Class: CECS 341 – Fall 2020
Professor: Maryam Seyyedhosseini
Student name: Thanh Nguyen
Student ID: 026843815

Homework 01

Performance Evaluation and Amdahl's Law

Due date: 11PM _ 09/20/2020

1. Problem 01:

Problem 1 (40 points):

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.
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.
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.
d. Another common performance figure is MFLOPS (millions of floating-point operations per second), defined as:

$$MFLOPS = No. FP\ operations/(execution\ time\ \times 1E6)$$

Assume that 40% of the instructions executed on both P1 and P2 are floating-point instructions. Find the MFLOPS figures for the processors.

P1: clock rate: 4GHz, average CPI: 0.9, execution: 5.0E9 instructions

P2: clock rate: 3GHz, average CPI: 0.75, execution: 1.0E9 instructions

a. Use the formula: $CPU\ Time = \frac{Instuctions\ Count \times CPI}{Clock\ Rate}$

$$CPU\ Time_{P1} = \frac{5 \times 10^9 \times 0.9}{4 \times 10^9} = 1.125s\ and\ \ CPU\ Time_{P2} = \frac{10^9 \times 0.75}{3 \times 10^9} = 0.25s$$

$$\frac{Performance_{P2}}{Performance_{P1}} = \frac{CPU\ Time_{P1}}{CPU\ Time_{P2}} = \frac{1.125}{0.25} = 4.5$$

So, P2 is 4.5 times faster than P1, but P2 has smaller clock rate then P1.

=> This fallacy is false for P1 and P2.

b. Use the formula: $CPU\ Time = \frac{Instuctions\ Count \times CPI}{Clock\ Rate}$

The CPU time of P1 using to execute a sequence of 1.0E9 instructions:

$CPU\ Time_{P1}{}' = \frac{10^9 \times 0.9}{4 \times 10^9} = 0.225s$

The number of instructions such that CPU Time$_{P2}$ = 0.225s is:

$$CPU\ Time_{P2} = \frac{number\ of\ instructions_2 \times 0.75}{3 \times 10^9} \leftrightarrow$$

$$0.225 = \frac{number\ of\ instructions_2 \times 0.75}{3 \times 10^9}$$

$$\leftrightarrow number\ of\ instructions_2 = \frac{0.225\ \times 3 \times 10^9}{0.75} = 9 \times 10^8$$

So, P1 can process more instructions than P2 in the same period of time.

c. Use the formula: $MIPS = \frac{Clock\ Rate}{CPI \times 10^6}$

$$MIPS_{P1} = \frac{4 \times 10^9}{0.9 \times 10^6} = 4444\ and\ \ MIPS_{P2} = \frac{3 \times 10^9}{0.75 \times 10^6} = 4000$$

So, P1 has larger MIPS than P2. However, the performance of P1 is slower than P2.

=> This statement is not true for P1 and P2.

d. Use the formula: $MFLOPS = \frac{No.FP\ operations}{execution\ time \times 10^6} = \frac{0.4 \times instrutions}{execution\ time \times 10^6}$ (Millions of floating-point operations per second)

$$MFLOPS_{P1} = \frac{0.4 \times 5 \times 10^9}{1.125 \times 10^6} = 1778\ and\ \ MIPS_{P2} = \frac{0.4 \times 10^9}{0.25 \times 10^6} = 1600$$

So, P1 has the bigger MFLOPS than P2.

2. ***<u>Problem 02:</u>***

Problem 2 (20 points):

A program P running on a single-processor system takes time T to complete. Let us assume that 40% of the program's code is associated with "data management housekeeping" (according to Amdahl) and, therefore, can only execute sequentially on a single processor. Let us further assume that the rest of the program (60%) is "embarrassingly parallel" in that it can easily be divided into smaller tasks executing concurrently across multiple processors (without any interdependencies or communications among the tasks).

a. Calculate T2, T4, T8, which are the times to execute program P on a two-, four-, eight-processor system, respectively.
b. Calculate T∞ on a system with an infinite number of processors. Calculate the speedup of the program on this system, where speedup is defined as $\frac{T}{T\infty}$. What does this correspond to?

