



PARALLEL AND DISTRIBUTED COMPUTING

M. ZAHID SARWAR

ASSISTANT PROFESSOR
COMPUTER SCIENCE DEPARTMENT

LECTURE # 01 INTRODUCTION

COURSE INTRODUCTION

OBJECTIVES

This lecture forms to understand the difference between performing computations in parallel (on multiple cores within one system) and in a distributed manner (across multiple networked computers).

Learning different ways to structure parallel and distributed programs, such as **data-parallel** (same instruction on different data) and **task-parallel** (different tasks on different processors).

CONTENTS

- Introduction to Parallel & Distributed Computing
- Advantages & Disadvantages of Parallel Computing
- Advantages & Disadvantages of Distributed Computing
- Difference between Parallel & Distributed Computing
- Use of Parallel & Distributed Computing
- Example of Parallel & Distributed Computing
- Applications
- Challenges
- Tools



PARALLEL COMPUTING

PARALLEL COMPUTING

- Parallel computing refers to the 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 parts.
- Usually similar parts that can be processed in one go.
- It is done by multiple CPUs communicating via shared memory, which combines results upon completion.
- It helps in performing large computations as it divides the large problem between more than one processor.

PARALLEL COMPUTING

- parallel computing uses multiple processors in a single computer to access shared memory.
- **How it works:**
 - Uses multiple processors or cores within a single machine to execute tasks simultaneously.
- **Memory:**
 - Processors share access to a common, single memory.
- **Communication:**
 - Information is exchanged through shared memory, which is very fast.
- **Best for:**
 - Tasks that require high performance and low latency within a single system, such as speeding up computations on a supercomputer.

PARALLEL COMPUTING EXAMPLES

- Examples of Parallel computing
 - Blockchains,
 - Smart phones,
 - Laptop computers,
 - Internet of Things,
 - Artificial intelligence and machine learning,
 - Space shuttle,
 - Supercomputers are the technologies that uses Parallel computing technology.

ADVANTAGES OF PARALLEL COMPUTING

- **Increased Speed:** In this technique, several calculations are executed concurrently hence reducing the time of computation required to complete large scale problems.
- **Efficient Use of Resources:** Takes full advantage of all the processing units it is equipped with hence making the best use of the machine's computational power.
- **Scalability:** Also the more processors built into the system, the more complex problems can be solved within a short time.
- **Improved Performance for Complex Tasks:** Best suited for activities which involve a large numerical calculation like, number simulation, scientific analysis and modeling and data processing.

DISADVANTAGES OF PARALLEL COMPUTING

- **Complexity in Programming:** Parallel writing programming that is used in organizing tasks in a parallel manner is even more difficult than that of serial programming.
- **Synchronization Issues:** Interaction of various processors when operating concurrently can become synchronized and result in problem areas on the overall communication.
- **Hardware Costs:** The implementation of parallel computing does probably involve the use of certain components such as multi-core processors which could possibly be costly than the normal systems.



DISTRIBUTED COMPUTING

DISTRIBUTED COMPUTING

- Distributed computing uses multiple, separate computers connected by a network that communicate via message passing.
- In distributed systems there is no shared memory .
- Usually similar parts that can be processed in one go.
- Computers communicate with each other through message passing.
- In distributed computing a single task is divided among different computers.

DISTRIBUTED COMPUTING

■ **How it works:**

- Uses multiple independent computers, often in different geographic locations, to work on a problem together.

■ **Memory:**

- Each computer has its own private memory (distributed memory).

■ **Communication:**

- Computers communicate by passing messages to each other over a network, which introduces more latency.

■ **Best for:**

- Applications that need large-scale data processing, scalability, and fault tolerance by distributing tasks across many machines.

DISTRIBUTED COMPUTING EXAMPLES

- Examples of Distributed computing
 - Artificial Intelligence and Machine Learning,
 - Scientific Research and High-Performance Computing, Financial Sectors,
 - Energy and Environment sectors,
 - Internet of Things,
 - Blockchain and Cryptocurrencies are the areas where distributed computing is used

ADVANTAGES OF DISTRIBUTED COMPUTING

- **Fault Tolerance:** The failure of one node means that this node is no longer part of the computations, but that is not fatal for the entire computation since there are other computers participating in the process thereby making the system more reliable.
- **Cost-Effective:** Builds upon existing hardware and has flexibility in utilizing commodity machines instead of the need to have expensive and specific processors for its use.
- **Scalability:** The distributed systems have the ability to scale and expand horizontally through the addition of more machines in the networks and therefore they can take on greater workloads and processes.
- **Geographic Distribution:** Distributed computing makes it possible to execute tasks at different points thereby eliminating latencies.

