CLIENT SERVER COMPUTING

BCA

IV SEMESTER

THE LEARNING HUB COMPUTER CENTRE

INSPIRE, LEARN, LEAD · LET'S GO

ACCORDING TO **LATEST SYLLABUS**

JAI NARAYAN VYAS UNIVERSITY, JODHPUR

CLIENT SERVER COMPUTING

A COMPLETE TEXTBOOK FOR BCA STUDENTS

COVERS

- Client/Server Fundamentals
- Client/Server Types & Distributed Computing
- Architectures, Middleware & MVC
- Client Services, RPC & Server Functionality
- Client/Server Database Computing & Architectures

EASY EXPLANATION | CONCEPTS WITH EXAMPLES | EXAM FOCUSED | DIAGRAMS & TABLES FOR BETTER UNDERSTANDING

UNIT – I

Introduction to Client/Server Computing

आज के आधुनिक कंप्यूटर नेटवर्क और इंटरनेट आधारित अनुप्रयोगों की नींव **Client-Server Computing Model** पर आधारित है। जब भी हम कोई वेबसाइट खोलते हैं, ई-मेल भेजते हैं, ऑनलाइन बैंकिंग करते हैं या सोशल मीडिया का उपयोग करते हैं, तब हम वास्तव में Client-Server Architecture का उपयोग कर रहे होते हैं।

इस यूनिट में हम Client-Server Computing की मूल अवधारणाओं, विभिन्न प्रकार के Servers तथा Client और Server के विभिन्न मॉडलों का अध्ययन करेंगे।

1. Introduction to Client/Server Computing

(क्लाइंट/सर्वर कम्प्यूटिंग का परिचय)

Client/Server Computing क्या है?

Client/Server Computing एक ऐसा कम्प्यूटिंग मॉडल है जिसमें कार्यों को दो भागों में विभाजित किया जाता है:

1. **Client (क्लाइंट)** – सेवा या जानकारी का अनुरोध करता है।
2. **Server (सर्वर)** – अनुरोध को स्वीकार करता है और आवश्यक सेवा प्रदान करता है।

सरल उदाहरण

जब आप अपने ब्राउज़र में कोई वेबसाइट खोलते हैं:

1. आपका ब्राउज़र (Client) वेबसाइट का अनुरोध भेजता है।
2. Web Server उस अनुरोध को स्वीकार करता है।
3. Server आवश्यक डेटा वापस भेजता है।
4. Browser उस डेटा को प्रदर्शित करता है।

परिभाषा

Client/Server Computing एक ऐसी प्रणाली है जिसमें एक कंप्यूटर सेवा का अनुरोध करता है और दूसरा कंप्यूटर वह सेवा प्रदान करता है।

Client और Server के उदाहरण

Client	Server
Web Browser	Web Server
Email Application	Mail Server
Banking App	Database Server
Mobile App	Application Server

Client/Server Computing की विशेषताएँ

1. Resource Sharing

संसाधनों को अनेक उपयोगकर्ताओं द्वारा साझा किया जा सकता है।

2. Centralized Management

डेटा और सेवाओं का नियंत्रण एक केंद्रीय स्थान से किया जाता है।

3. Scalability

आवश्यकता के अनुसार नए Clients या Servers जोड़े जा सकते हैं।

4. Security

डेटा सुरक्षा को बेहतर तरीके से लागू किया जा सकता है।

5. Reliability

यदि कोई Client विफल हो जाए तो पूरा सिस्टम प्रभावित नहीं होता।

Client/Server Architecture का चित्र

```
Client 1
    \
Client 2 ----> Server ----> Database
    /
Client 3
```

2. Basic Client/Server Computing Model

(मूल क्लाइंट-सर्वर मॉडल)

Client/Server Model में कार्य निम्न चरणों में होता है:

चरण 1:

Client Server को Request भेजता है।

चरण 2:

Server Request को Process करता है।

चरण 3:

Server Client को Response भेजता है।

उदाहरण

यदि कोई उपयोगकर्ता बैंक खाते का बैलेंस देखना चाहता है:

```
User → Banking App → Bank Server → Database
                          ↓
                     Account Data
                          ↓
Bank Server → Banking App → User
```

Components of Client/Server Model

1. Client

सेवा का अनुरोध करता है।

2. Server

सेवा प्रदान करता है।

3. Network

Client और Server के बीच संचार स्थापित करता है।

4. Database

डेटा को संग्रहीत करता है।

3. Server for Every Client

(प्रत्येक क्लाइंट के लिए सर्वर)

Client-Server Architecture में एक Server अनेक Clients को सेवाएँ प्रदान कर सकता है।

उदाहरण:

एक विश्वविद्यालय के Server से:

- छात्र Result देख सकते हैं।
- शिक्षक Attendance दर्ज कर सकते हैं।
- प्रशासनिक कर्मचारी Reports तैयार कर सकते हैं।

विशेषताएँ

- एक Server अनेक Clients को संभाल सकता है।
- Server Requests को क्रमबद्ध तरीके से Process करता है।
- आधुनिक Servers एक साथ हजारों Clients को सेवा प्रदान कर सकते हैं।

4. File Server

(फाइल सर्वर)

File Server क्या है?

