Chapter 4: Client/Server and Databases

Prof. Kirtankumar Rathod
Dept. of Computer Science
Indus University

INTRODUCTION

- Storing Data and the Database
  - Data is information in its simplest form, meaningless until related together in some fashion so as to become meaningful.
  - All data on computer is stored in one kind of database or another.

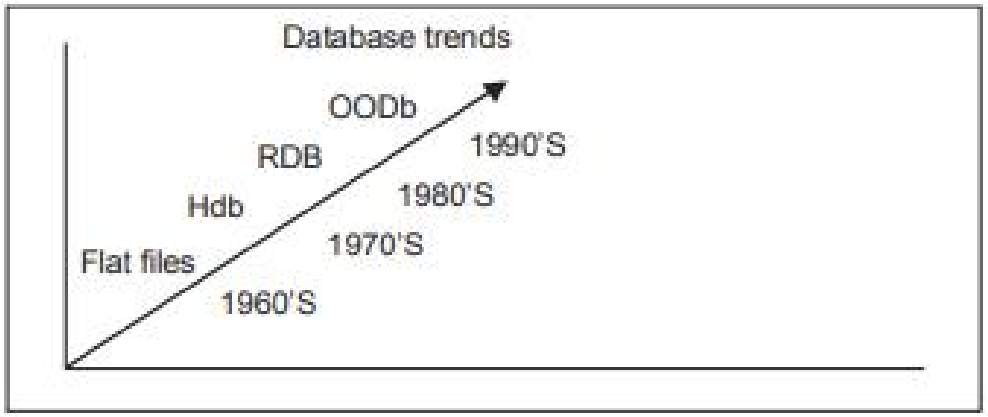

Fig.4.1: Evolution of Database Technologies

Database System Architectures

- Client/Server database system
- Distributed database system
- Parallel database system
- Centralized database system

CLIENT/SERVER IN RESPECT OF DATABASES

- A Database Management System (DBMS) lies at the center of most Client/Server systems in use today.
  - Provide transparent data access to multiple and heterogeneous clients.
  - Process client data requests at the local server.
  - Send only the SQL result to the clients over the network.

Client/Server Database Computing

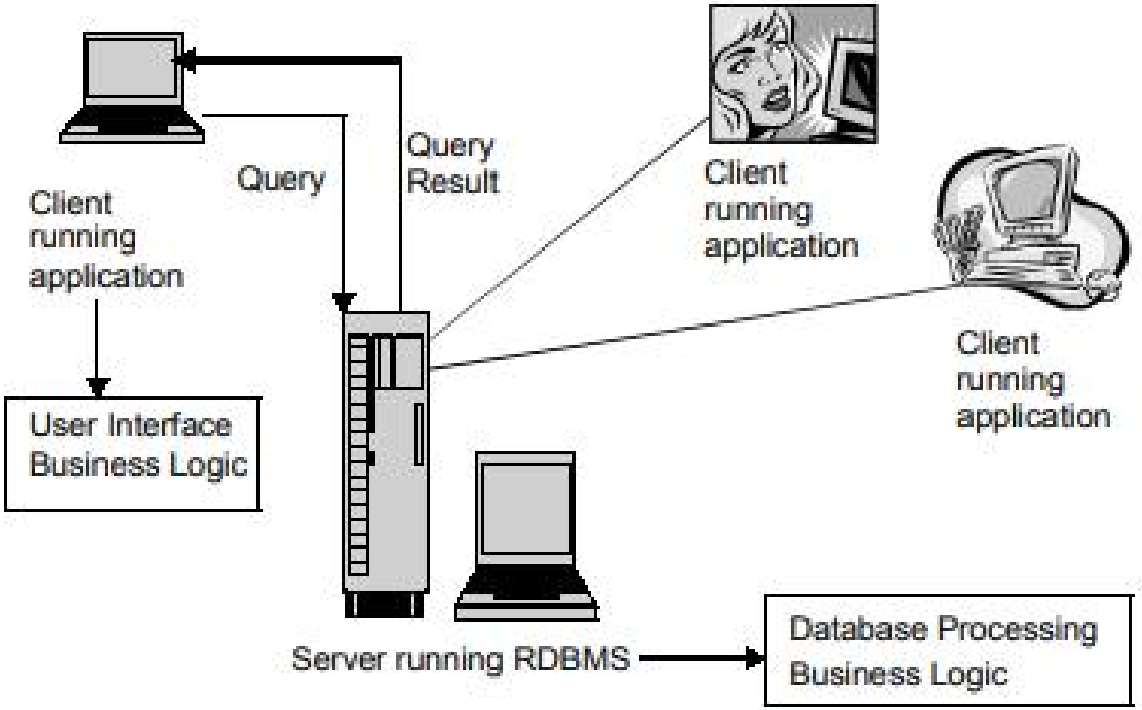

Fig. 4.2: Client/Server Database Computing

Client/Server Database Computing

- Client/Server database computing can be defined as the logical partition of the user interface, database management, and business; logic between the client computer and server computer.
- Business logic can be located on the server, on the client, or mixed between the two.
- Following are the reasons for its popularity.
  - Affordability
  - Speed
  - Adaptability
  - Simplified data access