DISADVANTAGES OF DISTRIBUTED COMPUTING

- **Complexity in Management:** The task of managing a distributed system itself can be made more difficult since it may require dealing with the latency and/or failure of a network as well as issues related to synchronizing the information to be distributed.
- **Communication Overhead:** Inter node communication requirements can actually hinder the package transfer between nodes that are geographically distant and hence the overall performance is greatly compromised.
- **Security Concerns:** In general, distributed systems are less secure as compared to centralized system because distributed systems heavily depend on a network.

KEY DIFFERENCES

Key differences

Feature	Parallel Computing	Distributed Computing
Hardware	Tightly coupled processors within a single machine.	Loosely coupled computers connected by a network.
Memory	Shared memory.	Private/Distributed memory.
Communication	Shared memory access.	Message passing over a network.
Fault Tolerance	Less fault-tolerant; a single point of failure can affect the whole system.	More fault-tolerant; tasks can be rerouted to other machines if one fails.
Scalability	Scales by adding more processors to a single machine.	Scales by adding more computers to the network.

PARALLEL COMPUTING VS DISTRIBUTED COMPUTING

Parallel Computing Vs Distributed Computing:

S.NO	Parallel Computing	Distributed Computing
1.	Many operations are performed simultaneously	System components are located at different locations
2.	Single computer is required	Uses multiple computers
3.	Multiple processors perform multiple operations	Multiple computers perform multiple operations
4.	It may have shared or distributed memory	It have only distributed memory
5.	Processors communicate with each other through bus	Computer communicate with each other through message passing.
6.	Improves the system performance	Improves system scalability, fault tolerance and resource sharing capabilities

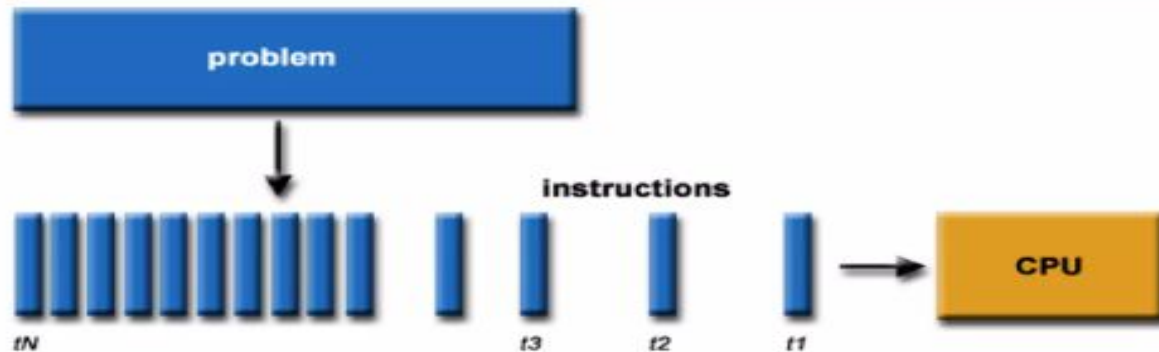
WHAT IS PARALLEL COMPUTING

What is Parallel Computing?

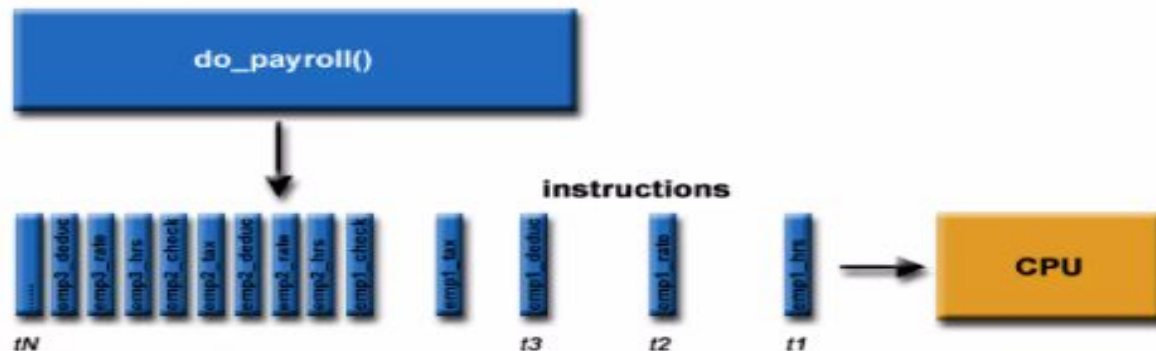
- ◆ Traditionally, software has been written for *serial* computation
 - To be run on a single computer having a single Central Processing Unit
 - A problem is broken into a discrete series of instructions
 - Instructions are executed one after another
 - Only one instruction may execute at any moment in time

