

UNIT II

Clients, Servers, Transactions and Operating Systems

1 Introduction

This unit covers the following topics:

- The Anatomy of a server program
- Operating System Basic and Extended Services for server applications
- Server Scalability
- Client Anatomy
- Client/server Hybrids - Comparison of two and three tier
- Client side, Server side and Middleware side
- Hardware and Software requirements
- Transaction servers - TP lite Vs TP Heavy

2 Client–Server Model

The client-server model is a distributed application structure that partitions tasks or workloads between providers of a resource or service, called servers, and service requesters, called clients.

- **Clients request services**
- **Servers provide services**
- **Communication via network**

3 The Anatomy of a Server Program

A server program is a software component that provides services and resources to client programs.

3.1 Key Characteristics

- **Listens for requests:** It constantly runs, waiting for client requests over a network.
- **Processes requests:** It receives a request, performs the necessary operations (e.g., querying a database, running business logic), and generates a response.
- **Manages resources:** It controls access to shared resources like databases, files, printers, and other hardware or software components.
- **High concurrency:** Server programs are designed to handle many client requests simultaneously, often using multitasking or multithreading within the operating system.

3.2 Components of Server Program Anatomy

1. Initialization
2. Listening for requests
3. Request handling
4. Business logic
5. Transaction management
6. Response generation

4 Operating System Services

Operating systems provide two categories of services for server applications:

- Basic Services
- Extended Services

4.1 Operating System: Basic and Extended Services for Server Applications

Server applications require specific services from their operating systems (OS) to function efficiently.

4.1.1 Basic Services

These are standard features of the OS:

- **Multitasking/multithreading:** Essential for handling numerous concurrent client requests.
- **Memory management:** Efficient allocation and management of memory for performance and stability.

- **Inter-process communication (IPC):** Mechanisms for different parts of the server program or different programs to communicate.
- **Task prioritization:** The ability to prioritize important client requests or background tasks.

Basic OS Services include:

- Process management
- Memory management
- File systems
- Networking
- Device management

4.1.2 Extended Services

These are modular, add-on software components layered on top of the base OS:

- **Database management systems (DBMS):** Software for managing data storage and retrieval.
- **Transaction processing monitors (TPMs):** Ensure the integrity of transactions across a network.
- **Middleware:** Software that facilitates communication between diverse clients and servers.
- **Security services:** Advanced authentication and authorization mechanisms like directory services (e.g., Active Directory).

Extended OS Services include:

- Multithreading
- Security
- Transaction support
- Fault tolerance
- Load balancing

5 Server Scalability

Scalability is the ability of a system to handle a growing amount of work, such as an increased number of clients or requests.

5.1 Types of Scalability

- Vertical scalability (Scale-up)
- Horizontal scalability (Scale-out)
- Load balancing

5.2 Detailed Description

- **Vertical Scalability:** Increasing the capacity of a single server by adding more resources (e.g., more CPU power, more RAM, faster storage).
- **Horizontal Scalability:** Adding more servers to the system and distributing the workload among them, often using techniques like load balancing.
- **Partitioning/Replication:** Servers can be logically partitioned by function or data, or replicated for redundancy and performance.

6 Client Anatomy

A client program is the consumer of services, initiating requests to a server.

6.1 Key Characteristics

- **User interface:** The client is typically responsible for the presentation layer, providing the interface with which the end user interacts.
- **Initiates communication:** The client starts the interaction by sending a request to the server.
- **Receives and presents data:** It receives the server's response and presents the information in a human-readable format.
- **Can be diverse:** Clients can run on various devices and operating systems (desktops, laptops, mobile phones, IoT devices, web browsers).

6.2 Components of Client Anatomy

- User Interface
- Presentation logic
- Communication module
- Local processing
- Session handling

7 Transactions in Client/Server

Transactions follow the ACID properties:

- **Atomicity**
- **Consistency**
- **Isolation**
- **Durability (ACID)**

8 Client/Server Hybrids

In many real-world scenarios, a single machine or application can function as both a client and a server, creating a hybrid model.

8.1 Examples of Hybrid Models

- **Peer-to-peer (P2P) systems:** A classic example where each node can act as both a client (requesting files) and a server (providing files).
- **Multi-tier architecture:** In a 3-tier system, the application server acts as a client to the database server, while acting as a server to the presentation layer client.
- **Local communication:** A client and server can run on the same physical computer and communicate with each other.

9 Summary

- Clients handle interaction
- Servers handle shared resources
- OS ensures performance and scalability