File Server ऐसा Server होता है जो नेटवर्क में Files को संग्रहीत, प्रबंधित और साझा करने का कार्य करता है।

कार्य

- File Storage
- File Sharing
- Backup
- Access Control

उदाहरण

किसी कॉलेज में:

- सभी छात्रों की Assignment Files
- Faculty Documents
- Examination Records

एक File Server पर संग्रहीत किए जा सकते हैं।

लाभ

- डेटा का केंद्रीकृत प्रबंधन।
- Backup आसान।
- Data Sharing सरल।

हानियाँ

- Server Failure होने पर Files उपलब्ध नहीं होंगी।
- अधिक Traffic होने पर गति कम हो सकती है।

5. Print Server

(प्रिंट सर्वर)

Print Server क्या है?

Print Server ऐसा Server है जो नेटवर्क में उपलब्ध Printers को नियंत्रित और प्रबंधित करता है।

कार्य

- Print Requests स्वीकार करना।
- Print Queue बनाए रखना।
- विभिन्न Users के Print Jobs को नियंत्रित करना।

उदाहरण

यदि किसी कार्यालय में 50 कर्मचारी एक ही Printer का उपयोग करते हैं, तो Print Server सभी Print Requests को क्रमबद्ध करता है।

लाभ

- Printer Sharing संभव।
- Print Queue Management।
- लागत में कमी।

6. Application Server

(एप्लिकेशन सर्वर)

Application Server क्या है?

Application Server ऐसा Server है जो Applications को चलाने और Business Logic को Process करने का कार्य करता है।

उदाहरण

- Online Banking
- E-Commerce Website
- University ERP System

कार्य

- Application Execution
- Data Processing
- Business Logic Management

लाभ

- Performance बेहतर होती है।
- Centralized Management संभव।

7. Mail Server

(मेल सर्वर)

Mail Server क्या है?

Mail Server ई-मेल संदेशों को भेजने, प्राप्त करने और संग्रहीत करने का कार्य करता है।

मुख्य कार्य

- Email भेजना।
- Email प्राप्त करना।

- Mail Storage।

प्रमुख Protocols

SMTP

Email भेजने के लिए।

POP3

Email प्राप्त करने के लिए।

IMAP

Server पर Email प्रबंधन के लिए।

उदाहरण

जब आप Gmail का उपयोग करते हैं, तो Mail Server आपकी Emails को संभालता है।

8. Directory Services Server

(डायरेक्टरी सर्विस सर्वर)

यह Server उपयोगकर्ताओं, समूहों और नेटवर्क संसाधनों की जानकारी संग्रहीत करता है।

कार्य

- User Authentication
- User Information Management
- Resource Information Storage

उदाहरण

किसी कंपनी में कर्मचारियों के:

- नाम
- विभाग
- ई-मेल
- अधिकार

Directory Server में संग्रहित होते हैं।

9. Web Server

(वेब सर्वर)

Web Server क्या है?

Web Server वह Server है जो Web Pages को संग्रहित करता है और उपयोगकर्ताओं को प्रदान करता है।

कार्य

- Web Pages उपलब्ध कराना।
- HTTP Requests को स्वीकार करना।
- Response भेजना।

कार्यप्रणाली

Browser (Client)

↓ HTTP Request

Web Server

↓

HTML Page

↓ HTTP Response

Browser

उदाहरण

जब कोई उपयोगकर्ता वेबसाइट खोलता है, तो Browser Web Server से डेटा प्राप्त करता है।

10. Database Server

(डेटाबेस सर्वर)

Database Server क्या है?

