1 Homework 1

1.1 Textbook Problem 4.15.5

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.

Processor	Clock Rate	CPI	Performance
P1	3 GHz	1.5	2 GHz
P2	2.5 GHz	1.0	2.5 GHz
P3	4.0 GHz	2.2	1.8 GHz

(a) Which processor has the highest performance expressed in instructions per second?
Answer:
Instructions per Second = Clock Rate/CPI

$$\begin{gathered}\text{P1} = 3\text{ GHz}/1.5 = 2\text{ GHz}\\ \text{P2} = 2.5\text{ GHz}/1.0 = 2.5\text{ GHz}\\ \text{P3} = 4.0\text{ GHz}/2.2 = 1.8\text{ GHz}\end{gathered}$$

P2 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.
Answer:
Number of Cycles = Clock Rate * Execution Time

$$\begin{gathered}\text{P1} = 3\text{ GHz} * 10\text{s} = 30\text{ Billion Cycles}\\ \text{P2} = 2.5\text{ GHz} * 10\text{s} = 25\text{ Billion Cycles}\\ \text{P3} = 4.0\text{ GHz} * 10\text{s} = 40\text{ Billion Cycles}\end{gathered}$$

Number of Instructions = Instructions per Second * Execution Time

$$\begin{gathered}\text{P1} = 2\text{ Hz} * 10\text{s} = 20\text{ Billion Instructions}\\ \text{P2} = 2.5\text{ GHz} * 10\text{s} = 25\text{ Billion Instructions}\\ \text{P3} = 1.8\text{ GHz} * 10\text{s} = 18\text{ Billion Instructions}\end{gathered}$$

(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?
Answer:
New CPI = 1.2 * Old CPI
New Execution Time = 0.7 * Old Execution Time
Clock Rate = (CPI * Instructions per Second) / Execution Time

$$\begin{gathered}\text{P1} = (20 * 1.8)\ /\ 7 \approx 5.14\text{ GHz}\\ \text{P2} = (25 * 1.2)\ /\ 7 \approx 4.29\text{ GHz}\\ \text{P3} = (18 * 2.64)\ /\ 7 \approx 6.79\text{ GHz}\end{gathered}$$

1.2 Textbook Problem 4.15.8

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.
Answer:
CPI = CPU Time / (Instruction Count * Clock Cycle Time)

$$\text{compiler A} = \frac{1.1s}{1.0E^9 * 1.0E^{-9}} = 1.1 \text{ CPI}$$

$$\text{compiler B} = \frac{1.5s}{1.2E^9 * 1.0E^{-9}} = 1.25 \text{ CPI}$$

(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?
Answer:
Clock Rate = (Instructions * CPI) / Execution Time

$$\text{Clock A} = \frac{1.0E^9 * 1.1}{1.1s}$$

$$\text{Clock B} = \frac{1.2E^9 * 1.25}{1.1s}$$

$$\text{Clock B / Clock A} \approx 1.37 \text{ Faster than Clock B}$$

(c) A new compiler is developed that uses only 6.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?
Answer:
Speedup = Old Execution Time / New Execution Time

$$\text{Speedup A} = \frac{1.0E^9 * 1.1}{6.0E^8 * 1.1} = 1.67$$

$$\text{Speedup B} = \frac{1.2E^9 * 1.25}{6.0E^8 * 1.1} = 2.27$$

1.3 Textbook Problem 4.15.10

Assume for arithmetic, load/store, and branch instructions, 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.

(a) 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.
Answer:
Processor 1 time = (2.56E9 * 1 + 1.28E9 * 12 + 2.56E8 * 5) / 2E9 = 9.6 seconds
Relative speedup = 9.6 / 9.6 = 1

Processor 2 time = ((2.56E9/1.4 * 1) + (1.28E9/1.4 * 12) + 2.56E8 * 5) / 2E9 = 7.04 seconds
Relative speedup = 9.6 / 7.04 = 1.36

Processor 4 time = ((2.56E9/2.8 * 1) + (1.28E9/2.8 * 12) + 2.56E8 * 5) / 2E9 = 3.86 seconds
Relative speedup = 9.6 / 3.86 = 2.49

