**LESSON 1:**
**Introduction to Parallel Computing**

## Parallel Computing

- **Multiple processors perform multiple tasks assigned to them simultaneously**.
- **Memory** can either be **shared** or **distributed**.
- Provides concurrency and saves time and money.

  **Concurrency** – is the execution of multiple instructions sequenced at the same time.

- The use of multiple processing elements simultaneously to solve any problem.

  **Serial Computing** – the sequential execution of all parts of the same task on a single processor.

## Key Concepts:

1. **Parallelism** – is the fundamental idea of **breaking down a problem into small parts** and **executing** them **concurrently**.

   **Types of Parallelism:**
   - **Task Parallelism** – different processors or cores work on different tasks or sub-problems.
   - **Data Parallelism** – the same operation is applied to multiple pieces of data simultaneously.
2. **Shared Memory** – all processors share a common memory. **Distributed Memory** – each processor has its own memory.
3. **Parallel Programming Models** – parallel computing often involves the use of specific programming models and libraries, such as OpenMPI (Message Passing Interface), and CUDA (for GPUs), to harness the power of parallel hardware.

## Distributed Computing

- **Multiple autonomous computers** which seem to the user as single system.
- There is no shared memory and computers communicate with each other through message passing.
- A single task is divided among different computers.

## Key Concepts:

1. **Distributed Systems** – consist of **multiple nodes connected through a network**. (can be small clusters to massive data)
2. **Concurrency and Consistency** – managing concurrent access to shared resources and ensuring data consistency **are critical challenges in distributed computing**.
3. **Scalability** – designed to scale horizontally by **adding more nodes to handle increased workloads**.
4. **Fault Tolerance** – often incorporate mechanisms to **handle node failures** and **ensure uninterrupted operation**.
5. **Distributed Algorithms** – **enable nodes** to **coordinate their actions** and **make collective decisions**.
6. **Message Passing** – communication between nodes in distributed systems typically **relies on message passing protocols**, such as **HTTP**, **RPC** (Remote Procedure Call), and **message queues**.

## Difference between Parallel Computing and Distributed Computing

| PARALLEL COMPUTING | DISTRIBUTED COMPUTING |
|---|---|
| Many operations are performed simultaneously | System components are located at different location |
| Single computer is required | Uses multiple computers |
| Multiple processors perform multiple operations | Multiple computers perform multiple operation |
| It may have shared or distributed memory | It has only distributed memory |
| Processors communicate with each other through bus | Computers communicate with each other through message passing |
| Improves the system performance | Improves system scalability, fault tolerance and resource sharing capabilities |

## Advantages of Parallel Computing over Serial Computing

1. Saves time and money.

2. Impractical to solve larger problems on Serial Computing
3. Take advantage of non-local resources when the local resources are finite.
4. Serial Computing 'wastes' the potential computing power, thus Parallel Computing makes better work of hardware.

**Why Parallel Computing?**

- The whole real world runs in a dynamic nature.
  **Ex**. Many things happen at the same time.
- Real-world data needs more dynamic simulation and modeling, and for achieving the same, parallel computing is the key.
- Parallel computing provides concurrency and saves time and money.
- Complex, large datasets and their management can be organized only and only using a parallel computing approach.
- Ensures the effective utilization of the resources.
- Impractical to implement real-time systems using serial computing.

**Application of Parallel Computing:**

- Database and Data Mining
- Real-time simulation of systems - Science and Engineering
- Advanced graphics, augmented reality and virtual reality

**Limitation of Parallel Computing:**

- It addresses such as communication and synchronization between multiple sub-tasks and processes which is difficult to achieve.
- The algorithms must be managed in such a way that they can be handled in a parallel mechanism.
- The algorithms or program must have low coupling and high cohesion. But it's difficult to create such programs.
- More technically skilled and expert programmers can code a parallelismbased program well.

**GROUP 1: Key Characteristics of Parallel Computing**

1. **Concurrency** - Parallel computing provides concurrency and saves time and money. It allows for the effective utilization of resources, ensuring that hardware is used effectively.
2. **Speedup** – reduces the time required to solve a problem by dividing it into smaller parts that can be solved simultaneously.
3. **Scalability** – Parallel computing can scale to handle larger problems by adding more processors.
4. **Task Parallelism** – a characteristic of a parallel program that entirely different calculations can be performed on either the same or different sets of data.
5. **Data Parallelism** - a characteristic of a parallel program where instructions from a single stream operate concurrently on several data.
6. **Mapping** - Mapping in parallel computing is used to solve embarrassing parallel problems by applying a simple operation to all elements of a sequence without requiring communication between the subtasks.