WHAT IS PARALLEL COMPUTING CON....

What is Parallel Computing?



For example:



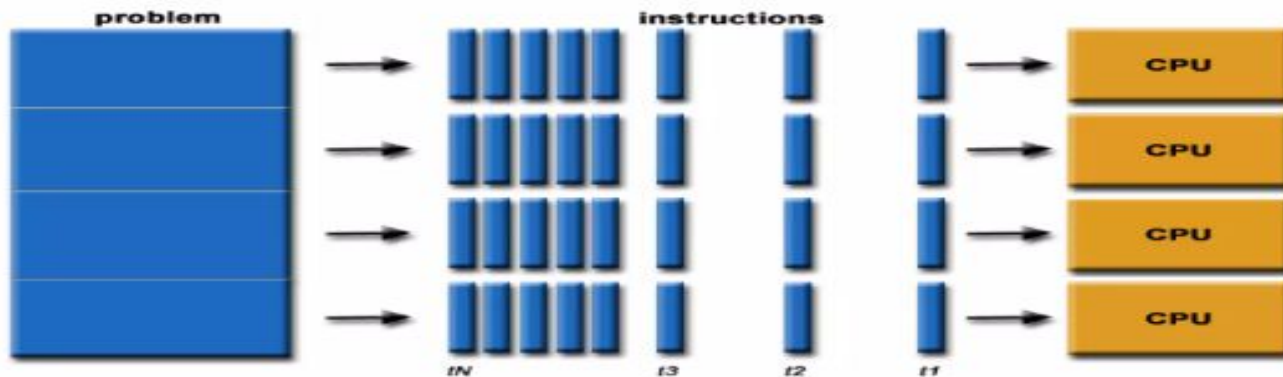
WHAT IS PARALLEL COMPUTING CON....

What is Parallel Computing?

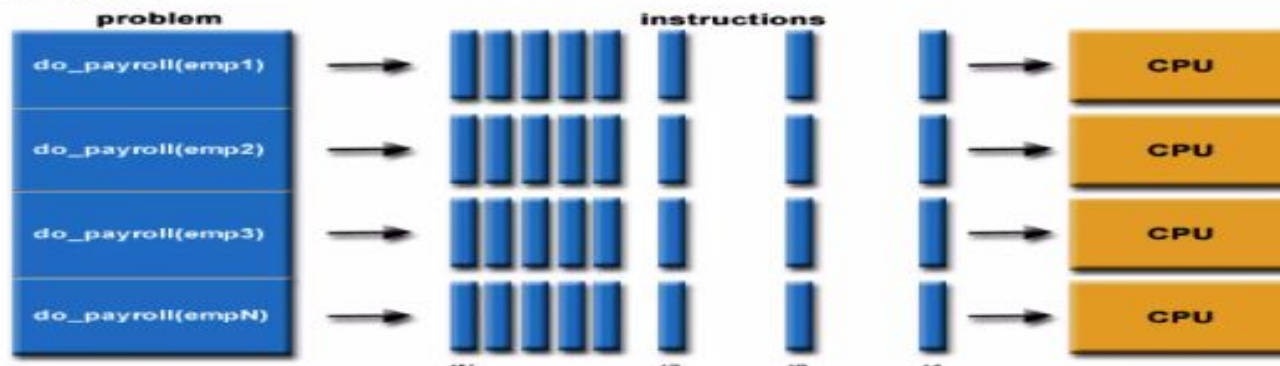
- ◆ *Parallel computing* is the **simultaneous use of multiple compute resources to solve a computational problem**
 - Accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others

WHAT IS PARALLEL COMPUTING CON....

