

A PHYSICAL CENTRALIZED ARCHITECTURE

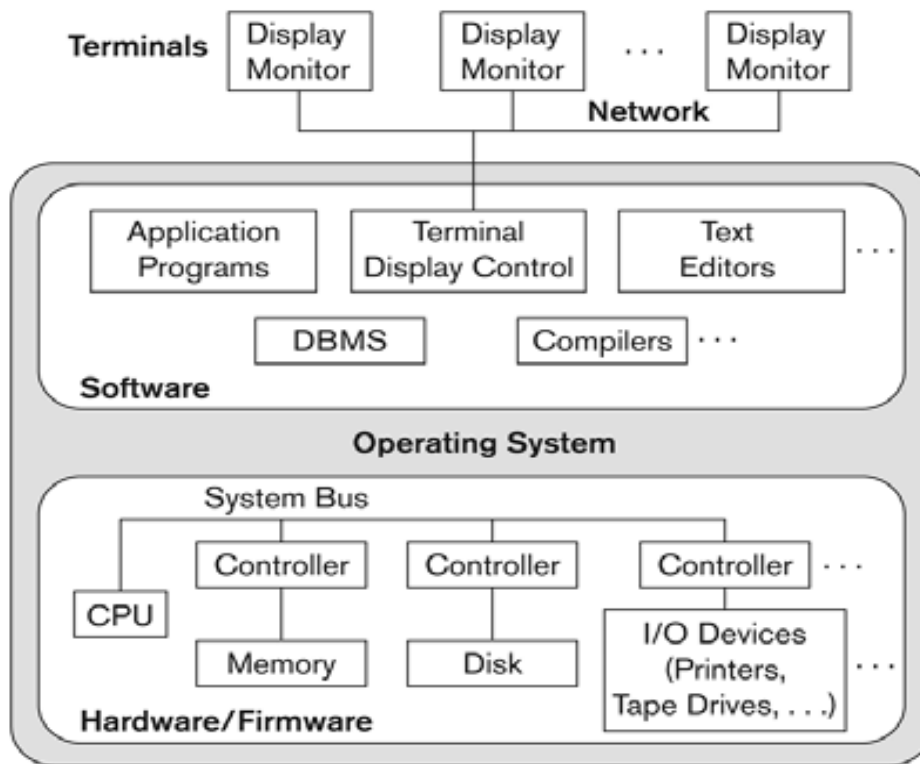


Figure 2.4
A physical centralized architecture.

Architectures for DBMS have followed trends similar to those generating computer system architectures. Earlier architectures used mainframe computers to provide the main processing for all system functions, including user application programs and user interface programs as well as all DBMS functionality. The reason was that most users accessed such systems via computer terminals that did not have processing power and only provided display capabilities. Therefore all processing was performed remotely on the computer system, and only display information and controls were sent from the computer to the display terminals, which were connected to central computer via various types of communication networks.

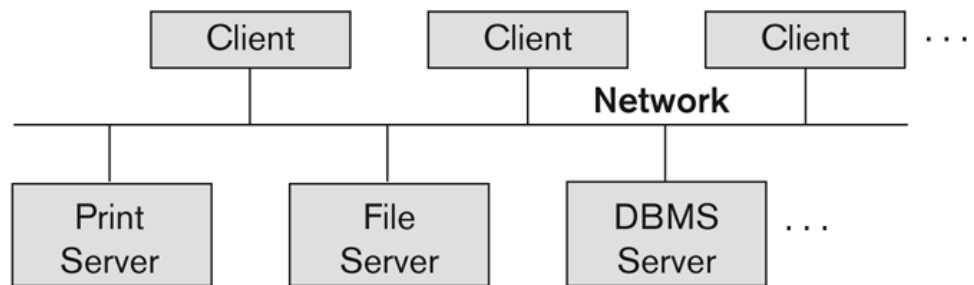
As prices of hardware declined, most users replaced their terminals with PCs and workstations. At first database systems used these computers similarly to how they have used display terminals, so that DBMS itself was still a Centralized DBMS in which all the DBMS functionality, application program execution and user interface processing were carried out on one Machine.

