DISTRIBUTED SYSTEMS (ICS-801)

Teacher Name:

DR. ANITA YADAV

Course Structure

Type	L	T	P	Credits
PEC	3	1	0	4

Prerequisite:

Course Content:

Unit-1:

Characterization of Distributed Systems: Introduction, Examples of distributed Systems, Resource sharing and the Web Challenges. **System Models:** Architectural models, Fundamental Models **Theoretical Foundation for Distributed System**: Limitation of Distributed system, absence of global clock, shared memory, Logical clocks, Lamport's & vectors logical clocks, Causal ordering of messages, global state, termination detection.

Distributed Mutual Exclusion: Classification of distributed mutual exclusion, Requirement of mutual exclusion theorem, Token based and non-token based algorithms, Performance metric for distributed mutual exclusion algorithms.

Unit-2:

Distributed Deadlock Detection: System model, resource vs communication deadlocks, deadlock prevention, avoidance, Detection & resolution, centralized dead lock detection, distributed dead lock detection, Path pushing algorithms, Edge chasing algorithms, **Agreement Protocols**: Introduction, System models, classification of Agreement Problem, Byzantine agreement problem, Consensus problem, Interactive consistency Problem, Solution to Byzantine Agreement problem, Application of Agreement problem, Atomic Commit in Distributed Database system.

Unit-3:

Distributed Objects and Remote Invocation: Communication between distributed objects, Remote procedure call, Events and notifications, Java RMI case study. **Security**: Overview of security techniques, Cryptographic algorithms, Digital signatures Cryptography pragmatics, Case studies:

Needham Schroeder, Kerberos, SSL & Millicent. **Distributed File Systems**: File service architecture, Sun Network File System, The Andrew File System, Recent advances.

Unit-4:

Transactions and Concurrency Control: Transactions, Nested transactions, Locks, Optimistic Concurrency control, Timestamp ordering, Comparison of methods for concurrency control. **Distributed Transactions**: Flat and nested distributed transactions, Atomic Commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery. Replication: System model and group communication, Fault - tolerant services, highly available services, Transactions with replicated data.

Unit-5:

Distributed Algorithms: Introduction to communication protocols, Balanced sliding window protocol, Routing algorithms, Destination based routing, APP problem, Deadlock free Packet switching, Introduction to Wave & traversal algorithms, Election algorithm. **CORBA Case Study**: CORBA RMI, CORBA services.

Text and References Books:

1. Singhal & Shivaratri, "Advanced Concept in Operating Systems", McGraw Hill
2. Coulouris, Dollimore, Kindberg, "Distributed System: Concepts and Design", Pearson Ed.
3. Gerald Tel, "Distributed Algorithms", Cambridge University Press
4. Nancy A. Lynch, "Distributed Algorithms", Elsevier Publication

Course Outcomes:

1. Understand the concepts of distributed systems in solving real world problems. (Understand)
2. Understand and apply various concepts of synchronization and agreement protocols. (Understand, Apply)
3. Understand and develop various distributed applications using remote procedure calls and remote method invocation for real time problems. (Understand, Apply)
4. Configure, deploy and access network file system. (Understand, Apply)
5. Develop distributed algorithms for MAC, routing and transport layer protocols. (Apply)

Lesson Plan

Lecture Plan

L1 Characterization of Distributed Systems: Introduction, Examples of distributed Systems, Resource sharing and the Web Challenges.

L2 System Models: Architectural models, Fundamental Models Theoretical Foundation for Distributed System:

L3 Limitation of Distributed system, absence of global clock, shared memory,

L4 Logical clocks, Lamport‟s & vectors logical clocks,

L5 Causal ordering of messages,

L6 Global state, termination detection.

L7 Distributed Mutual Exclusion: Classification of distributed mutual exclusion, Requirement of mutual exclusion theorem,

L8 Token based and non-token based algorithms, Performance metric for distributed mutual exclusion algorithms.

L9 Distributed Deadlock Detection: System model,

L10 Resource vs communication deadlocks,

L11 Deadlock prevention, avoidance, Detection & resolution,

L12 Centralized dead lock detection,

L13 Distributed dead lock detection, Path pushing algorithms, Edge chasing algorithms,

L14 Agreement Protocols: Introduction, System models,

L15 Classification of Agreement Problem, Byzantine agreement problem,

L16 Consensus problem, Interactive consistency Problem,

L18 Solution to Byzantine Agreement problem, Application of Agreement problem,

L19 Distributed Objects and Remote Invocation: Communication between distributed objects,

L20 Remote procedure call, Events and notifications, Java RMI case study.

L21 Security: Overview of security techniques, Cryptographic algorithms, Digital signatures Cryptography pragmatics,

L22 Case studies: Needham Schroeder, Kerberos, SSL & Millicent. D

L23 Distributed File Systems: File service architecture,

L24 Sun Network File System, The Andrew File System, Recent advances.

L25 Transactions and Concurrency Control: Transactions, Nested transactions,

L26 Locks, Optimistic Concurrency control, Timestamp ordering,

L27 Comparison of methods for concurrency control.

L28 Distributed Transactions: Flat and nested distributed transactions,

L29 Atomic Commit protocols,

L30 Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery.

L31 Replication: System model and group communication,

L32 Fault - tolerant services, highly available services, Transactions with replicated data.

L33 Distributed Algorithms -1

L34 Distributed Algorithms -2

L35 Introduction to communication protocols, Balanced sliding window protocol,

L36 Routing algorithms,

L37 Destination based routing,

L38 APP problem, Deadlock free Packet switching,

L39 Introduction to Wave & traversal algorithms,

L40 Election algorithm. CORBA Case Study: CORBA RMI, CORBA services.

Assignments

Distributed Systems (ICS-801)

Assignment# 1

Q. 1.
Using Dijkstra's Shortest Paths to Route Packets in a Network Overlay.
[This is a group assignment: Teams of max 4]
The objective of this assignment is to get you familiar with coding in a distributed setting where you need to manage the underlying communications between nodes. As part of this assignment you will be:

(1) constructing a logical overlay over a distributed set of nodes, and then
(2) computing shortest paths using Dijkstra's algorithm to route packets in the system.

Q.2: Consider that a file server is up half of the time due to hardware problems. How many more replicated servers have to be used to give an availability of at least 99 percent?

Q.3: Which protocol do you think it should be used to implement the following services, TCP or UDP? Explain you choices.

a. telnet, ssh
b. ftp, scp
c. rwho, finger
d. http
e. RPC

Assignment # 2

Q1. Why would you design a distributed system? List some advantages of distributed systems?
Q2. What is a Single-point-of-failure and how can distribution help here?
Q3. What kind of reliable connection is provided by a TCP/IP based socket? Is this reliability enough for distributed calls or when does it break down?
Q4. Can remote calls ever be "by reference"? Or is it only "reference semantics" that is achieved by shipping things back and forth and make it look like a local call did directly manipulate a remote object?
Q5. Explain the concept of an object reference in a distributed system. Why do remote objects need one and who creates it?
Q6. What is a distributed deadlock and why are they hard to detect?
Q7. Explain distributed 2-phase commit. Why is it called a voting algorithm?
Q8. How would you configure a Network File System (NFS)? Enlist its advantages.