

MODULE: **DISTRIBUTED SYSTEMS**

CODE: **BSCH-4-2-10**

Stage: IV

Credit Points: 4 semester credits / 6 quarter units

Overview and Aims

The aim of this module is to give you a clear overview of the challenges and design goals of distributed systems. You will gain an exposure to the issues that must be dealt with while constructing flexible distributed applications. Though emphasis in on the conceptual basis, you will also be taught different approaches to implement distributed applications.

Learning Outcomes

Upon successful completion of this module, you should be able to:

1. explain challenges involved in designing and developing distributed systems
 2. compare different architectural and communication models
 3. write distributed applications using distributed object model and web services
 4. explain clock synchronisation and system state capturing strategies
 5. evaluate appropriate distributed algorithms for leader election and mutual exclusion
- describe state-of-the-art distributed search techniques

Module Content

Distributed systems overview

Design goals and challenges– heterogeneity, openness, scalability, consistency, transparency, mobility, fault tolerance, security

Fundamental concepts– multicomputer vs. multiprocessor systems, tightly vs. loosely coupled systems, horizontal vs. vertical distribution, Middleware based systems
Architectural models– client server model, multiple servers, thin clients, peer to peer model, mobile devices and spontaneous networking

Inter-processes communication

Synchronous and Asynchronous communication, TCP Sockets, Remote Procedure Calls, message passing, Message Oriented Middleware

Distributed object model

Distributed objects, Object Request Broker (ORB), object references, Interfaces, naming and binding, method invocation

Java RMI, CORBA and Distributed Component Object Model

Distributed object model– design issues

Service oriented architecture

Web services, XML-RPC, Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), Universal Description, Discovery and Integration (UDDI)

Service oriented architecture– design issues

Clock synchronisation

Concept of time in distributed systems, physical vs. logical time,

Clock synchronisation algorithms, Network Time Protocol (NTP), Vector clocks

Distributed system states

Global state predicates– deadlocks, termination detection, garbage collection,
Distributed snapshots– Chandy and Lamport snapshot algorithm, “possibility” and
“definitely” evaluation

Coordination, agreement and distributed search

Distributed Algorithms– fairness, liveness, safety

Distributed Algorithms for– leader election, mutual exclusion and distributed search