Let the time taken to execute on a single processor system be T.

a. Use the formula $T_{improved} = \frac{T_{affected}}{improvement\ factor} + T_{unaffected}$

$$T_2 = \frac{0.6T}{2} + 0.4T = 0.7T \text{ and } T_4 = \frac{0.6T}{4} + 0.4T = 0.55T \text{ and } T_8 = \frac{0.6T}{8} + 0.4T = 0.475T$$

b. Speedup is defined as $\frac{T}{T_\infty}$ with $T_\infty = \frac{0.6T}{\infty} + 0.4T = 0.4T$

⇨ $Speedup = \frac{T}{T_\infty} = \frac{T}{0.4T} = 2.5$

⇨ This corresponds to the speedup if only the serial portion had been executed.

3. ***<u>Problem 03:</u>***

Problem 3 (15 points):

Assume that we are considering enhancing a machine by adding a vector mode to it. When a computation is performed in vector mode, it is 20 times faster than the normal mode of execution. We call percentage of time that could be spent using vector mode the percentage of vectorization.

a. What percentage of vectorization is needed to achieve a speedup of 2?
b. What percentage of vectorization is needed to achieve one-half of the maximum speedup attainable from using vector mode?
c. Suppose you have measured the percentage of vectorization for programs to be 70%. The hardware design group says they can double the speed of vector rate with a significant additional engineering investment. You wonder whether the compiler crew could increase the use of vector mode as another approach to increasing performance. How much of an increase in the percentage of vectorization (relative to the current usage) would the compiler team need to obtain the same performance gain? Which investment would you recommend?

a. Use the formula: $Speedup\ = \frac{1}{(1-F)+\frac{F}{Improvement\ Factor}}$

$$\leftrightarrow 2 = \frac{1}{(1-F)+\frac{F}{20}} \leftrightarrow F = \frac{10}{19} = 0.5263$$

So, the percentage of vectorization needed to achieve a speedup of 2 is 52.63%.

b. We have the maximum possible speedup is 20, so one-half of this value would be 10.

Use the formula: $Speedup\ = \frac{1}{(1-F)+\frac{F}{Improvement\ Factor}} \leftrightarrow 10 = \frac{1}{(1-F)+\frac{F}{20}} \leftrightarrow F = \frac{18}{19} = 0.9474$

So, the percentage of vectorization is needed to achieve one-half of the maximum speedup is 94.74%.

c. Use the formula: $Speedup\ = \frac{1}{(1-F)+\frac{F}{Improvement\ Factor}}$

For the hardware speedup, after we doubled hardware speed, the Improvement Factor is equal to 40. Therefore, the speedup is:

$$Speedup\ = \frac{1}{(1-F)+\frac{F}{Improvement\ Factor}} = \frac{1}{(1-0.7)+\frac{0.7}{40}} = \frac{400}{127} = 3.1496$$

The percentage of vectorization would need to achieve this speedup by improving the compiler is:

$$Speedup\ = \frac{1}{(1-F)+\frac{F}{Improvement\ Factor}} \leftrightarrow \frac{400}{127} = \frac{1}{(1-F)+\frac{F}{20}} \leftrightarrow F = 0.7184$$

As a result, the hardware design group only has to improve its Improvement Factor from 0.7 to 0.7184. It means they just need an improvement of only 1.84% to achieve the same overall speedup. Therefore, I recommend using the compiler to crew to gain the extra vectorization.

Student name: Thanh Nguyen
Student ID: 026843815

4. Problem 04:

Problem 4 (15 points):

Assume a program requires the execution of 50×10^6 FP (Floating Point) instructions, 110×10^6 INT (integer) instructions, 80×10^6 L/S (Load/Store) instructions, and 16×10^6 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2GHz clock rate.

a. By how much must we improve the CPI of FP (Floating Point) instructions if we want the program to run two times faster?
b. By how much must we improve the CPI of L/S (Load/Store) instructions if we want the program to run two times faster?
c. By how much is the execution time of the program improved if the CPI of INT (Integer) and FP (Floating Point) instructions are reduced by 40% and the CPI of L/S (Load/Store) and Branch is reduced by 30%?

