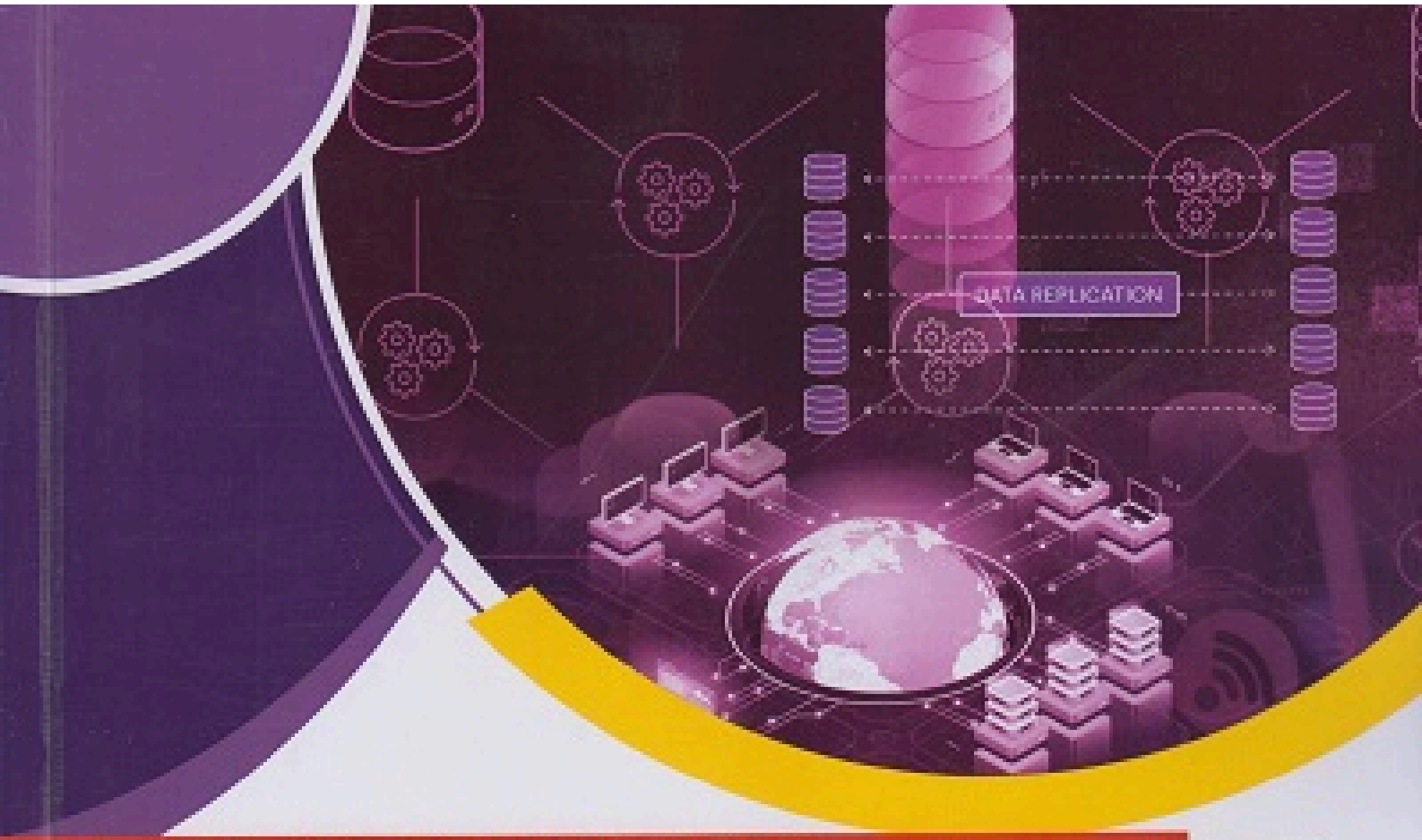


According to New Credit System Syllabus

SPPU

Final Year (B.E.) Degree Course In

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Semester - VIII

DISTRIBUTED COMPUTING

Dr. RAJIV R. BHANDARI
PRASHANT B. KOLI
Dr. SHRADDHA R. KHONDE
SURBHI P. AHIRE
Dr. DEVYANI JADHAV BHAMARE

Includes

- Model Question Papers For Practice (In Sem-30 Marks & End Sem-70 Marks)

www.pragationline.com
niralipune@pragationline.com
www.facebook.com/niralibooks
@nirali.prakashan

