

## Module 1: Introduction to Parallel and Distributed Computing

**Parallel Computing** refers to the use of **multiple processors** or computing resources to solve computing problems simultaneously. Allows for the **simultaneous execution of multiple instructions** or tasks, thus significantly improving the speed and efficiency.

**Distributed Computing** refers to a computing model in which a **group of interconnected computers work together** to solve a computing problem. **Distributes the workload across multiple machines**, often geographically dispersed, and coordinates their activities to achieve a common goal.

In **Traditional Computing**, a single processor executes instructions sequentially. A **single computer** performs all the processing.

### Difference between Parallel and Distributed Computing

**Parallel Computing** is commonly used for computers with **shared processors**.

**Distributed Computing** is commonly used in **networking** wherein **more than one computers** are shared to complete a task.

### Types of Parallel Computing

#### 1. Instruction-level Parallelism

- This involves executing multiple instructions simultaneously.

##### a. Pipelining

- Instructions are divided into stages, and multiple instructions can be in different stages of execution simultaneously.

##### b. Superscalar Execution

- Multiple execution units are employed, allowing for parallel execution of independent instructions

#### c. Vectorization

- It is a technique used to exploit data-level parallelism by performing the same operation on multiple data elements simultaneously.

#### 2. Thread-level Parallelism

- it involves dividing a program into multiple threads (smaller units of execution) that can run independently

#### 3. Task-level Parallelism

- A problem is divided into multiple tasks or subtasks that can be executed independently.

### Benefits of Parallel Computing

#### 1. Increased Performance

- By harnessing multiple processors, parallel computing can provide significant speedup and improved performance compared to sequential processing.

#### 2. Large-scale Problem Solving

- It enables the processing of vast amounts of data and the exploration of complex models and simulations.

#### 3. Resource Utilization

- Parallel computing makes efficient use of computing resources by utilizing idle processors for executing independent tasks concurrently. It maximizes the utilization of available hardware, leading to improved efficiency and cost-effectiveness.

#### 4. Flexibility

- Parallel computing offers flexibility in terms of hardware choices. It can leverage various

architectures, such as multi-core processors, distributed systems, and GPU accelerators, to meet the specific requirements of different applications.