Database Server डेटा को संग्रहित, प्रबंधित और उपलब्ध कराने का कार्य करता है।

उदाहरण

- बैंक खातों की जानकारी।
- छात्र परिणाम।
- कर्मचारी रिकॉर्ड।

लाभ

- Data Security
- Centralized Storage
- Fast Access

उदाहरण

किसी बैंक में लाखों ग्राहकों की जानकारी Database Server में संग्रहित होती है।

11. Transaction Server

(ट्रांजैक्शन सर्वर)

Transaction Server उन कार्यों को संभालता है जिनमें डेटा का सुरक्षित और विश्वसनीय आदान-प्रदान आवश्यक होता है।

उदाहरण

- Online Banking
- ATM Transactions
- Railway Reservation

मुख्य विशेषताएँ

- Data Consistency
- Reliability
- Security

12. Client/Server – Fat or Thin

(फैट और थिन क्लाइंट)

Fat Client

Fat Client में अधिकांश Processing Client Computer पर होती है।

विशेषताएँ

- अधिक Processing Power आवश्यक।
- Server पर कम भार।
- Offline कार्य संभव।

उदाहरण

Desktop Accounting Software

लाभ

- तेज प्रतिक्रिया।
- Network पर कम निर्भरता।

हानियाँ

- Maintenance कठिन।
- Hardware लागत अधिक।

Thin Client

Thin Client में अधिकांश Processing Server पर होती है।

विशेषताएँ

- Client केवल Interface प्रदान करता है।
- Processing Server पर होती है।

उदाहरण

Web Browser आधारित Applications।

लाभ

- Maintenance आसान।
- Hardware लागत कम।

हानियाँ

- Server पर अधिक भार।
- Network पर निर्भरता अधिक।

Fat Client और Thin Client में अंतर

Fat Client	Thin Client
Processing Client पर	Processing Server पर
Hardware शक्तिशाली चाहिए	साधारण Hardware पर्याप्त
Maintenance कठिन	Maintenance आसान
Offline कार्य संभव	Offline कार्य कठिन

13. Stateless and Stateful Servers

(स्टेटलेस और स्टेटफुल सर्वर)

Stateless Server

Stateless Server प्रत्येक Request को स्वतंत्र रूप से Process करता है।

यह पिछली Requests की जानकारी नहीं रखता।

उदाहरण

HTTP Protocol

यदि उपयोगकर्ता दो Requests भेजता है, तो Server उन्हें अलग-अलग Requests मानता है।

लाभ

- सरल डिजाइन।
- Scalability बेहतर।

हानियाँ

- Session Management कठिन।

Stateful Server

Stateful Server प्रत्येक Client की स्थिति (State) को याद रखता है।

उदाहरण

Online Shopping Website

यदि आपने Cart में कोई वस्तु जोड़ी है, तो Server उस जानकारी को याद रखता है।

लाभ

- Personalization संभव।
- Session Tracking आसान।

हानियाँ

- Memory उपयोग अधिक।
- Server Failure का प्रभाव अधिक।

Stateless और Stateful Server में अंतर

Stateless	Stateful
पिछली जानकारी नहीं रखता	पिछली जानकारी रखता है
कम Memory उपयोग	अधिक Memory उपयोग
Scalability बेहतर	Scalability कम
सरल Architecture	जटिल Architecture

UNIT – II

1. Client/Server Types

1.1 Single Client / Single Server

यह Client/Server Architecture का सबसे सरल रूप है।

इसमें:

- केवल एक Client होता है।
- केवल एक Server होता है।
- Client Server से सेवा प्राप्त करता है।

उदाहरण

एक कंप्यूटर किसी प्रिंटर सर्वर से जुड़कर प्रिंट निकालता है।

Client -----> Server

विशेषताएँ

- सरल संरचना।
- कम लागत।
- छोटे नेटवर्क के लिए उपयुक्त।

लाभ

- स्थापना आसान।
- Maintenance सरल।

हानियाँ

- Scalability कम।
- Server Failure होने पर पूरी सेवा बंद हो जाती है।

1.2 Multiple Clients / Single Server

इस मॉडल में एक Server अनेक Clients को सेवा प्रदान करता है।

उदाहरण

कॉलेज ERP सिस्टम:

- छात्र Result देखते हैं।
- शिक्षक Attendance भरते हैं।
- प्रशासन रिपोर्ट तैयार करता है।

सभी एक ही Server का उपयोग करते हैं।

Client 1

Client 2 -----> Server

Client 3

Client 4

विशेषताएँ

- Resource Sharing संभव।
- Centralized Data Management।

लाभ

- लागत कम।
- Data Consistency बनी रहती है।

हानियाँ

- Server पर अधिक भार।
- Server Failure से सभी Clients प्रभावित होते हैं।

1.3 Multiple Clients / Multiple Servers

इस मॉडल में अनेक Clients तथा अनेक Servers होते हैं।

उदाहरण

ई-कॉमर्स वेबसाइट:

- Web Server
- Database Server
- Application Server
- Payment Server

Clients

↓

Load Balancer

↓ ↓ ↓

Server1 Server2 Server3

लाभ

- High Availability
- Fault Tolerance
- Scalability

हानियाँ

- अधिक लागत।
- प्रबंधन जटिल।

2. Integration with Distributed Computing

Distributed Computing क्या है?

