DBMS 5 Multi- User Architectures

Architectures that are used to implement multi- user:

- teleprocessing,   
- file-server, and   
- client–server.

**Teleprocessing**

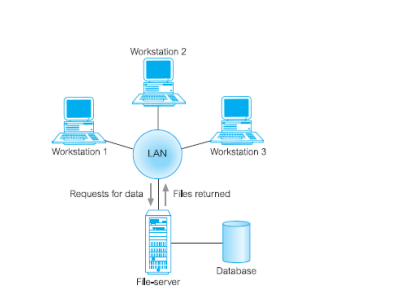
* The traditional architecture where there is one computer with a single CPU and a number of terminals.
* They are cabled to the central computer. The terminals send messages via the communications control subsystem of the operating system to the user’s application program, which in turn uses the services of the DBMS.
* **Disadvantage:**
* this architecture placed a tremendous burden on the central computer, which not only had to run the application programs and the DBMS, but also had to carry out a significant amount of work on behalf of the terminals (such as formatting data for display on the screen).

Note:

* **Downsizing:** replacing expensive mainframe computers with more cost-effective networks of personal computers that achieve the same, or even better, results. This trend has given rise to the next two architectures: **file-server and client–server**

**File – server Architecture**

* The processing is distributed about the network (LAN).
* The applications and DBMS run on each workstation, requesting files from the file – server.
* The file – server acts simply as a shared hard disk drive.
* This approach can generate a significant amount of network traffic, which can lead to performance problems.

****

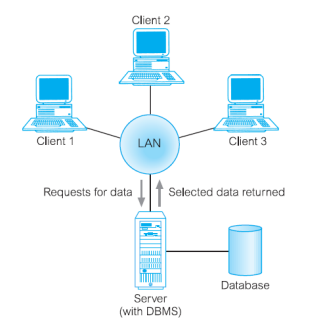
**Disadvantages of File server:-**

* There is a large amount of network traffic.
* A full copy of DBMS is required on each workstation
* Concurrency, recovery and integrity control are more complex because there can be multiple DBMS accessing the same files.

**Traditional Two-Tier Client–Server Architecture**

To overcome the disadvantages of the first two approaches and accommodate an increasingly decentralized business environment, the client–server architecture was developed.

Client–server refers to the way in which software components interact to form a system.

****

Data-intensive business applications consist of four major components:

* the database,
* the transaction logic,
* the business and data application logic,
* the user interface.

The traditional two-tier client–server architecture provides a very basic separation of these components.

* The client (tier 1) is primarily responsible for the presentation of data to the user, Presentation services handle user interface actions and the main business and data application logic.
* and the server (tier 2) is primarily responsible for supplying data services to the client.

The processing involves checking authorization, ensuring integrity, maintaining the system catalog, and performing query and update processing. In addition, it also provides concurrency and recovery control.

There are many **advantages** to this type of architecture. For example:

• It enables wider access to existing databases.

• Increased performance – if the clients and server reside on different computers then different CPUs can be processing applications in parallel. It should also be easier to tune the server machine if its only task is to perform database processing.

• Hardware costs may be reduced – it is only the server that requires storage and processing power sufficient to store and manage the database.

• Communication costs are reduced – applications carry out part of the operations on the client and send only requests for database access across the network, resulting in less data being sent across the network

• Increased consistency – the server can handle integrity checks, so that constraints need be defined and validated only in the one place, rather than having each application program perform its own checking.

• It maps on to open systems architecture quite naturally.

**Disadvantages**

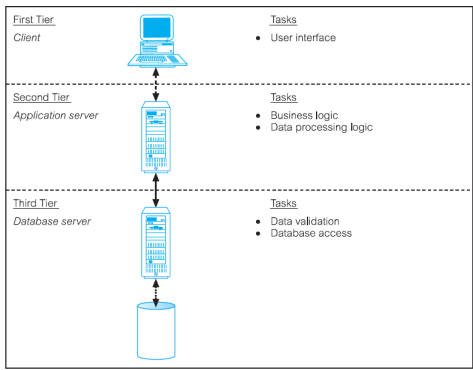
• A ‘fat’ client, requiring considerable resources on the client’s computer to run effectively. This includes disk space, RAM, and CPU power.

• A significant client-side administration overhead

**Three-Tier Client–Server Architecture**

This new architecture proposed three layers

1. User interface layer (Client/end user)
2. The business logic and data processing layer. This middle tier runs on a server and is often called the **application server.**
3. Database server: A DBMS, which stores the data required by the middle tier. This tier may run on a separate server called the database server.



The three-tier design has many advantages over traditional two-tier or single-tier designs, which include: • The need for less expensive hardware because the client is ‘thin’.

• Application maintenance is centralized with the transfer of the business logic for many end-users into a single application server. This eliminates the concerns of software distribution that are problematic in the traditional two-tier client–server model.

• The added modularity makes it easier to modify or replace one tier without affecting the other tiers.

• Load balancing is easier with the separation of the core business logic from the database functions.

• An additional advantage is that the three-tier architecture maps quite naturally to the Web environment, with a Web browser acting as the ‘thin’ client, and a Web server acting as the application server.

The three tier architecture can be extended to in tires for more flexibility and scalability.

Middle tier architecture of 23-tier could be split into two  
- Web server  
- Application server.

It is an important tier for **Transaction Processing Monitors** ( it is a program that controls data transfer between clients and servers in order to provide a consistent environment), usually used in a very high vol transaction environment.

**Advantages of TP Monitor:**

1. Transaction routing: increase scalability by directing transactions to specific DBMS.
2. Managing Distributed transactions: it can manage data held in multiple DBMS.
3. Load Balancing: balancing client requests by directing client server calls to least loaded server.
4. Funneling: DMBS connections as and when required, allowing large no.of users to acces available DBMS with smaller no.of connections and less resources.
5. Increased reliability: TP monitor acts as transaction manager with DBMS as resource manager if it fails then TP Monitor resubmit the transaction to another DBMS or hold till DBMS is available.