**GROUP 2: Key Characteristics of Distributed Computing**

1. **Openness**
2. **Concurrency** – shared resources
3. **Scalability** – horizontal scaling
4. **Fault Tolerance**
5. **Transparency**
6. **Heterogeneity**

**GROUP 3: Changes in Parallel Computing**

**19th Century.**

- **The Analytical Engine** - The origins of true **MIMD (Multiple Instruction, Multiple Data) parallelism** can be traced back to **Luigi Federico Menabrea**'s work on Charles Babbage's Analytic Engine, which dates to the 19th century.

- **(1957) Gamma 60 - Compagnie des Machines Bull** introduced the Gamma 60 computer architecture, which was specifically designed for parallelism. It utilized a **fork-join**

**model** and a "**Program Distributor**" to coordinate data between independent processing units connected to central memory.

- **(1958) Early Discussions on Parallel Programming** – In April, **Stanley Gill (Ferranti)** discussed **parallel programming** and the need for branching and waiting. Also in the same year, IBM researchers **John Cocke** and **Daniel Slotnick** discussed the use of **parallelism in numerical calculations**.

- **(1962) Burroughs D825** - Burroughs Corporation introduced the **D825**, a **four-processor computer** that accessed up to **16 memory modules through a crossbar switch**.

- **(1967) Amdahl and Slotnick Debate** - Amdahl and Slotnick published a debate about the **feasibility of parallel processing** at an **American Federation of Information Processing Societies Conference**. It was during this debate that Amdahl's law was coined to define **the limit of speed-up due to parallelism**.

- **(1969) Honeywell Multics** - Honeywell introduced its **first Multics system**, a **symmetric multiprocessor system** capable of running up to **eight processors in parallel**.

**1977 – Present**

- **(1970) Emergence of Supercomputers**
  During the late 1970s and into the 1980s, supercomputers like the **Cray-1 and Cray-2** became prominent, **emphasizing parallel processing for high-performance computing**. o C.mmp Project
  o SIMD Parallel Computers

- **1972** o **ILLIAC IV**

- **1976** o **ILLIAC IV Completion**

- **1980's Connection Machines -** In the mid-1980s, **Thinking Machines Corporation** developed the **Connection Machine series**, which featured **massively parallel architectures for scientific computing**.

- **1984** o **Synapse N+1**

- **1990's Beowulf Clusters** - In the 1990s, Beowulf clusters, a form of **commodity-based parallel computing**, gained popularity. These clusters used off-the-shelf hardware and opensource software to create costeffective parallel computing systems.

- **2000's Multi-Core Processors** - the **shift** to **multi-core processors** in **personal computers** and **servers** became a common practice, leading to a **widespread adoption of parallelism in software development**.

- **2010's GPGPU Computing** - **GeneralPurpose Graphics Processing Unit (GPGPU)** computing became prevalent. Graphics cards were repurposed for general parallel computing tasks due to their highly parallel architecture.

- **2019 Exascale Computing** - the concept of exascale computing gained prominence. **Supercomputers** capable of **performing one exaFLOP** (a **quintillion floating-point operations per second**) were under development, showcasing the continued advancement in parallel processing capabilities.

- **2020's Quantum Computing** - While not fully parallel in the traditional sense, quantum computing has been a significant development. Quantum computers **leverage quantum parallelism** to **solve complex problems** faster than classical computers.

- **Present AI and Deep Learning**
  Parallelism plays a vital role in the field of artificial intelligence and deep learning. **Training neural networks** often involves distributed computing across **multiple GPUs or TPUs**, leading to **significant advancements in machine learning**.

## GROUP 4: Benefits of Parallel Computing

1. **Parallel Computing Models the Real World**
   - Real-worlds events are non-linear and lack a sequential pattern.

- Crucial for simultaneous data processing.
- Applications span weather, transportation, banking, industry, agriculture, seas, ice caps, and healthcare.

2. **Saves Time**
   - Parallel computing can lead to wasteful tasks for fast processors.
   - It's akin to inefficiently transporting 20 oranges one by one in a Ferrari from Maine to Boston, highlighting the need for more efficient grouping and processing.

3. **Saves Money**
   - Parallel computing reduces costs through time savings.
   - Efficiency gains may seem small on a small scale but become significant when scaling up systems with billions of processes, such as bank software.
   - These efficiency improvements result in substantial cost savings.

