Drake Bridgewater & Ryan Phillips
May 26, 2014
CS 472 HW 6

(6.1)

What is computer performance and why is it so difficult to define?

> Computer performance is difficult to define and is dependent on the slowest component. As our text book explains "computer's performance are used by the designer to improve the computers architecture and organization by locating bottlenecks and eliminating them or minimizing their effects." [1]

(6.2)

A system consists of a CPU, cache memory, main store, and hard disk drive. Where are time and effort best spent improving the system's performance? What factors affect your answer?

> The answer really depends on what kind of performance you are wanting to improve. An example of this is the latest push for SSDs as a better technology for harddrives. This does relatively little in increasing the performance of a lot of classic computing scenarios that are CPU or GPU bound. But for the average consumer's workflow in which many different programs are loaded in and out of memory, the time the user spends waiting to start working is important. So then a new goal could be to try to eliminate any noticeable wait times for basic tasks. And the truth of the matter is that each of the components in a system play a part in accomplishing this, but whatever the current bottleneck is, that is where the focus should be. And this is a constantly shifting target.

(6.4)

A data transmission system transmits data in the the form of a master frame containing 16 sub-frames. Each sub-frame includes a 1024-bit data word and a 12-bit error-correcting code. The master frame itself contains a 32-bit error correcting code. What is the efficiency of this system?

(6.6)

Why is clock rate a poor metric of computer performance? What are the relative strengths and weaknesses of clock speed as a performance metric?

Because it's too simple. Clock rate does affect computer performance, but it's merely one of many factors.

Strengths: All other factors being equal (i.e. within the same family and architecture) clockspeed can help you determine the relative performance. And in the purely theoretical/abstract case, there is a direct connection between clockspeed and computing power.

Weaknesses: In real-world scenarios, increasing the speed of the clock means that many operations that used to be able to fit in one clock cycle might now take more than one cycle, negating the performance improvements. This is because "some actions such as accessing memory require a minimum amount of time". [1]

(6.7)

The timing diagram in Figure P6.7 illustrates a system in which operations occur as three consecutive clock cycles. Actions taking place in clock cycle 1 are scalabe; that is, if the clock cycle time chances, the actions can be speeded up or slowed down corresondingly. In cycle 2, the action process 1 requires 25 ns and in clock cylce 3 the action proces 2 requres 32 ns. If the clock cycle is less than the time required for proeess 1 or preoccess 2, then one ore more wait ctycle havew to be inserted for the process to complete.

What is the time to complete an operation if the clock cycle time is

a. 50 ns b. 40 ns c. 30 ns d. 20 ns e. 10 ns

(6.9)

Can you think of a better metric than MIPS or clock speeds that gives a good impression of the power of a processor (without having to use benchmarks)?

The problem with MIPS is that you don't know "what exactly was achieved by an instruction being executed"; [1] the complexity of instructions varies.

So then we could move up one level of abstraction, and try to come up with a standard which takes these differences into account. You would have an increased

number of basic tasks which will be performed by some number of MIPS. For example, table 6.2 in the book compares the instructions needed to perform an expression on two hypothetical computers. Well, the number of times this expression could be executed would become the new metric. This does start to come close to what many benchmarks do, but the 'operations' are much simpler. The problem then is deciding how to weight these operations, and which operations to include in your test set.

(6.11)

Overlclocking a computer means operating ait at a higher clock rate than that specified by its manufacturer; for example, a 2 GHz chip might be clocked at 2.1 GHz to squeeze more performance out of it.

Does overclocking disprove the famous aphorism "Thereś no such thing as a free lunch," or is there a hidden cost? If so, what is the cost of overclocking?

One of the main downsides of overclocking is that is may cause some systems to become more unstable. Additionally, a system is designed to dissipate a certain maximum amount of heat. Overclocking effectively increases the amount of heat produced, meaning that the system is more likely to overheat (if the cooling system isn't improved).

(6.12)

The following figures define the typical operating parameters of a processor.

Operation, Frequency, Cycles Arithmatic/logical instructions, 45%, 1 Register load operations, 20%, 3 Register store operations, 10%, 2 All branch instructions, 25%, 2

If the clock rate could be reduced by 15%, it would require only 2 cycles to perform a register load. Would that be a good idea?

(6.13)

A computer has the following parameters:

Operation, Frequency, Cycles Arithmatic/logical instructions, 45%, 1 Register load operations, 20%, 5 Register store operations, 10%, 2 All branch instructions, 25%, 8

If the average performance of the computer (in terms of its CPI) is to be increased by 20% while executing the same instruction mix, what target must be achieved for the cycles per conditional branch instruction?

(6.14)

A program is run on a computer with the following parameters.

Clock cycle time, 10 ns Instructions with 1 cycle, 70% Instructions with 2 cycles, 20% Instructions with 3 cycles, 10%

What is the MIPS rating of this computer?

Using 100 for n:

$100/(140 * 10^{-}9 * 10^{6}) \ = 715 MIPS$

(6.16)

In a particular system, a CPU is used for 78% of the time and a disk drive for 22% of the time. A designer has two options:

a. improve the disc performance by 40% and the CPU performance by 20%

b. improve the disc performance by 10% and the CPU performance by 80%

Which is the better option, and why?

Option b. Since the CPU is more heavily used that the disk drive, improvements to the CPU carry more weight. In this particular example option b increases the overall performance by over twice that of option a. This is about 65% improvement vs. 30% improvement.

(6.17)

For the following systems that have both serial and parallel activies, calculate the speedup ratio.

a 10 processors, $f_s = 0.1$

b 100 processors, $f_s = 0.1$

c 5 processors, $f_s = 0.4$

d 100 processors, $f_s = 0.01$

(6.18)

A system has a single core processor that costs \$150. Suppose that adding more cores to the chip costs \$10 per additional processor. (Note: For this system, the value of f_s is 0.10).

If it is considered worthwhile adding cores until te incremental speedup ratio increases by less than 5% over the orignal (unmodified) performance, what is the optimum number of processors? What percentag increase in cost is required to achieve this performance?

(6.22)

A coprocessor is added to a computer to speed up the execution time of string-processing instructions by a factor of 3.5. What fraction of the execution time must use these string-processing instructions in order to achieve an average speedup of 1.5?

(6.25)

A computer spends 25% of its time accessing a hard disk. It spends 20% of the time doing floating point. The hard disk is replaced by two disks operating in parallel and the floating point unit is replaced by one four times faster. The speed up is given by

equation here

So, the speedup for the disk is S_{disk} - 2/2(2 * 0.75 + 1 - 0.75) = 1/1.75 = 1.429

The speedup for the floating-point unit is $S_{floating-point}$ = 4/(f * 0.80 + 1 - 0.80) = 4/1 = 4/3.4 = 1.176. The total speedup ratio is the product o the individual speedups which is 1.429 * 1.176 = 1.681. Is this answer correct?

(6.26)

Someone decided to use the following C code as part of a benchmark to determine the performance of a computer including its memory. It has two potential faults. What are they?

```
for (i = 0; i < 100; i++){
    p = q * s + 12345
    x = 0.0;
    for (j = 0; j < 60000; j++){
    x = x + A[j] * B[j];
    }
}
```

(6.31)

For two benchmarks, x and y, show that their arithmatic mean is always higher than, or the sams as, the geometric mean.

References

[1] Alan Clements. *Computer Organization and Architecture*. Global Engineering: Christopher M. Shortt, themes and variations edition, 2014.