What is Parallel Computing?



For example:



WHAT IS PARALLEL COMPUTING CON....

What is Parallel Computing?

- ◆ The computational problem should be:
 - Solved in less time with multiple compute resources than with a single compute resource

- ◆ The compute resources might be
 - A single computer with multiple processors
 - Several networked computers
 - A combination of both

WHAT IS PARALLEL COMPUTING CON....

What is Parallel Computing?

◆ The Real World is Massively Parallel

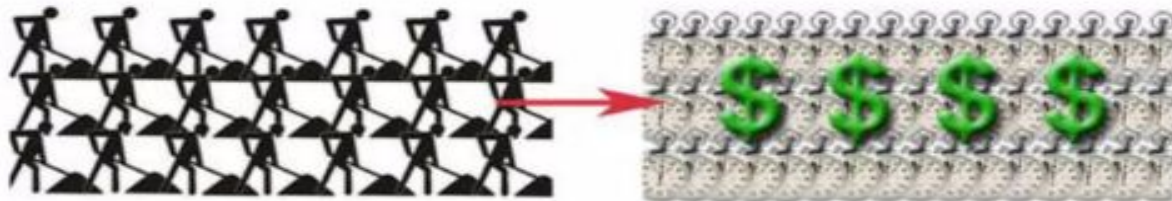
- In the natural world, **many complex, interrelated events are happening at the same time**, yet within a temporal sequence
- Compared to serial computing, parallel computing is much **better suited for modeling, simulating and understanding complex, real world phenomena**

WHY PARALLEL COMPUTING

Why Use Parallel Computing?

◆ Save time and money

- In theory, throwing more resources at a task will shorten its time to completion, with potential cost savings
- Parallel computers can be built from cheap, commodity components



WHY PARALLEL COMPUTING CON...

Why Use Parallel Computing?

◆ Solve larger problems

- Many problems are so large and complex that it is **impractical or impossible to solve them on a single computer**, especially given limited computer memory
 - en.wikipedia.org/wiki/Grand_Challenge problems requiring PetaFLOPS and PetaBytes of computing resources.
- Web search engines and databases processing millions of transactions per second

WHY PARALLEL COMPUTING CON...

Why Use Parallel Computing?

◆ Provide concurrency

- A single compute resource can only do one thing at a time.

Multiple computing resources can be doing many things simultaneously

- For example, the Access Grid (www.accessgrid.org) provides a global collaboration network where people from around the world can meet and conduct work "virtually"

WHY PARALLEL COMPUTING CON...

Why Use Parallel Computing?

◆ Use of non-local resources

- Using compute resources on a wide area network, or even the Internet when local compute resources are insufficient
- SETI@home (setiathome.berkeley.edu) over 1.3 million users, 3.4 million computers in nearly every country in the world.
Source: www.boincsynergy.com/stats/ (June, 2013).
- Folding@home (folding.stanford.edu) uses over 320,000 computers globally (June, 2013)

WHY PARALLEL COMPUTING CON...

Why Use Parallel Computing?

◆ Limits to serial computing

■ Transmission speeds

- The speed of a serial computer is directly dependent upon how fast data can move through hardware.
- Absolute limits are the speed of light and the transmission limit of copper wire
- Increasing speeds necessitate increasing proximity of processing elements

■ Limits to miniaturization

- Processor technology is allowing an increasing number of transistors to be placed on a chip
- However, even with molecular or atomic-level components, a limit will be reached on how small components can be

WHY PARALLEL COMPUTING CON...

Why Use Parallel Computing?

◆ Limits to serial computing

- Economic limitations
 - It is increasingly **expensive to make a single processor faster**
 - **Using a larger number of** moderately fast commodity **processors** to achieve the same or better performance **is less expensive**
- Current computer architectures are increasingly relying upon hardware level parallelism to improve performance
 - Multiple execution units
 - Pipelined instructions
 - Multi-core

TYPES OF PARALLEL COMPUTING

Types of Parallel Computing System

Bit Level Parallelism: Uses larger “*words*” which is fixed sized piece of data

Instruction Level Parallelism: allow processors to execute more than one instruction per clock cycle

Task-Level Parallelism: run computer code across multiple processors to run multiple tasks at the same time on same data.

TYPES OF DISTRIBUTED COMPUTING

Distributed Systems

There are two types of architecture in distributed systems

1. *General Purpose*

Including PC, Laptops etc.