4. **Solve More Complex/Larger Problems**
   - Computing is advancing rapidly, with web apps handling millions of transactions per second through AI and big data.
   - Grand challenges like cybersecurity and affordable solar energy require immense processing power in the petaFLOPS range.
   - Parallel computing is the key to achieving these goals more rapidly.

5. **Leverage Remote Resources**
   - Humans generate an immense amount of daily data, totaling 2.5 quintillion bytes.
   - Parallel processing, using multiple computers with multiple cores, enables the efficient analysis of this vast amount of real-time data.
   - It surpasses the capabilities of serial processors working in isolation.

## GROUP 5: Changes in Distributed Computing

- **1960s-1970s: Early Concepts and Research** o This era set the theoretical foundation for future networking and communication advancements.

- **1980s: Client-Server Architecture** o The era saw the ascent of clientserver architecture, with clients requesting services or resources from centralized servers.

- **1990s: Grid and Cluster Computing** o Grid computing used networked computers in the 1990s to solve complex computational problems.

- **Late 1990s-2000s: Rise of Cloud Maturity and Cloud Computing** o In the late 1990s and early 2000s, with **Application Service Providers** (ASPs) selling software and services online, the concept of cloud computing emerged.

- **2010s: Cloud Maturity and Microservices** o Cloud computing advanced significantly, providing various services like PaaS and SaaS. Additionally, containerization with technologies like Docker and orchestration using Kubernetes gained popularity.

- **2010s-2020s: Edge and Serverless Computing**
  o In the 2010s, the focus shifted toward bringing computation closer to data sources.

- **Present and Future: Quantum Computing and Ethical Challenges** o In the present era, ongoing developments in distributed machine learning, artificial intelligence, and quantum computing are shaping the landscape of distributed computing.

## GROUP 6: Benefits of Distributed Computing

1. **Performance**
2. **Resilience and Redundancy**
3. **Efficiency**
4. **Scalability**
5. **Cost-effectiveness**

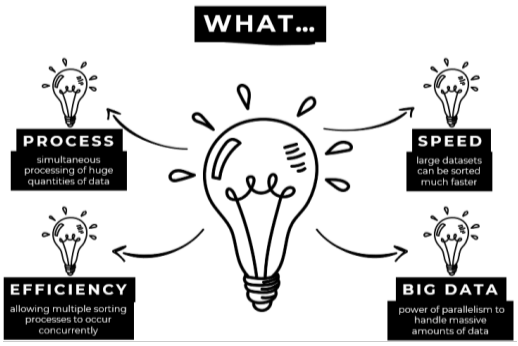
## GROUP 7: Parallel Programming Task in Shared Memory System

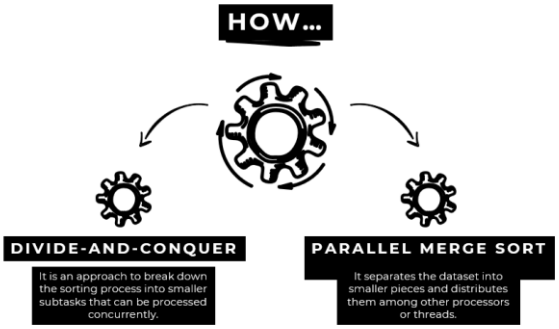
| Parallel Programming | Shared Memory System |
|---|---|
| • Method of computation<br>• Useful for solving large problems<br>• Widely used in high-performance computing | • refers to memory accessible simultaneously by multiple programs.<br>• allow processors to share a common view of data |

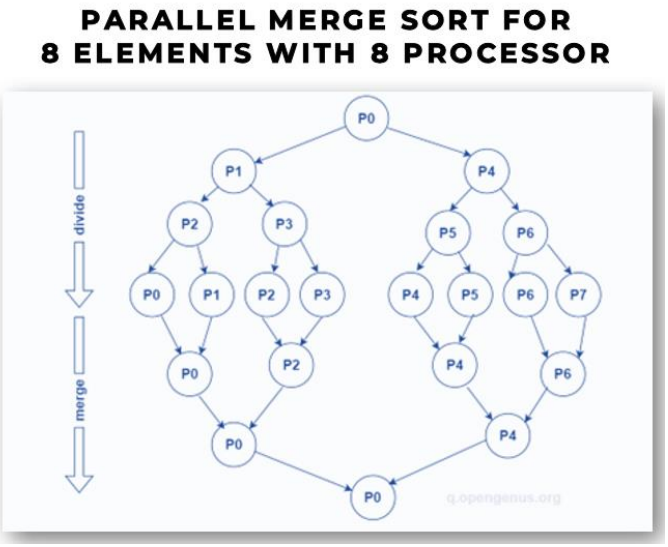
**Steps on how to solve parallel programming tasks in a shared memory system**

