

ICS 233 – Spring 2007

Computer Architecture & Assembly Language

Assignment 5: Performance and Metrics

Due: Wednesday, May 2nd, 2007

1. (4 pts) We wish to compare the performance of two different computers: M1 and M2. The following measurements have been made on these computers:

Program	Time on M1	Time on M2
1	2.0 seconds	1.5 seconds
2	5.0 seconds	10.0 seconds

Program	Instructions executed on M1	Instructions executed on M2
1	5×10^9	6×10^9

- a) (1 pt) Which computer is faster for each program, and how many times as fast is it?
- b) (1 pt) Find the instruction execution rate (instructions per second) for each computer when running program 1.
- c) (1 pt) The clock rates for M1 and M2 are 3 GHz and 5 GHz respectively. Find the CPI for program 1 on both machines.
- d) (1 pt) Suppose that program 1 must be executed 1600 times each hour. Any remaining time should be used to run program 2. Which computer is faster for this workload? Performance is measured here by the throughput of program 2.

Solution:

- a) **For program 1, M2 is $2.0/1.5 = 1.33$ times as fast as M1.**
For program 2, M1 is $10.0/5.0 = 2$ times as fast as M2.

- b) **For program 1:**
Execution rate on M1 = $5 \times 10^9 / 2.0 = 2.5 \times 10^9$ IPS (Instructions Per Second).
Execution rate on M2 = $6 \times 10^9 / 1.5 = 4 \times 10^9$ IPS.

- c) **CPI = Execution time \times Clock rate / Instruction Count**
For program 1:
CPI on M1 = $2.0 \times 3 \times 10^9 / (5 \times 10^9) = 1.2$ cycles per instruction
CPI on M2 = $1.5 \times 5 \times 10^9 / (6 \times 10^9) = 1.25$ cycles per instruction

- d) **Running program 1 1600 times each hour:**
On M1, time required for program 1 = $1600 \times 2.0 = 3200$ seconds
On M2, time required for program 1 = $1600 \times 1.5 = 2400$ seconds

Time left to run program 2 each hour:
On M1, time left for program 2 = $3600 - 3200 = 400$ seconds
On M2, time left for program 2 = $3600 - 2400 = 1200$ seconds

In that time left, program 2 can run:
On M1, program 2 can run $400/5 = 80$ times
On M2, program 2 can run $1200/10 = 120$ times

Thus M2 performs better on this workload than M1.

2. (2 pts) Suppose you wish to run a program P with 7.5×10^9 instructions on a 5 GHz machine with a CPI of 1.2.
- (1 pt) What is the CPU execution time?
 - (1 pt) When you run program P, it takes 3 seconds of wall time to complete. What is the percentage of the CPU time program P received?

Solution:

a) **CPU execution time = Instruction Count \times CPI / Clock rate**
CPU execution time = $7.5 \times 10^9 \times 1.2 / (5 \times 10^9) = 1.8$ seconds

b) **% of CPU time = $1.8 / 3 = 0.6$ or 60% of the total wall time**

3. (4 pts) Consider two different implementations, M1 and M2, of the same instruction set. There are five classes of instructions (A, B, C, D and E) in the instruction set. M1 has a clock rate of 4 GHz and M2 has a clock rate of 6 GHz.

Class	CPI on M1	CPI on M2
A	1	2
B	2	2
C	3	2
D	4	4
E	3	4

- (2 pts) Assume that peak performance is defined as the fastest rate that a computer can execute any instruction sequence. What are the peak performances of M1 and M2 expressed in instructions per second?
- (2 pts) If the number of instructions executed in a certain program is divided equally among the classes of instructions, except that for class A, which occurs twice as often as each of the others, how much faster is M2 than M1?

Solution:

a) **For peak performance, the machine will be executing the fastest set of instructions. M1 will be executing instructions of class A only, and M2 will be executing instructions that belong to class A, B, or C.**

Peak performance of M1 = $4 \times 10^9 / 1 = 4 \times 10^9$ IPS = 4000 MIPS

Peak performance of M2 = $6 \times 10^9 / 2 = 3 \times 10^9$ IPS = 3000 MIPS

- b) **Average CPI on M1 = $(2 \times 1 + 2 + 3 + 4 + 3) / 6 = 14 / 6 = 2.33$**
Average CPI on M2 = $(2 \times 2 + 2 + 2 + 4 + 4) / 6 = 16 / 6 = 2.67$

Since the instruction count is the same:

$$\frac{\text{Performance of M2}}{\text{Performance of M1}} = \frac{\text{Clock Rate of M2}}{\text{Clock Rate of M1}} \times \frac{\text{CPI on M1}}{\text{CPI on M2}}$$

M2 is faster than M1 by a factor of $(6 \text{ GHz} / 4 \text{ GHz}) \times (2.33 / 2.67) = 1.31$

4. (5 pts) Consider two different implementations, M1 and M2, of the same instruction set. There are three classes of instructions (A, B, and C) in the instruction set. M1 has a clock rate of 6 GHz and M2 has a clock rate of 3 GHz. The CPI for each instruction class on M1 and M2 is given in the following table:

Class	CPI on M1	CPI on M2	C1 Usage	C2 Usage	C3 Usage
A	2	1	40%	40%	60%
B	3	2	40%	20%	15%
C	5	2	20%	40%	25%

The above table also contains a summary of the usage of instruction classes generated by three different compilers: C1, C2, and C3. Assume that each compiler generates the same number of instructions for a given program.

