Unit 1

Introduction

- Computers used to operate independently.
- 2 things changed that:
  - Microprocessors
  - High speed networks
- Also, miniaturisation
- Now, easy to create a distributed system.
- A DS is a collection of autonomous computing elements that appear to its users as a single coherent system.

Characteristics

- Multiple Entities
- Heterogenity
- Concurrency
- Resource Sharing

Examples

- Web Search
- MMOGs
- Financial Trading

Design Goals

- Supporting resource sharing
- Making distribution transparent
- Being open
- Being scalable

Types

- Distributed Computing Systems
- Distributed Information Systems
- Pervasive Systems

Challenges

- Heterogenity
- Openness
- Scalability
- Security
- Failure Hiding
- Concurrency
- Transparency
- Quality of Service

Architectural Styles

- Layered
- Object-based
- Data-centric
- Event driven

Middleware

- Layer of software placed on top of the OS of all nodes.
- Computer: OS, DS: Middleware
- Difference is that middleware is offered in a networked form.

Middleware Organisation

- 2 design patterns.
  - Wrappers
  - Interceptors

System architectures

- Centralised
  - Simple client-server
    - Two processes
      - Server offers service
      - Client requests service
  - Multitiered
    - Simple organisation: 2 devices - client and server.
    - 3 logical levels: UI, application, data
    - Thus, split functionality over 3 devices - client, application server, database server.
- Decentralised: Peer-to-peer
  - Modern class of system architecture.
  - Enables horizontal distribution. Client/Server split into logically equal parts. Each part operates on its subset of data.
  - Communication between processes is symmetric. Each process is simultaneously a client and a server.
  - How to organise nodes in an overlay network. 2 types: structured and unstructured.

Network File System

- Many Distributed File Systems are organised as client-server architectures. NFS too.
- Standardised view of local file system.
- Heterogenous collection of processes can share a common file system.
- Remote file service.
- Clients offered transparent access to remote file server.
- Clients don’t know the location. They’re given an interface similar to a local file system.