MCC2313

Advanced Computer System & Architecture

2013/14

Tutorial

1. Computer A runs program gcc in 3.2 seconds. Computer B runs the same program in 2.9 seconds. Which computer has better performance and by how much? What about computer C, which runs the program in 3.1 seconds?

2. Which computer has a faster execution time for program perl? Computer A, which runs the program in 730M cycles. Or computer B, which runs the program in 810M cycles? What if A has a clock rate of 500MHz and B has a clock rate of 550Mhz?

3. Which computer is faster and by how much? Computer A runs lisp in 100M instructions, the CPI is 2.4 and the clock cycle time is 1.5 ns. Computer B runs lisp in 115M instructions, the CPI is 2.1 and the clock cycle time is 1.4 ns. The manufacturers of computer B decided to run their clock to 1000MHz. But this caused design changes that raised the CPI to 2.6, and caused the compiler to produce 118M instructions. Was the clock change worthwhile?

4. We are considering an enhancement to the processor of a web server. The new CPU is 20 times faster on search queries than the old processor. The old processor is busy with search queries 70% of the time, what is the speedup gained by integrating the enhanced CPU?

5. Booth's algorithm, used to perform binary multiplication of X * Y using addition, subtraction and arithmetic right shift, is given below:
   1. A ← 0, Q ← X, Q-1 ← 0, M ← Y
   2. For I=1 to n do
      a. IfQ0Q-1 =01thenA←A+M
      b. IfQ0Q-1 =10thenA←A–M
      c. Arithmetic Right Shift (A || Q)

   The answer is stored in the combination of (A || Q)

   Given that 5% of all instructions in a given benchmark are 32-bit binary multiplications (no other multiplications take place), how much faster is the machine with the hardware multiplication circuit over a machine that must perform the multiplication using Booth's algorithm?
   Assume that each instruction (line) in the algorithm takes 1 clock cycle to perform (including the for-loop mechanisms and each if-then statement).
   Assume that a machine which can calculate binary multiplication using a hardware circuit has a multiply instruction with a CPI of 15.

The frequency of the enhancement (the multiplication circuit) is 5%
The machine that uses Booth's algorithm to perform multiplication executes 1 + 4 * 32 instructions (at one cycle each) or has a 129 cycle multiply
The machine with the hardware multiplier takes 15 cycles per multiple
The speedup in the enhanced mode is then 129 / 15 = 8.6

6. Consider two different implementations of the same instruction set architecture. There are four classes of instructions, A, B, C, and D. The clock rate and CPI of each implementation are given in the following table.

	Clock rate	CPI class A	CPI class B	CPI class C	CPI class D
P1	1.5 GHz	1	2	3	4
P2	2 GHz	2	2	2	2

Given a program with 10^6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C and 20% class D, which implementation is faster?

7. For the multi-cycle MIPS
   Load 5 cycles
   Store 4 cycles
   R-type 4 cycles
   Branch 3 cycles
   Jump 3 cycles

   If a program has
   50% R-type instructions
   10% load instructions
   20% store instructions
   8% branch instructions
   2% jump instructi ons

   then what is the CPI?

8. An architect wants to consider the effect of adding a register-memory addressing mode to a load/store machine. The idea is to replace sequences such as
   lw $s2, 16 ($s4)
   add $s5, $s1, $s2
   by a new single instruction:
   add_new $s5, $s1, 16($s4)

   Assume that the new instruction will cause clock cycle time to increase by 25%. Assume the following instruction mix.

Instruction Class	**Frequency**
Loads	25%
Stores	12%
ALU ops	35%
Branches	28%

Assume that the new instruction affects only the clock speed and not the CPI. *What percentage of the loads* must be eliminated for the machine with the new instruction to have at least the same performance?

9. Suppose that a program is being run on a processor consists of the following instruction mix:

Operation	**Frequency**	**Clock cycle count per instruction**
ALU operations	35%	4
Loads	20%	2
Stores	25%	2
Branches	20%	3

With the current processor, only **20%** of all ALU operations write results directly to memory, the remaining ALU operations write the results into registers. A designer decides to investigate a modified architecture for the processor by adding new ALU operations such that **all results** from ALU operations are written into registers, and, hence, require additional Store commands. The new ALU operations have a clock cycle of 2. By what **percentage** the modified processor's clock cycle should be faster/slower than the current processor's clock cycle so that both processors have the same execution time?

Answer:

1. Performance is relative so we want to measure P_B/P_A which is the inverse of $ET_A/ET_B = 3.2/2.9 = 1.103$. Thus the performance of computer B is 10% better that computer A. Computer C is 3% (3.2/3.1 = 1.03) better than computer A. And computer B is 7% (3.1/2.9 = 1.07) faster than computer C.

2. CPU execution time = # CPU cycles * CPU cycle time. So from only the number of cycles we can't compare the computers. As the CPU clock rate is known we can now compute the execution time. The clock period or clock cycle time is 1/(clock rate). So the execution time on A is 730/500 = 1.46 seconds. $ET_B = 810/550 = 1.47$. So Computer A is faster. It is about 0.5% faster.

3. $ET_A = 100{,}000{,}000*2.4*1.5ns = 0.36$ seconds. $ET_b = 115{,}000{,}000*2.1*1.4ns = 0.338$ seconds. Thus B is faster than A by 6%. By changing the clock, $ET_B = 118{,}000{,}000*2.6*1ns = 0.307$. So the change has improved computer B by 0.338/0.307 = 1.1, 10% (for the program perl).

