CSE 471 Winter 2000

Computer Design and Organization
Problem Set #4

Due: Wednesday March 8

This assignment is to make you think about some design decisions for cache hierarchies.

Consider the following characteristics of a cache hierarchy. We first consider a single level cache for which the following measurements have been gathered.

1. The I-cache miss rate is negligible
2. 30% of the instructions are memory operations (read/write)
3. There is a single level D-cache and:
   - The hit rate is 95% and is the same for reads and writes
   - 25% of the memory operations are writes
   - A read hit takes one cycle; a write hit takes two cycles
   - The block size is 4 words
4. On a D-cache miss, it takes:
   - 4 cycles to send the address of the missing block
   - 24 cycles for the access time in DRAM for 4 words (the DRAM is 4-way interleaved)
   - 4 cycles to transmit data from the DRAM to the cache. The bus is 64-bit (2 words) wide.

Let the "Memory Cycles Per Instruction" be a metric defined as:

$$MCPI = \frac{Total\ Memory\ Access\ Penalty}{Total\ Number\ of\ Instructions\ Executed}$$

Assuming there is no penalty for a write miss (we'll take care of these later) *what is the MCPI* of the above system? You can assume that all non-memory operations take 1 cycle, i.e., there is no memory penalty for read hits.

If one wanted to improve (decrease) MCPI due to reads, what would be best:

1. Introduce a victim cache with a hit rate of 20%. A read takes two cycles when it misses the D-cache (L1) and hits in the victim cache. The miss penalty when there are misses in the D-cache and the victim cache is the same as before.

2. Introduce a second level cache L2 with a (local) hit rate of 50%. A miss in the L1 cache that hits in L2 takes 6 cycles to be resolved. It takes 3 cycles to detect a miss in both L1 and L2. After that, the penalty to resolve the miss is the same as if there were only L1 (i.e., the data is sent to both L1 and L2).

We return now to the single level D-cache. Assume first a **write-back write-allocate** policy and that at any given time 30% of the blocks in the cache are dirty. Assume also that there is a write buffer such that the write-backs of replaced dirty blocks do not prevent read/write misses to be processed immediately.

If load/store instructions are

- uniformly distributed among the sequence of instructions
- read/write misses are uniformly distributed among the memory operations

is the assumption that write-backs of replaced dirty blocks do not prevent misses to be processed immediately a correct one? Justify your answer quantitatively. For example, you can compute the average occupancy of the bus due to read/write misses and how much more occupancy is due to write-backs.

Do you think that the "uniform distribution assumption" is a good one in reality? Explain why or why not.

Assume now a **write-through write-around** policy.

Assume first an "infinite" write buffer, i.e., write misses take no time. Under the (questionable) assumption that the miss ratios are the same in the (write-back write-allocate) and (write-through write-around) policies, *which of the two policies would give a lower MCPI* (now writes are included). Justify your answer qualitatively.

Now assume no write buffer, i.e., a write miss must proceed directly to memory. *Does bus occupancy become a bottleneck?* Justify your answer quantitatively. As before, you can assume the "uniform distribution assumption".

Do you think your analysis is a step towards justifying todays design of cache hierarchies that have write-through lock-up free L1 caches and write-back L2 caches?