Basic 2-tier Client-Server Architectures

- Specialized Servers with Specialized functions
- Print server
- File server
- DBMS server
- Web server
- Email server
- Clients can access the specialized servers as needed

Logical two-tier client server architecture

Figure 2.5
Logical two-tier
client/server
architecture.



Clients

- Provide appropriate interfaces through a client software module to access and utilize the various server resources.
- Clients may be diskless machines or PCs or Workstations with disks with only the client software installed.
- Connected to the servers via some form of a network.
- (LAN: local area network, wireless network, etc.)

DBMS Server

- Provides database query and transaction services to the clients
- Relational DBMS servers are often called SQL servers, query servers, or transaction servers
- Applications running on clients utilize an Application Program Interface (**API**) to **access server databases via** standard interface such as:

- ODBC: Open Database Connectivity standard
- JDBC: for Java programming access
- Client and server must install appropriate client module and server module software for ODBC or JDBC

Two Tier Client-Server Architecture

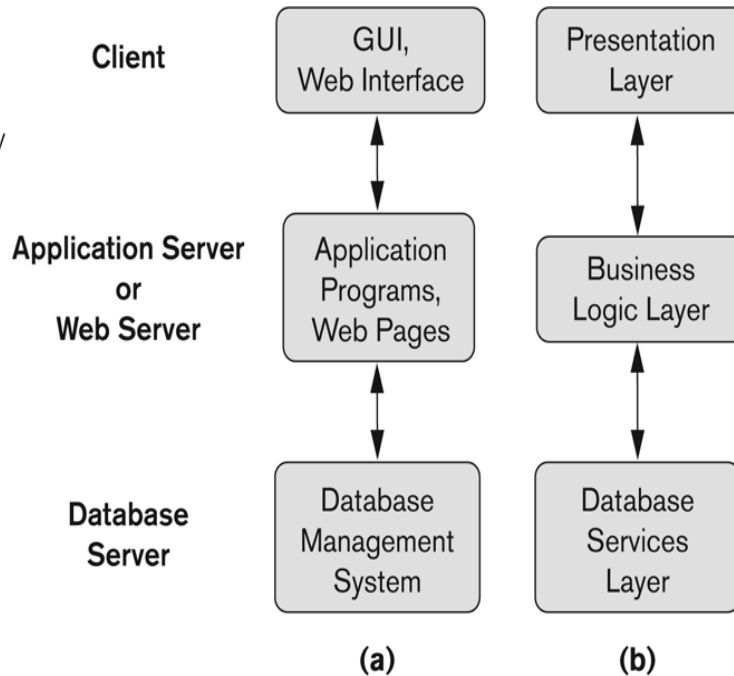
- A client program may connect to several DBMSs, sometimes called the data sources.
- In general, data sources can be files or other non-DBMS software that manages data. Other variations of clients are possible: e.g., in some object DBMSs, more functionality is transferred to clients including data dictionary functions, optimization and recovery across multiple servers, etc.

Three Tier Client-Server Architecture

- Common for Web applications
- Intermediate Layer called Application Server or Web Server:
- Stores the web connectivity software and the business logic part of the application used to access the corresponding data from the database server
- Acts like a conduit for sending partially processed data between the database server and the client.
- Three-tier Architecture Can Enhance Security:
- Database server only accessible via middle tier
- Clients cannot directly access database server

Figure 2.7

Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.



Classification of DBMSs

- Based on the data model used
- Traditional: Relational, Network, Hierarchical.
- Emerging: Object-oriented, Object-relational.
- Other classifications
- Single-user (typically used with personal computers) vs. multi-user (most DBMSs).
- Centralized (uses a single computer with one database) vs. distributed (uses multiple computers, multiple databases)

Variations of Distributed DBMSs (DDBMSs)

- Homogeneous DDBMS
- Heterogeneous DDBMS
- Federated or Multidatabase Systems
- Distributed Database Systems have now come to be known as client-server based database systems because:

- They do not support a totally distributed environment, but rather a set of database servers supporting a set of clients.

Cost considerations for DBMSs

- Cost Range: from free open-source systems to configurations costing millions of dollars
- Examples of free relational DBMSs: MySQL, PostgreSQL, others

Source : <http://elearningatria.files.wordpress.com/2013/10/introduction.pdf>