CLIENT/SERVER DATABASE ARCHITECTURE

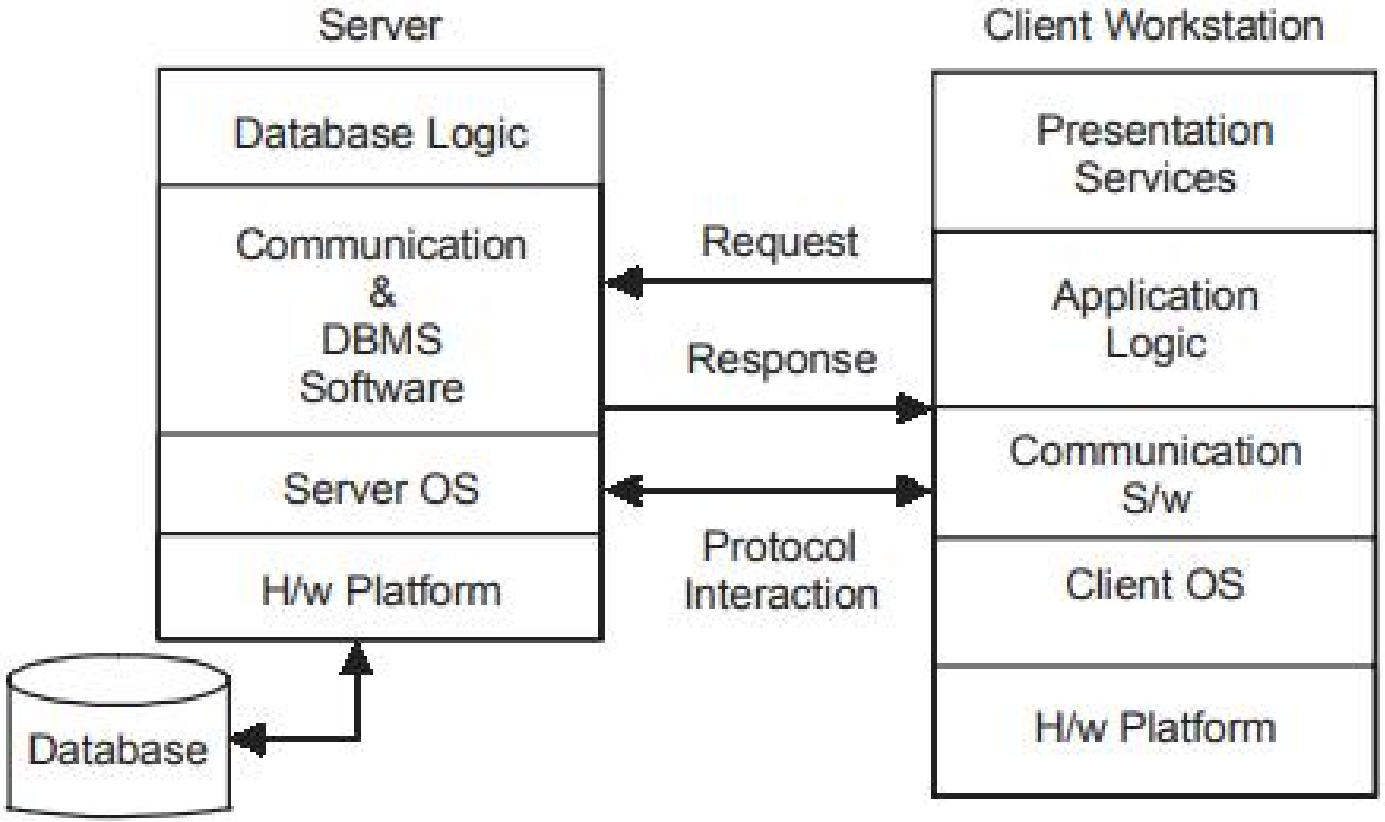

Fig.4.3: Client/Server Database Architecture

Various types of available Client/Server Database Architecture

1. Process-per-client architecture
2. Multi-threaded architecture
3. Hybrid architecture

1. Process-per-client architecture

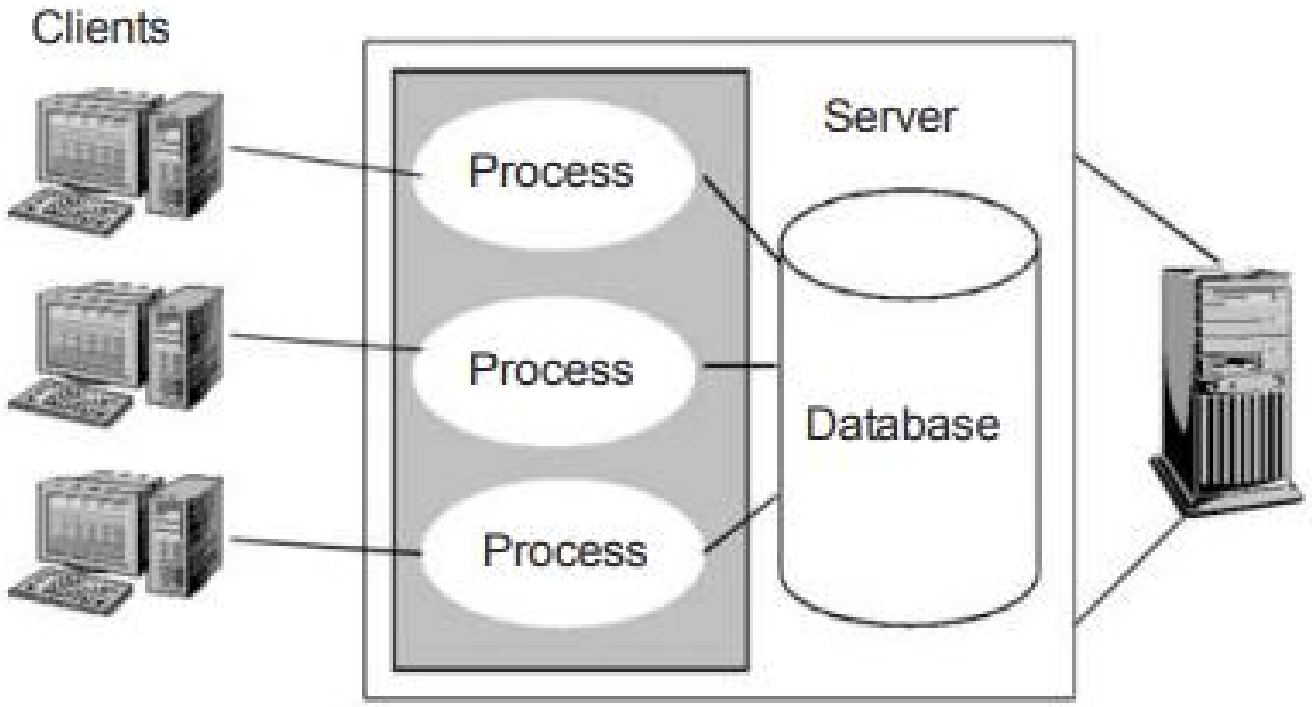

Fig.4.4: Process-per-client Architecture

2. Multi-threaded architecture

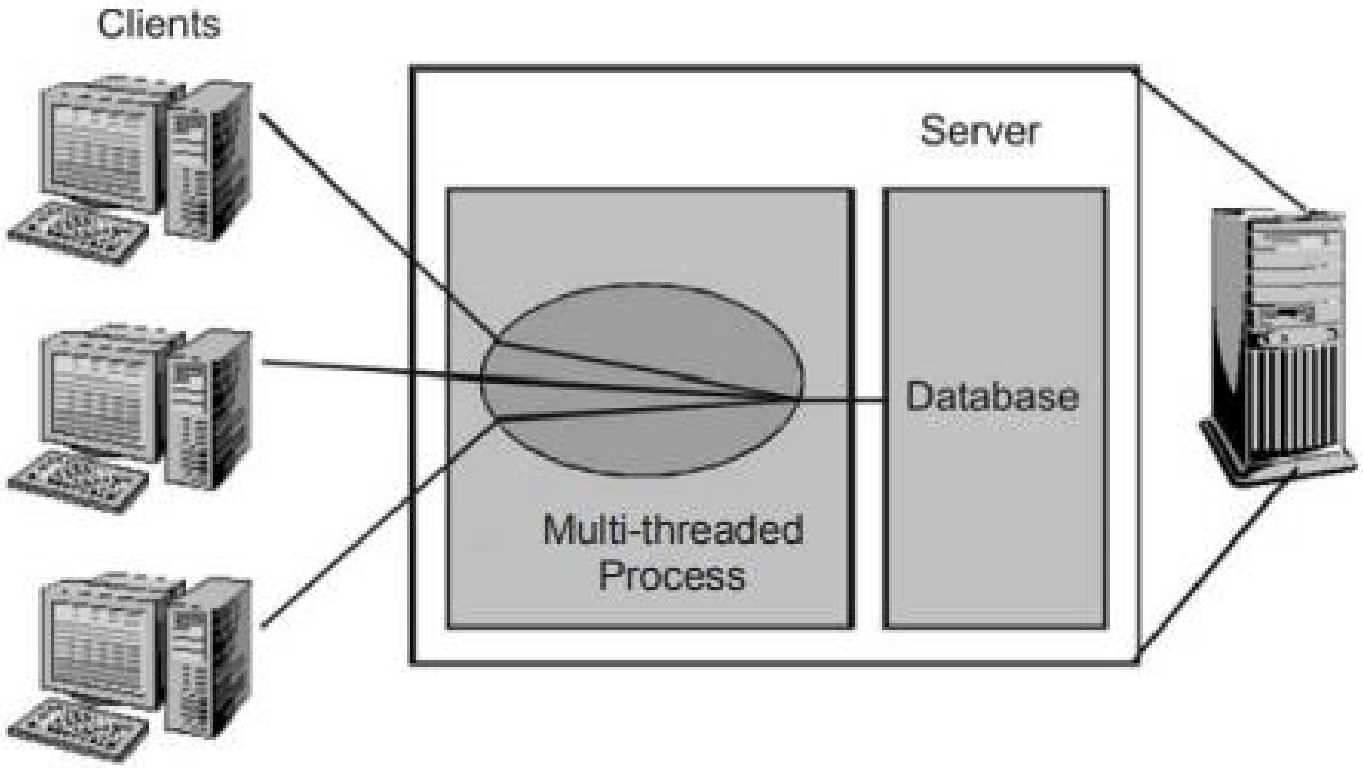

Fig.4.5: Multi-threaded Architecture

3. Hybrid architecture

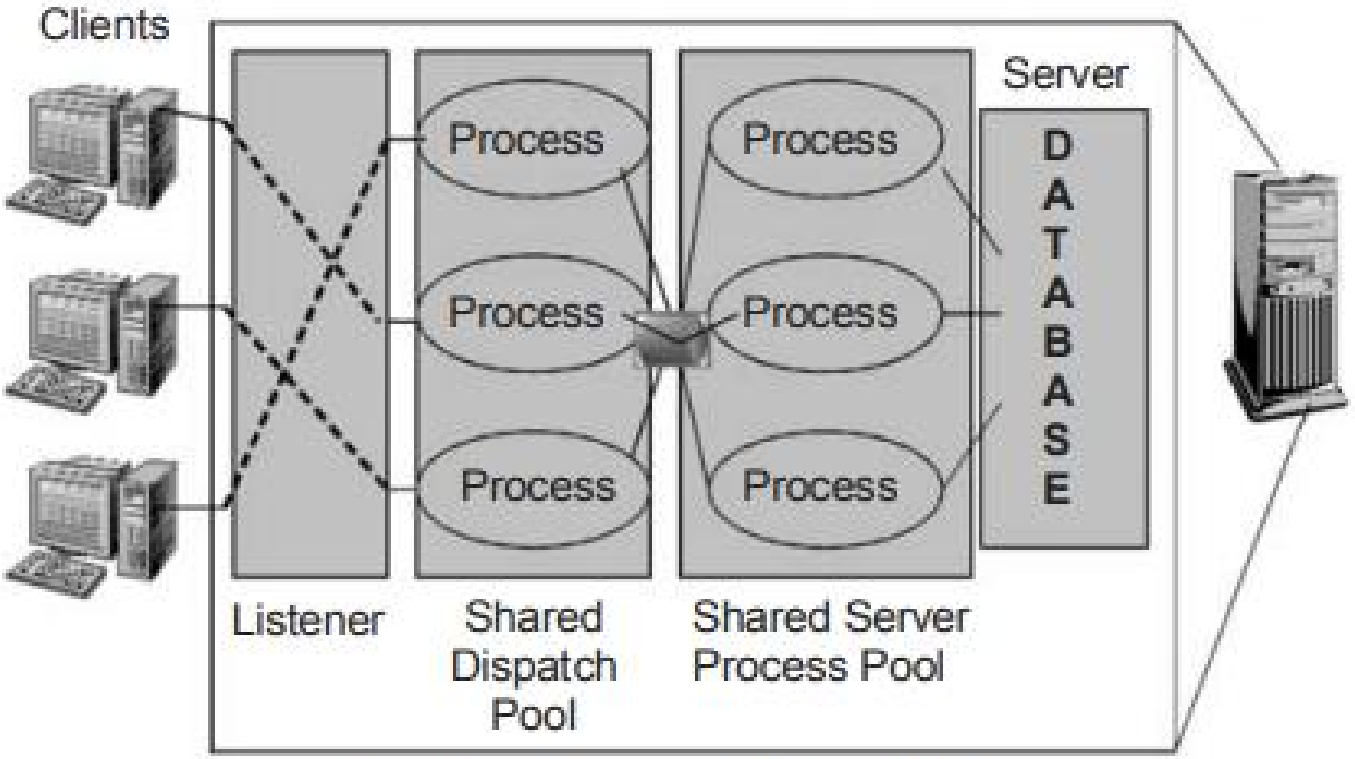

Fig.4.6: Hybrid Architecture

DATABASE MIDDLEWARE COMPONENT

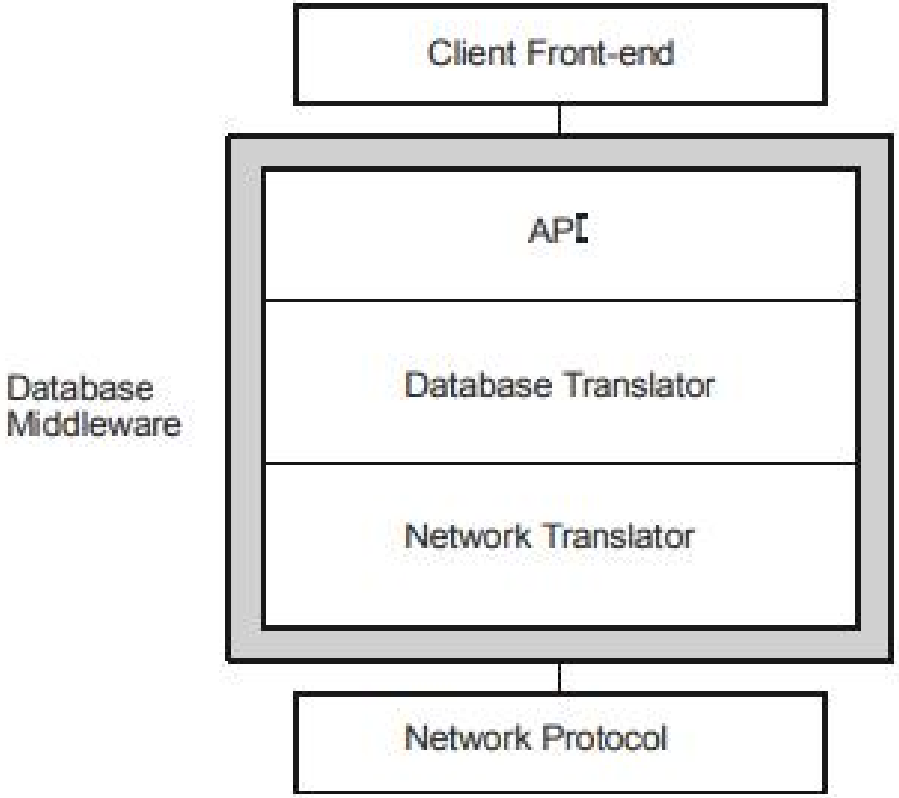

Fig.4.7: Database Middleware Components

Application programming interface

- The application-programming interface is public to the client application.
- The middleware API allows the programmer to write generic SQL code instead of code specific to each database server.
- The server can be changed without requiring that the client applications be completely rewritten.

Database translator

- The database translator translates the SQL requests into the specific database server syntax.
- The database translator layer takes the generic SQL request and maps it to the database server’s SQL protocol.
- If the SQL request uses data from two different database servers, the database translator layer will take care of communicating with each server,

Network translator

- The network translator manages the network communication protocols.
- If a client application taps into the two databases, one that uses TCP/IP and another that uses IPX/SPX, the network layer handles all the communications detail of each database transparently to the client application.

