L2 cache misses. Assume a 1 MB L2 cache with 64 byte blocks and a refill path that is 16 bytes wide. Assume that the L2 can be written with 16 bytes every 4 processor cycles, the time to receive the first 16 byte block from the memory controller is 130 cycles, each additional 16 byte block from main memory requires 16 cycles, and data can be bypassed directly into the read port of the L2 cache. Ignore any cycles to transfer the miss request to the L2 cache and the requested data to the L1 cache.
a. [12] <2.2> How many cycles would it take to service an L2 cache miss with and without critical word first and early restart?
b. [15] <2.2> Do you think critical word first and early restart would be more important for L1 caches or L2 caches, and what factors would contribute to their relative importance?

7. [12/12/15] You are designing a write buffer between a write-through L1 cache and a write-back L2 cache. The L2 cache write data bus is 16 B wide and can perform a write to an independent cache address every 4 processor cycles.
a. [12] How many bytes wide should each write buffer entry be?
b. [15] What speedup could be expected in the steady state by using a merging write buffer instead of a nonmerging buffer when zeroing memory by the execution of 32-bit stores if all other instructions could be issued in parallel with the stores and the blocks are present in the L2 cache?
c. [15] What would the effect of possible L1 misses be on the number of required write buffer entries for systems with blocking and nonblocking caches?

7. [10/10/10/10] A cache acts as a filter. For example, for every 1000 instructions of a program, an average of 20 memory accesses may exhibit low enough locality that they cannot be serviced by a 2 MB cache. The 2 MB cache is said to have an MPKI (misses per thousand instructions) of 20, and this will be largely true regardless of the smaller caches that precede the 2 MB cache. Assume the following cache/latency/MPKI values: 32 KB/1/100, 128 KB/2/80, 512 KB/4/50, 2 MB/8/40, 8 MB/16/10. Assume that accessing the off-chip memory system requires 200 cycles on average. For the following cache configurations, calculate the average time spent accessing the cache hierarchy.
a. 32 KB L1; 8 MB L2; off-chip memory
b. 32 KB L1; 512 KB L2; 8 MB L3; off-chip memory
c. 32 KB L1; 128 KB L2; 2 MB L3; 8 MB L4; off-chip memory
d. What do you observe about the downsides of a cache hierarchy that is too shallow or too deep?