CPU Performance Notes
4/12/10

Source

Compiler

machine code

OS

CPU

MEM

# operations

# instructions / op

speed of instr. execution

speed of I/O (disk, filesystem)

i/o
Latency

- how long does a particular task take?
- response time
- time / work

Throughput

- total work done per unit time
- work / time

Swap for a faster processor?

L & T improve (L↓, T↑)

add more processors (same speed)?

only T will improve
**Elapsed Time:** total response time for a job

- this & other jobs on CPU
- I/O
- overhead
- idle time

**CPU Time:** total time spent executing this task

$$\text{Elapsed} \geq \text{CPU}$$
CPU hardware governed by constant rate clock:

Clock period [\(\mu s\)] e.g. 250 ps

Clock frequency \([\text{cycles/s}]\) e.g. 2 GHz
CPU Time = \[\text{CPU Clock Cycles} \times \text{Clock Cycle Time}\]
\[\text{[s]}\] \[\text{[cc]}\] \[\text{[s/cc]}\]

\[\text{= CPU Clock Cycles} / \text{Frequency}\]
\[\text{[cc]}\] \[\text{[cc/s]}\]

To improve performance:

1. reduce # clk cycles
2. increase clock rate
Example problem:

- Program P
  - machine A
    - 2GHz [c/s]
    - 10s [s]
  - machine B
    - Aim for los CPU time
    - Increasing clk rate \( \rightarrow \)
    - \( 1.2 \times \) # cycles
    - How fast must mach. B's clock be?
Example Solution

\[
\text{Freq}_B = \frac{\text{Cycles}_B [\text{cc}]}{\text{Time}_B [s]} = \frac{1.2 \times \text{Cycles}_A}{6 \text{s}}
\]

\[
\text{Cycles}_A = \text{Time}_A \cdot \text{Freq}_A
\]

\[
= \frac{[\text{cc}]}{[s]} \cdot \frac{[\text{cc/s}]}{[\text{cc/s}]} = 10 \text{s} \cdot 2 \times 10^9 = 2 \times 10^{10} \text{[cc]}
\]

\[
\text{Freq}_B = \frac{1.2 \times 2 \times 10^{10}}{6 \text{s}} = \frac{2.4 \times 10^{10}}{6} = 4 \times 10^9
\]
Clock Cycles = C x CPI

(beware: static v. dynamic)

- Typically compute a weighted average
  
  Average Cycles Per Instruction (CPI)

Instruction Count (dynamic) (IC)
CPU Time = \((LC \times CPI) \times \text{CycleTime}\) 
\[\text{s} \quad \text{[ins]} \quad \text{[cc/ins]} \quad \text{[s/cc]}\]

= \((LC \times CPI) / \text{Frequency}\) 
\[\text{[ins]} \quad \text{[cc/ins]} \quad \text{[cc/s]}\]
**CPI Problem**

machine A

- cycle time = 250 ps
- CPI = 2.0

machine B

- cycle time = 500 ps
- CPI = 1.2

(same ISA)

Which is faster?

**Soln**

\[
\begin{align*}
\text{Time}_A &= \frac{1C}{\text{Instruction Count}} \times CPI_A \times \text{Cycle Time}_A \\
&= 1C \times 2.0 \times 250 \text{ ps} = 1C \times 500 \text{ ps} \\
\text{Time}_B &= 1C \times CPI_B \times \text{Cycle Time}_B \\
&= 1C \times 1.2 \times 500 \text{ ps} = 1C \times 600 \text{ ps}
\end{align*}
\]

\[
\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{1C \times 600 \text{ ps}}{1C \times 500 \text{ ps}} = 1.2
\]

by this much

"A is 1.2x faster than B"
Amdahl's Law

\[ T_{\text{improved}} = \frac{T_{\text{affected}}}{\text{improvement factor}} + T_{\text{unaffected}} \]

*Example*: On machine A, multiplication accounts for 80s out of 100s total CPU Time. How much improvement in multiplication to make the program 5 times as fast (5x speedup overall).

*Soln*: Infinite improvement s.t. multiplication accounts for 0s of execution.

Corollary of Amdahl's: Make the common case fast.
Performance Summary

CPU Time = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}

- Algorithm, language & compiler affect these
- ISA affects all three
- Performance depends on all terms