ACCESS TO MULTIPLE DATABASES

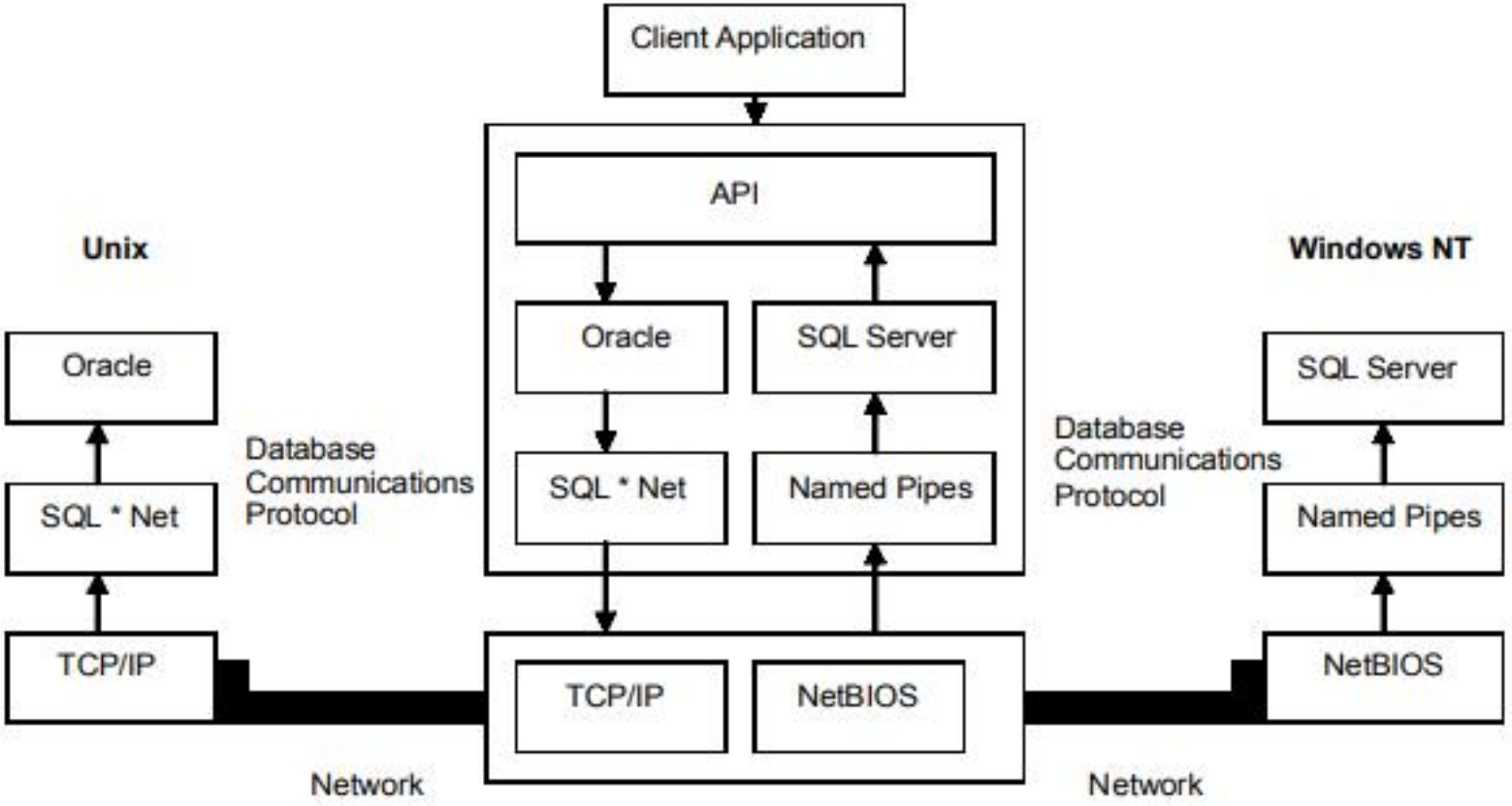

Fig.4.9: Multiple Database Server Access Through Middleware

ACCESS TO MULTIPLE DATABASES

- The Oracle database server uses SQL *Net as its communications protocol with the client; the SQL Server uses Named Pipes as the communications protocol.
- Named Pipes is a inter-process communication (IPC) protocol common to multitasking operating systems such as UNIX.
- The SQL Server runs under the Windows NT operating system and uses NetBIOS as its network protocol.

DISTRIBUTED CLIENT/SERVER DATABASE SYSTEMS

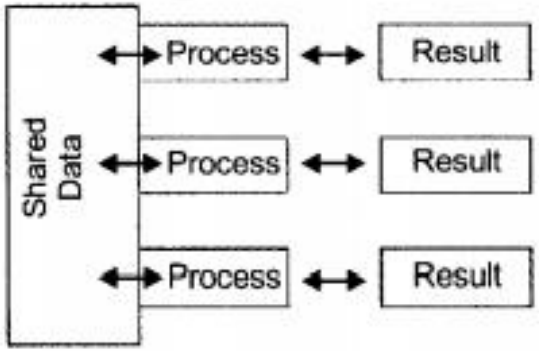

Fig.4.10(*a*): Distributed Data

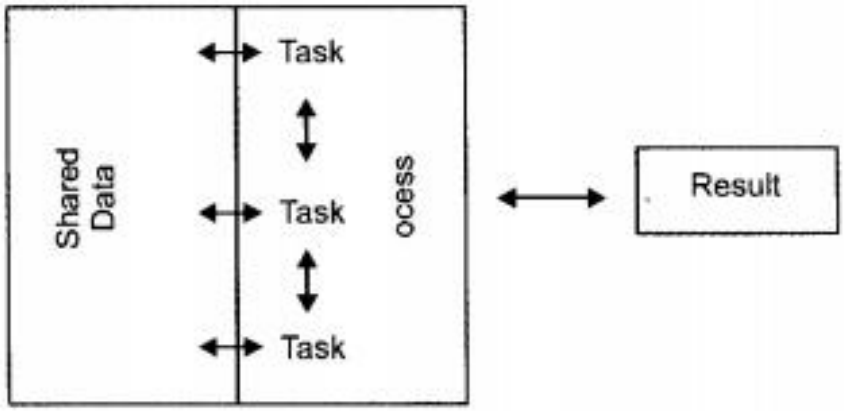

Fig.4.10(*b*): Distributed Processing

DISTRIBUTED CLIENT/SERVER DATABASE SYSTEMS

- Distributed data refers to the basic data stored in the server, which is distributed to different members of the work team.
- While distributed processing refers to the way different tasks are organized among members of the work team.

DISTRIBUTED CLIENT/SERVER DATABASE SYSTEMS

- The data in the database can be partitioned in several ways and process can be centralized, partitioned or replicated in many differe

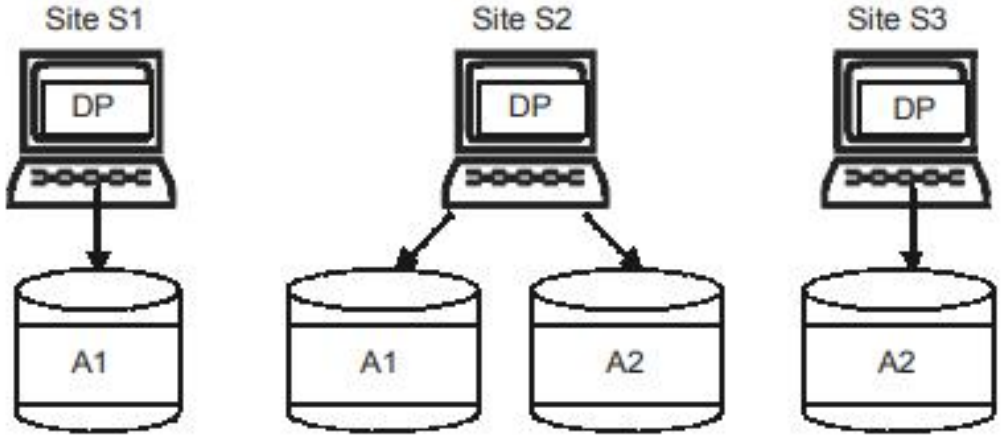

Fig.4.11: Data Replication

DISTRIBUTED DBMS

- The DBMS must provide distributed database transparency features like:
  1. Distribution transparency
  2. Transaction transparency
  3. Failure transparency
  4. Performance transparency
  5. Heterogeneity transparency

DISTRIBUTED DBMS

- Interaction between client and server might proceed as follows during the processing of an SQL query:
  - The client **passes a user query** and decomposes it into a number of independent site queries. Each site query is sent to the appropriate server site.
  - Each **server process the local query** and sends the resulting relation to the client site.
  - The **client site combines the results** of the subqueries to produce the result of the originally submitted query.

DISTRIBUTED DBMS

- In a typical DBMS, it is customary to divide the software module into three levels:
  - **L1:** The **server software** is responsible for **local data management** at site, much like centralized DBMS software.
  - **L2:** The **client software** is responsible for most of **the distributions**; it access data distribution information from the DBMS catalog and process all request that requires access to more than one site. It also handles all user interfaces.
  - **L3:** The **communication software** provides the communication primitives that are used by the client to **transmit commands and data** among the various sites as needed.

