Performance

Dr. Prasenjit Chanak

Department of Computer Science and Engineering
Indian Institute of Technology (BHU), Varanasi
UP, 221005

Defining Performance

- When we say one computer has better performance than another, what do we mean?
- If you were running a program on two different desktop computers, you'd say that the faster one is the desktop computer that gets the job done first.
- If you were running a datacenter that had several servers running jobs submitted by many users, you'd say that the faster computer was the one that completed the most jobs during a day.
- As an individual computer user, you are interested in reducing **response time**—the time between the start and completion of a task—also referred to as execution time

Defining Performance

- **Response time** also called **execution time**: The total time required for the computer to complete a task, including disk accesses, memory accesses, I/O activities, operating system overhead, CPU execution time, and so on
- Datacenter managers are often interested in increasing **throughput** or **bandwidth**—the total amount of work done in a given time
- **Throughput** Also called **bandwidth**: Another measure of performance, it is the number of tasks completed per unit time

Throughput and Response Time

- Do the following changes to a computer system increase throughput, decrease response time, or both?
  - Replacing the processor in a computer with a faster version
  - Adding additional processors to a system that uses multiple processors for separate tasks—for example, searching the web
- Decreasing response time almost always improves throughput
- In case 1, both response time and throughput are improved
- In case 2, no one task gets work done faster, so only throughput increases

Contd.

- To maximize performance, we want to minimize response time or execution time for some task
- Performance and execution time for a computer X:

$$\text{Performance}_X = \frac{1}{\text{Execution time}_X}$$

- This means that for two computers X and Y, if the performance of X is greater than the performance of Y

$$\text{Performance}_X > \text{Performance}_Y$$
$$\frac{1}{\text{Execution time}_X} > \frac{1}{\text{Execution time}_Y}$$
$$\text{Execution time}_Y > \text{Execution time}_X$$

- The execution time on Y is longer than that on X, if X is faster than Y.

Contd.

- In discussing a computer design, we oft en want to relate the performance of two different computers quantitatively. We will use the phrase “X is *n* times faster than Y”—or equivalently “X is *n* times as fast as Y”—to mean

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

- If X is *n* times as fast as Y, then the execution time on Y is *n* times as long as it is on X:

$$\frac{\text{Performance}_X}{\text{Performance}_Y} = \frac{\text{Execution time}_Y}{\text{Execution time}_X} = n$$

Measuring Performance

- Program *execution time* is measured in seconds per program
- The most straightforward definition of time is called *wall clock time*, *response time*, or *elapsed time*. These terms mean the total time to complete a task, including disk accesses, memory accesses, *input/output* (I/O) activities, operating system overhead—everything
- **CPU execution time** also called **CPU time**: The actual time the CPU spends computing for a specific task
- **User CPU time:** The CPU time spent in a program itself
- **System CPU time:** The CPU time spent in the operating system performing tasks on behalf of the program

Clock Cycle

- In particular, computer designers may want to think about a computer by using a measure that relates to how fast the hardware can perform basic functions. Almost all computers are constructed using a clock that determines when events take place in the hardware
- These discrete time intervals are called **clock cycles**
- **Clock cycle** also called **tick**, **clock tick**, **clock period**, **clock**, or **cycle**: The time for one clock period, usually of the processor clock, which runs at a constant rate.
- **Clock period:** The length of each clock cycle

Contd.

- A simple formula relates the most basic metrics (clock cycles and clock cycle time) to CPU time:

$$\begin{array}{c}\text{CPU execution time}\\ \text{for a program}\end{array} = \begin{array}{c}\text{CPU clock cycles}\\ \text{for a program}\end{array} \times \text{Clock cycle time}$$

- Alternatively, because clock rate and clock cycle time are inverses

$$\begin{array}{c}\text{CPU execution time}\\ \text{for a program}\end{array} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

- This formula makes it clear that the hardware designer can improve performance by reducing the number of clock cycles required for a program or the length of the clock cycle