CSE3009 - Parallel and Distributed Computing

Course Type: LTP Credits: 4

Prepared by
Dr Komarasamy G
Senior Associate Professor
School of Computing Science and Engineering
VIT Bhopal University

Course Objectives

1. To provide contemporary knowledge to students in parallel and distributed environment.
2. To provide students with abilities to analyze and design parallel and distributed applications.
3. In the development of parallel and distributed applications, apply core computer science concepts and algorithms.
4. To illustrate middleware technologies to support distributed applications.
5. To identify Distributed and parallel programs to improve performance and reliability.

Course Outcomes

Students will be able to

- CO1: Analyze parallel and distributed system using the principles and concepts.[KL4]
- CO2: Apply parallelize problems for load balancing. [KL3]
- CO3 : Explain the challenges and opportunities that parallel and distributed systems present. [KL2]
- CO4: Explain middleware technologies. RPC, RMI, and object-based[KL2]
- CO5: Illustrate middleware technologies to support distributed applications. [KL2]
- CO6: Identify Distributed and parallel programs to improve performance and reliability. [KL2]

Unit-1

Parallelism Fundamentals – Key Concepts and Challenges – Overview of Parallel computing – Flynn's Taxonomy – Multi-Core Processors – Shared vs Distributed memory.

Performance of Parallel Computers, Performance Metrics for Processors, Parallel Programming Models, Parallel Algorithms.

Parallelism Fundamentals

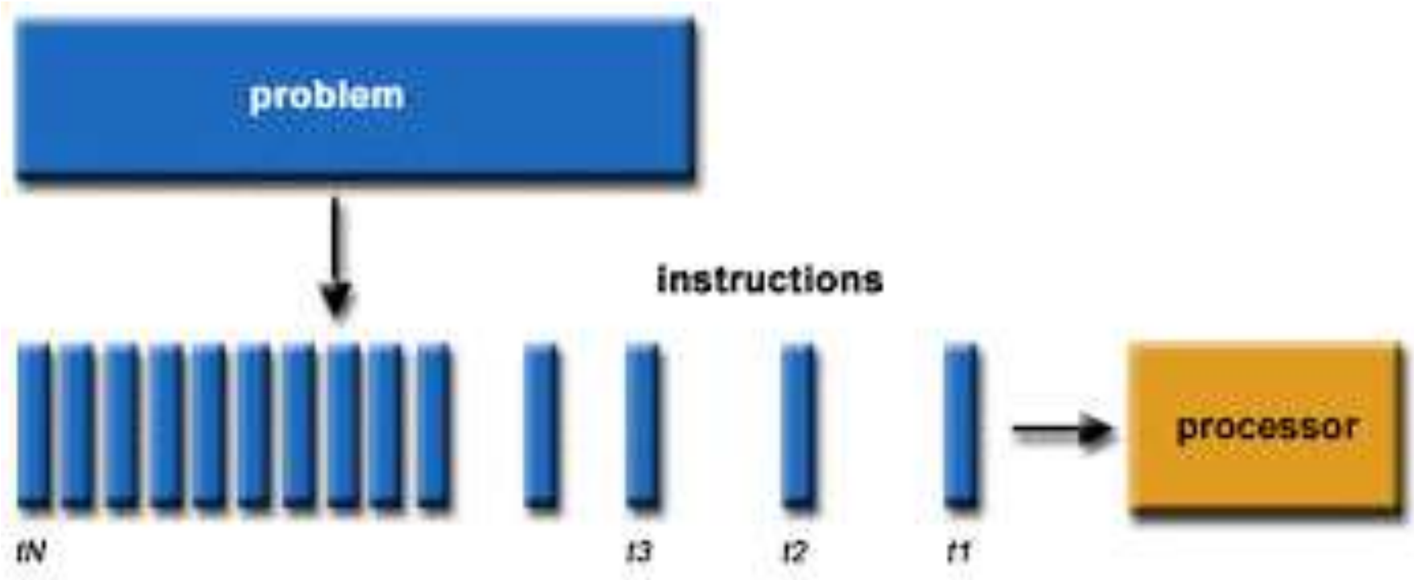
- **Conventional architectures** – **consists of a processor, memory system, and the data path.**
- Each of these components present significant performance bottlenecks.
- Parallelism addresses each of these components insignificant ways.
- Different applications utilize different aspects of parallelism - e.g., data intensive applications utilize high aggregate throughput, server applications utilize high aggregate network bandwidth, and scientific applications typically utilize high processing and memory system performance.
- It is important to understand each of these performance bottleneck.