Distributed Computing में कई Computers मिलकर एक बड़े कार्य को पूरा करते हैं।

Client/Server और Distributed Computing का संबंध

Client/Server Architecture Distributed Computing का एक महत्वपूर्ण भाग है।

उदाहरण

Cloud Computing में:

- Application Server
- Database Server
- Storage Server

अलग-अलग स्थानों पर हो सकते हैं लेकिन उपयोगकर्ता को एक ही सिस्टम जैसा अनुभव होता है।

लाभ

- Processing Speed बढ़ती है।
- Reliability बढ़ती है।
- Load Distribution संभव होता है।

3. Alternatives to Client/Server Systems

कुछ परिस्थितियों में Client/Server Architecture के स्थान पर अन्य प्रणालियों का उपयोग किया जाता है।

3.1 Peer-to-Peer (P2P) System

इसमें प्रत्येक Computer Client और Server दोनों का कार्य करता है।

उदाहरण

File Sharing Systems

लाभ

- Server की आवश्यकता नहीं।
- कम लागत।

हानियाँ

- Security कम।
- Data Management कठिन।

3.2 Centralized Computing

सभी Processing एक केंद्रीय Mainframe Computer पर होती है।

3.3 Cloud Computing

संसाधन इंटरनेट के माध्यम से उपलब्ध कराए जाते हैं।

UNIT – III : Classification of Client/Server Systems

1. Classification of Client/Server Systems

Client/Server Systems को सामान्यतः निम्न आधारों पर वर्गीकृत किया जाता है:

- Processing Distribution
- Data Distribution
- Application Logic Distribution

2. Two-Tier Computing

परिचय

Two-Tier Architecture में दो स्तर होते हैं:

1. Client
2. Database Server

Client <-----> Database Server

कार्यप्रणाली

- Client User Interface प्रदान करता है।
- Database Server Data Storage और Processing करता है।

उदाहरण

Desktop Application और Database Server।

लाभ

- सरल Architecture।
- Fast Communication।

हानियाँ

- Scalability कम।
- Large Systems के लिए उपयुक्त नहीं।

3. Middleware

Middleware क्या है?

Middleware वह Software Layer है जो Client और Server के बीच संचार स्थापित करती है।

कार्य

- Communication Management
- Security
- Data Conversion
- Transaction Management

उदाहरण

- Message Queue
- Application Middleware

लाभ

- विभिन्न Platforms को जोड़ता है।
- System Integration आसान बनाता है।

4. Three-Tier Computing

Three-Tier Architecture आधुनिक Web Applications में सबसे अधिक उपयोग होती है।

तीन स्तर

Presentation Layer

User Interface।

Business Layer

Business Logic।

Database Layer

Data Storage।

Client

↓

Application Server

↓

Database Server

लाभ

- Security बेहतर।
- Scalability अधिक।
- Maintenance आसान।

उदाहरण

Online Banking Systems।

5. Model View Controller (MVC)

MVC एक Software Design Pattern है।

Components

Model

Data तथा Business Logic।

View

User Interface।

Controller

User Input को नियंत्रित करता है।

User

↓

Controller

↓

Model

↓

View

लाभ

- Code Reusability
- Maintenance आसान
- Development तेज

6. Principles Behind Client/Server Systems

Client/Server Systems निम्न सिद्धांतों पर आधारित हैं:

1. Resource Sharing
2. Service Oriented Architecture
3. Scalability
4. Reliability
5. Security
6. Interoperability

7. Client/Server Topologies

Star Topology

सभी Clients Server से जुड़े होते हैं।

Bus Topology

सभी Devices एक Common Cable का उपयोग करती हैं।

Ring Topology

सभी Computers एक वृत्ताकार संरचना में जुड़े होते हैं।

Mesh Topology

प्रत्येक Device अन्य Devices से जुड़ी होती है।

8. Existing Client/Server Architecture

आधुनिक Client/Server Systems निम्न तकनीकों पर आधारित हैं:

- Cloud Computing
- Web Services
- Service Oriented Architecture (SOA)
- Microservices Architecture

UNIT – IV : Client Services and Network Technologies

1. Client Services

Client Services वे सेवाएँ हैं जो Client Side पर उपलब्ध होती हैं।

उदाहरण

- User Interface
- Data Validation
- Input Processing

2. Request for Services

Client किसी Service की आवश्यकता होने पर Server को Request भेजता है।

प्रक्रिया

Client Request

↓

Server Processing

↓

Server Response

3. RPC (Remote Procedure Call)

RPC क्या है?

