Definition & concept

Lecture - 05

Performance of a Computer

- **Response Time**

  - The time between the start and completion of a task.

    It is also known as **execution time**.

  - This includes disk accesses, memory accesses, I/O

    activities, operating system overhead, CPU execution

    time, etc.

- **Throughput**

  - The total amount of job done in a fixed amount of time.

  - Decreasing response time almost always improves

    throughput.

Relation between Performance and Execution Time

- $\text{Performance}_X = 1/\ \text{Execution time}_X$

- For two computers X and Y, if the performance of X is greater than the performance of Y, then $\text{Performance}_X > \text{Performance}_Y$

$= 1/\ \text{Execution time}_X > 1/\ \text{Execution time}_Y$

$= \text{Execution time}_Y > \text{Execution time}_X$

- If X is n times faster than Y, then

$\text{Performance}_X\ /\ \text{Performance}_Y = n$

or, $\text{Execution time}_Y\ /\ \text{Execution time}_X = n$

Relative Performance

- If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?

$\text{Performance}_A / \text{Performance}_B$

$= \text{Execution time}_B / \text{Execution time}_A$

$= 15/10$

$= 1.5$

A is 1.5 times faster than B.

Measuring Performance

- Time is the measure of computer performance.

- Computers are often shared. In such cases, the system may try to optimize throughput rather than minimizing the elapsed time for one program.

- **CPU execution time** or **CPU time** is the time the CPU spends computing a task not including the time spent waiting for I/O or running other programs.

Measuring Performance

- CPU time- 1. User CPU time

  2. System CPU time

- CPU time spent in the program is called User CPU time, and
- The CPU time spent in the operating system performing tasks requested by the program is called System CPU time.

Measuring Performance

- Suppose, after executing a program we get the following measurements:

  User CPU time – 90.7 seconds

  System CPU time – 12.9 seconds

  Elapsed time – 159 seconds

  CPU Utilization is (90.7+12.9)/159 = 65%

  - about 65.2% of the elapsed time, the CPU was actively processing the program

- The term **system performance** is used to refer to elapsed time on an unloaded system while **CPU performance** refers to user CPU time on an unloaded system.

Clock Cycles

- Almost all computers are constructed using a clock that runs at a constant rate and determines when events take place in the hardware.

- Clock period is the time for a complete clock cycle.

- Clock rate is the inverse of clock period.

CPU Time

- CPU time for a program can be expressed in two ways:

  CPU time = CPU clock cycles for a program * clock cycle time

  or

  CPU time = CPU clock cycles for a program / Clock rate

- If we know the number of clock cycles and the number of instructions execute (Instruction count, IC), we can calculate the average number of clock cycles per instruction (CPI).

  CPI = CPU clock cycles for a program / Instruction count

- Instructions per clock (IPC) is the inverse of CPI.

CPU Time

- **CPU time = Instruction count * Cycles per instruction * Clock cycle time**

  or

  CPU time = Instruction count * Cycles per instruction / Clock rate

  T = N * CPI / f

- It is possible to compute the CPU clock cycles by looking at the different types of instructions and using their individual clock cycle counts.

  $$\textbf{CPU clock cycles} = \Sigma(CPI_i \times C_i)$$

  where C_i is the count of the number of instructions of class i executed, CPI_i is the average number of cycles per instruction for that instruction class, and n is the number of instruction classes.

Comparing Code Segments

	CPI for this instruction class		
	A	B	C
CPI	1	2	3

Code Sequence	Instruction count for instruction class		
	A	B	C
1	2	1	2
2	4	1	1

a) Which code sequence executes the most instructions?

b) Which will be faster?

c) What is the CPI for each sequence?

- Sequence 1 executes 5 instructions and sequence 2 executes 6 instructions.
- CPU clock cycles_1 = (2X1)+(1X2)+(2X3) = 10 cycles

  CPU clock cycles_2 = (4X1)+(1X2)+(1X3) = 9 cycles

  So code sequence 2 is faster.
- Since sequence 2 takes fewer overall clock cycles but has more instruction, it must have a lower CPI.

  CPI_1 = 10/5 = 2

  CPI_2 = 9/6 = 1.5

Practice Problem

Our favorite program runs in 10 seconds on computer A, which has a 2 GHz clock. We are trying to help a computer designer build a computer, B, which will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing computer B to require 1.2 times as many clock cycles as computer A for this program. What clock rate should we tell the designer to target?

CPU Time

- CPU performance depends on three characteristics:

  - **Clock cycle time (Clock rate) :** Hardware technology

  - **Clock cycles per instruction:** Architecture

  - **Instruction count (N) :** software technology

Components affect the CPU Performance

Hardware or software component	Affects what?
Algorithm	Instruction count
Programming language	Instruction count CPI
Compiler	Instruction count CPI
Instruction set architecture	Instruction count Clock rate CPI

Speedup Techniques

Feature	Objective
Cache Memory	To provide the CPU with faster access to instruction and data.
Pipelined Processing	To increase performance by allowing the processing of several instructions to be partially overlapped.
Superscalar Processing	To increase performance by allowing several instructions to be processed in parallel (full overlapping).

Sequential and Pipelined Processing