CO3201 Database Systems
Assignment 01

NAME : **J.R.REMILTAN**
REGISTATION NO : **EN102828**
INDEX NO. : **21/ENG/048**

CONDUCTED DATE : **26/04/2024**
DATE OF SUBMISSION : **03/05/2024**

i) Centralized DBMS architecture

- Below diagram show physical components of the Centralized DBMS architecture. Combines everything into single system including DBMS software, hardware, application programs and user interface processing software.
- Earlier Days, mainframe computers were used to process all system function.
  And reason for that was the computer terminals used by users were not capable of processing only capable of displaying any information to the user.
- All the processing were performed remotely on the computer system. Only display information sent to the terminals. And this terminal connected to the terminals via a network.
- As a prices of the hardware started to reduced. The terminals were replaced by personal computer and workstations.
- Initially, database systems operated similarly to display terminals on computers. However, the DBMS was centralized, meaning that all the functionality of the DBMS, user processing, and execution of the application program were performed on one machine. The display information was then sent to display terminals via the network.

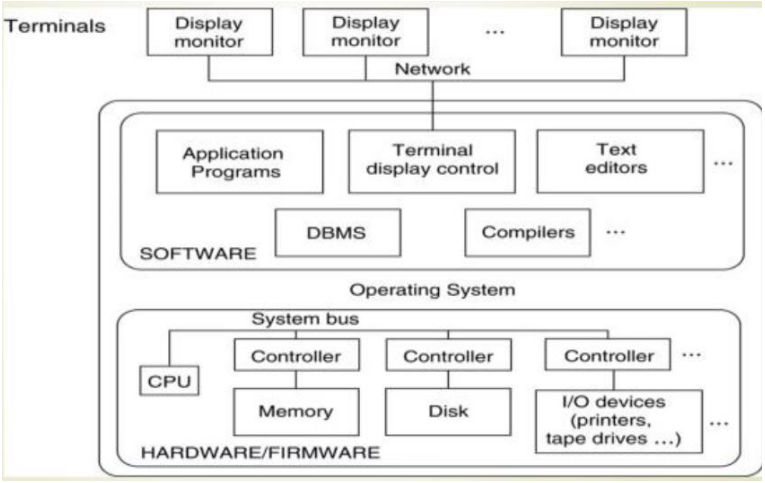

Figure 01: Physical components of the Centralized DBMS architecture

ii) Basic Client/Server Architecture

- The client server architecture was developed to deal with large number of PCs, file server, workstation, printers, database server, and webserver are connected via a network.
- **Main Goal** of this define the specialized servers with specific functionalities. For example print server, file server, DBMS server and etc.
  - For example it is possible to connect a number of PCs as client to the file server that maintain the files of the client machine.
  - Another example printer server dedicated to all the print requests forwarded to this machine.

Likewise resources provided by specialized servers can be accessed by many client machines.

- **Client**: A client is a user machine that provides user interface capabilities and local processing. This means that client machines offer users the appropriate interface to access and utilize servers. Clients can range from diskless machines to PCs or workstations with only client softwareinstalled. They are connected to servers via various forms of networks, such as **LAN or wireless networks**.

- **Server**: Provides services to client machines. Like access to the database, printing and etc. Sometimes called query and transaction servers.
- Two main types of basic DBMS architectures were created on this underlying client/server framework: two-tier and three-tier.

iii) Two-Tier Client/Server Architecture

- In the two-tier architecture, the presentation layer runs on the client device (PC, mobile etc.) and the data is stored on a server which is known as the second tier. As the data is not directly exposed to the end user, it also provides a mean for additional security.
- When DBMS access is required the application programs establishes a connection with the DBMS (server side).
- Server level usually encompasses operations like disk page data storage, recovery, buffering, and disk page caching. On the other hand, the client level typically handles the user interface, data dictionary operations, global query optimization, multi-server recovery, assembling complicated objects from buffered data, and other associated tasks.
- DBMS on the server side can be called by applications from the client side through the Application program interface (API) provided by Open Database Connectivity (ODBC). In this manner, a client application can use the ODBC API to connect to many RDBMS and send queries and transaction requests. The client software issued this request, which the server processed. Consequently, the client and server connection is created using the ODBCAPI.
- JDBC is also one such standard which is specified for Java client programs
- Advantages of this architecture
    - Simplicity
    - Compatibility of other system
- Disadvantages of this architecture
    - Give the poor connectivity

