CSE2312-002/003, Fall 2014, Homework 1

Due: September 4, 2014

The following problems are from Patterson and Hennessy, Chapter 1.

1.3. Describe the steps that transform a program written in a high-level language such as C into a representation that is directly executed by a computer processor (e.g., machine language).

1.4. Assume a color display using 8 bits for each of the primary colors (red, green, blue) per pixel and a frame size of 1280 x 1024.

a. What is the minimum size in bytes of the frame buffer to store a frame?
b. How long would it take, at a minimum, for the frame to be sent over a 100 Mbit/s network?

1.5. Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a. Which processor has the highest performance expressed in instructions per second?
b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.
c. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

1.7 (a and b only). Compilers can have a profound impact on the performance of an application. Assume that for a program, compiler A results in a dynamic instruction count of 1.0E9 and has an execution time of 1.1 s, while compiler B results in a dynamic instruction count of 1.2E9 and an execution time of 1.5 s.

a. Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.
b. Assume the compiled programs run on two different processors. If the execution times on the two processors are the same, how much faster is the clock of the processor running compiler A's code versus the clock of the processor running compiler B's code?

1.9 (1.9.1 and 1.9.2 only). Assume for arithmetic, load/store, and a branch instruction, a processor has CPIs of 1, 12, and 5, respectively. Also assume that on a single processor a program requires the execution of 2.56E9 arithmetic instructions, 1.28E9 load/store instructions, and 256 million branch instructions. Assume that each processor has a 2 GHz clock frequency.

Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per processor is divided by 0.7 x p (where p is the number of processors) but the number of branch instructions per processor remains the same.

1.9.1. Find the total execution time for this program on 1, 2, 4, and 8 processors, and show the relative speedup of the 2, 4, and 8 processor result relative to the single processor result.

1.9.2. If the CPI of the arithmetic instructions was doubled, what would the impact be on the execution time of the program on 1, 2, 4, or 8 processors?

1.12 (all). The utilization of a subset of the performance equation as a performance metric is a pitfall. To illustrate this, assume the following two processors. P1 has a clock rate of 4 GHz, average CPI of 0.9, and requires the execution of 5.0E9 instructions. P2 has a clock rate of 3 GHz, an average CPI of 0.75, and requires the execution of 1.0E9 instructions.

1.12.1. One usual fallacy is to consider the computer with the largest clock rate as having the largest performance. Check if this is true for P1 and P2.

1.12.2. Another fallacy is to consider that the processor executing the largest number of instructions will need a larger CPU time. Considering that processor P1 is executing a sequence of 1.0E9 instructions and that the CPI of processors P1 and P2 do not change, determine the number of instructions that P2 can execute in the same time that P1 needs to execute 1.0E9 instructions.

1.12.3. A common fallacy is to use MIPS (millions of instructions per second) to compare the performance of two different processors, and consider that the processor with the largest MIPS has the largest performance. Check if this is true for P1 and P2.

1.12.4. Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as: *MFLOPS = No. FP operations / (execution time X 1E6)*, but this figure has the same problems as MIPS. Assume that 40% of the instructions executed on both P1 and P2 are floating-point instructions. Find the MFLOPS figures for the programs.

1.15. When a program is adapted to run on multiple processors in a multiprocessor system, the execution time on each processor is comprised of computing time and the overhead time required for locked critical section and/or to send data from one processor to another.

Assume a program requires t= 100 s of execution time on one processor. When run *p* processors, each processor requires t/p s, as well as additional 4 s of overhead, irrespective of the number of processors. Compute the pre-processor execution time for 2, 4, 8, 16, 32, 64, and 128 processors. For each case, list the corresponding speedup relative to a single processor and the ratio between actual speedup versus ideal speedup (speedup if there was no overhead).

Bonus.

In 1959, an IBM 7090 had a memory of 128KB, could execute 500,000 instructions/sec, and could be bought for three million dollars. In 2012, one particular desktop PC model has 16GB of memory, has an Intel Core i7 3630QM processor that can execute 113 billion instructions/sec, and could be bought for $2,000.

Using these two computers as references, formulate your own versions of Moore's law for instructions per second, memory, and price. In particular, specify:

- How often should the rate of instructions per second double, to be consistent with the observed improvement in performance from 1959 to 2012?
- How often should memory capacity double, to be consistent with the observed improvement in performance from 1959 to 2012?
- How often should computer prices get halved, to be consistent with the observed drop in price from 1959 to 2012?