Computer Performance

"X is N% faster than Y."

$$\frac{\text{Execution Time of Y}}{\text{Execution Time of X}} = 1+\frac{N}{100}$$

Amdahl's law for overall speedup

$$\text{Overall Speedup} = \frac{1}{(1-F)+\frac{F}{S}}$$

F = The fraction enhanced

S = The speedup of the enhanced fraction

Using Amdahl's law

Overall speedup if we make 90% of a program run 10 times faster.

F = 0.9 S = 10

$$\text{Overall Speedup} = \frac{1}{(1-0.9)+\frac{0.9}{10}} = \frac{1}{0.1+0.09}=5.26$$

Overall speedup if we make 80% of a program run 20% faster.

F = 0.8 S = 1.2

$$\text{Overall Speedup} = \frac{1}{(1-0.8)+\frac{0.8}{1.2}} = \frac{1}{0.2+0.66}=1.153$$

*) http://www.cs.iastate.edu/~prabhu/Tutorial/title.html

You have a system that contains a special processor for doing floating-point operations. You have determined that 50% of your computations can use the floating-point processor. The speedup of the floating pointing-point processor is 15.

a) Overall speedup achieved by using the floating-point processor.

F = 0.5 S = 15

Overall speedup = $\dfrac{1}{(1-0.5)+\dfrac{0.5}{15}} = \dfrac{1}{0.5+0.033} = 1.876$

b) Overall speedup achieved if you modify the compiler so that 75% of the computations can use the floating-point processor.

F = 0.75 S = 15

Overall speedup = $\dfrac{1}{(1-0.75)+\dfrac{0.75}{15}} = \dfrac{1}{0.25+0.05} = 3.33$

c) What fraction of the computations should be able to use the floating–point processor in order to achieve an overall speedup of 2.25?

F = ? S = 15

$$2.25 = \frac{1}{(1-F)+\dfrac{F}{15}}$$

$$= \frac{15}{15-15F+F} = \frac{15}{15-14F}$$

$$2.25(15-14F) = 15$$

$$33.75 - 31.5F = 15$$

$$31.5F = 18.75$$

$$F = \frac{18.75}{31.5} = 0.595 \quad \text{or} \quad 60\%$$

You have a system that contains a special processor for doing floating-point operations. You have determined that 60% of your computations can use the floating-point processor. When a program uses the floating-point processor, the speedup of the floating-point processor is 40% faster than when it doesn't use it.

a) Overall speedup by using the floating-point processor.

$F = 0.6 \qquad S = 1.4$

$$\text{Overall speedup} = \frac{1}{(1-0.6)+\frac{0.6}{1.4}} = \frac{1}{0.4+0.429} = 1.206$$

b) In order to improve the speedup you are considering two options:

- Option 1: Modifying the compiler so that 70% of the computations can use the floating-point processor. Cost of this option is $50K.
- Option 2: Modifying the floating-point processor . The speedup of the floating-point processor is 100% faster than when it doesn't use it. Assume in this case that 50% of the computations can use the floating–point processor. Cost of this option is $60K.

Which option would you recommend? Justify your answer quantitatively.

$F = 0.7 \qquad S = 1.4$

✓

$$\text{Overall speedup} = \frac{1}{(1-0.7)+\frac{0.7}{1.4}} = \frac{1}{0.3+0.5} = 1.25$$

$$Cost/Speedup = \frac{\$50K}{1.25} = \$40K \longrightarrow \text{Option 1}$$

$F = 0.5 \qquad S = 2$

$$\text{Overall speedup} = \frac{1}{(1-0.5)+\frac{0.5}{2}} = \frac{1}{0.5+0.25} = 1.33$$

$$Cost/Speedup = \frac{\$60K}{1.33} = \$45.1K \longrightarrow \text{Option 2}$$

Therefore, Option 1 is better because it has a smaller Cost/Speedup ratio.

1) Suppose you have a load/store computer with the following instruction mix:

Operation	**Frequency**	**No. of Clock cycles**
ALU ops	35%	1
Loads	25%	2
Stores	15%	2
Branches	25%	3

a) Compute the CPI.

$$CPI_{old} = (0.35*1) + (0.25*2) + (0.15*2) + (0.25*3) = 1.9$$

b) We observe that 35% of the ALU ops are paired with a load, and we propose to replace these ALU ops and their loads with a new instruction. The new instruction takes 1 clock cycle. With the new instruction added, branches take 5 clock cycles, Compute the CPI for the new version.

$$0.35*0.35 = 0.1225$$

$$CPI_{new} = \frac{(0.35-0.1225)*1 + (0.25-0.1225)*2 + 0.15*2 + 0.25*5 + 0.1225*1}{(1.0.1225)}$$

$$= \frac{2.155}{0.8775} = 2.455$$

c) If the clock of the old version is 20% faster than the new version, which version has faster CPU Execution time and by how much percent?

$$\frac{CCT_{new}}{CCT_{old}} = 1.2 \qquad CCT_{new} = 1.2 * CCT_{old}$$

$$\begin{aligned} CPU\ Exec.Time_{old} &= 1.9 * IC_{old} * CCT_{old} \\ CPU\ Exec.Time_{new} &= 0.8775 * IC_{old} * 2.46 * 1.2 * CCT_{old} \\ &= 2.59 * IC_{old} * CCT_{old} \end{aligned}$$

So, old version is faster $\quad \frac{2.59}{1.9} = 1.36$

By 36%

2) For the purpose of solving a given application problem, you benchmark a program on two computer systems. On system A, the object code executed 80 million Arithmetic Logic Unit operations (ALU ops), 40 million load instructions, and 25 million branch instructions. On system B, the object code executed 50 million ALU ops, 50 million loads, and 40 million branch instructions. In both systems, each ALU op takes 1 clock cycles, each load takes 3 clock cycles, and each branch takes 5 clock cycles.

a) Compute the relative frequency of occurrence of each type of instruction executed in both systems.

	A	B
ALU ops	$\frac{80}{145} = 0.55$	$\frac{50}{140} = 0.36$
Loads	$\frac{40}{145} = 0.28$	$\frac{50}{140} = 0.36$
Branches	$\frac{25}{145} = 0.17$	$\frac{40}{140} = 0.28$

1) Suppose you have a load/store computer with the following instruction mix:

Operation	**Frequency**	**No. of Clock cycles**
ALU ops	35%	1
Loads	25%	2
Stores	15%	2
Branches	25%	3

a) Compute the CPI.

$$CPI_{old} = (0.35*1) + (0.25*2) + (0.15*2) + (0.25*3) = 1.9$$

b) We observe that 35% of the ALU ops are paired with a load, and we propose to replace these ALU ops and their loads with a new instruction. The new instruction takes 1 clock cycle. With the new instruction added, branches take 5 clock cycles, Compute the CPI for the new version.

$$0.35*0.35 = 0.1225$$

[1] http://www.cs.iastate.edu/~prabhu/Tutorial/title.html