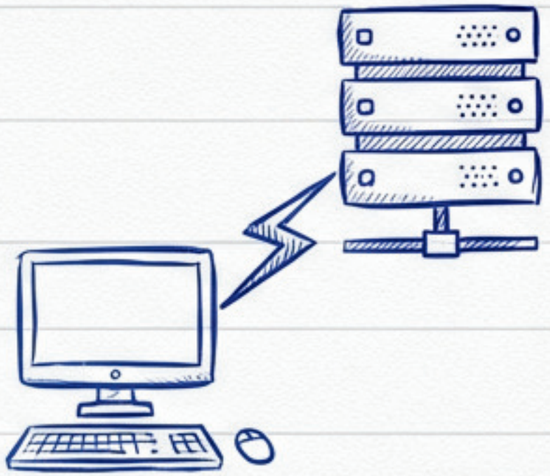
CSA 6008T - BCA

CLIENT SERVER COMPUTING

Introduction to Client/Server computing & Architecture Notes

Introduction, Basic Model, Server Types, Client Types, Topologies, Architecture, Middleware, MVC, Databases.

Notes by:
Kamal Kishor



1. Introduction to Client/Server Computing

Client/Server Computing is a distributed computing model where tasks are divided between 'clients' (requesters) and 'servers' (service providers).

- **Client**

A client is a computer or program that requests services.

  - Example: Web browser, email client.

- **Server**

A server is a powerful system that provides services.

  - Example: Web server, database server.

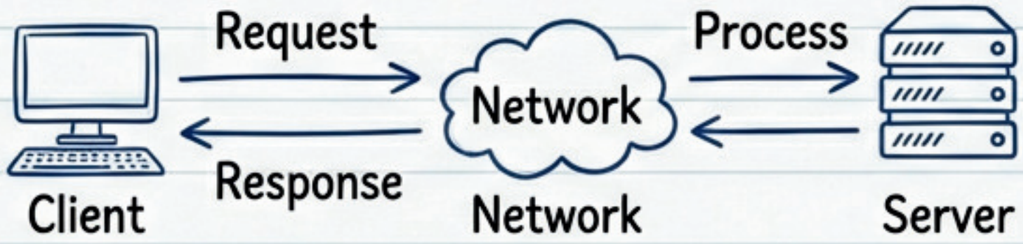
Key Idea

Client sends request → Server processes → Server sends response

2. Basic Client/Server Computing Model

Components:

1. Client: User interface & request generator
2. Server: Processes requests
3. Network: Medium for communication



Working:

- Client sends request using network
- Server processes the request
- Server sends result back to client

Advantages:

- Centralized data, • Better security,
- Easy maintenance, • Scalability.

3. Server for Every Client Model

- Each client is assigned a dedicated server.
- No sharing between clients.

Disadvantages:

- High cost
- Poor scalability
- Wastes resources

4. Types of Servers (Part 1)

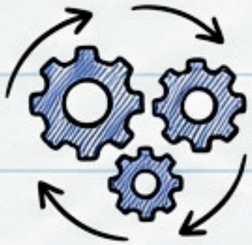
4.1 File Server

- Stores files centrally.
- Clients download files.
- Example: Shared folders.

4.2 Print Server

- Manages printers.
- Handles print jobs from multiple clients.

4.3 Application Server



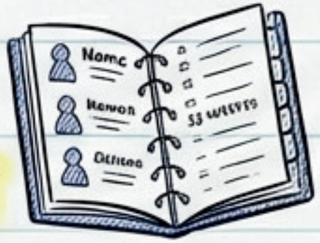
- Runs application logic.
- Reduces load on client.
- Example: Java EE Server.

4.4 Mail Server

- Manages email sending/receiving.
- Example: SMTP, POP3, IMAP.

4. Types of Servers (Part II)

4.5 Directory Services Server



- Stores user & resource information.
- Example: LDAP, Active Directory.

4.6 Web Server

- Delivers web pages.
- Uses HTTP/HTTPS.
- Example: Apache, Nginx.

4.7 Database Server

- Stores & manages databases.
- Example: MySQL, Oracle.

4.8 Transaction Server

- Handles transactions.
- Ensures ACID properties.
- Used in banking systems.

5. Client Types: Fat Client vs Thin Client

Fat Client

- Processing done on client.
- Less dependency on server.
- Example: Desktop software.

