CS252 HOMEWORK 4

Due Tues. 11/20/07

Problem 1:

Data prefetching is a technique that allows the "consumer" of data to request the data to its cache *before* it needs them. With multiprocessors, we can think of the "dual" of this technique where the "producer" of the data "evicts" the data from its cache *after* it is done with them. An extension of such "postflush" could be to send the data to the next processor that needs the data, in cases where that can be determined. Such technique is called producer-initiated communication.

a) When is data prefetching likely to be beneficial?

b) Assume a shared-memory multiprocessor system that takes 300 cycles for a cache-to-cache transfer.

A program running on this machine spends 60% of its time stalled on memory accesses (i.e. where cache-to-cache transfer is needed) and 20% of its time stalled on synchronization. Of those memory stalls, 20% are due to producer-consumer data access patterns where the writer of data can identify the processor that will read the value next. In these cases, producer-initiated communication can directly transfer data to the cache of the next processor needing the data. This will reduce the latency of these memory accesses from 300 cycles for a cache-to-cache transfer to 1 cycle for a cache hit. Another 30% of the memory stalls are due to migratory data patterns where data move from one processor to another, but the migration path is unclear to the source processor. In this case, a producer-initiated communication can reduce the latency of the memory access from 300 cycles to 100 cycles.

Further assume that all the synchronization is due to tree barriers and that the tree barrier overhead can be reduced by 40% with producer-initiated communication. Assuming no other overheads, what is the reduction in execution time with producer-initiated communication?

Problem 2:

As caches increase in size, blocks often increase in size as well.

a) If a large instruction cache has larger data blocks, is there still a need for instruction prefetching? Explain the interaction between instruction prefetching and increased block size in instruction caches.

b) Is there a need for data prefetch instructions when data blocks get larger? Explain.

Problem 3:

You are building a system around a processor with in-order execution that runs at 1.067 GHz and has a CPI of 0.7 without cache misses (no memory stalls whatsoever). The only instructions that read or write data from memory are loads (20% of all instructions) and stores (5% of all instructions).

The memory system for this computer is composed of a split L1 cache that has a 1 cycle access time on hits -- i.e., there is no penalty for a cache hit. 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 all processor stalls for writes.

The 512 KB write-back, unified L2 cache has 64-byte blocks and an access time of 15 ns (not counting time to transfer any data). It is connected to the L1 cache by a 128-bit data bus that runs at 267 MHz and can transfer one 128-bit word per bus cycle. Of all memory references sent to the L2 cache in this system, 80% are satisfied without going to main memory. Also, 50% of all blocks replaced are dirty. The 128-bit-wide main memory has an access time of 60 ns (not counting time to transfer any data), after which bus words may be transferred at the rate of one per cycle on the 128-bit-wide 133 MHz main memory bus. The main memory requires separate access sessions for read and write operations (i.e. a read following a write starts with 60 ns latency after the write completes). There is no early restart or critical word first used.

Assume that the bus is able to support the required traffic (no bus contention). Also assume that cache miss penalties are cumulative, i.e. there is no overlap between the miss penalty for the L1 cache and the L2 cache when a memory access must go to memory.

a) Calculate the average miss penalty (in ns) of
    i) L2 cache miss penalty
    ii) L1 I-cache read miss
    iii) L1 D-cache read miss

b) What is the average memory access time for instruction accesses?

c) What is the average memory access time for data reads?

d) What is the average memory access time for data writes?

e) What is the overall CPI, including memory accesses?

f) You are considering replacing the 1.067 GHz CPU with one that runs at 2.133 GHz, but is otherwise identical. How much faster does the system run with a faster processor? Assume the L1 cache still has no hit penalty, and that the speed of the L2 cache, main memory, and buses remains the same in absolute terms (e.g., the L2 cache still has a 15 ns access time and a 267 MHz bus connecting it to the CPU and L1 cache).