Clock rate = 2GHz

	Floating Point	Integer	Load/Store	Branch
IC	50×10^6	110×10^6	80×10^6	16×10^6
CPI	1	1	4	2

a. Use the formula: $CPU\ Time = \frac{\sum_{i=1}^{4} IC_i \times CPI_i}{Clock\ Rate}$

$$CPU\ Time_{new} = \frac{CPU\ Time_{old}}{2} \leftrightarrow$$

$$\frac{CPI_{FP\ new} \times IC_{FP} + CPI_{INT} \times IC_{INT} + CPI_{L/S} \times IC_{L/S} + CPI_{branch} \times IC_{branch}}{Clock\ Rate} = \frac{CPI_{FP\ old} \times IC_{FP} + CPI_{INT} \times IC_{INT} + CPI_{L/S} \times IC_{L/S} + CPI_{branch} \times IC_{branch}}{2 \times Clock\ Rate}$$

$$\leftrightarrow 2\left(CPI_{FP\ new} \times IC_{FP} + CPI_{INT} \times IC_{INT} + CPI_{L/S} \times IC_{L/S} + CPI_{branch} \times IC_{branch}\right) = CPI_{FP\ old} \times IC_{FP} + CPI_{INT} \times IC_{INT} + CPI_{L/S} \times IC_{L/S} + CPI_{branch} \times IC_{branch}$$

$$\leftrightarrow 2CPI_{FP\ new} \times IC_{FP} + CPI_{INT} \times IC_{INT} + CPI_{L/S} \times IC_{L/S} + CPI_{branch} \times IC_{branch} = CPI_{FP\ old} \times IC_{FP}$$

$$CPI_{FP\ new} = \frac{CPI_{FP\ old} \times IC_{FP} - CPI_{INT} \times IC_{INT} - CPI_{L/S} \times IC_{L/S} - CPI_{branch} \times IC_{branch}}{2IC_{FP}}$$

$$= \frac{1 \times 50 \times 10^6 - 1 \times 110 \times 10^6 - 4 \times 80 \times 10^6 - 2 \times 16 \times 10^6}{2 \times 50 \times 10^6} = \frac{-103}{25} < 0$$

⇨ It is impossible to make the program runs two times faster.

b. According to problem a, we have:

$$CPI_{L/S\,new} = \frac{CPI_{L/S\,old} \times IC_{L/S} - CPI_{INT} \times IC_{INT} - CPI_{FP} \times IC_{FP} - CPI_{branch} \times IC_{branch}}{2IC_{L/S}}$$

$$= \frac{4 \times 80 \times 10^6 - 1 \times 110 \times 10^6 - 1 \times 50 \times 10^6 - 2 \times 16 \times 10^6}{2 \times 80 \times 10^6} = \frac{128}{160} = 0.8$$

⇨ Because $\frac{CPI_{L/S\,old}}{CPI_{L/S\,new}} = \frac{4}{0.8} = 5$, the CPI of L/S must be improved by 5 times to make the program runs two times faster.

c.

$$CPU\ Time_{old} = \frac{CPI_{FP} \times IC_{FP} + CPI_{INT} \times IC_{INT} + CPI_{L/S} \times IC_{L/S} + CPI_{branch} \times IC_{branch}}{Clock\ Rate}$$

$$= \frac{1 \times 50 \times 10^6 + 1 \times 110 \times 10^6 + 4 \times 80 \times 10^6 + 2 \times 16 \times 10^6}{2 \times 10^9} = \frac{32}{125}$$

$$= 0.256s$$

$$CPU\ Time_{new}$$

$$= \frac{0.6CPI_{FP} \times IC_{FP} + 0.6CPI_{INT} \times IC_{INT} + 0.7CPI_{L/S} \times IC_{L/S} + 0.7CPI_{branch} \times IC_{branch}}{Clock\ Rate}$$

$$= \frac{0.6 \times 50 \times 10^6 + 0.6 \times 110 \times 10^6 + 0.7 \times 4 \times 80 \times 10^6 + 0.7 \times 2 \times 16 \times 10^6}{2 \times 10^9} = \frac{107}{625}$$

$$= 0.1712s$$

After improving, the program runs 1.5 times faster. This is the speedup of 33.125%

5. ***<u>Problem 05:</u>***

Problem 5 (10 points):

Processor A has a clock rate of 3.6 GHz and voltage 1.25V. Assume that, on average, it consumes 90W of dynamic power.

Processor B has a clock rate of 3.4 GHz and voltage of 0.9V. Assume that, on average, it consumes 40W of dynamic power.

For each processor find the average capacitive loads.

A: Clock Rate: 3.6GHz, Voltage: 1.25V, dynamic power: 90W

B: Clock Rate: 3.4GHz, Voltage: 0.9V, dynamic power: 40W

Use the formula: Power = Capacitive load x Voltage^2 x Frequency => $C = \frac{Power}{Voltage^2 \times f}$

$$C_A = \frac{90}{3.6 \times 10^9 \times 1.25^2} = 16nF \; and \; C_B = \frac{40}{3.4 \times 10^9 \times 0.9^2} = 14.5nF$$