Distributed Computing – Spring 2018

An Institute Elective Course (UG3 / UG4)

Course Level: L1 L-T-P-C: 3 -1-0-4

Description: Distributed Computing course is an elective (bouquet) course offered to the Under Graduate students.

Pre-requisite: Good Understanding of Operating Systems is essential

Syllabus:
This course is divided into the following modules:

1. **Introduction:** Parallel multiprocessor/multicomputer systems, Message-passing systems versus shared memory systems, Primitives for distributed communication, Synchronous versus asynchronous executions, Design issues and challenges
2. **A model of distributed computations:** A model of distributed executions, Models of communication networks, Global state of a distributed system, Models of process communications
3. **Logical time:** A framework for a system of logical clocks, Scalar time, Vector time, Efficient implementations of vector clocks, Physical clock synchronization
4. **Global state and snapshot recording algorithms:** System model and definitions, Snapshot algorithms for FIFO channels, Variations of the Chandy–Lamport algorithm, Snapshots in a causal delivery system, Monitoring global state, Finding consistent global snapshots in a distributed computation
5. **Topology abstraction and overlays:** Classifications and basic concepts, Complexity measures and metrics, Program structure
6. **Message ordering and group communication:** Message ordering paradigms, Asynchronous execution with synchronous communication, Synchronous program order on an asynchronous system, Group communication, Causal order (CO), Total order
7. **Termination detection:** System model of a distributed computation, Termination detection using distributed snapshots, A spanning-tree-based termination detection algorithm, Message-optimal termination detection, Termination detection in a faulty distributed system
8. **Distributed mutual exclusion algorithms:** Lamport's algorithm, Ricart–Agrawala algorithm, Quorum-based mutual exclusion algorithms, Maekawa's algorithm, Token-based algorithms, Suzuki–Kasami's broadcast algorithm

9. **Deadlock detection in distributed systems:** System model Models of deadlocks, Chandy–Misra–Haas algorithm for the AND model, Chandy–Misra–Haas algorithm for the OR model
10. **Distributed shared memory:** Abstraction and advantages, Memory consistency models, Shared memory mutual exclusion, Wait-free atomic snapshots of shared objects
11. **Checkpointing and rollback recovery:** Background and definitions, Issues in failure recovery, Checkpoint-based recovery, Log-based rollback recovery,
12. **Consensus and agreement algorithms:** Problem definition. Agreement in a failure-free system (synchronous or asynchronous), Agreement in (message-passing) synchronous systems with failures, Agreement in asynchronous message-passing systems with failures, Wait-free shared memory consensus in asynchronous systems
13. **Authentication in distributed systems:** Background and definitions, Protocols based on symmetric cryptosystems, Protocols based on asymmetric cryptosystems, Password-based authentication, Authentication protocol failures
14. **Self-stabilization:** Definition of self-stabilization, Issues in the design of self-stabilization algorithms, Methodologies for designing self-stabilizing systems, Self-stabilizing distributed spanning trees, Self-stabilizing algorithms for spanning-tree construction, A probabilistic self-stabilizing leader election algorithm, Self-stabilization as a solution to fault tolerance, Factors preventing self-stabilization, Limitations of self-stabilization
15. **Case Studies:** Discrete Event Simulations, Token based Message passing, Distributed Sorting on a line network, Distributed routing algorithms,
16. **Case Studies:** GFS, HDFS, Map Reduce and Spark, Google File System and HDFS, Distributed Execution using Map Reduce,
17. **Case Studies:** Sensor Networks, Authentication & Security in DS, Introduction to Sensor Networks, Distributed Algorithms for Sensor Networks, Authentication in Distributed Systems, Security in Distributed Systems and Block Chain

Text Books:

1) Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2008, Cambridge, UK, ISBN (e-Book): 978-0-511-39341-9

Reference Books:

1) Nancy A. Lynch, Distributed Algorithms, Morgan Kaufmann, USA, ISBN: 1-55860-348-4, 1996
2) Andrew S. Tanenbum and Maarten Van Steen, Distributed Systems: Principles and Para: Principles and Paradigms, Pearson Education India; Second edition, 2015, ISBN: 978-9332549807
3) George Coulouris, Jean Dollimore and Tim Kindberg, Distributed Systems: Concepts and Design, Pearson Education India; 4th edition, 2008, ISBN: 978-8131718407