Thin Client

- Processing done on server.
- Client is lightweight.
- Example: Browser-based apps.

Feature	Fat Client	Thin Client
Processing	Client	Server
Cost	High	Low
Maintenance	Difficult	Easy

6. Stateless vs Stateful Servers

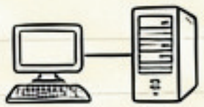
Stateless Server: No client info stored. Each request is independent. Example: HTTP.

Stateful Server: Maintains client session data. More complex. Example: Online banking session.

7. Client/Server Types & Alternatives

7. Client/Server Types

- **7.1 Single Client / Single Server:** One client connects to one server.



- **7.2 Multiple Clients / Single Server:** Many clients use one server. Common in real systems.

- **7.3 Multiple Clients / Multiple Servers:** Load distributed across servers. High scalability.

8. Integration with Distributed Computing

- Client/Server is part of distributed systems.

- Resources shared over network.
- Improves performance and reliability.

9. Alternatives to Client/Server Systems

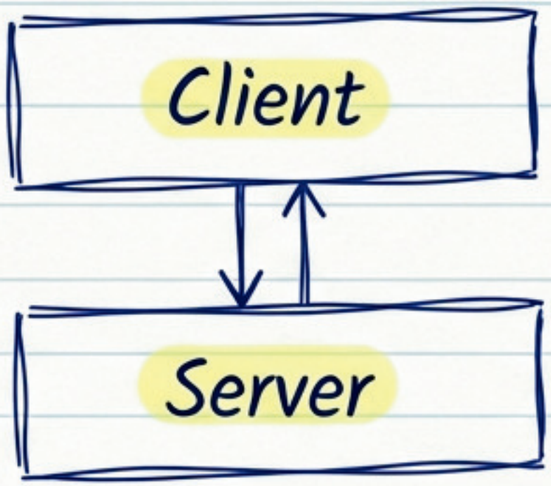
- **Peer-to-Peer (P2P)**

- **Mainframe Computing**

- **Cloud Computing**

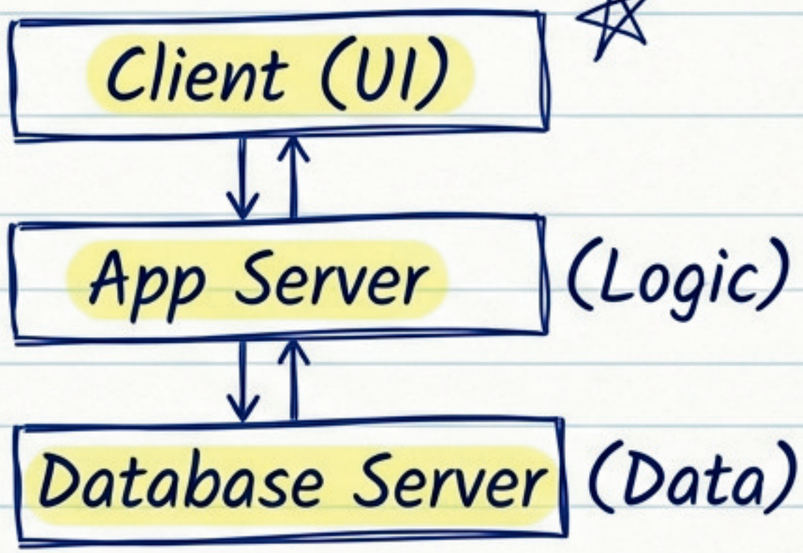
10. Classification of Client/Server Systems

10.1 Two-Tier Architecture



- Client ↔ Server.
- Client handles UI & logic.
- Server handles database.

10.2 Three-Tier Architecture



Advantages:

- Better security
- Easy scalability
- Separation of concerns.