WEB/DATABASE SYSTEM FOR CLIENT/SERVER APPLICATIONS

- A client machine that runs a web browser issues a request for information in the form of a URL reference.
- This reference triggers a program at the web server that issues the correct database command to a database server.
- The output returned to the web server is converted into a HTML format and returned to t

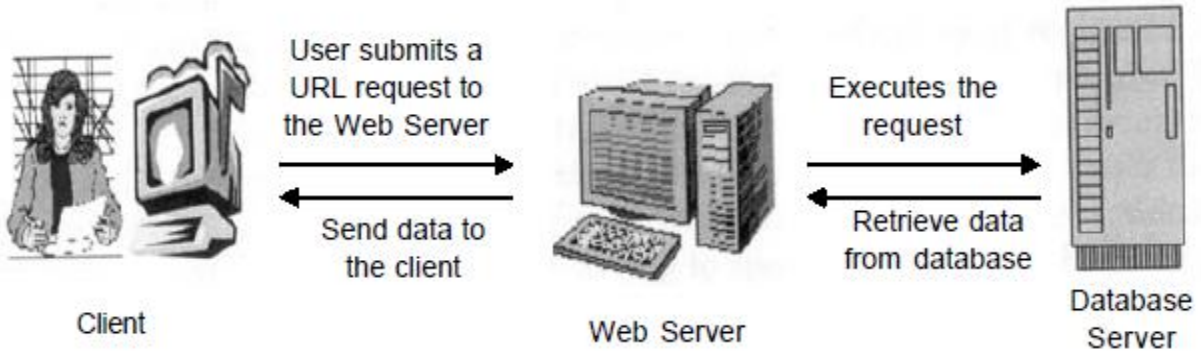

Fig.4.12: Web Database System Integration

Web/Database vs. Traditional Database

- The advantages of a web/database system
  - *Administration*
  - *Deployment*
  - *Speed*
  - *Information presentation*
- The disadvantages of a web/database system
  - *Functionality*
  - *Operations*

Chapter 5: Client/Server Application Components

Prof. Kirtankumar Rathod
Dept. of Computer Science
Indus University

TECHNOLOGIES FOR CLIENT/SERVER APPLICATION

- Rich transaction processing
- Roaming agents
- Rich data management
- Intelligent self-managing entities
- Intelligent middleware

Database centered systems

- Decision-Support Systems (DSS)
- Online Transaction Processing (OLTP)

Groupware

- Multimedia Document Managements (MMDM)
- Workflow
- Scheduling (or Calendaring)
- Conferencing
- Electronic Mail (E-mail)

CATEGORIES OF CLIENT/SERVER APPLICATIONS

1. Host-based processing
2. Server-based processing
3. Client-based processing
4. Cooperative processing

CATEGORIES OF CLIENT/SERVER APPLICATIONS

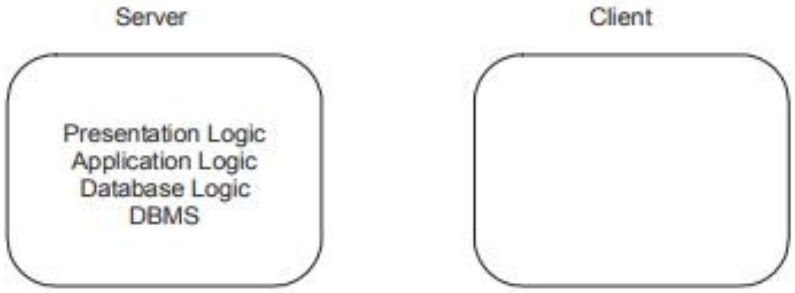

Fig.5.1: Host-base Processing

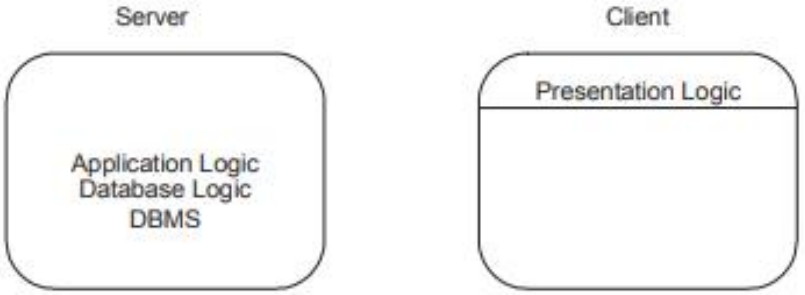

Fig.5.2: Server-base Processing

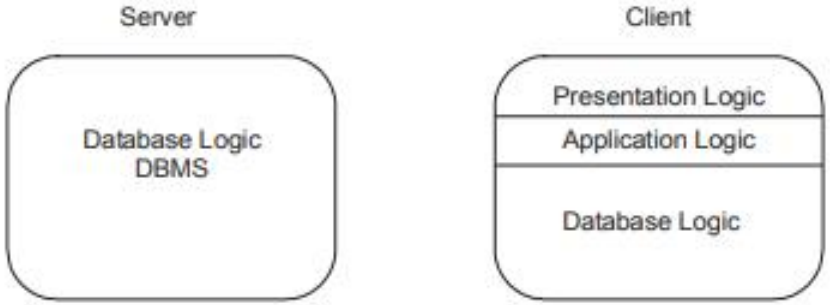

Fig. 5.3: Client-base Processing

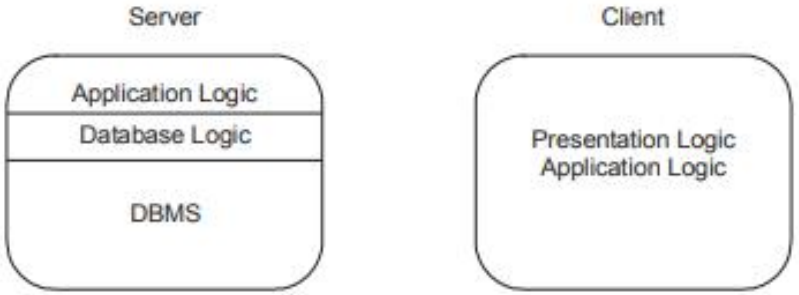

Fig. 5.4: Cooperative Processing

CLIENT SERVICES

- Responsible for managing the user interface.
- Provides presentation services.
- Accepts and checks the syntax of user inputs. User input and final output, if any, are presented at the client workstation.
- Acts as a consumer of services provided by one or more server processors.

1. Inter Process Communication

- The communication between two processes take place via buffer.
  - Message passing.
  - Direct communication.
  - Indirect communication.
  - Remote procedures call.