RPC ऐसी तकनीक है जिसमें Client किसी Remote Server पर उपलब्ध Function को इस प्रकार Call करता है जैसे वह Local Machine पर मौजूद हो।

उदाहरण

Client -----> RPC -----> Server Function

लाभ

- Distributed Systems को सरल बनाता है।
- Programming आसान बनती है।

4. Windows Services

Windows Services वे Programs हैं जो Background में चलते हैं।

उदाहरण

- Print Spooler
- Update Service
- Network Service

विशेषताएँ

- User Login के बिना भी कार्य करते हैं।
- Automatically Start हो सकते हैं।

5. Print Services

Print Services Network Printers को प्रबंधित करती हैं।

कार्य

- Print Queue Management
- Job Scheduling
- Printer Sharing

6. Remote Boot Services

Remote Boot Services द्वारा Computer Network के माध्यम से Operating System Boot कर सकता है।

उपयोग

- Diskless Workstations
- Computer Labs

7. Other Remote Services

- Remote Login
- Remote Desktop
- Remote File Access

8. Utility Services

Utility Services System Management में सहायता करती हैं।

उदाहरण

- Backup Service
- Antivirus Service
- Monitoring Service

9. Server Detailed Functionality

Server के मुख्य कार्य:

- Request Processing
- Data Storage
- Security Management
- Authentication
- Transaction Handling

10. Network Client/Server Technology

यह तकनीक नेटवर्क के माध्यम से Clients और Servers को जोड़ती है।

मुख्य Components

- Network Interface Card
- Switch
- Router
- Protocols

11. Databases – Storing Data

Database डेटा को व्यवस्थित रूप से संग्रहीत करती है।

उदाहरण

- Student Records
- Employee Records
- Banking Transactions

12. Database System Architectures

Centralized Database

डेटा एक स्थान पर होता है।

Distributed Database

डेटा कई स्थानों पर वितरित होता है।

Client/Server Database

डेटा Server पर तथा Applications Client पर होती हैं।

UNIT – V : Client/Server Database Computing

1. Client/Server in Respect of Databases

Client/Server Systems में Database Server सबसे महत्वपूर्ण भूमिका निभाता है।

Client का कार्य

- Query भेजना।
- परिणाम प्राप्त करना।

Server का कार्य

- Query Processing।
- Data Management।
- Security।

2. Client/Server Databases

Client/Server Database Architecture में Database Server और Client Application अलग-अलग होते हैं।

लाभ

- Security बेहतर।
- Performance बेहतर।

- Centralized Management।

3. Client/Server Database Computing

इस मॉडल में:

- Client Application Request भेजती है।
- Database Server Query Execute करता है।
- परिणाम Client को भेजे जाते हैं।

Client

↓ SQL Query

Database Server

↓ Result

Client

4. Database Computing vs Mainframe Computing

Database Computing	**Mainframe Computing**
Distributed Processing	Centralized Processing
Flexible	कम Flexible
Scalability अधिक	Scalability कम
Maintenance आसान	Maintenance कठिन

5. PC/File Server Computing

इस मॉडल में Files Server पर होती हैं लेकिन Processing Client पर होती है।

लाभ

- सरल व्यवस्था।
- कम लागत।

हानियाँ

- Network Traffic अधिक।
- Performance कम।

6. Client/Server Database Architecture

6.1 Process-Per-Client Architecture

इस Architecture में प्रत्येक Client के लिए अलग Process बनाई जाती है।

लाभ

- Isolation बेहतर।
- Reliability अधिक।

हानियाँ

- Memory उपयोग अधिक।

6.2 Multi-Threaded Architecture

एक Server Process कई Threads का उपयोग करके अनेक Clients को संभालता है।

लाभ

- Resource Utilization बेहतर।
- Performance अधिक।

हानियाँ

- Synchronization समस्याएँ उत्पन्न हो सकती हैं।

6.3 Hybrid Architecture

यह Process-Based और Thread-Based Architecture का मिश्रण है।

लाभ

- Performance और Reliability दोनों बेहतर।

उपयोग

- Large Scale Enterprise Systems
- Banking Systems
- Cloud Services