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

A TEXT BOOK OF

DISTRIBUTED COMPUTING

FOR
SEMESTER – VIII

FINAL YEAR (B.E.) DEGREE COURSE IN ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Strictly According to New Credit System Syllabus of Savitribai Phule Pune University
(w.e.f. 2023-2024)

Dr. RAJIV R. BHANDARI
Ph.D. (CSE)
Associate Professor,
Dept. of AI & DS Engg.,
SNJB's KBJ COE,
Chandwad, NASHIK.

PRASHANT B. KOLI
M. Tech (Comp. Engg.)
Assistant Professor,
Dept. of AI & DS Engg.,
SNJB's KBJ COE,
Chandwad, NASHIK.

Dr. SHRADDHA R. KHONDE
Ph.D. (Comp. Engg.)
Assistant Professor,
Dept. of Comp. Engg.,
Modern Education Society's Wadia College of Engg.,
PUNE.

SURBHI P. AHIRE
M.E. (Comp. Engg.)
Assistant Professor,
Dept. of AI & DS Engg.,
D.Y.Patil Institute of Engg. Mgt. & Research,
Akurdi, PUNE.

Dr. DEVYANI JADHAV BHAMARE
Ph.D. (CSE)
Assistant Professor,
Dept. of AI & DS Engg.,
Sanjivani College of Engineering,
Kopargaon, PUNE.

NIRALI PRAKASHAN
ADVANCEMENT OF KNOWLEDGE

N6722

DISTRIBUTED COMPUTING (BE, AI & DS) **ISBN : 978-93-6109-434-7**

First Edition : January 2024
© : Authors

The text of this publication or any part thereof, should not be reproduced or transmitted in any form or stored in any computer storage system or device for distribution including photocopy, recording, taping or information retrieval system or reproduced on any disc, tape, perforated media or other information storage device etc., without the written permission of Authors with whom the rights are reserved. Breach of this condition is liable for legal action.

Every effort has been made to avoid errors or omissions in this publication. In spite of this, errors may have crept in. Any mistake, error or discrepancy so noted and shall be brought to our notice shall be taken care of in the next edition. It is notified that neither the publisher nor the authors or seller shall be responsible for any damage or loss of action to any one, of any kind, in any manner, therefrom. The reader must cross check all the facts and contents with original Government notification or publications.

Published By :
NIRALI PRAKASHAN
Abhyudaya Pragati, 1312 Shivaji Nagar
Off J.M. Road, PUNE 411005
Tel : (020) 25512336/37/39
Email : niralipune@pragationline.com

Polyplate

Printed By :
YOGIRAJ PRINTERS AND BINDERS
Survey No. 10/1A, Ghule Industrial Estate
Nanded Gaon Road
Nanded, Pune 411041

DISTRIBUTION CENTRES

PUNE

Nirali Prakashan
(For orders outside Pune)
S. No. 28/27, Dhayari Narhe Road, Near Asian College
Pune 411041, Maharashtra
Tel : (020) 24690204; Mobile : 9657703143
Email : bookorder@pragationline.com

Nirali Prakashan
(For orders within Pune)
119, Budhwar Peth, Jogeshwari Mandir Lane
Pune 411002, Maharashtra
Mobile : 9657703145, 9890997937
Email : niralilocal@pragationline.com

MUMBAI

Nirali Prakashan
Rasdhara Co-op. Hsg. Society Ltd., 'D' Wing Ground Floor, 385 S.V.P. Road
Girgaum, Mumbai 400004, Maharashtra
Mobile : 7045821020, Tel : (022) 2385 6339 / 2386 9976
Email : niralimumbai@pragationline.com

DISTRIBUTION BRANCHES

DELHI

Nirali Prakashan
Room No. 2 Ground Floor
4575/15 Omkar Tower, Agarwal Road
Darya Ganj, New Delhi 110002
Mobile : 9555778814/9818561840
Email : delhi@niralibooks.com