Parallelism Fundamentals

- **Parallel computing** -> **process of executing several processors an application or computation simultaneously.**

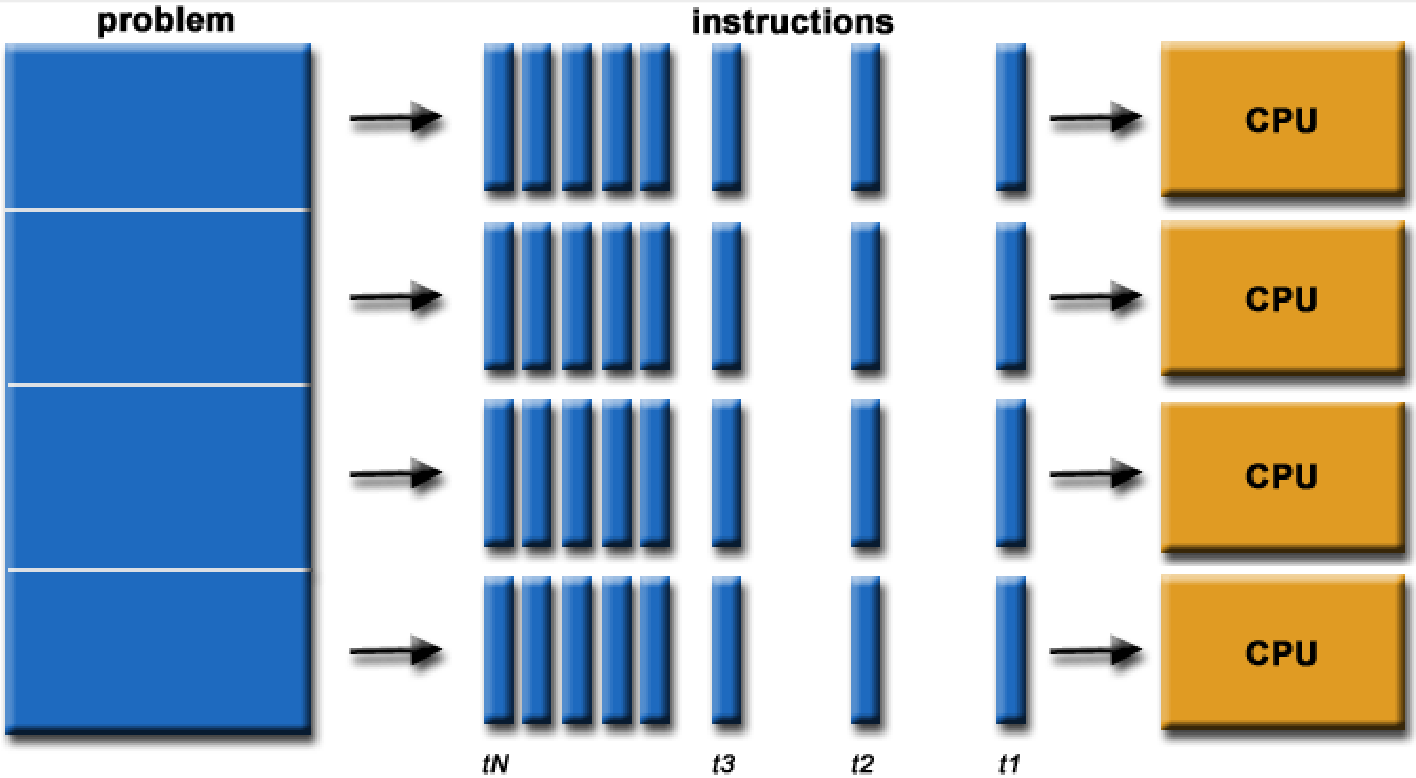
- Generally, it is a kind of computing architecture where the large problems break into independent, smaller, usually similar parts that can be processed in one go.

Parallelism Fundamentals



- **Traditionally software has been written for <u>serial computations</u>:**
- **To be run on a single computer having a single Central Processing Unit (CPU)**
- **A problem is broken into a discrete set of instructions**
  - **Instructions are executed one after another**
  - **Only one instruction can be executed at any moment in time**

Parallelism Fundamentals



Parallel computing is the simultaneous use of multiple compute resources to solve a computational problem.

- **To be run using multiple CPUs**
- **A problem is broken into discrete parts that can be solved concurrently**
- **Each part is further broken down to a series of instructions**
- **Instructions from each part execute simultaneously on different CPUs**

Parallelism Fundamentals

- Parallel computing also helps in **faster application processing and task resolution by increasing the available computation power of systems.**
- The parallel computing principles are used by most supercomputers employ to operate.
- The operational scenarios that need massive processing power or computation, generally, parallel processing is commonly used there.
- The earliest computer software is **written for serial computation** as they are able to execute a single instruction at one time, but parallel computing is different where it executes several processors an application or computation in one time.

