Jacob Branaugh, Brenn Kucey
November 22, 2013
CS 472 HW 5

(6.5) The time taken by machines A, B, and C to execute a given task is

- A 16m, 9s
- B 14m, 12s
- C 12m, 17s What is the performance of each of these machines relative to machine A?

- $Perf_{AtoA} = \frac{969s}{969s} = 1$
- $Perf_{BtoA} = \frac{969s}{852s} = 1.14$, 14% faster
- $Perf_{CtoA} = \frac{969s}{737s} = 1.31$, 31% faster

(6.6) Why is clock rate a poor metric of computer performance? What are the relative strengths and weaknesses of clock speed as a performance metric?

- All clock speed tells you is how often the processor clock ticks. If two processors are identical, but one has a slightly higher clock speed, it is usually an indicator that the faster processor will perform slightly better. Factors such as cache sizes, architecture, computing power of the processor core, and power draw have a much greater say in determining how well a processor performs, as these listed attributes have a much greater effect on performance.

(6.12) The following figures define the typical operating parameters of a processor:

Operation	Frequency	Cycles
Arithmetic/logic instructions	45%	1
Register load operations	20%	3
Register store operations	10%	2
All branch instructions	25%	2

If the clock rate could be reduced by 15%, it would require only 2 cycles to perform a register load. Would that be a good idea?

- Yes. While a drop in clock speed will only make a small performance distance at best, wasting one less cycle on load operations will boost the overall system performance.

(6.13) A computer has the following parameters:

Operation	Frequency	Cycles
Arithmetic/logic instructions	65%	1
Register load operations	10%	5
Register store operations	5%	2
All branch instructions	20%	8

If the average performance of the computer (in terms of its CPI) is to be increased by 20% while executing the same instruction mix, what target must be achieved for the cycles per conditional branch instruction?

- $CPI_A = .65 + .5 + .1 + 1.6 = 2.85$
- $CPI_B = \frac{CPI_A}{1.2} = 2.375$
- $cycles_{branch} \leq \frac{2.375-.65-.5-.1}{.2} = 5.625 \Rightarrow cycles_{branch} = \mathbf{5}$

(6.17) For the following systems that have both serial and parallel activities, calculate the speedup ratio.

- 10 processors, $f_s = 0.1$

- $S = \frac{p}{pf_s+1-f_s} = \frac{10}{10\times0.1+1-0.1} = 5.26$

- 100 processors, $f_s = 0.1$

- $S = \frac{100}{100\times0.1+1-0.1} = 9.17$

- 5 processors, $f_s = 0.4$

- $S = \frac{5}{5\times0.4+1-0.4} = 1.92$

- 100 processors, $f_s = 0.01$

- $S = \frac{100}{100\times0.01+1-0.01} = 50.25$

(6.18) A system has a single core processor that costs $150. Suppose that adding more cores to the chip costs $10 per additional processor. (*Note*: For this system, the value of f_s is 0.1). If it is considered worthwhile adding cores until the incremental speedup ratio increases by less than 5% over the original performance, what is the optimum number of processors? What percentage increase in cost is required to achieve this performance?

- $S_i = \frac{1}{1\times0.1+1-0.1} = 1$

- Since the initial speed up is 1, we need to increase the number of processors until the speed up increase is less than .05. We wrote a quick python program to calculate this, and the event occurs at 34 processors, which means we want a 33 core processor. The additional 32 processors cost $320, bringing the total cost to $470 and the cost percentage increase to about 313%.

(6.22) A coprocessor is added to a computer to speed the execution time of string-processing instructions by a factor of 3.5. What fraction of the execution time must use these string-processing instructions in order to achieve an average speed up of 1.5?

- $S = \frac{1}{1-fraction+\frac{fraction}{factor}} \Rightarrow fraction = \frac{\frac{1}{S}-1}{\frac{1}{factor}-1} = .466$

- 46.6% of the execution time must be used on string processing to achieve the desired speed up.

(7.12) Figure 7.12 from the text demonstrates the execution of a conditional branch instruction in a flow-through computer. The grayed out sections of the computer are not required by a conditional branch instruction. Can you think of any way in which these unused elements of the computer could be used during the execution of a conditional branch?

- They could be used to perform arithmetic operations on register contents and store them back in the register file. A memory store could also be performed.

(7.15) What modification would have to be made to the architecture of the computer in the figure to implement operand shifting (as part of a normal instruction) like the ARM?

- Modifications have to be made to allow a literal to be passed in with a register. Multiplying or dividing by a literal power of 2 will shift left or right, respectively.