BENGALURU

Nirali Prakashan
Maitri Ground Floor, Jaya Apartments, No. 99, 6th Cross, 6th Main, Malleswaram, Bengaluru 560003 Karnataka; Mob : 9686821074
Email : bengaluru@niralibooks.com

NAGPUR

Nirali Prakashan
Above Maratha Mandir, Shop No. 3, First Floor, Rani Jhanshi Square, Sitabuldi Nagpur 440012 (MAH)
Tel : (0712) 254 7129
Email : nagpur@niralibooks.com

KOLHAPUR

Nirali Prakashan
438/2, Bhosale Plaza, Ground Floor, Khasbag, Opp. Balgopal Talim, Kolhapur 416 012 Maharashtra
Mob : 9850046155
Email : kolhapur@niralibooks.com

JALGAON

Nirali Prakashan
34, V. V. Golani Market, Navi Peth, Jalgaon 425001, Maharashtra
Tel : (0257) 222 0395
Mob : 94234 91860
Email : jalgaon@niralibooks.com

SOLAPUR

Nirali Prakashan
R-158/2, Avanti Nagar, Near Golden Gate, Pune Naka Chowk
Solapur 413001, Maharashtra
Mobile 9890918687
Email : solapur@niralibooks.com

marketing@pragationline.com | www.pragationline.com

Also find us on www.facebook.com/niralibooks

PREFACE

It gives us great pleasure to present the book '**Distributed Computing**' for the students of **Final Year (BE) Degree Course in Artificial Intelligence and Data Science** of Savitribai Phule Pune University. This book is strictly as per the **New Credit System Syllabus 2020 Pattern** effective from the Academic Year 2023-24.

As per the New Revised Examination Scheme, which has been implemented from the previous academic year, the In-Semester Examination allots 30 marks for the first two units and the End-Semester Examination allots 70 marks for the entire syllabus.

The theory subject will have 3 Credits.

In the book, all the basic concepts are explained in a simplified manner. Subject matter is presented in a conceptual manner rather than mathematical, as required by the new examination system. It is our objective to keep the presentation systematic and clear through explanatory notes and figures.

Main feature of this book is, Complete Coverage of the New Credit System Syllabus with Exercises and Model Question Papers of In-Semester and End-Semester Examinations.

We are sure that this book will cater to all needs of the students for this subject.

We also take this opportunity to express our sincere thanks to Shri. Dineshbhai Furia, Shri. Jignesh Furia, Mrs. Nirali Verma and entire team of Nirali Prakashan, namely Mrs. Deepali Lachake (Co-ordinator), who really have taken keen interest and untiring efforts in publishing this text.

We wish to make a special mention of the valuable contribution made by Late Shri M. P. Munde during the last two decades in reaching out to students, parents and teachers which eventually made **Nirali Prakashan, a brand of trust and quality in technical books**. Without his perseverance and zeal, successive editions would not have been possible.

The advice and suggestions of our esteemed readers to improve the text are most welcome and will be highly appreciated.

Pune **Authors**

SYLLABUS

Unit I: Introduction to Distributed Computing (06 hrs)

Fundamentals of Distributed Computing: Characteristics of Distributed Systems: Issues, Goals and Types of distributed systems, Distributed System Models.

Introduction to Artificial Intelligence and Data Science in Distributed Computing:

Distributing computational tasks, Handling large volumes of data and leveraging parallel processing capabilities, Issues related to data storage and retrieval, Data consistency, Communication overhead, Synchronization and fault tolerance.

Use Cases and Applications of Integrating AI and Data Science in Distributed Systems:

Predictive Maintenance, Fraud Detection, Intelligent Transportation Systems, Supply Chain Optimization, Energy Management, Healthcare and Medical Diagnostics, Customer Behavior Analysis and Natural Language Processing (NLP).

Case Studies : Introduction to Distributed Computing in E-commerce.

Unit II: Distributed Data Management and Storage (06 hrs)

Overview of Distributed Computing Frameworks and Technologies

Parallel Computing, Distributed Computing Models, Message Passing, Distributed File Systems: Hadoop Distributed File System (HDFS) and Google File System (GFS), Cluster Computing: (AWS), Microsoft Azure and Google Cloud Platform (GCP), Message Brokers and Stream Processing, Edge Computing.

