Pokhara University
Faculty of Science and Technology

Course Code.: CMP 380 | Full marks: 100
Course title: **Distributed System and Cloud Computing (3-1-2)** | Pass marks: 45
Nature of the course: Theory/Practical | Time per period: 1 hour
Total periods: 45
Level: Bachelor | Program: BE

3. Course Description

The "Distributed System and Cloud Computing" course is designed for students to provide a comprehensive understanding of the principles, algorithms, and technologies underpinning distributed systems and cloud computing. This course integrates theoretical concepts with practical applications to prepare students for real-world challenges in building and managing distributed and cloud-based systems. The course will cover fundamental principles of distributed systems, advanced cloud computing techniques, and emerging trends in both fields. It also course provides different aspects of Distributed System including overview of distributed communication technologies, clock synchronization, coordination and agreement in distribution system. It als highlights the issues of security and failure handling in distributed system. Laboratory exercises will provide hands-on experience with relevant tools and technologies, reinforcing the theoretical knowledge gained through lectures.

4. Specific Objectives:

By the end of this course, students will be able to:

Course Objectives
The general objectives of this course includes

- In-depth understanding of distributed system and cloud computing principles and practices, develop distributed applications.

The course is designed with the following specific objectives:

- To understand and Grasp the core principles and architectures of distributed systems.
- To analyze and apply algorithms for synchronization, consistency, and fault tolerance in distributed environments.
- To comprehend key cloud computing models and services, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
- To implement and manage cloud resources and applications using popular cloud platforms.
- To investigate and understand the impact of current trends and technologies in distributed systems and cloud computing, such as edge computing, serverless architectures, and containerization.
- To gain hands-on experience through lab exercises that involve building, deploying, and managing distributed and cloud-based applications.

5. Contents in Detail

Specific Objectives	Contents
• To understand the Basics of Distributed Systems and articulate the key goals of distributed systems including scalability, transparency, reliability, and fault tolerance. • To understand the distributed system architecture including client-server model, peer-to-peer model and hybrid model.	**Unit I: Introduction to Distributed Systems (4 hours)** 1.1 Distributed Systems: Definition, characteristics, and types of distributed systems. 1.2 Goals of Distributed Systems: Scalability, transparency, reliability, and fault tolerance. 1.3 Distributed System Architectures 1.3.1 Client-server model: architecture, advantages, and limitations 1.3.2 Peer-to-peer model: decentralized approach, applications 1.3.3 Hybrid models: combining client-server and peer-to-peer
• To able to describe the concept and methods of message passing in distributed systems and define multicast communication and its use in distributed systems.	**Unit II: Communication in Distributed Systems (5 hours)** 2.1 Inter-process Communication 2.1.1 Message passing 2.1.2 Remote Procedure Calls (RPC) 2.1.3 Communication between distributed objects 2.1.4 Remote Method Invocation (RMI) 2.1.5 Events and Notifications 2.2 Group Communication 2.2.1 Multicast 2.2.2 Publish/subscribe systems 2.2.3 Consistency models.

• To understand the basic working principles of different logical and physical clock synchronization algorithms and Distributed co-ordination	**Unit III: Synchronization, Coordination and Agreement (10 hours)** 3.1 Time Synchronization 3.1.1 Time and time synchronization 3.1.2 Physical Clock Synchronization 3.1.2.1 Berkeley Algorithm 3.1.2.2 Cristian's Algorithm 3.1.2.3 Network Time Protocol (NTP) 3.1.3 Logical Clocks Synchronization 3.1.3.1 Events and ordering 3.1.3.2 Partical, causal and total ordering of messages 3.1.3.3 Global State and State Recording 3.1.3.4 Lamport logical clock 3.1.3.5 Vector clock 3.2 Distributed Co-ordination 3.2.1 Distributed mutual exclusion 3.2.2 Mutual Exclusion Algorithms 3.2.2.1 Central Coordinator Algorithm 3.2.2.2 Token Ring Algorithm 3.2.2.3 Lamport's Algorithm 3.2.2.4 Ricart-Agrawala (Non-Token based and token based) 3.2.3 Distrbuted Leader Election 3.2.3.1 Bully algorithm 3.2.3.2 Ring Algorithms
• To understand Fault Models and Fault Tolerance mechanisms in distributed system, replication and checkpointing techniques contribute to fault tolerance. • To understand the concept of distributed consensus and its importance in ensuring agreement among distributed processes.	**Unit IV: Fault Tolerance, Recovery and Distributed Transaction (7 Hours)** 4.1 Fault Models 4.1.1 Crash, Omission and Byzantine failures. 4.1.2 Fault Tolerance Techniques: 4.1.2.1 Replication and Checkpointings 4.1.3 Recovery Mechanisms: 4.1.3.1 Rollback, checkpointing and recovery protocols 4.2 Agreement in faulty system 4.2.1 Distributed consensus 4.2.2 Byzantine Generals Problem 4.3 Distributed Transaction 4.3.1 Concurrency control mechanism 4.3.2 Automic Commitment Protocol 4.4 Distributed Deadlock 4.4.1 Overview of distributed deadlock (Resource