a) Message passing

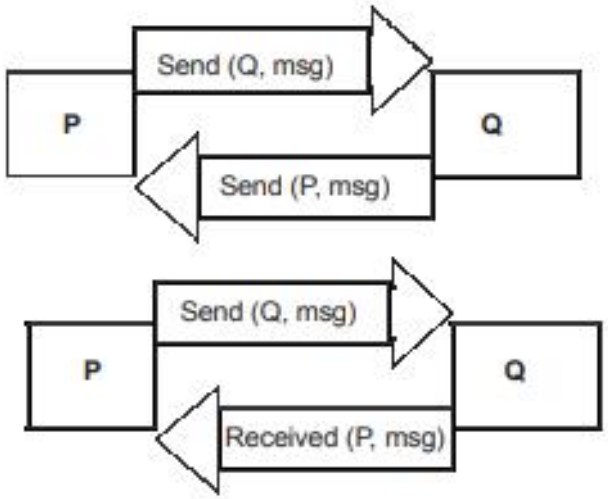

Fig.5.5: Message Passing

b) Indirect Communication

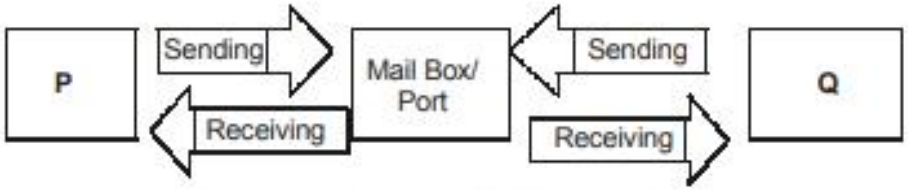

Fig. 5.6: Indirect Communication

c) Remote procedures call

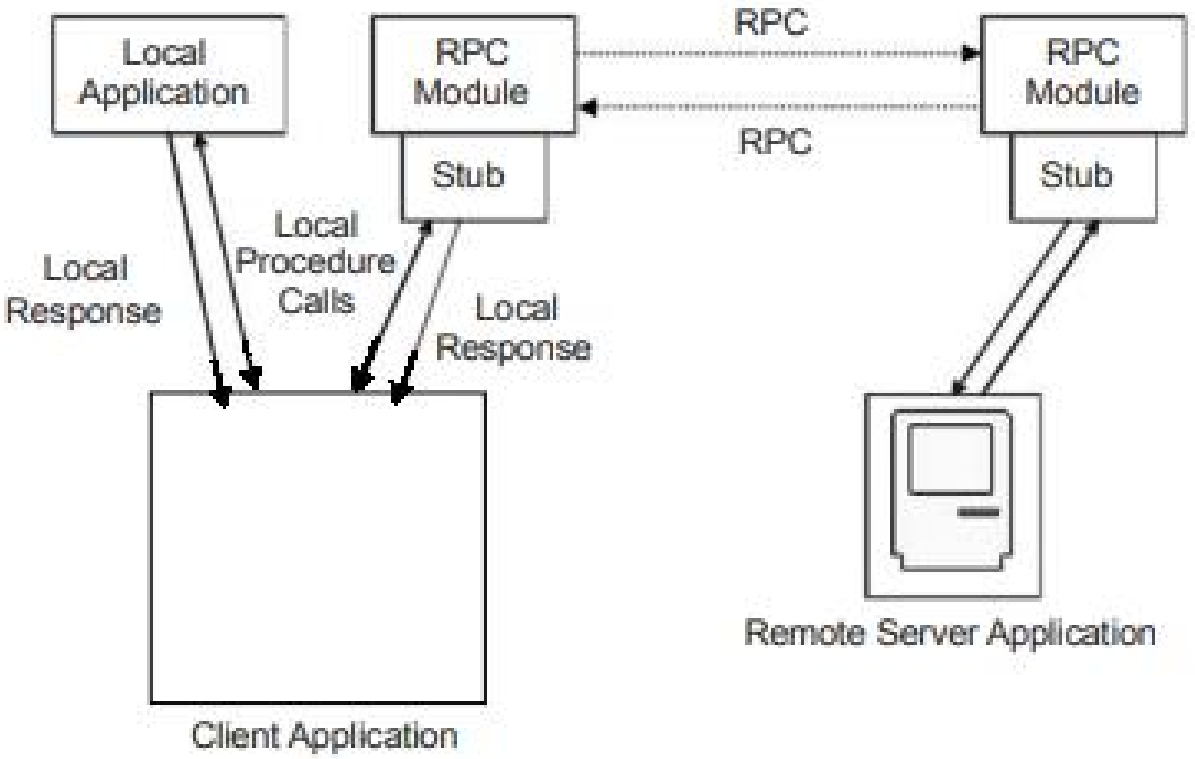

Fig. 5.7: RPC Mechanism

c) Remote procedures call

- A remote procedure is uniquely identified by the triple: (program number, version number, procedure number), the program number identifies a group of related remote procedures, each of which has a unique procedure number.

- A stub procedure is added in the callers users address space.

2. Remote Services

- The workstation is responsible to provide various remote services.
- like remote login, remote command execution, remote backup services, remote tape drive access and remote boot services, and remote data access are important.
- From a client workstation backup services may be invoked remotely.

3. Window Services

- With the help of interface software client application places data into virtual screen, and then the windows services handles manipulation and placement of application windows.
- The client user is fully in grip of his desktop and can give priority to the most important tasks at hand simply by positioning the window of interest to the workstation.
- Print complete, E-mail receipt, Fax available, and application termination are examples of alerts that might generate a pop-up window to

4. Dynamic Data Exchange (DDE)

- DDE is usually described as a conversation between two applications, a client application and a server application.
- For an example, if an application wants to connect a Microsoft Excel spreadsheet with Microsoft Word for windows report in such a way that changes to the spreadsheet are reflected automatically in the report, in that case Microsoft Word for windows is the client and Microsoft Excel is the server.
- A DDE link may be automatic or manual. An automatic link is refreshed whenever the source data changes, provided both the client and server applications are running. A manual link is

5. Object Linking and Embedding (OLE)

- OLE is a software package accesses data created from another through the use of a viewer or launcher.
- These viewers and launchers must be custom built for every application.
- Paintbrush can act only as a server.
- Microsoft has released its OLE 2.0 Software Development Kit (SDK).

6. Common Object Request Broker Architecture (CORBA)

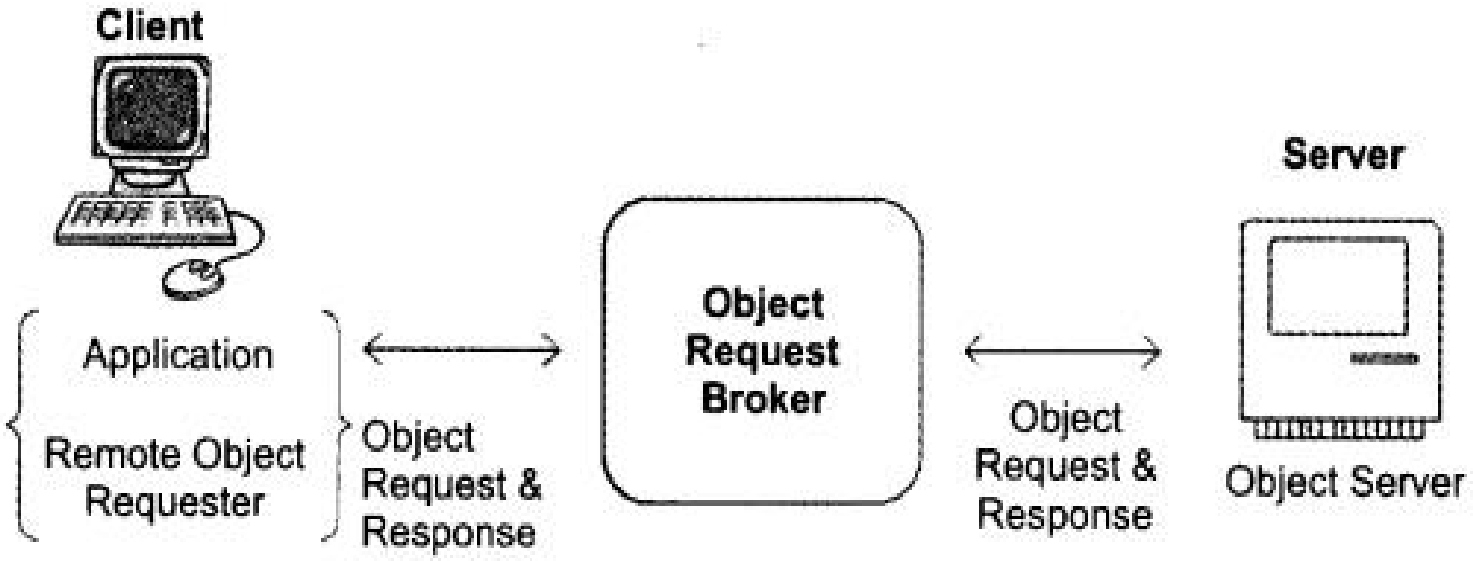

Fig. 5.9: Object Request Broker

6. Common Object Request Broker Architecture (CORBA)

- An object oriented architecture that provides mechanism to share/call the objects over a mixed network.
- A client send request to ORB, which act as a directory of all remote services.
- A CORBA server means the components execute methods for a particular object on behalf of other components (Clients).

7. Print/Fax Services

- Client generates print/fax requests to the printer/fax machine without knowing whether they are free or busy.

- These requests are redirected by the NOS redirector software and managed by the print/fax server queue manager.