Data Replication and Consistency Model: Eager Replication, Lazy Replication, Quorum-Based Replication, Consensus-Based Replication, Selective Replication, Strong Consistency, Eventual Consistency, Read-your-writes Consistency, Consistent Prefix Consistency, Causal Consistency.

Distributed Data Indexing and Retrieval Techniques: Distributed Hash Tables (DHTs), Distributed Inverted Indexing, Range-based Partitioning, Content-based Indexing, Peer-to-Peer (P2P) Indexing, Hybrid Approaches.

Case Studies : Distributed Data Management and Storage in Healthcare.

Unit III: Distributed Computing Algorithms (06 hrs)

Distributed Computing Algorithms: Communication and coordination in distributed systems.

Distributed consensus algorithms (Other consensus algorithms) ● Viewstamped Replication ● RAFT ● ZAB ● Mencius ● Many variants of Paxos (Fast Paxos, Egalitarian Paxos etc).

Fault tolerance and recovery in distributed systems.

Load Balancing and Resource Allocation Strategies: Weighted Round Robin, Least Connection, Randomized Load Balancing, Dynamic Load Balancing, Centralized Load Balancing, Distributed Load Balancing, Predictive Load Balancing.

Applying AI Techniques to Optimize Distributed Computing Algorithms: Machine Learning for Resource Allocation, Reinforcement Learning for Dynamic Load Balancing, Genetic Algorithms for Task Scheduling, Swarm Intelligence for Distributed Optimization.

Case Studies : Distributed Computing Algorithms in Weather Prediction.

Unit IV: Distributed Machine Learning and AI (06 hrs)

Introduction to Distributed Machine Learning Algorithms: Types of Distributed Machine Learning: Data Parallelism and Model Parallelism, Distributed Gradient Descent, Federated Learning, All-Reduce, Hogwild, Elastic Averaging SGD.

Software to Implement Distributed ML: Spark, GraphLab, Google TensorFlow, Parallel ML System (Formerly Petuum), Systems and Architectures for Distributed Machine Learning.

Integration of AI Algorithms in Distributed Systems: Intelligent Resource Management, Anomaly Detection and Fault Tolerance, Predictive Analytics, Intelligent Task Offloading.

Case Studies : Distributed Machine Learning and AI in Fraud Detection.

Unit V: Big Data Processing in Distributed Systems **(06 hrs)**

Big Data Processing Frameworks in Distributed Computing: Hadoop, Apache Spark, Apache Storm, Samza, Flink.

Parallel and Distributed Data Processing Techniques: Single Instruction Single Data (SISD), Multiple Instruction Single Data (MISD), Single Instruction Multiple Data (SIMD), Multiple Instruction Multiple Data (MIMD), Single Program Multiple data (SPMD), Massively Parallel Processing (MPP).

Scalable Data Ingestion: Types of data ingestion, Benefits, Challenges, tools, Transformation in distributed systems.

Real-time Analytics and Streaming Analytics: Types of real time analytics, Types of streaming analytics, Comparison of real time analytics and streaming analytics, Applying AI and data science for large-scale data processing and analytics.

Case Studies : Big Data Processing in Distributed Systems for Social Media Analytics.

Unit VI: Distributed Systems Security and Privacy **(06 hrs)**

Security Challenges in Distributed Systems, Insider Threats, Encryption and Secure Communication: TLS/SSL, PKI, VPN, AMQP, Privacy Preservation Techniques: Differential Privacy, Homomorphic Encryption, Secure Multi-Party Computation (SMPC), Federated Learning, Anonymization and Pseudonymization, Access Control and Data Minimization, AI-based Intrusion Detection and Threat Mitigation Techniques: Anomaly Detection, Behavior-based Detection, Threat Intelligence and Analysis, Real-time Response and Mitigation, Adaptive Security, User and Entity Behavior Analytics (UEBA), Threat Hunting and Visualization.

Case Studies : Distributed Systems Security and Privacy in Healthcare.

