

Enterprise-Scale Software Development (COMP5348)

Semester 1, 2009

Tutorial Week 5 Example Solution

Part A: Paper-based exercises

Q1. You have a 1GB file on an external disk. If the bandwidth obtainable when reading sequentially from the disk is 25MB/s, how long will it take to read the file (suppose that there is initial set-up time of 5ms, for disk seek etc)? Suppose instead that the file is read in a sequence of separate operations in random order, where each operation reads a 1KB block and requires its own set-up time. How long will it take to complete reading every block of the file?

*The time for just the data transfer is $1\text{ GB} / 25\text{MB/s} = 41\text{ s}$; in the sequential case we also have one lot of set-up, so the overall time taken is still 41 s (to 2 significant places). In the random read case we do 1024 set-ups, so the total time is $41 + 1024 * 1024 * 0.005 = 5284\text{ s}$ (ie 5300 s to 2 sig figs, or about 1.5 hr). Another way to calculate this is that each block takes $1\text{KB} / 25\text{MB/s} = 30\mu\text{s}$ for data transfer, plus 5ms for set up, so the time per block is 5.03 ms; doing the whole file take $1024 * 1024$ blocks, or 5274s, or 5300s to sensible precision (the difference between the calculations is due to approximation, and isn't visible with answers given at sensible precision).*

Q2 Suppose requests arrive every 100ms, and that 90% of requests are of type A, and 10% of requests are of type B. Each type A request requires processing that executes 2M instructions (say that on average these instructions use 2 cycles each), and it also requires 10ms of disk activity; type B requests need no I/O but they need 100M instructions (many of these are expensive multiplications, and the average is 4 cycles per instruction). If the CPU speed is 2 GHz, what load is being placed on the CPU and the disk by these requests? If the workload arrival rate becomes faster, estimate when and why saturation might occur.

*Each type A request needs 4M cycles of CPU, or approx 2ms; it also needs 10ms of disk activity. Each type B request needs 400M cycles, or 200ms of CPU. The average request thus needs $(90 * 2 + 10 * 200) / 100 = 21.8\text{ms}$ of CPU, and it needs $(90 * 10 + 10 * 0) / 100 = 9\text{ms}$ of disk activity. This can easily be done within the 100ms inter-arrival time. If however the inter-arrival time drops to a bit less than 21.8ms, then the CPU will be saturated.*

Q3. Suppose we can buy three different system configurations

Configuration X: 2 GHz CPU, 2GB of slow main memory, 128kB of cache; cost is \$1250

Configuration Y: 2 GHz CPU, 2GB of slow memory, 3MB of cache; cost is \$1500

Configuration Z: 2 GHz CPU, 2GB of slow memory, 64MB of cache; cost is \$6000

Suppose that excluding times for load/store, each instruction takes 2 cycles. The access time for cache is 10ns, and for slow main memory is 70ns.

Suppose that our workload does on average 1 load or store per 10 instructions, and it has locality properties so that the hit rate is 50% for the small cache (configuration X), 95% for configuration Y, and 100% for configuration Z. What is the average performance (in instructions/sec) for each configuration? What is the performance per dollar for each configuration?

*The time to execute 1 instruction (not counting any memory I/O) is 2 cycles, or 1ns. Each instruction requires 0.1 load/store, on average. A load/store takes 10ns if there is a cache hit, and 10+70=80ns if there is a miss followed by read from main memory. In config X, this means the average load/store takes $(50*10+50*80)/100=45ns$. In config Y, it takes $(90*10+10*80)/100=17ns$. In config Z it takes 10ns. Thus the total time for 1 instruction (including load/store) is $1+0.1*45=5.5ns$ in configuration X; it is $1+0.1*17=2.7ns$ in config Y, and it is $1+0.1*10=2ns$ in config Z.*

Thus in config X, the performance in instructions/sec = $1/5.5ns=181$ million instructions/sec. The performance per dollar is $(181M\text{instr}/\text{sec})/\$1250=140000$ inst/sec/\$.

In config Y, the performance is $1/2.7ns = 370M$ instr/sec; the performance per dollar is 250000 instr/sec/\$.

In config Z, performance is $1/2ns = 500M$ instr/sec, performance per dollar is 83000 instr/sec/\$.

Thus config Z gives best performance, config X gives lowest (best) price, and config Y gives best performance/dollar.

Part B: Run perfmon on the lab machine, and explore the behaviour when opening a large file from disk (drive C) compared to opening it from across the network (from drive U). See <http://adminfoo.net/2007/04/windows-perfmon-top-ten-counters.html>