8. Database Services

- Client/Server model provides integration of data and services allow clients to be isolated from inherent complexities such as communication protocols.
- Client application submit database request to the server using SQL statements

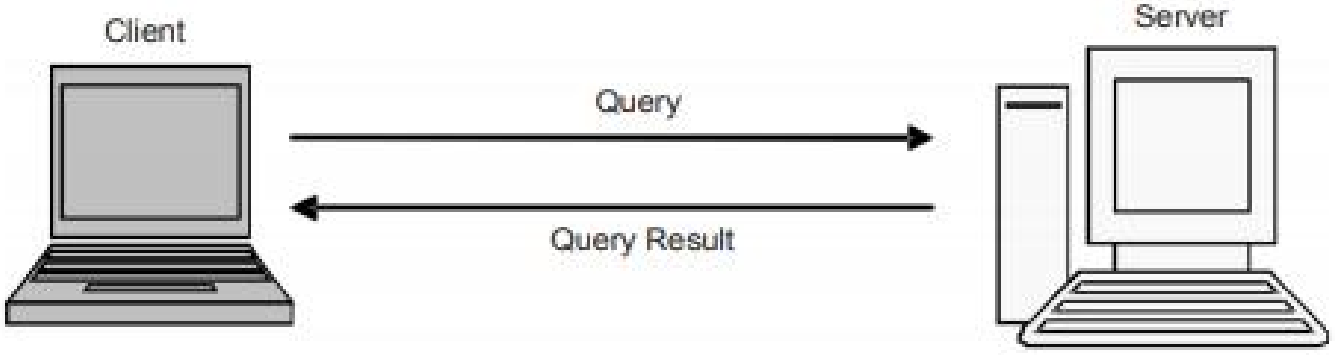

Fig. 5.11: Execution of SQL

SERVER SERVICES

- Some of the main operations that server perform are listed below:
  - Accepts and processes database requests from client.
  - Checks authorization.
  - Ensure that integrity constraints are not violated.
  - Performs query/update processing and transmits response to client.
  - Maintains system catalog.
  - Provide concurrent database access.
  - Provides recovery control.

1. Application services

- Application servers provide business services to support the operation of the client workstation.
- A collection of application servers may work in concert to provide an entire business function.
- On larger and more complicated systems, server responsibility may be distributed among several different types of servers.

2. File services

- A file server can store any type of data, and so on simpler systems, may be the only server necessary.
- Catalog functions are also provided by the file server to support file naming and directory structure.
- File services are responsible to handle access to the virtual directories and files located on the client workstation and to the server’s permanent storage.
- File server manages databases, software’s,

3. Database services

- Concurrency control is managed by the application program, which issues lock requests and lock checks, and by the database server, which creates a lock table that is interrogated whenever a record access lock check is generated.
  - Performance optimization tools.
  - Dynamic transaction backout.
  - Roll back from, roll forward to last backup.
  - Audit file recovery.
  - Automatic error detection and recovery.
  - File reclamation and repair tools.
  - Support for mirrored databases.
  - Capability to split database between physical disk drives.
  - Remote distributed database management features.
  - Maintenance of accurate and duplicate audit files on any LAN node

4. Print/fax/image services

- The server can accept input from many clients, queue it according to the priority of the request and handle it when the device is available.
- In concert with workflow management techniques, images can be captured and distributed to the appropriate client workstation from the image server.

5. Communications services

- Client/server applications require LAN and WAN communication services.
- Basic LAN services are integral to the NOS.
- WAN services are provided by various communications server products.

6. Security systems services

- Every user should be required to log in with a user ID and password.
- The enterprise on the desk implies that a single logon ID and logon sequence is used to gain the authority once to access all information and process for the user has a need and right of access.
- The encryption and decryption are done using the DES algorithm and the user password.
- This ensures that no unauthorized user can access stored data or communications data.

CLIENT/SERVER APPLICATION: CONNECTIVITY

- The communication middleware software provides the means through which clients and servers communicate to perform specific actions.

- The Net BIOS command would allow the client process to establish a session with the database server, send specific control information, send the request, and so on.

Role and Mechanism of Middleware

- Role of middleware component can be exactly understand by the way in which Client/Server computing being used , we know that there are number of approaches are there like host-based processing, server based processing, cooperative processing and client based processing.

- A Middleware component resides on both Client/Server machine enabling an application or user at a client to access a variety of services provided by server.