2. *Special purpose*

Mainframe computers or super computers



TOOLS FOR PARALLEL & DISTRIBUTED COMPUTING

TOOLS FOR PARALLEL & DISTRIBUTED COMPUTING

Tools

- Hadoop
- Apache Spark
- MPI (Message Passing Interface)
- OpenMP (Open Multi-Processing)



TOOLS FOR PARALLEL & DISTRIBUTED COMPUTING

- There are several tools and frameworks available for parallel and distributed computing, catering to various programming languages and application domains.
- Some popular tools include:
- **MPI (Message Passing Interface):**
 - MPI is a standard specification for message-passing libraries used in parallel computing.
 - It provides a programming model for distributed memory systems and enables communication between parallel processes running on different nodes.

TOOLS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ **OpenMP (Open Multi-Processing):**

- OpenMP is an API that supports multi-platform shared-memory parallel programming in C, C++, and Fortran.
- It allows developers to parallelize loops, sections of code, and tasks across multiple threads within a single compute node.

■ **CUDA (Compute Unified Device Architecture):**

- CUDA is a parallel computing platform and programming model developed by NVIDIA for GPU-accelerated computing.
- It enables developers to harness the computational power of NVIDIA GPUs for parallel processing tasks, such as scientific simulations and deep learning.

TOOLS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Hadoop:**

- Hadoop is an open-source framework for distributed storage and processing of large datasets across clusters of commodity hardware.
- It includes components like Hadoop Distributed File System (HDFS) for storage and MapReduce for parallel processing.

■ **Apache Spark:**

- Spark is a fast and general-purpose distributed computing system that provides high-level APIs for in-memory data processing.
- It supports various programming languages like Java, Scala, Python, and R and offers libraries for stream processing, machine learning, and graph processing.

TOOLS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ **TensorFlow:**

- TensorFlow is an open-source machine learning framework developed by Google.
- It supports parallel and distributed training of machine learning models across multiple GPUs and CPUs, allowing for scalable model training and inference.

■ **Apache Kafka :**

- Kafka is a distributed streaming platform used for building real-time data pipelines and streaming applications.
- It enables high-throughput, fault-tolerant messaging between distributed systems and supports parallel processing of data streams.

TOOLS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ **MPI4Py:**

- MPI4Py is a Python binding for MPI, allowing Python developers to write parallel and distributed computing applications using the MPI standard.
- It provides Python interfaces for MPI functions and enables communication between Python processes running on different nodes.
- These are just a few examples of tools and frameworks for parallel and distributed computing.
- Depending on the specific requirements of your application and the programming language you're using, there are many other options available for harnessing the power of parallel and distributed computing.



APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING

APPLICATIONS

Applications of Distributed and Parallel Computing

Applications

- High-Performance Computing (HPC)
- Big Data Analytics
- Scientific Simulations
- Genomics and Bioinformatics
- Financial Modeling
- Internet of Things
- Social Media



APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING

- Parallel and distributed computing find applications across a wide range of domains, enabling the efficient processing of large-scale datasets and complex computational tasks.
- Some common applications include:
- **High-Performance Computing (HPC):**
 - Parallel computing is essential for achieving high performance in computationally intensive tasks such as weather forecasting, seismic analysis, and computational chemistry.
 - HPC clusters leverage parallel processing to tackle complex calculations and simulations efficiently.

APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Big Data Analytics:**

- Parallel and distributed computing are instrumental in analyzing massive datasets to extract insights and patterns.
- Applications include data mining, machine learning, and predictive analytics, where parallel processing enables the efficient training of models and the processing of large volumes of data.

APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ Scientific Simulations:

- Parallel computing is widely used in scientific simulations for tasks such as climate modeling, computational fluid dynamics, and molecular dynamics simulations.
- Distributed computing enables researchers to divide simulations into smaller tasks and run them across multiple nodes for faster computation.

APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Genomics and Bioinformatics:**

- Parallel and distributed computing are used in genomics and bioinformatics for tasks like DNA sequencing, sequence alignment, and protein structure prediction.
- Distributed computing platforms enable researchers to process and analyze large genomic datasets quickly and accurately..

APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Financial Modeling:**

- Parallel computing is employed in financial modeling for tasks like risk analysis, portfolio optimization, and algorithmic trading.
- Distributed computing platforms enable financial institutions to analyze market data in real time and make informed decisions based on complex models and algorithms.

APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ Internet of Things (IoT):

- Distributed computing is crucial for processing and analyzing data generated by IoT devices in real time.
- Applications include smart cities, industrial automation, and healthcare monitoring, where distributed computing platforms enable the aggregation, processing, and analysis of sensor data from diverse sources.

APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Cloud Computing:**

- Parallel and distributed computing form the backbone of cloud computing platforms, enabling scalable and on-demand access to computing resources.
- Cloud providers leverage distributed computing architectures to deliver services like infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS) to users worldwide.

APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

- **Content Delivery Networks (CDNs):**
 - Distributed computing is used in CDNs to deliver content efficiently to users by caching and distributing content across geographically distributed servers.
 - This ensures fast and reliable access to web content, streaming media, and other online services.
- These are just a few examples of the diverse applications of parallel and distributed computing.
- Across industries, these paradigms play a critical role in driving innovation, improving efficiency, and enabling the processing of vast amounts of data.



CHALLENGES WITH PARALLEL & DISTRIBUTED COMPUTING

CHALLENGES

Challenges with Parallel and Distributed Computing

Challenges

- Parallelization Overhead
- Load Balancing
- Data Dependencies
- Communication Overhead
- Scalability
- Fault Tolerance
- Resource Management



CHALLENGES WITH PARALLEL & DISTRIBUTED COMPUTING

- Parallel computing presents several challenges that need to be addressed to ensure efficient and effective execution of parallelized tasks.
- Some of the key challenges include:
- **Parallelization Overhead :**
 - Parallelizing tasks incurs overhead due to the need for synchronization, communication, and management of parallel processes.
 - This overhead can sometimes outweigh the benefits of parallelization, especially for tasks with small computational requirements.

CHALLENGES WITH PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Load Balancing:**

- Distributing tasks evenly across multiple processing units or nodes is challenging, as the workload may not be evenly distributed or may vary dynamically over time.
- Load balancing algorithms are needed to ensure that each processing unit receives a fair share of the workload, optimizing overall performance.

CHALLENGES WITH PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Data Dependencies:**

- Dependencies among tasks can hinder parallel execution, as certain tasks may need to wait for others to complete before they can proceed.
- Identifying and managing data dependencies is crucial for efficient parallelization, as excessive synchronization can lead to bottlenecks and decreased parallel performance.

CHALLENGES WITH PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Communication Overhead:**

- Communication between parallel processes incurs overhead due to latency, bandwidth limitations, and network congestion.
- Minimizing communication overhead is essential for achieving scalable parallel performance, often requiring optimization techniques such as message aggregation, pipelining, and asynchronous communication.

CHALLENGES WITH PARALLEL & DISTRIBUTED COMPUTING CON...

■ Scalability:

- Ensuring that parallel algorithms and systems scale efficiently with increasing problem sizes and computational resources is a significant challenge.
- Scalability issues can arise due to limitations in algorithm design, data distribution strategies, and hardware architectures, requiring careful consideration during system design and implementation.

CHALLENGES WITH PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Fault Tolerance:**

- Parallel computing systems are susceptible to failures, including hardware failures, network failures, and software errors.
- Implementing fault tolerance mechanisms, such as checkpointing, replication, and recovery strategies, is essential for ensuring the reliability and availability of parallel computing systems, particularly in large-scale distributed environments.

CHALLENGES WITH PARALLEL & DISTRIBUTED COMPUTING CON...

■ **Programming Complexity:**

- Developing parallel algorithms and applications can be complex and error-prone, requiring specialized programming models, languages, and libraries.
- Parallel programming paradigms such as shared memory and message-passing introduce additional complexities, including race conditions, deadlocks, and synchronization issues, which must be carefully managed to avoid bugs and ensure correctness.

APPLICATIONS FOR PARALLEL & DISTRIBUTED COMPUTING CON...

■ Resource Management:

- Efficiently managing computational resources, such as CPU cores, memory, and network bandwidth, is critical for achieving optimal performance in parallel computing systems.
- Resource management challenges include task scheduling, memory allocation, and network bandwidth provisioning, which require sophisticated algorithms and policies to balance competing demands and priorities effectively.

QUESTION ANSWERS



THANK YOU