Processor 8 time = ((2.56E9/5.6 * 1) + (1.28E9/5.6 * 12) + 2.56E8 * 5) / 2E9 = 2.25 seconds
Relative speedup = 9.6 / 2.25 = 4.27

(b) 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?
Answer:
Processor 1 time = (2560 * 2 + 1280 * 12 + 256 * 5) / 2E9 = 10.88 ms
Processor 2 time = ((2560/1.4 * 2) + (1280/1.4 * 12) + 256 * 5) / 2E9 = 7.95 ms
Processor 4 time = ((2560/2.8 * 2) + (1280/2.8 * 12) + 256 * 5) / 2E9 = 4.3 ms
Processor 8 time = ((2560/5.6 * 2) + (1280/5.6 * 12) + 256 * 5) / 2E9 = 2.47 ms

(c) To what should the CPI of load/store instructions be reduced in order for a single processor to match the performance of four processors using the original CPI values?
Answer:
Reduced CPI = x / Original CPI
Execution Time = (2.56E9 * 1 + 1.28E9 * x + 2.56E8 * 5) / 2E9 = 3.86 seconds
Clock Cycle = 2.56E9 * 1 + 1.28E9 * x + 2.56E8 * 5 = 3.84E9 + 1.28E9 * x

$$3.86 = (3.84 + 1.28 * x) / 2.0E9$$
$$1.94 = 0.64x$$
$$x = 3.03$$
$$\text{Reduced CPI} = 3.03/12 = 0.2525$$

It should be reduced by 25.25% for a single processor to match the performance of 4

1.4 Textbook Problem 4.15.13

COD Section 1.11 (Fallacies and pitfalls) cites as a pitfall the utilization of a subset of the performance equation as a performance metric. To illustrate this, consider 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.

(a) One usual fallacy is to consider the computer with the largest clock rate as having the highest performance. Check if this is true for P1 and P2.
Answer:
Execution Time = (Instruction Count * CPI) / ClockRate
Performance = 1 / Execution Time

P1 = (5.0E9 * 0.9) / 4 GHz = 1.125 seconds
Performance = 1 / 1.125 = 0.89
P2 = (1.0E9 * 0.75) / 3 GHz = 0.25 seconds
Performance = 1 / 0.25 = 4

We can see that P2 still has the higher performace even though P1 has a higher clock rate.

(b) 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.
Answer:
P1 Execution Time = 1.0E9 * 0.9 / 4 GHz = 0.225 seconds
Instruction Count = (Execution Time * Clock Rate) / CPI
P2 Instruction Count = (0.225 * 3 GHz) / 0.75 = 9.0E8
P2 executes 9.0E8 instructions in the same time that P1 executes 1.0E9 instructions.

(c) 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.
Answer:
MIPS = Clockrate / (CPI) * E6

MIPS P1 = 4E9 GHz / (0.9) * E6 = 4.44E3 MIPS
MIPS P2 = 3E9 GHz / (0.75) * E6 = 4.0E3 MIPS

P1 has the higher MIPS but P2 has the higher performance. Performance doesn't rely on MIPS.

(d) Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as:
MFLOPS = No. FP operations / (execution time $\times$ 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.
Answer:
FP Operations = Instruction Count * Clock Rate
Time = FP Operations / Clock Rate

FP Operations P1 = 5.0E9 * 0.4E9 = 1.8E9
FP Operations P2 = 1.0E9 * 0.4 * 0.75 = 0.3E9

Time P1 = 1.8E9 / 4E9 = 0.45 seconds
Time P2 = 0.3E9 / 3E9 = 0.1 seconds

MFLOPS P1 = 1.8E9 / (0.45 * 1E6) = 4000 MFLOPS
MFLOPS P2 = 0.3E9 / (0.1 * 1E6) = 3000 MFLOPS

1.5 Textbook Problem 4.15.14

Another pitfall cited in COD Section 1.11 (Fallacies and pitfalls) is expecting to improve the overall performance of a computer by improving only one aspect of the computer. Consider a computer running a program that requires 250 s, with 70 s spent executing FP instructions, 85 s executing L/S instructions, 40 s executing branch instructions, and the remaining time is spent executing integer instructions.