4. Calculation:

$$Speedup = \frac{1}{(1 - Fraction_{enhanced}) \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

$Fraction_{enhanced} = 70\ \% = 0.70$

$Speedup_{enhanced} = 20$

$$Speedup = \frac{1}{(1-0.70) + \frac{0.70}{20}} = \frac{1}{0.335} = 2.985$$

5.

$$Speedup = \frac{1}{(1-0.05) + \frac{0.05}{8.6}} = \frac{1}{0.955} = 1.047 \text{ or a 4.7\% speedup}$$

6.

The No. instructions for all four classes are:
Class A: 105 Class B: 2*105 Class C: 5*105 Class D: 2*105
CPU time = (No. instructions × CPI) / Clock rate
P1:
CPU time Class A: (105 * 1)/ 1.5*109 = 0.67*10-4 CPU time Class B: (2*105 * 2)/ 1.5*109 = 2.67*10-4 CPU time Class C: (5*105 * 3)/ 1.5*109 = 10*10-4 CPU time Class D: (2*105 * 4)/ 1.5*109 = 5.33*10-4 Total CPU time is 18.67*10-4
P2:
CPU time Class A: (105 * 2)/ 2*109 = 10-4 CPU time Class B: (2*105 * 2)/ 2*109 = 2*10-4 CPU time Class C: (5*105 * 2)/ 2*109 = 5*10-4 CPU time Class D: (2*105 * 2)/ 2*109 = 2*10-4 Total CPU time is 10*10-4
P2 is faster with a shorter CPU time.

7. CPI = (4x50 + 5x10 + 4x20 + 3x8 + 3x2)/100 = 3.6

8. **CPU time** = (CPU clock cycles) x (Clock cycle time)
= (CPI x Instruction count) x (Clock cycle time)
CPI values are the same, say C.
If clock cycle time for the previous is 100, clock cycle time for the proposed becomes 125.
For at least the same performance, CPU times should be equal:

CPU Time (previous) = CPU Time (proposed)
C x Instruction count (previous) x 100 = C x Instruction count (proposed) x 125 Instruction count (proposed) = 0.8 x Instruction count (previous)

If we have 100 instructions for the previous, we must have 80 instructions for the proposed. Therefore, we must eliminate 20% of **total** instructions.
Since we can only eliminate loads (25%), we must eliminate **80% (20/25) of the loads** to be able to eliminate 20% of total instructions.

9. Execution time = CPU time
CPI_{old} = 35% * 4 + 20% * 2 + 25% * 2 + 20% * 3 = 2.9

$$\text{CPU time}_{old} = \text{CPI}_{old} * \text{Instruction Count}_{old} * \text{Clock Cycle}_{old}$$
$$= 2.9 * \text{Instruction Count}_{old} * \text{Clock Cycle}_{old}$$

With modified architecture:

ALU operations: (35% * 20%) operations use the new ALU operations with 2 clock cycles per instruction, and 35% - (35% * 20%) operations use the original ALU operations with 4 clock cycles per instruction

Load operations: No change

Store operations: (25% + (35% * 20%)) operations with 2 clock cycles

Branches: No change

Overall program becomes 100% + (35% * 20%) = 107%

$$\text{CPI}_{new} = [(35\% * 20\%) * 2 + (35\% - (35\% * 20\%)) * 4 + 20\% * 2 + (25\% + (35\% * 20\%)) * 2 + 20\% * 3] / 107\%$$
$$= 2.9 / 1.07 = 2.71$$

$$\text{CPU time}_{new} = \text{CPI}_{new} * \text{Instruction Count}_{new} * \text{Clock Cycle}_{new}$$
$$= 2.71 * (107\% * \text{Instruction Count}_{old}) * \text{Clock Cycle}_{new}$$
$$= 2.9 * \text{Instruction Count}_{old} * \text{Clock Cycle}_{new}$$

Since we require that (CPU time_{new}) is equal to (CPU time_{old}), then

$$\text{CPU time}_{new} = \text{CPU time}_{old}$$
$$2.9 * \text{Instruction Count}_{old} * \text{Clock Cycle}_{new} = 2.9 * \text{Instruction Count}_{old} * \text{Clock Cycle}_{old}$$
$$\text{Clock Cycle}_{new} = \text{Clock Cycle}_{old}$$

Thus, Clock Cycle_{new} must be equal to Clock Cycle_{old} for the (CPU time_{new}) to be equal to (CPU time_{old}).

Hence, the modified processor's clock cycle must be **0%** faster than the original processor's clock cycle for the execution time to be the same.

Sources:

Questions 1-3 : http://homedir.jct.ac.il/~citron/ca/ex7_ans.html

Questions 4-5 :
http://www.csci.csusb.edu/schubert/tutorials/csci610/w06/TKleffelAmdahl06w.pdf

Question 6: http://www.ecs.umass.edu/ece232/hw/PracProbs_solutions.pdf

Question 7: http://www.divms.uiowa.edu/~ghosh/2-2-06.pdf

Question 8: http://mimoza.marmara.edu.tr/~isil.oz/courses/cse338/quiz_338_S12_1.pdf

Question 9:
http://www.google.com/url?sa=t&rct=j&q=instruction%20count%20old%20cpi%20new&source=web&cd=4&ved=0CD4QFjAD&url=http%3A%2F%2Ffaculty.kfupm.edu.sa%2Fcoe%2Fmarwan%2Frichfiles%2FCOE308_T031_Quiz01_Solution_20030930.doc&ei=GoKYUPTuA4OIrAeUu4CYBA&usg=AFQjCNH3z7UMSgQUVNC9c61bIy6i43x5cQ