CONTENTS

Unit I: Introduction to Distributed Computing **1.1-1.22**

1.1 Fundamentals of Distributed Computing 1.1
- 1.1.1 Introduction 1.1
- 1.1.2 Characteristics of Distributed Computing 1.1
- 1.1.3 Goals in Distributed System 1.2
- 1.1.4 Issues of Distributed System 1.2

1.2 Types of Distributed Systems 1.3
- 1.2.1 Client-server Architecture 1.3
- 1.2.2 Peer-to-Peer Networks (P2P) 1.3
- 1.2.3 Middleware System 1.4
- 1.2.4 Three-tier System 1.4

1.3 Distributed System Models 1.4
- 1.3.1 Distributed Computing System 1.4
- 1.3.2 Distributed Information System 1.5
- 1.3.3 Distributed Pervasive System 1.6

1.4 Introduction to Artificial Intelligence and Data Science in Distributed Computing 1.6
- 1.4.1 Collaboration between AI and Data Science in Distributed Computing 1.6
- 1.4.2 Distributing Computational Tasks 1.7
- 1.4.3 Handling Large Volume of Data Set 1.8
- 1.4.4 Understanding Parallel Processing 1.10
- 1.4.5 Data Consistency 1.11
- 1.4.6 Communication Overhead 1.12
- 1.4.7 Data Storage Issues and Retrieval 1.13

1.5 Synchronization and Fault Tolerance in Distributed Systems 1.14
- 1.5.1 Synchronization 1.14
- 1.5.2 Fault Tolerance 1.14

1.6 Use Cases and Applications of Integrating AI and Data Science in Distributed Systems 1.14
- 1.6.1 Predictive Maintenance (PdM) 1.15
- 1.6.2 Fraud Detection 1.15
- 1.6.3 Intelligent Transportation Systems (ITS) 1.16
- 1.6.4 Supply Chain Optimization 1.17
- 1.6.5 Energy Management 1.18
- 1.6.6 Healthcare and Medical Diagnostics 1.18
- 1.6.7 Customer Behavior Analysis and Natural Language Processing (NLP) 1.19

- **Case Study:** Enhancing Scalability and Performance in E-Commerce through Distributed Computing 1.20
- **Exercise** 1.21

Unit II: Distributed Data Management and Storage **2.1-2.26**

2.1 Overview of Distributed Computing Frameworks and Technologies 2.1
- 2.1.1 Parallel Computing 2.1
- 2.1.2 Types of Parallel Computing 2.1
- 2.1.3 Distributed Computation Models 2.2

2.2 Distributed File System 2.3
- 2.2.1 Hadoop Distributed File System (HDFS) 2.4
- 2.2.2 Google File System (GFS) 2.4
- 2.2.3 Google Cloud Platform (GCP) 2.6

2.3 Cluster Computing 2.7
- 2.3.1 Types of Clusters 2.7

2.3.2 Amazon Web Services (AWS) 2.8
2.3.3 Microsoft Azure 2.8
2.3.4 Message Broker 2.10
2.3.5 Stream Processing 2.10
2.3.6 Edge Computing 2.11
2.3.7 Applications of the Edge Computing 2.13
2.4 Data Replication and Consistency Model 2.14
2.4.1 Eager Replication 2.15
2.4.2 Lazy Replication 2.15
2.4.3 Quorum Based Replication 2.16
2.4.4 Selective Replication 2.17
2.4.5 Consensus-Based Replication 2.17
2.4.6 Comparison of Data Replication 2.18
2.5 Consistency Models 2.19
2.5.1 Strong Consistency 2.19
2.5.2 Eventual Consistency 2.20
2.5.3 Read Your Writes Consistency 2.20
2.5.4 Consistent Prefix Consistency 2.21
2.5.5 Casual Consistency 2.21
2.6 Distributed Data Indexing and Retrieval Techniques 2.21
2.6.1 Distributed Hash Tables (DHTs) 2.21
2.6.2 Distributed Inverted Indexing 2.22
2.6.3 Range-based Partitioning 2.22
2.6.4 Content-based Indexing 2.23
2.6.5 Peer-to-Peer (P2P) Indexing 2.24
2.6.6 Hybrid Approaches 2.24
- **Case Study:** Distributed Data Management and Storage in Healthcare 2.25
- **Exercise** 2.26