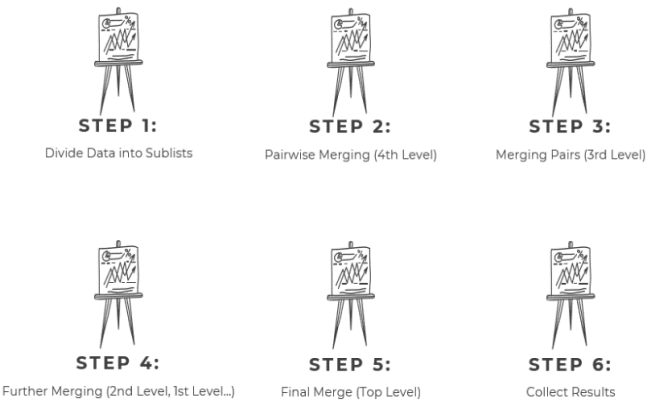
1. Identify Parallelizable Tasks
2. Establish Shared Memory
3. Allocate tasks to processors.
4. Synchronize Task Execution
5. Collect Results

## GROUP 8: Steps on how to solve Parallel Algorithm and Sorting.



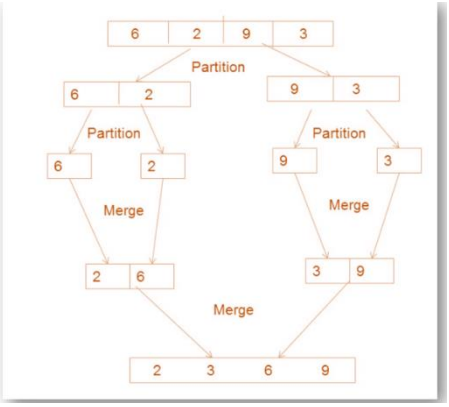




**EXAMPLE:**

Parallel mergesort has a time complexity of O(n log n) in the worst, average, and best cases.



1. The use of multiple processing elements simultaneously to solve the problem. **PARALLEL COMPUTING**

2. The execution of multiple instructions sequenced at the same time. **CONCURRENCY**

3. The fundamental idea of breaking down a problem into small parts and executing them concurrently. **PARALLELISM**

_ 4-5. Types of Parallelism **Task** & **Data** Parallelism

6. In a single task is divided among different computers. **DISTRIBUTED COMPUTING**

7. The sequential execution of all parts of the same task on a single processor. **SERIAL COMPUTING**

8. In parallel computing, the system components are located at different locations. (TRUE or FALSE) **FALSE**

9. Computers communicate with each other through message passing in distributed computing. (TRUE or FALSE) **TRUE**

10. One of the advantages of parallel computing over serial computing is that it saves time and money as many resources working together will reduce the time and cut potential costs. (TRUE or FALSE) **TRUE**

11. Parallel computing can_____ to handle large problems by adding more processors. **SCALE**

12. In April 1958, branching and waiting, _____ discussed parallel programming and the need for branching and waiting. **STANLEY GILL**

13. In the _ Beowulf clusters, a form of commodity-based parallel computing, gained popularity. **1990s**

14. GPGPU stands for **GENERAL PURPOSE GRAPHICS PROCESSING UNIT**

15-16. Parallelism plays a vital role in the field of and **AI** AND **MACHINE/DEEP LEARNING**

17. Parallel computing reduces costs through time-saving. (TRUE or FALSE) **TRUE**

18. Distributed computing is the key to achieving the goals of solving more complex and larger problems. (TRUE or FALSE) **FALSE**

19. ASPs stands for **APPLICATION SERVICE PROVIDERS**

20. In the ____, the focus shifted toward bringing computation closer to data resources. **2010s**

21. In the present era, ongoing developments in distributed machine learning, artificial intelligence, and _____. are shaping the landscape of distributed computing. **QUANTUM COMPUTING**

_ 22. In distributed computing clusters are scalable by adding new hardware when needed. (TRUE or FALSE) **TRUE**

23. Distributed computing can help improve efficiency by having each computer in a cluster handle different parts of a task simultaneously. (TRUE or FALSE) **FALSE**

24. Parallel computing often involves the use of specific programming models and libraries (TRUE or FALSE) **TRUE**

25. This system consists of multiple nodes connected through a network. **DISTRIBUTED SYSTEM**