Course Title: Distributed Systems

Course Type: Main Course
Unit Type: Theoretical
Units: 3
Prerequisites: Advanced Computer Architecture
Project: None
Teaching Hours: 48

Objectives:
Distributed systems are common. Computer scientists and engineers need to understand what are the principles and paradigms underlying distributed systems software and be familiar with several real-world examples. This course systematically examines the underlying principles and how they are applied to a wide variety of distributed systems in depth.

Syllabus:

Principles:

1. **Introduction**: Definition, Goals, Hardware and Software Concepts, Client-Server Model
2. **Communication**: Layered Protocols, Remote Procedure Call, Remote Object Invocation, Message-Oriented Communication, Stream-Oriented Communication
3. **Processes**: Threads, Clients, Servers, Code Migration, Software Agents
4. **Naming**: Naming Entities, Locating Mobile Entities, Removing Unreferenced Entities
5. **Synchronization**: Clock Synchronization, Logical Clocks, Global State, Election Algorithms, Mutual Exclusion, Distributed Transactions
6. **Consistency and Replication:** Data- and Client-Centric Consistency Models, Distribution Protocols, Consistency Protocols
7. **Fault Tolerance:** Process Resilience, Reliable Client-Server and Group Communication, Distributed Commit, Recovery
8. **Exemplar Distributed Systems:** Object-Based, Document-Based, Coordination-Based

Practices:

1. **Socket Programming**: Using socket APIs to establish communication links between remote and local processes
2. **Cluster Construction**: Clustering a number of networked homogenous/heterogenous computers.
3. **MPI:** Programming with the Message Passing Interface (MPI) that presents a standardized and portable message-passing repertoire of primitives designed to program parallel programs to run on parallel and distributed computing architectures
4. **Consensus:** Introducing a variety of algorithms for achieving consensus on a single data value amongst a number of distributed processes or systems

1. Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems: Principles and Paradigms, 3rd Ed., Prentice-Hall (2017)
2. George Coulouris and Jean Dollimore, Distributed Systems Concepts and Design, 5th Ed., Addison-Wesley (2011)
3. Maarten Van Steen and Andrew S. Tanenbaum, Reliable Distributed Systems: Technologies, Web Services and Applications, Springer (2005)
4. Hagit Attiya and Jennifer Welch, Distributed Computing: Fundamentals, Simulations and Advanced Topics, John-Wiley (2004)
5. Blaise Barney, Introduction to Parallel Computing, Lawrence Livermore National Laboratory 6, No. 13 (2007)
6. Peter Pacheco, Parallel Programming with MPI, Morgan-Kaufmann (1997)
7. Thomas Rauber and Gudula Rünger, Parallel Programming for Multicore and Cluster Systems, Springer (2010)
8. Nagel E. Wolfgang, High Performance Computing in Science and Engineering, Springer (2007)
9. Michel Charpentier, Mamoun Filali, Philippe Mauran, Gérard Padiou and Philippe Quéinnec, Abstracting Communication to Reason about Distributed Algorithms, In Proceedings of International Workshop on Distributed Algorithms, pp. 89-104, Springer, Berlin, Heidelberg (1996)