Unit III: Distributed Computing Algorithms 3.1-3.24

3.1 Communication and Coordination in Distributed System 3.1
3.2 Distributed Consensus Algorithms 3.2
3.3 Variant of Paxos 3.2
3.3.1 Fast Paxos 3.3
3.3.2 Egalitarian Paxos (EPaxos) 3.3
3.4 Raft 3.4
3.5 ZAB in Distributed Computing 3.5
3.6 Viewstamped Replication Protocol 3.10
3.7 Mencius Protocol in Distributed System 3.10
3.8 Load Balancing and Resource Allocation Strategies 3.11
3.8.1 Weighted Round Robin (WRR) 3.11
3.8.2 Least Connections 3.12
3.8.3 Randomized Load Balancing 3.13
3.8.4 Dynamic Load Balancing 3.13
3.8.5 Centralized Load Balancing 3.14
3.8.6 Distributed Load Balancing 3.14
3.8.7 Predictive Load Balancing 3.15
3.9 Applying AI Techniques to Optimize Distributed Computing Algorithms 3.16
3.9.1 Machine Learning for Resource Allocation 3.16

3.9.2 Reinforcement Learning for Dynamic Load Balancing 3.19
3.9.3 Genetic Algorithms for Task Scheduling 3.20
3.9.4 Swarm Intelligence for Distributed Optimization 3.21
- **Case Study:** Distributed Computing Algorithm in Weather Prediction 3.23
- **Exercise** 3.24

Unit IV: Distributed Machine Learning and AI **4.1-4.16**

4.1 Introduction to Distributed Machine Learning Algorithms 4.1
4.2 Types of Distributed Machine Learning 4.1
4.2.1 Data Parallelism in PyTorch 4.1
4.2.2 Model Parallelism in PyTorch 4.2
4.3 Distributed Gradient Decent 4.3
4.3.1 Visualization of Gradient Descent 4.3
4.3.2 Gradient Descent Algorithm 4.3
4.4 Federated Learning, All Reduce and Hogwild 4.3
4.4.1 Federated Learning 4.3
4.4.2 All Reduce 4.4
4.4.3 Hogwild 4.5
4.5 Elastic Averaging SGD 4.6
4.5.1 Working of EASGD 4.6
4.5.2 Applications of EASGD 4.6
4.6 Software to Implement Distributed ML 4.6
4.6.1 Spark 4.6
4.6.2 GraphLab 4.7
4.6.3 TensorFlow 4.10
4.6.4 Parallel ML System-Petuum Design 4.10
4.6.5 Systems and Architectures for Distributed Machine Learning 4.11
4.7 Integration of AI algorithms in Distributed Systems 4.12
4.7.1 Intelligent Resource Management 4.13
4.7.2 Anomaly Detection and Fault Tolerance 4.14
4.7.3 Intelligent Task Offloading 4.14
- **Case Study :** Distributed Machine Learning and AI in Fraud Detection 4.15
- **Exercise** 4.16

Unit V: Big Data Processing in Distributed Systems **5.1-5.16**

5.1 Big Data Processing Frameworks in Distributed Computing 5.1
5.1.1 Apache Hadoop 5.1
5.1.2 Apache Spark 5.2
5.1.3 Apache Flink 5.3
5.1.4 Apache Storm 5.3
5.1.5 Apache Kafka 5.4
5.1.6 Apache Samza 5.4
5.1.7 Apache Hive 5.4
5.1.8 Apache HBase 5.5
5.1.9 Google BigQuery 5.5
5.2 Parallel and Distributed Data Processing Techniques 5.6
5.2.1 Parallel Data Processing 5.6
5.2.2 Distributed Data Processing 5.6
5.3 Fynn's Taxonomy with Processing Nodes and Data Stream Technique 5.6
5.3.1 Single Instruction, Single Data (SISD) 5.7
5.3.2 Single Instruction, Multiple Data (SIMD) 5.7
5.3.3 Multiple Instruction, Single Data (MISD) 5.8
5.3.4 Multiple Instruction, Multiple Data (MIMD) 5.8

