Sean Penney and Paul Atkinson
May 26, 2014
CS 472 HW 6

(6.1) There are many criteria for judging computer performance. These include efficiency, throughput, latency, relative performance, and time and rate. Performance is difficult to define because of the different criteria, and also because different people are concerned with different performance criteria.

(6.2) It depends. If you upgrade a 7,200 RPM HDD to a SSD, it will be a massive performance increase, as the hard drive is usually the bottleneck for a lot of PC operations. However, once a program is actually loaded, hard drive does not help performance very much, so something like a large cache memory would make the programs run faster.

(6.4) Efficiency = bits of data words / total bits sent = 16384 / 16608 = 98.65

(6.6) Because the computer architecture is the main thing that determines performance. You can buy 10 year old chips that can run at 4 GHz, but it will be much much worse then almost any CPU you can buy nowadays. However, clock speed is useful when comparing two CPUs with the same architecture, as they are directly comparable.

(6.7) a. Cycle 1, 2, and 3 would all take 50 ns so 150 ns total.

b. Cycle 1, 2, and 3 would all take 40 ns so 120 ns total.

c. Cycles 1 and 2 would take 30 ns. Cycle 3 would still take 32 ns, so 92 ns total.

d. Cycle 1 would take 20 ns, but cycle 2 would take 25 ns and cycle 3 would take 32 ns. 77 ns total.

e. Cycle 1 would take 10 ns, but cycle 2 and 3 would take 25 and 32 ns. 67 ns total.

(6.9) While it isn't a direct indicator of how powerful a processor is, the cache size does impact how powerful a processor is and can be used as a metric to compare processors.

(6.11) Some operations are not scalable, and it is difficult to alter the clock rate dynamically from clock pulse to clock pulse. Due to this, computers may become unstable if you try to overclock them.

(6.12) No, because it could overload the processor.

(6.13) Arithmetic/logical instructions, CPI = .65

Register load operations, CPI = .50

Register store operations, CPI = .10

Conditional branch instructions, CPI = 1.6

Total CPI = 2.85

Need to increase to 3.42, or by .57

CPI for conditional branch instructions needs to be 2.17

2.17 / .2 = 11 cycles

(6.14) (.7*1 + .2*2 + .1*3)*10 ns = 14 ns

(6.16) Assuming the CPU has a large cache, I would go with option b. If the cache is large enough, data from the hard drive would be loaded into the cache, then a fast CPU would be able to access the data fast.

(6.17) a. 5.26

b. 9.17

c. 1.92

d. 50.25

(6.18) Speedup ratio = p / (p*fs + 1 - fs), where fs = .10

It would not be worth adding cores until the speedup ratio increases by less than 5

At 32 processors, this point will be reached. The speedup ratio for 31 processors is 7.75 and 7.80 for 32 processors.

(6.22) Assuming all non string operations are not affected: 20 percent of operations have to be string ops.

(6.25) No, the formula is not applied correctly. p refers to the number of processors, and f_s refers to the fraction of the program that can run only on one processor. The part of the program spent not accessing the hard drive can not necessarily be run on multiple processors, which the problem assumes. The same incorrect assumption is made for the floating point operations.

(6.26) The compiler is probably going to optimize parts of this code out (mainly p, which does not do anything), so if they want to force the operation the variables have to be volatile. The second problem with this code is that testing the CPUs ability to both get memory and do calculations at the same time does not give you a very good idea about what the performance of the CPU is going to be. These tests should be performed seperately.

(6.31) $(\sqrt{x} - \sqrt{y})^2 \geq 0$

$x - 2\sqrt{xy} + y \geq 0$

$x + y \geq 2\sqrt{xy}$

$\frac{x+y}{2} \geq \sqrt{xy}$