Time v/s Throughput

Time to complete task:

- execution time, response time, latency

Tasks per unit time:

- throughput, bandwidth

Carrier	Latency	Throughput
Driving Home	15 mins	1092M bits/sec
Cable Modem	30 ms	<1M bits/sec

(Carry 10 12GB tapes... you can take more!)

Performance

Performance measured as the inverse of execution time.

$$Performance_X = \frac{1}{Execution\ time_X}$$

X is N times faster than Y:

$$\frac{Performance_X}{Performance_Y} = \frac{Execution\ time_Y}{Execution\ time_X} = N$$

Measuring Time

Elapsed time, wall-clock time:

- Actual time from start to completion of task
- Depends on CPU, disk, operating system (OS)
- Affected when multiple users share a system

CPU time:

- Only look at CPU performance
- Suitable if multiple users running simultaneously
- Possible to measure OS component

When times are accurately accounted for, elapsed time is CPU time + Idle time.

Performance Metrics

Different metrics appropriate at different levels

- Application
  - Answers per month, operations per second
- Compiler
  - millions of instructions per second (MIPS)
  - millions of floating-point ops per second (MFLOPS)
- CPU Organization
  - cycles per instruction
- FSM, implementation
  - cycles per second (clock rate)

Relating Different Levels

- Application: time for one operation

  $\Rightarrow$ 1 operation

- Compiler: expands operation into number of instructions

  $\Rightarrow N$ instructions

- CPU Organization: each instruction takes some number of cycles.

  $\Rightarrow N \times CPI$ cycles

  CPI: average number of cycles per instruction for instruction mix

- FSM, implementation: clock period

  $\Rightarrow N \times CPI \times seconds/cycle$ seconds

Cycles Per Instruction (CPI)

CPI is the *average* clock cycles per instruction

Different instructions might take different number of cycles!

$\Rightarrow$ look at *instruction mix*

(frequency of different instruction types)

If p_i is the probability of instruction type i, and CPI_i is the cycles taken by an instruction of type i:

$$CPI = \sum_{i=1}^{n} p_i \times CPI_i$$

Example CPI Calculation

Simple ISA with four types of instructions:

Instr. Type	Frequency p_i	Cycles CPI_i	Contribution $p_i \times CPI_i$	%Time taken
ALU	50%	1	0.5	33%
Load	20%	2	0.4	27%
Store	10%	2	0.2	13%
Branch	20%	2	0.4	27%

Total CPI: 1.5

Loads take up 0.4/1.5 = 0.27 = 27% of total time.

Which Machine Is Faster?

Given the following instruction mix and cycle counts:

Type	Frequency	Cycles	CPI Contribution
Load	30%	2	0.6
Store	15%	2	0.3
Branch	15%	2	0.3
ALU	40%	1	0.4

Total CPI: 1.6

Adding an ALU operation with one memory operand:

- Clock cycle period 1.25 $\times$ original period
- Half the loads + corresponding ALU ops replaced
- New operation takes 2 cycles

Solution 1: Find New IC, CPI, CC

New Instruction Count (IC):

$$
\begin{aligned}
IC_{new} &= IC - \text{instr replaced} + \text{instr added} \\
&= IC - (0.5 \times 30\% \times 2)IC + (0.5 \times 30\%)IC \\
&= 0.85 \times IC
\end{aligned}
$$

New CPI: we have a new instruction mix!

Type	Frequency	Cycles	CPI Contribution
Load	15	2	0.3
Store	15	2	0.3
Branch	15	2	0.3
ALU	25	1	0.25
ALU Mem	15	2	0.3

Solution 1: Find New IC, CPI, CC

Totals: 85 (Freq. column), 1.45 (CPI contribution)

$$\Rightarrow CPI_{new} = 1.45/0.85 = 1.7 = (1.7/1.6)CPI$$

Also, new clock cycle time:

$$CC_{new} = 1.25 \times CC$$

Therefore, execution time:

$$\begin{aligned} \text{Execution Time}_{new} &= IC_{new} \times CPI_{new} \times CC_{new} \\ &= (0.85 \times IC) \times ((1.7/1.6) \times CPI) \\ &\quad \times (1.25 \times CC) \\ &= 1.13 \times (IC \times CPI \times CC) \\ &= 1.13 \times \text{Execution Time} \end{aligned}$$

Solution 2: Compute CPU Cycles

$$Cycles = IC \times CPI = 1.6 \times IC$$

$$Cycles_{new} = Cycles - \textbf{saved} + \textbf{added}$$

$$\begin{aligned} \textbf{saved} &= \textbf{loads saved} + \textbf{alu saved} \\ &= (30\% \times 0.5 \times 2 + 30\% \times 0.5 \times 1) \times IC \\ &= 0.45 \times IC \end{aligned}$$

$$\begin{aligned} \textbf{added} &= \textbf{ALU memory cycles} \\ &= (30\% \times 0.5 \times 2) \times IC \\ &= 0.3 \times IC \end{aligned}$$

$$\begin{aligned} Cycles_{new} &= 1.6 \times IC - 0.45 \times IC + 0.3 \times IC \\ &= 1.45 \times IC \end{aligned}$$

Solution 2: Compute CPU Cycles

Therefore, execution time:

$$\begin{aligned}\text{Execution Time}_{new} &= Cycles_{new} \times CC_{new} \\ &= (1.45 \times IC) \times (1.25 \times CC) \\ &= 1.81 \times (IC \times CC)\end{aligned}$$

$$\begin{aligned}\text{Execution Time} &= IC \times CPI \times CC \\ &= 1.6 \times (IC \times CC)\end{aligned}$$

$\Rightarrow$ **original machine is** 1.13 **times faster.**

Solution 3: "100 typical instructions"

		Original			New	
	cycles per instr	instr count	cycle count	instr delta	instr count	cycle count
Load	2	30	60	-15	15	30
Store	2	15	30		15	30
Branch	2	15	30		15	30
ALU	1	40	40	-15	25	25
ALU Mem	2	0	0	+15	15	30
Total		100	160		85	145

Original time: 160 cycles $\times\ CC$

New time: 145 cycles $\times\ (1.25 \times CC)$

$\Rightarrow$ original machine is 1.13 times faster.