5.3.5 Single Program, Multiple Data (SPMD) 5.8
5.3.6 Massively Parallel Processing (MPP) 5.8
5.4 Scalable Data Ingestion 5.9
5.4.1 Types of Data Ingestion 5.10
5.4.2 Advantages of Data Ingestion 5.10
5.4.3 Challenges in Data Ingestion 5.10
5.4.4 Tools for Data Ingestion in Distributed Systems 5.10
5.4.5 Data Transformation 5.11
5.5 Real-Time Analytics and Streaming Analytics 5.11
5.5.1 Key Differences between Real Time Analytics and Streaming Analytics 5.11
5.5.2 Types of Real-Time Analytics 5.12
5.5.3 Types of Streaming Analytics 5.12
5.5.4 Difference between Real Time Analytics and Streaming Analytics 5.13
5.5.5 Applying AI and Data Science for Large-Scale Data Processing and Analytics 5.14
- **Case Study :** Big Data Processing in Distributed Systems for Social Media Analytics 5.15
- **Exercise** 5.16

Unit VI: Distributed Systems Security and Privacy **6.1-6.18**

6.1 Security Challenges in Distributed Systems 6.1
6.1.1 Goals of Distributed System Security 6.1
6.1.2 Security Requirements and Attacks Related to Distributed Systems 6.1
6.1.3 Issues of Distributed System Security 6.1
6.2 Insider Threats/ Attacks 6.2
6.3 Encryption and Secure Communication 6.3
6.3.1 TLS/SSL 6.3
6.3.2 PKI 6.4
6.3.3 VPN 6.5
6.3.4 AMQP 6.6
6.4 Privacy Preservation Techniques 6.7
6.4.1 Differential Privacy 6.7
6.4.2 Homomorphic Encryption 6.8
6.4.3 Secure Multi-Party Computation (SMPC) 6.9
6.4.4 Federated Learning 6.9
6.4.5 Anonymization and Pseudonymization 6.10
6.4.6 Access Control and Data Minimization 6.10
6.5 AI-based Intrusion Detection and Threat Mitigation Techniques 6.13
6.5.1 Anomaly Detection 6.13
6.5.2 Behavior-based Detection 6.13
6.5.3 Threat Intelligence and Analysis 6.13
6.5.4 Real-time Response and Mitigation 6.14
6.5.5 Adaptive Security 6.14
6.5.6 User and Entity Behavior Analytics (UEBA) 6.16
6.5.7 Threat Hunting and Visualization 6.16
- **Case Study :** Distributed System Security and Privacy in Healthcare 6.17
- **Exercise** 6.18

- **Model Question Papers**
  - ➢ **In-Sem. Exam. (30 Marks)** **P.1-P.1**
  - ➢ **End-Sem. Exam. (70 Marks)** **P.2-P.2**

❋ ❋ ❋

UNIT - I

INTRODUCTION TO DISTRIBUTED COMPUTING

1.1 FUNDAMENTALS OF DISTRIBUTED COMPUTING

Distributed computing refers to the use of multiple interconnected computers or processors that work together to solve a complex problem or perform a task. In the context of Artificial Intelligence (AI), distributed computing plays a crucial role in handling the computational demands of large-scale and resource-intensive AI applications.

1.1.1 Introduction

- **Parallel Processing:** Distributed computing in AI often involves parallel processing, where tasks are divided into smaller sub-tasks and executed concurrently across multiple processors or machines. This parallelism helps in achieving faster computations and handling large datasets.
- **Scalability:** Scalability is a key advantage of distributed computing in AI. As the complexity of AI models and datasets increases, distributed systems can easily scale by adding more computational resources, allowing for efficient handling of growing workloads.
- **Large-Scale Data Processing:** AI applications often require the processing of massive datasets. Distributed computing enables the parallel processing of data across multiple nodes, reducing the time required for data analysis, training and inference.
- **Fault Tolerance:** Distributed systems are designed to be resilient to failures. If one node or machine fails, the system can continue to operate with the remaining nodes. This fault tolerance is crucial in ensuring the reliability of AI applications, especially in mission-critical scenarios.
- **Distributed Machine Learning:** Training complex machine learning models, especially deep neural networks, can be computationally intensive. Distributed computing facilitates the parallel training of models across multiple machines, reducing the time needed for model convergence.
- **Decentralized AI Architectures:** In some cases, AI systems are designed with decentralized architectures, where components of the AI model or algorithm are distributed across different nodes. This approach can enhance privacy, reduce latency and improve the overall performance of the AI system.
- **Resource Optimization:** By distributing computation across multiple nodes, distributed computing allows for better resource utilization. This is particularly important in AI applications where optimizing computational resources is essential for efficiency and cost-effectiveness.
- **Distributed Inference:** In real-time AI applications, such as those involving computer vision or natural language processing, distributed systems can be used to perform inference across multiple nodes simultaneously, enabling quicker response times.
- **Edge Computing Integration:** Distributed computing is closely linked with edge computing in AI. Edge devices can perform local computations and the results can be aggregated and processed in a distributed manner, minimizing the need for centralized data processing.

