ECS103 Prelim Period • Parallel and Distributed Computing

Overview and Introduction

We have witnessed the technology industry evolve a great deal over the years.

Earlier computer systems could complete only one task at a time.

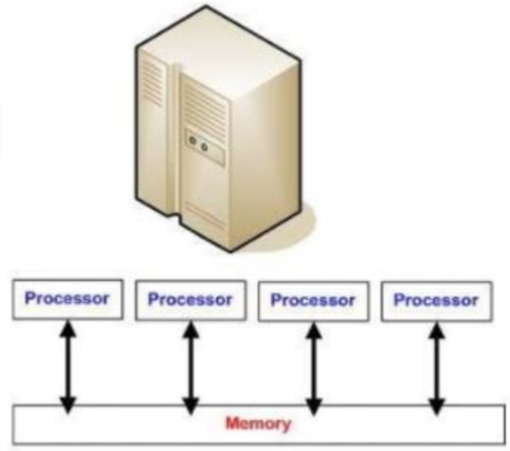
Today, we multitask on our computers like never before.

With improving technology, even the problem handling expectations from computers has risen.

This has given rise to many computing methodologies – **parallel computing** and **distributed computing** are two of them.

Parallel & Distributed Computing

Is a model that divides a task into multiple sub-tasks and executes them simultaneously to increase the speed and efficiency.



In parallel computing multiple processors performs multiple tasks assigned to them simultaneously.

Memory in parallel systems can either be shared or distributed.

Parallel computing provides concurrency and saves time and money.

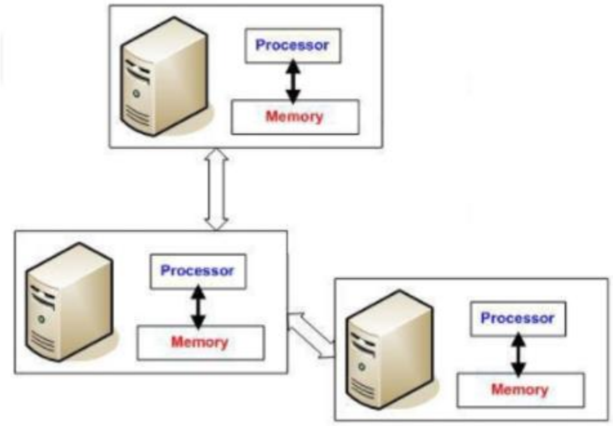
Is a field that studies distributed systems.

Distributed systems are systems that have multiple computers located in different locations.

In distributed computing we have multiple autonomous computers which seems to the user as single system.

In distributed systems there is no shared memory and computers communicate with each other through message passing.

In distributed computing a single task is divided among different computers.



ECS103 • Next Lesson

Difference between Parallel Computing and Distributed Computing

Parallel Computing	Distributed Computing
Dependency Between Process	
Many operations are performed simultaneously	System components are located at different locations
Number of Computer Systems Involved	
Single computer is required	Uses multiple computers
Usage	
Multiple processors perform multiple operations	Multiple computers perform multiple operations
Resource Sharing	
It may have shared or distributed memory	It have only distributed memory
Tasks	
Processors communicate with each other through bus	Computer communicate with each other through message passing.
Scalability	
Improves the system performance	Improves system scalability, fault tolerance and resource sharing capabilities

What Are They Used?

Parallel computing is often used in places requiring **higher** and **faster processing power**.

For example, supercomputers.
Since there are no lags in the passing of messages, these systems have high speed and efficiency.

Distributed computing is used when computers are **located** at **different geographical locations**.

In these scenarios, speed is generally not a crucial matter.
They are the preferred choice when scalability is required.