Parallelism Fundamentals

- There are **many reasons** to use parallel computing, such as save **time and money, provide concurrency, solve larger problems**, etc.
- Furthermore, **parallel computing reduces complexity.**

In the real-life example of parallel computing,

- there are two queues to get a ticket of anything;
- if **two cashiers are giving tickets to 2 persons simultaneously,**
- **it helps to save time as well as reduce complexity.**

Key Concepts and Challenges

- **The compute resources can include:**
  - A single computer with multiple processors;
  - An arbitrary number of computers connected by a network;
  - A combination of both.
- **The computational problem usually demonstrates characteristics such as the ability to be:**
  - Broken apart into discrete pieces of work that can be solved simultaneously;
  - Execute multiple program instructions at any moment in time;
  - Solved in less time with multiple compute resources than with a single compute resource.

Types of parallel computing

1. Bit-level parallelism:

- The form of parallel computing in which every task is dependent on **processor word size.**
- In terms of performing a task on large-sized data, it **reduces the number of instructions** the processor must execute. There is a need to split the operation into series of instructions.
- **For example**,
- There is an **8-bit processor**, and you want to do an operation on **16-bit numbers**.
- First, it must operate the 8 lower-order bits and then the 8 higher-order bits.
- Therefore, two instructions are needed to execute the operation. The operation can be performed with one instruction by a 16-bit processor.

2. Instruction-level parallelism:

- In a single **CPU clock cycle**, the processor decides in instruction-level parallelism **how many instructions are implemented at the same time.** For each clock cycle phase, a processor in instruction-level parallelism can have the ability to address that is less than one instruction. The software approach in instruction-level parallelism functions on static parallelism, where the computer decides which instructions to execute simultaneously.

3. Task Parallelism:

- Task parallelism is the **form of parallelism in which the tasks are decomposed into subtasks.**
- Then, each subtask is allocated for execution. And, the execution of subtasks is performed concurrently by processors.

Applications of Parallel Computing

- **There are various applications of Parallel Computing, which are as follows:**
- One of the primary applications of parallel computing is **Databases and Data mining.**
- The real-time simulation of systems is another use of parallel computing.
- The technologies, such as Networked videos and Multimedia.
- Science and Engineering.
- Collaborative work environments.
- The concept of parallel computing is used by **augmented reality, advanced graphics, and virtual reality.**

Advantages of Parallel computing

- In parallel computing, more resources are used to complete the task that led to decrease the **time and cut possible costs.**
- Comparing with Serial Computing, parallel computing can solve **larger problems in a short time.**
- For simulating, modeling, and understanding complex, real-world phenomena, parallel computing is much appropriate while comparing with serial computing.
- There are multiple problems that are very large and may impractical or impossible to solve them on a single computer;
- It allows you to do several things in a time by using multiple computing resources.
- Parallel computing is suited for hardware as serial computing wastes the potential computing power.

Disadvantages of Parallel Computing

- It addresses Parallel architecture that can be **difficult to achieve.**
- In the case of clusters, better **cooling technologies are needed** in parallel computing.
- It requires the managed algorithms, which could be handled in the parallel mechanism.
- The multi-core architectures consume high power consumption.
- The parallel computing system needs **low coupling and high cohesion, which is difficult to create.**
- The code for a parallelism-based program can be done by the most technically skilled and expert programmers.

Disadvantages of Parallel Computing

- Although parallel computing helps you out to resolve computationally and the data-exhaustive issue with the help of using multiple processors, **sometimes it affects the conjunction of the system and some of our control algorithms** and does not provide good outcomes due to the parallel option.
- Due to synchronization, thread creation, data transfers, and more, the extra cost sometimes can be quite large; even it may be exceeding the gains because of parallelization.
- Moreover, for improving performance, the parallel computing system needs different code tweaking for different target architectures.

Overview or Why Parallel computing

- The whole real-world runs in **dynamic nature** i.e. many things happen at a certain time but at different places concurrently. This data is extensively huge to manage.
- 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 and using parallel computing's approach.**
- Ensures the effective utilization of the resources.
- The hardware is guaranteed to be used effectively whereas in serial computation only some part of the hardware was used and the rest rendered idle.
- Also, it is impractical to implement real-time systems using serial computing.

How does Computer Hardware Work

- **How does Computer Hardware Work|| 3D animation.**
- **https://youtu.be/d86ws7mQYIg?si=ZWP7OoRO9AyObOGI**