(a) By how much is the total time reduced if the time for FP operations is reduced by 20%?
Answer:
Total time reduced = ammount of time reduced when FP operations are reduced by 20%

FP time = 70 seconds
FP time reduced = 70 * 0.2 = 14 seconds

The total time is reduced by 14 seconds.

(b) By how much is the time for INT operations reduced if the total time is reduced by 20%? Assume the time for other operations reminds the same.
Answer:
INT Reduced Time = 0.8 * 250 = 200 seconds
Total Time = 70 + 85 + 40 = 195 seconds
Total Time Reduced = 200 - 195 = 5 seconds
5/55 x 100 = 90.9%

The time for INT operations is reduced by 90.9%

(c) Can the total time can be reduced by 20% by reducing only the time for branch instructions?
Answer:
Reduced Time = 0.8 * 250 = 200 seconds
Total Time = 70 + 85 + 55 = 210 seconds

No, the total time cannot be reduced by 20% just by reducing the time for branch instructions.

1.6 Textbook Problem 4.15.15

Assume a program requires the execution of 50 × 106 FP instructions, 110 × 106 INT instructions, 80 × 106 L/S instructions, and 16 × 106 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2 GHz clock rate.

(a) By how much must we improve the CPI of FP instructions if we want the program to run two times faster?
Answer:
Time = (FP * CPI + INT * CPI + L/S * CPI + Branch * CPI) / Clock Rate

Using the equation: Time = (59 + 110 + 320 + 32) / 2E9 = 0.256 seconds

CPI = (256 - 32 - 320 - 110) / 50 = -4.12

Since the CPI cannot be negative, we cannot improve the CPI of FP instructions to make the program run two times faster.

(b) By how much must we improve the CPI of L/S instructions if we want the program to run two times faster?
Answer:
Use the same process in part (a) but with L/S instructions instead of FP instructions.

Time = ((50 + 110 + 80 * CPI + 32) * 10E6) / (2 * 10E9)

CPI = (256 - 110 - 50 - 32) / 80 = 0.8

The CPI of L/S instructions must be improved by 0.8 to make the program run two times faster.

(c) By how much is the execution time of the program improved if the CPI of INT and FP instructions is reduced by 40% and the CPI of L/S and Branch is reduced by 30%?
Answer:

Time = (50 * 0.6 + 110 * 0.6 + 80 * 2.8 + 16 * 1.4) * 10E6 / 2E9 = 0.1712 seconds
Improved Time = 0.256 / 0.1712 = 1.495

The time is improved by 1.495 times.

1.7 Textbook Problem 4.15.16

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 sections 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 an additional 4 s of overhead, irrespective of the number of processors. Compute the per-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).

Answer:

Processor 2 time = 100 / 2 + 4 = 54 seconds
Speed up = 100 / 54 = 1.85
Ideal Speed up = 100 / 50 = 2
Ratio = 1.85 / 2 = 0.925

Processor 4 time = 100 / 4 + 4 = 29 seconds
Speed up = 100 / 29 = 3.45
Ideal Speed up = 100 / 25 = 4
Ratio = 3.45 / 4 = 0.8625

Processor 8 time = 100 / 8 + 4 = 16.5 seconds
Speed up = 100 / 16.5 = 6.06
Ideal Speed up = 100 / 12.5 = 8
Ratio = 6.06 / 8 = 0.7575

Processor 16 time = 100 / 16 + 4 = 10.25 seconds
Speed up = 100 / 10.25 = 9.75
Ideal Speed up = 100 / 6.25 = 16
Ratio = 9.75 / 16 = 0.6094

Processor 32 time = 100 / 32 + 4 = 7.125 seconds
Speed up = 100 / 7.125 = 14.04
Ideal Speed up = 100 / 3.125 = 32
Ratio = 14.04 / 32 = 0.4388

Processor 64 time = 100 / 64 + 4 = 5.5625 seconds
Speed up = 100 / 5.5625 = 17.97
Ideal Speed up = 100 / 1.5625 = 64
Ratio = 17.97 / 64 = 0.2804

Processor 128 time = 100 / 128 + 4 = 4.781 seconds
Speed up = 100 / 4.781 = 20.9
Ideal Speed up = 100 / 0.781 = 128
Ratio = 20.9 / 128 = 0.1633