Introduction to Computer Architecture

Assignment 3

Due May 13, 2014

1. [50 = 10 + 15 + 15 + 10] You are building a system around a processor with in-order execution that runs at 1.1GHz and has a CPI of 0.7 excluding memory accesses. 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 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 unified L2 cache has 64-byte blocks and an access time of 15 ns. It is connected to the L1 cache by a 128-bit data bus that runs at 266 MHz and can transfter one 128-bit word per bus cycle. Of all memory references sent to the L2 cache in blocks replaced are dirty. The 128-bit-wide main memory has an access latency one per cycle on the 128-bit-wide 133 MHz main memory bus.

a. What is the average memory access time for instruction accesses?
b. What is the average memory access time for data reads?
c. What is the average memory access time for data writes?
d. What is the overall CPI, including memory accesses?

2. [50 = 10 + 10 + 15 +15] Virtual machines can lose performance from a number of events, such as the execution of privileged instructions, TLB misses, traps, and I/O. These events are usually handled in system code. Thus, one way of estimating the slowdown when running under a VM is the percentage of application execution time in a system versus user mode. For example, an application spending 10% of its execution in a system mode might slow down by 60% when running on a VM. Figure 2.32 lists the early performance of various system calls under native execution, pure virtualization, and paravirtualization for LMbench using Xen on an Itanium system with times measured in microseconds (courtesy of Matthew Chapman of the University of New South Wales).

a. What types of programs would be expected to have smaller slowdowns when running under VMs?
b. If slowdowns were linear as a function of system time, given the slowdown above, how much slower would a program spending 20% of its execution in system time be expected to run?
c. What is the median slowdown of the system calls in the table above under pure virtualization and paravirtualization/

d. Which functions in the table above have the largest slowdowns? What do you think the cause of this could be?

Benchmark	Native	Pure	Para
Null call	0.04	0.96	0.50
Null I/O	0.27	6.32	2.91
Stat	1.10	10.69	4.14
Open/close	1.99	20.43	7.71
Install sighandler	0.33	7.34	2.89
Handle signal	1.69	19.26	2.36
Fork	56.00	513.00	164.00
Exec	316.00	2084.00	578.00
Fork + exec sh	1451.00	7790.00	2360.00