Role and Mechanism of Middleware

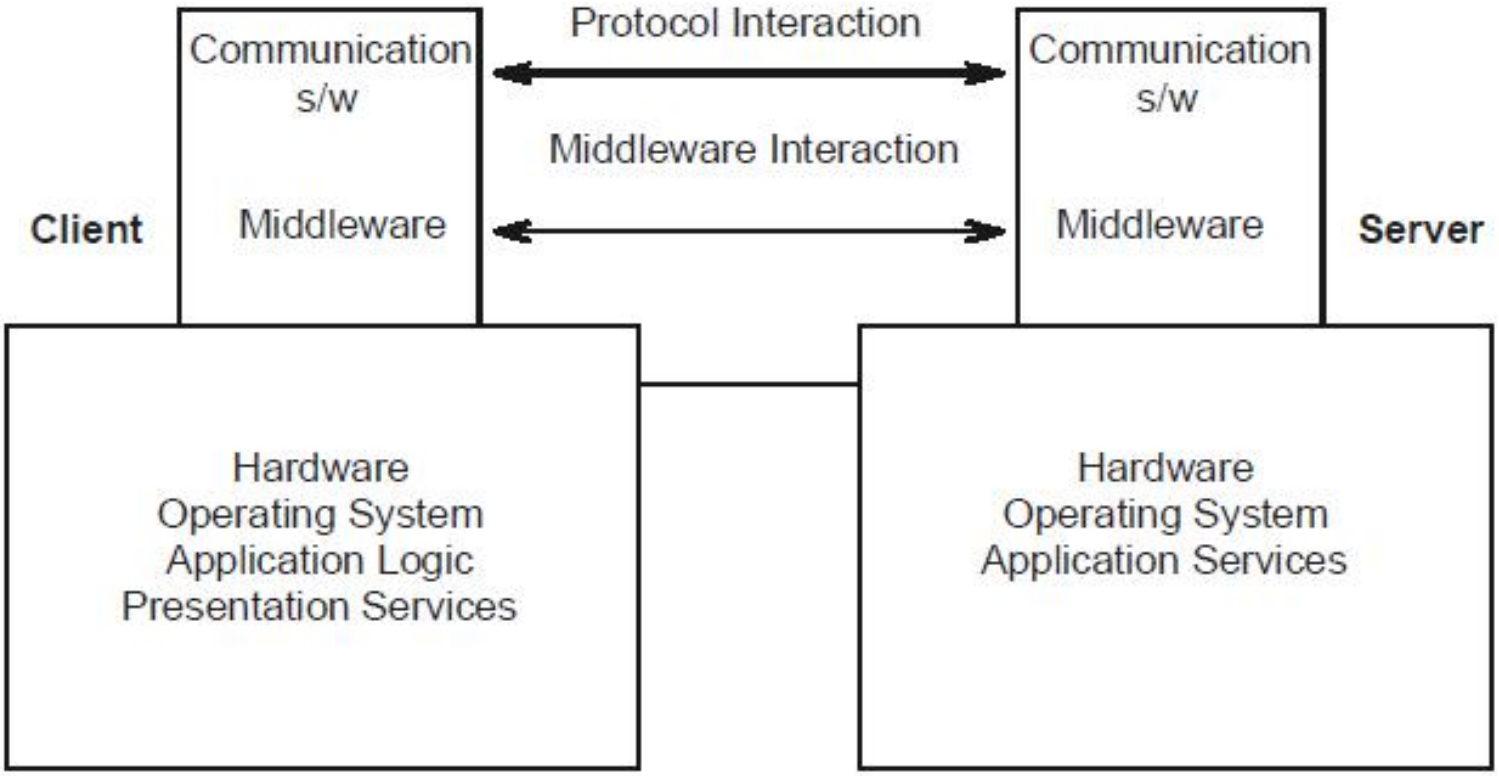

Fig. 5.12(a): Middleware Role in Client/Server Architecture

Role and Mechanism of Middleware

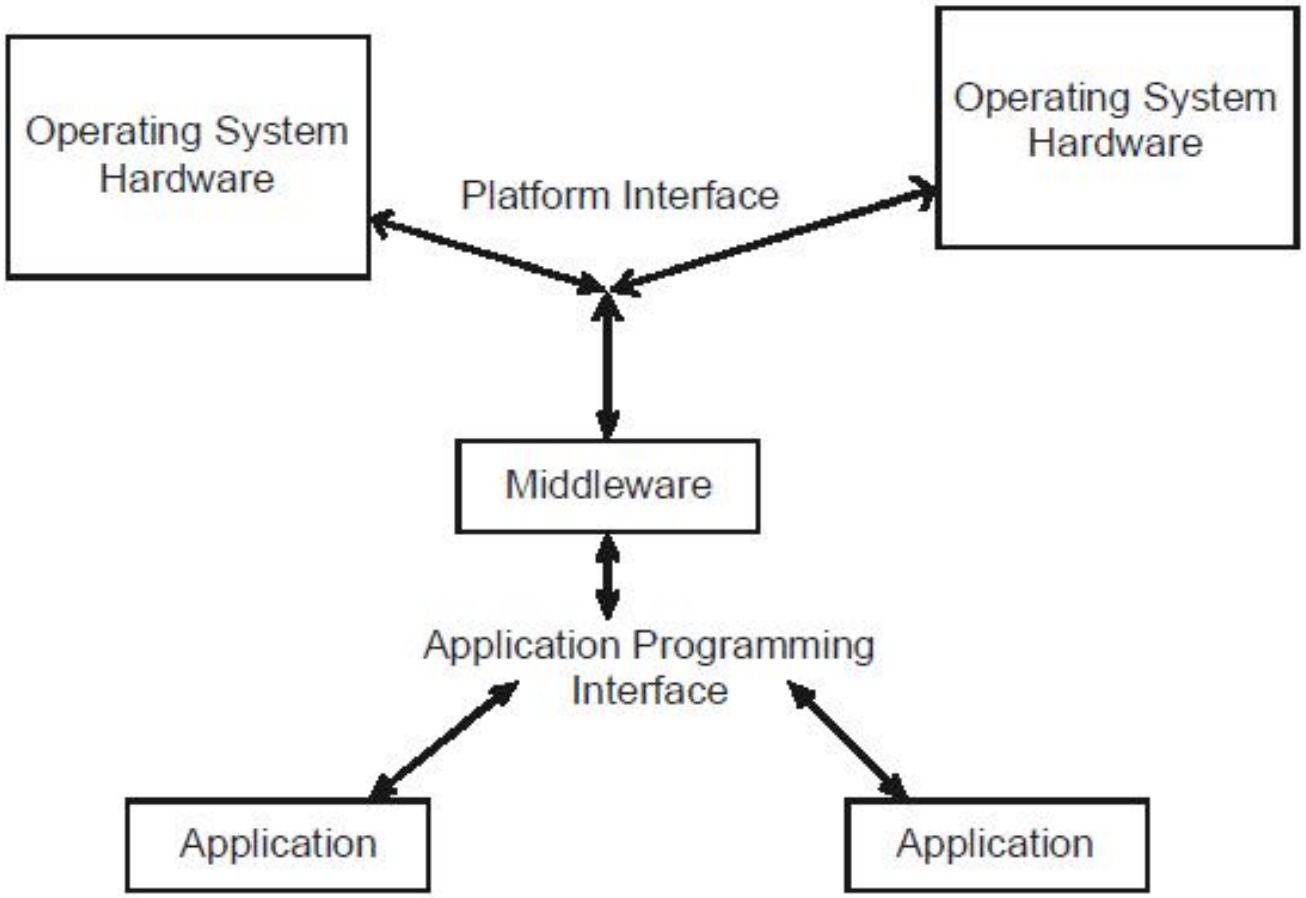

Fig. 5.12(*b*): Middleware Role in Client/Server Architecture

Role and Mechanism of Middleware

- All applications operate over a uniform application programming interface. The middleware is responsible for routing client requests to the appropriate server.

- Also middleware used to overcome operating system as well as network incompatibility.

CLIENT/SERVER APPLICATION: LAYERED ARCHITECTURE

- In most client/server systems, there is heavy emphasis on providing a graphical user interface that is easy to use, easy to learn, yet powerful and flexible

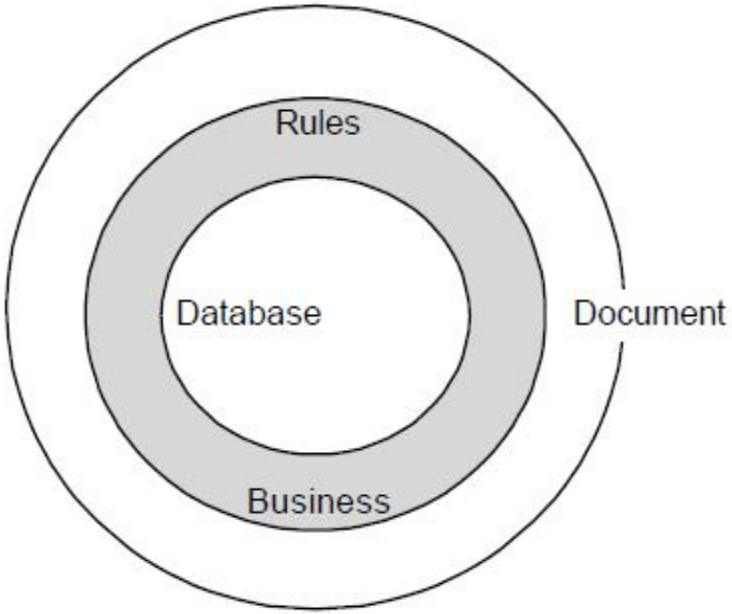

Fig. 5.13: Three-layered Application Architecture

Design Approach

- The phrase ‘client/server’ reflects the role played by an application’s functions as they interact with one another.

- One or more of these functions is to provide a service, typically in the form of a database server that is commonly used by other functions across the application(s).

- A client/server application operates across multiple platforms, i.e. a server platform for
- the database, and a client platform for the application.

Interface in Three **Layers**

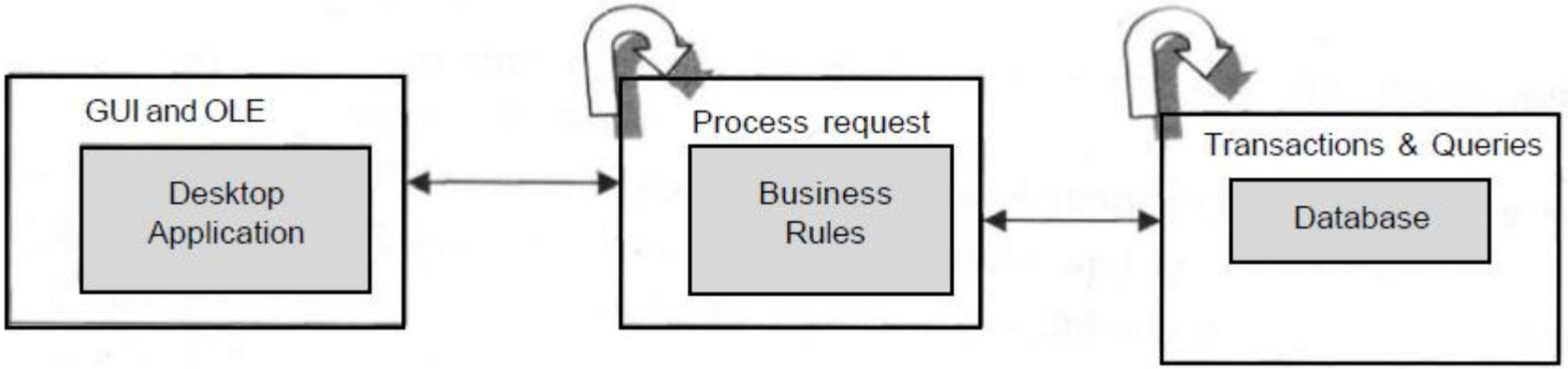

Fig. 5.14: Interface in a Three-layered Application Architecture

1. Graphical user interface.
2. Process request interface.
3. Transaction and query manager interface.

Interface in Three Layers

- An interface enables a component in one layer to communicate with a component in another layer; it also enables a component to interact with another component in the same layer.

- Cooperating components in a layered application design provide the following:

  A framework for building highly flexible applications that can be changed easily to meet the changing needs of business.

  A high level of software reuse.

  Easier development of large, complex applications that can sustain high throughput levels in both decision support and transaction environments.

  Easier development of distributed applications that support centrally and self-managed teams.