- a) (1 pt) Using C1 compiler on both M1 and M2, how much faster is M1 than M2?
b) (1 pt) Using C2 compiler on both M1 and M2, how much faster is M2 than M1?
c) (1 pt) If you purchase M1, which compiler would you use?
d) (1 pt) If you purchase M2, which compiler would you use?
e) (1 pt) Which computer and compiler combination give the best performance?

Solution:

- a) **Using C1 compiler:**

$$\text{Average CPI on M1} = 0.4 \times 2 + 0.4 \times 3 + 0.2 \times 5 = 3.0$$

$$\text{Average CPI on M2} = 0.4 \times 1 + 0.4 \times 2 + 0.2 \times 2 = 1.6$$

M1 is faster than M2 by a factor of $(6 \text{ GHz} / 3 \text{ GHz}) \times (1.6 / 3.0) = 1.0667$

- b) **Using C2 compiler:**

$$\text{Average CPI on M1} = 0.4 \times 2 + 0.2 \times 3 + 0.4 \times 5 = 3.4$$

$$\text{Average CPI on M2} = 0.4 \times 1 + 0.2 \times 2 + 0.4 \times 2 = 1.6$$

M2 is faster than M1 by a factor of $(3 \text{ GHz} / 6 \text{ GHz}) \times (3.4 / 1.6) = 1.0625$

- c) **Using C3 compiler on M1:**

$$\text{Average CPI on M1} = 0.6 \times 2 + 0.15 \times 3 + 0.25 \times 5 = 2.9$$

We should use compiler C3 because it gives the lowest CPI on M1.

d) Using C3 compiler on M2:

$$\text{Average CPI on M2} = 0.6 \times 1 + 0.15 \times 2 + 0.25 \times 2 = 1.4$$

We should use compiler C3 because it gives the lowest CPI on M2.

e) Compiler C3 compiler gives the best average CPI on both M1 and M2.

Performance is proportional to Clock rate / CPI, because I-count is the same.

$$\text{Performance of M1 relative to M2} = (6 \text{ GHz} / 3 \text{ GHz}) \times (1.4 / 2.9) = 2.8/2.9$$

Therefore, M2 gives the best performance using Compiler C3.

5. (2 pts) A benchmark program runs for 100 seconds. We want to improve the speedup of the benchmark by a factor of 3. We enhance the floating-point hardware to make floating-point instructions run 5 times faster. How much of the initial execution time would floating-point instructions have to account for to show an overall speedup of 3 on this benchmark?

Let f be the fraction of time spent in floating point instructions. Then, after the enhancement, the time taken by the machine will be:

$$100 \times (f/5 + (1-f)) = 100/3$$

$$(f/5 + (1-f)) = 1/3$$

$$1 - 1/3 = f - f/5$$

$$4f/5 = 2/3$$

$$f = (2/3) (5/4) = 5/6$$

Therefore, 5/6 of the initial 100 seconds, or 83.33 seconds, must be spent executing floating-point instructions.

6. (3 pts) Consider the following fragment of MIPS code. Assume that **a** and **b** are arrays of words and the base address of **a** is in **\$a0** and the base address of **b** is in **\$a1**. How many instructions are executed during the running of this code? If ALU instructions (**addu** and **addiu**) take 1 cycle to execute, load/store (**lw** and **sw**) take 5 cycles to execute, and the branch (**bne**) instruction takes 3 cycles to execute, how many cycles are needed to execute the following code (all iterations). What is the average CPI?

```

    addu  $t0, $zero, $zero      # i = 0
    addu  $t1, $a0, $zero       # $t1 = address of a[i]
    addu  $t2, $a1, $zero       # $t2 = address of b[i]
    addiu $t3, $zero, 101       # $t3 = 101 (max i)
loop: lw   $t4, 0($t2)           # $t4 = b[i]
    addu  $t5, $t4, $s0         # $t5 = b[i] + c
    sw    $t5, 0($t1)           # a[i] = b[i] + c
    addiu $t0, $t0, 1           # i++
    addiu $t1, $t1, 4           # address of next a[i]
    addiu $t2, $t2, 4           # address of next b[i]
    bne  $t0, $t3, loop        # loop if (i != 101)
```

$$\text{Total number of instructions executed} = 4 + 101 \times 7 = 711.$$

$$\text{ALU} = 4 + 101 \times 4 = 408, \text{ lw and sw} = 101 \times 2 = 202, \text{ and bne} = 101 \text{ instructions.}$$

$$\text{Total number of cycles} = 408 \times 1 + 202 \times 5 + 101 \times 3 = 1721.$$

$$\text{Average CPI} = 1721 / 711 = 2.42$$