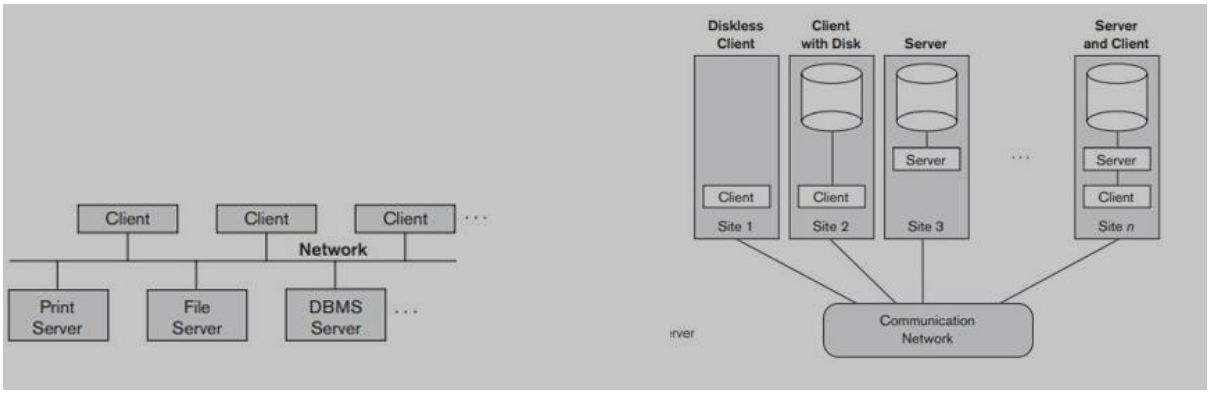

Figure 02: Logical and Physical two-tier client/server architecture

IV) Three-Tier and n-Tier Architectures for Web Application

- **Three-Tier Architecture** is the most popular client server architecture .Many web applications are used this Architecture.
- It improving the security by checking client's credentials before forwarding the request. Encrypt the data at the server before transmission decrypt data at the client.

- Additional intermediate layer is added between the client and database server, which is why it is called Three-tier client/server architecture. This additional layer is called the Application server, sometimes referred to as the Web server based on the application. The Application server stores rules used to access data from the database server. It also accepts requests from the client, processes the requests, and sends the database commands to the database server. This intermediate layer acts as a buffer for passing the processed data from the database server to the client. On the client side, data is processed further and filtered to make it presentable for the users in the GUI format.
- Advantages of this architecture are Scalability, Data integrity, Security and three-tier architecture tends to be faster than the two-tier architecture
- Disadvantages of this architecture is complexity of implementation and communication due to the middle layers

- **N-tier architecture**: The presentation, processing, and data operations are separated into logically and physically distinct tiers in this multi-tiered client-server architecture. This architecture is frequently used to create simple web applications as well as for the unified development of both on-premises and cloud-based applications.
- Layers are a way to separate responsibilities and manage dependencies. Each layer has a specific responsibility. A higher layer can use services in a lower layer, but not the other way around.
- Tiers operate independently on distinct machines and are physically divided. A tier can make direct calls to another layer. It's not necessary for every layer to be hosted in its own tier, however it might be. On a single tier, there could be multiple levels hosted. Although physically dividing the layers increases resilience and scalability, it also increases latency due to more network traffic.

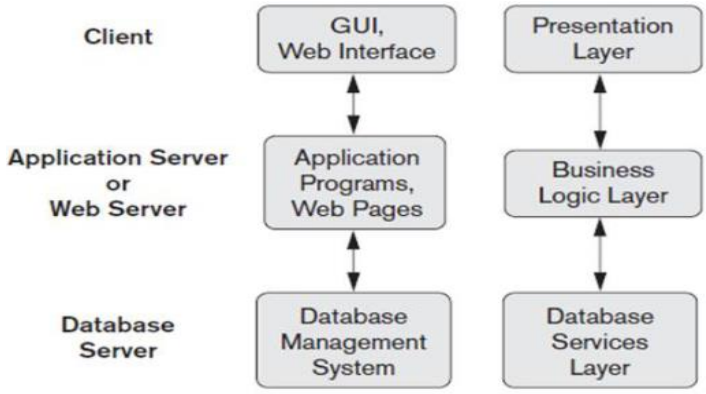

Figure 03: Three-tier architecture

REFERENCES

[1] ghogen, "N-tier Data Applications Overview - Visual Studio (Windows)," learn.microsoft.com. https://learn.microsoft.com/en-us/visualstudio/data-tools/n-tier-data-applications-overview?view=vs-2022

[2]Vasav, "Client Server Architecture: Types, Examples, & Benefits," www.redswitches.com, Oct. 20, 2023. https://www.redswitches.com/blog/client-server-architecture/