1.1.2 Characteristics of Distributed Computing

Distributed computing refers to the use of multiple computers or nodes that work together to solve a problem or perform a task. Here are some key characteristics of distributed computing:

- **Concurrency:** Distributed systems often involve multiple nodes working concurrently. Tasks are divided among different nodes and these nodes can execute their tasks independently of each other.
- **Fault Tolerance:** Distributed systems need to be resilient to failures. If one node fails, the system should be able to continue functioning without a complete breakdown. This is achieved through redundancy, replication and fault-tolerant mechanisms.

- **Scalability:** Distributed systems should be scalable to accommodate an increasing number of nodes or users. Scalability can be achieved by adding more nodes to the system and distributing the workload effectively.
- **Interprocess Communication:** Communication is a critical aspect of distributed computing. Nodes need to exchange information and coordinate their activities. There are various communication mechanisms, such as message passing, Remote Procedure Calls (RPC) and distributed objects.
- **Transparency:** Ideally, the distribution of the system should be transparent to the end-users and application developers. This means that users should not be aware of the underlying distribution of resources and the system should appear as a single, unified entity.
- **Heterogeneity:** Distributed systems often consist of nodes with different hardware, operating systems and software. They should be capable of working together despite these differences. Middleware and standardized communication protocols are often used to address heterogeneity.
- **Consistency and Replication:** Ensuring consistency of data across distributed nodes can be challenging. Replication is often used to mitigate this challenge by maintaining multiple copies of data on different nodes. However, maintaining consistency among these replicas is a complex task.
- **Load Balancing:** To ensure efficient resource utilization, distributed systems employ load balancing mechanisms. These mechanisms distribute the workload evenly among nodes to prevent overloading of specific resources.
- **Security:** Distributed systems must address security concerns such as data integrity, confidentiality and authentication. Security mechanisms, such as encryption and access control, are implemented to protect data and ensure the integrity of communication.
- **Decentralization:** Distributed systems are often designed to be decentralized, meaning that there is no single point of control. This can improve fault tolerance and reduce the risk of system-wide failures.

1.1.3 Goals in Distributed System

What Made Computational Systems Distributed?

At the very beginning computers were huge and expensive machines. Computer science at the time, as the art of computer programming was born upon such machines.

Four Goals for a Distributed System

1. **Making (distributed, remote) Resources Available for Use**

   Making distributed, remote resources available for use typically involves setting up a system or infrastructure that allows users to access and utilize resources located in different locations.

2. **Allowing Distribution of Resources to be Hidden Whenever Unnecessary**

   Hiding non-relevant properties of the system's components and structure is called transparency.

   Types of Transparency

   - Location hide the location of a resource
   - Migration hide the change of location of a resource
   - Relocation hide the motion of a resource
   - Replication hide that a resource is replicated
   - Concurrency hide the sharing of a resource by multiple users
   - Failure hide the failure and subsequent recovery of a resource

3. **Promoting Openness**

   Promoting openness in a distributed system is crucial for fostering collaboration, transparency and effective communication among the various components and participants.

4. **Promoting Scalability**

   Scaling a distributed system involves designing and implementing a system architecture that can handle increased load and demand.

1.1.4 Issues of Distributed System

There are also multiple challenges of distributed systems that determine the performance of the overall system. There are also multiple challenges of distributed systems that determine the performance of the overall system.

(I) Heterogeneity

- One of the difficulties with a distributed system is heterogeneity, which describes variations in the hardware, software or network configurations of individual nodes.
- Coordination and communication may be hampered as a result. Service-oriented architecture, virtualization, middleware and standardization are methods for handling heterogeneity.