	and communication deadlock) 4.4.2 Deadlock prevention 4.4.2.1 Timestamp Ordering 4.4.2.2 Prevention Through Resource Allocation 4.4.3 Deadlock detection 4.4.3.1 Centralized Deadlock Detection 4.4.3.2 Distributed Deadlock Detection 4.4.4 Resolving deadlock
• To understand and describe the architecture, operation and use cases of different distributed file systems including GFS, HDFS. • To explore the architecture , design principles and use cases of NoSQL databases such as Cassandra and MongoDB. • To Identify the key features and use cases of big data processing frameworks including spark and Hadoop.	**Unit V: Distributed Data Storage and Big Data (7 Hours)** 5.1 Distributed File Systems 5.1.1 Google File System (GFS): Architecture, Operations,features and use cases 5.1.2 Hadoop Distributed File System (HDFS): Architecture, Operations, features and use cases 5.2 NoSQL Databases: 5.2.1 Cassandra: Architecture, design principles, Data Model and Query Language, features and use cases 5.2.2 MongoDB: Architecture, design principles, Data Model and Query Language, features and use cases 5.3 Big Data Processing Frameworks 5.3.1 Hadoop: Key features and use cases 5.3.2 Spark : Key Features and use cases
• To review and explain the basic concepts of cloud computing, cloud service models and deployment models. • To study and understand the concepts of cloud service architecture and design including SOA, Microservices, Serverless architecture and Resource Management. • To understand the principles and concepts of cloud service management including SLA and Cloud security.	**Unit VI: Cloud Computing (12 hours)** 6.1 Review of virtualization and cloud computing of cloud computing 6.2 Implementing Monitoring, Elasticity and High Availability in cloud environment 6.3 Building Serverless Architecture 6.4 Automating Cloud Architecture 6.5 Cloud Architecture and design 6.5.1 Service-oriented architecture (SOA) 6.5.2 Microservices, serverless architecture 6.5.3 Resource Management and Orchestration 6.5.3.1 Resource Allocation: auto-scaling, load balancing and elasticity 6.5.3.2 Orchestration Tools: Kubernetes,

	Docker Swarm. 6.5.4 Cloud Service Management 6.5.4.1 Service Level Agreements (SLAs): Definition, importance, and SLA monitoring. 6.5.4.2 Cloud Security: Threats, identity, and access management (IAM), encryption, and compliance.

7. Methods of Instruction

The Distributed System and Cloud Computing course will employ a comprehensive approach to instruction, integrating theoretical knowledge with practical application. Interactive lectures will introduce core concepts of distributed system principles, algorithms and principles and technologies of cloud computing,while hands-on lab session, project work and demonstrations will provide real-time/real life examples of these theories in action. Hands-on lab sessions will reinforce learning, allowing students to implement distributed and cloud applications. Real-world case studies and group projects will foster collaboration and critical thinking.This multifaceted approach aims to equip students with both the theoretical foundation and practical skills necessary for proficient software engineer in a real-world context.

8. List of Practical Works

1. Implemention of a simple client-server application with socket programming.
2. Implementation of a RPC and RMI.
3. Setting up a distributed system environment using virtual machines or cloud instances.
4. Implement and apply distributed algorithms for synchronization, mutual exclusion and leader election.
5. Developing and testing a fault-tolerant distributed system.
6. Configuring and using a distributed file system (e.g., HDFS).
7. Building and deploying a basic cloud application.
8. Implementing auto-scaling and resource management in a cloud environment.

9. Evaluation system and Students' Responsibilities

Evaluation System

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc.

In this course, students will be required to develop a network application as part of their subject project. This project will involve designing, implementing, and testing a network application that

can operate across in Unix/Windows or both platform.. The project will require students to apply the concepts and techniques learned throughout the course.By completing this project, students will gain hands-on experience in building real-world network applications, reinforcing their understanding of course material and preparing them for professional challenges in network programming.

The tabular presentation of the evaluation system is as follows.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		30	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	10%			
Lab Report/Project Report	20%			
Practical Exam/Project Work	50%			
Viva	20%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Students' Responsibilities:
Each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

Prescribed Books and References

1. "Distributed Systems: Principles and Paradigms" by Andrew S. Tanenbaum and Maarten Van Steen
2. "Distributed Systems: An Algorithmic Approach" by Sanjay Kumar and Devesh Tiwari
3. "Distributed Systems Concepts and Design" by Georce Coulouris, Jean Dollimore, Tim Kindberg, 5th Edition, Pearson Education

4. “Cloud Computing: Concepts, Technology & Architecture” by Thomas Erl, Zaigham Mahmood, and Ricardo Puttini
5. “Cloud Computing: Principles, Systems and Applications” edited by Nikos Antonopoulos and Lee Gillam

Notes/Online Resources:

1. Google Cloud Platform Documentation : Link**:** **Google Cloud Documentation**
2. AWS Documentation and Tutorials : Link**:** **AWS Documentation**
3. Kubernetes Documentation : Link**:** **Kubernetes Official Documentation**
4. Docker Documentation : Link**:** **Docker Documentation**
5. Arxiv.org - Research Papers on Distributed Systems and Cloud Computing : Link**:** **Arxiv Distributed Systems**