ECS103 • Next Lesson

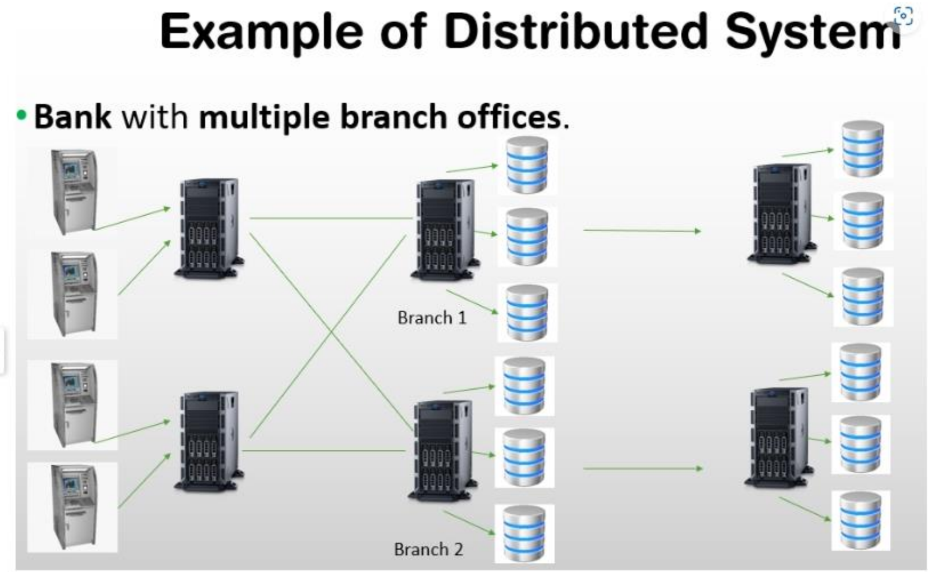
Distributed Computing

Is a **collection** of **autonomous computers** linked by a **computer network** and **equipped** with **distributed system software**.

Distributed System Software enables **computer** to **coordinate** their **activities** and to **share resources** of **system hardware**, **software** and **data**.

Application of Distributed System

Finance and Commerce
E-commerce (amazon, e-bay, online banking)
Information society
Search engine, Wikipedia, social networking
Creative industry and entertainment
Online gaming, music, YouTube
Healthcare
Online patient record, health informatics
Education
E-learning
Transport and Logistics
GPS, Google Maps



Example of Distributed System

- **Bank** with **multiple branch offices**.

Requirement of Distributed System

Security and Reliability

Consistency of replicated data

Concurrent transaction

Fault tolerance

Distributed Computing Models

Fundamental Models – Based on some fundamental's properties	**Architectural Models – Based on Architectural style**
Interaction model	Client-server model
Failure model	Peer-to-peer model
Security model	

Interaction model
Computation occurs within processes

The **process interact** by **passing messages resulting** in:

Communication (information flow)

Coordination (synchronization and ordering of activities) between processes.
Interaction model reflects the facts that communication takes place with delays.

Performance of **communication channels**

Latency - the **delay between** the **sending** of a **message** by **one process** and its **receipt** by **another** is **referred** to as **latency**.

Bandwidth - is the total amount of **information** that can be **transmitted over it** in a **given time**.

Jitter - is the **variation** in the **time taken** to **deliver** a **series** of **messages**. This is **relevant** to **real-time** and **multimedia traffic**.

Failure model

Failure model defines and **classifies** the **faults**.
It is important to understand the kinds of **failures** that may occur in a **system**.

1. **Process Crash**

A process halts and remains halted.
Other processes may not be able to detect this state.

2. **Process Fail-Stop**

A process halts and remains halted.
Other processes ca detect that the process has filed.

3. **Omission**

A message inserted in an outgoing message buffer never arrives at the other ends incoming message buffer.

4. **Arbitrary Failure**

Process/ channel exhibits messages at arbitrary behavior.
It may send/ transmit arbitrary messages at arbitrary time.

5. **Timing Failure**

Clock drift exceeds allowable bounds.

Security model

There are **several potential threats** a **system designer** need to aware of:

1. **Threats to processes**

- an attacker send a request or response using false identity.

2. **Threats to communication channels**

- an attacker may listen to message and save

3. **Denial of service**

- an attacker may overload a server by making excessive request.

ECS103 • Next Lesson

Client-server model

In a typical application, the server is **concurrent** and can **handle several clients simultaneously**.

Servers may in **turn** be **clients** of **other servers**.

Categories of Client-Server Computing

There are four main categories of client-server computing:

1. **One-Tier architecture**: **consists** of a **simple program running** on a **single computer without requiring access** to the **network**.
2. **Two-Tier architecture**: **consists** of the **client**, the **server**, and the **protocol** that **links** the **two tiers**.
3. **Three-Tier architecture**: consists of a **presentation tier**, which is the **User Interface layer**, the **application tier**, which is the **service layer** that **performs detailed processing**, and the **data tier**, **which consists** of a **database server** that **stores information**.
4. **N-Tier architecture**: **divides** an **application** into **logical layers**, which **separate responsibilities** and **manage dependencies**, and **physical tiers**, which **run** on **separate machines**, improve **scalability**, and add **latency** from the **additional network communication**.

ECS103 • Next Lesson

Peer-to-Peer model

P2P model does not **distinguish between client/ server**, **instead each node** can **either** be a **client** and **servers depending** on **whether** the **node** is **requesting** and **providing** the **services**.