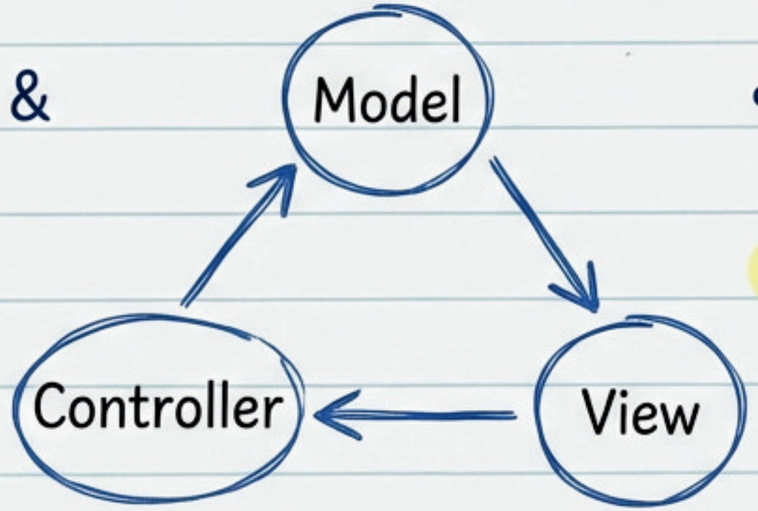
11. Middleware

Middleware is software that acts as a bridge between client and server.

- Functions: Communication, Data translation, Security, Transaction management.
- Examples: RPC, CORBA, Message Queues.

12. MVC (Model View Controller)

- Model: Handles data & business logic.
- Controller: Handles user input.



- View: User interface.

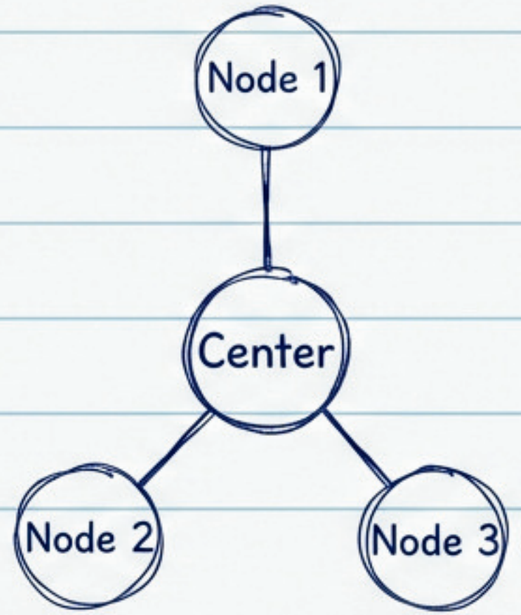
Advantage: Clean separation, Easy maintenance.

13. Principles Behind Client/Server Systems

- ✓ Scalability
- ✓ Reliability
- ✓ Interoperability
- ✓ Transparency
- ✓ Security

14. Client/Server Topologies

- Centralized
- Distributed
- Hybrid



15. Existing Client/Server Architecture

- File-based systems
- Database-based systems
- Web-based systems

16. Client Services

Types:

- Request for services
- Remote Procedure Call (RPC)
- Windows services
- Print services
- Remote boot services
- Utility services

17. Server Functionality (Detailed)

- Request handling
- Resource management
- Security enforcement
- Concurrency control
- Transaction management

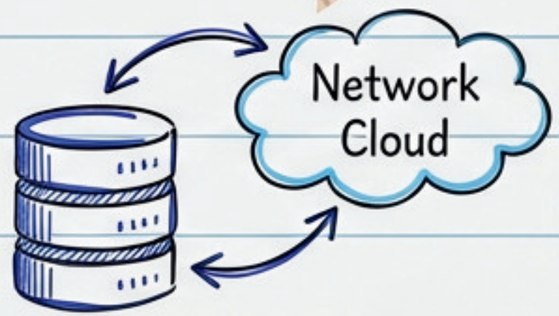
18. Network Client/Server Technology & Databases

Database Storage:

- Centralized DB,
- Distributed DB

Database System Architecture:

- Single-tier,
- Two-tier,
- Three-tier



19. Client/Server in Respect of Databases

Client/Server Databases

- Client sends queries.
- Server executes queries.

Database Computing vs Mainframe (Comparison)

Feature	Client/Server	Mainframe
Cost 	Low	High
Flexibility	High	Low

20. Client/Server Database Architecture

20.1 Process-per-Client Architecture

- One process for each client.
- High overhead.

20.2 Multithreaded Architecture

- Multiple clients handled by threads.
- Efficient resource usage.

20.3 Hybrid Architecture

- Combines both approaches.
- Best performance.

Summary: Advantages & Disadvantages

✓ Advantages of Client/Server Computing

- Resource sharing
- Scalability
- Better security
- Centralized management

✗ Disadvantages

- Server dependency
- Network failure issues
- Initial setup cost