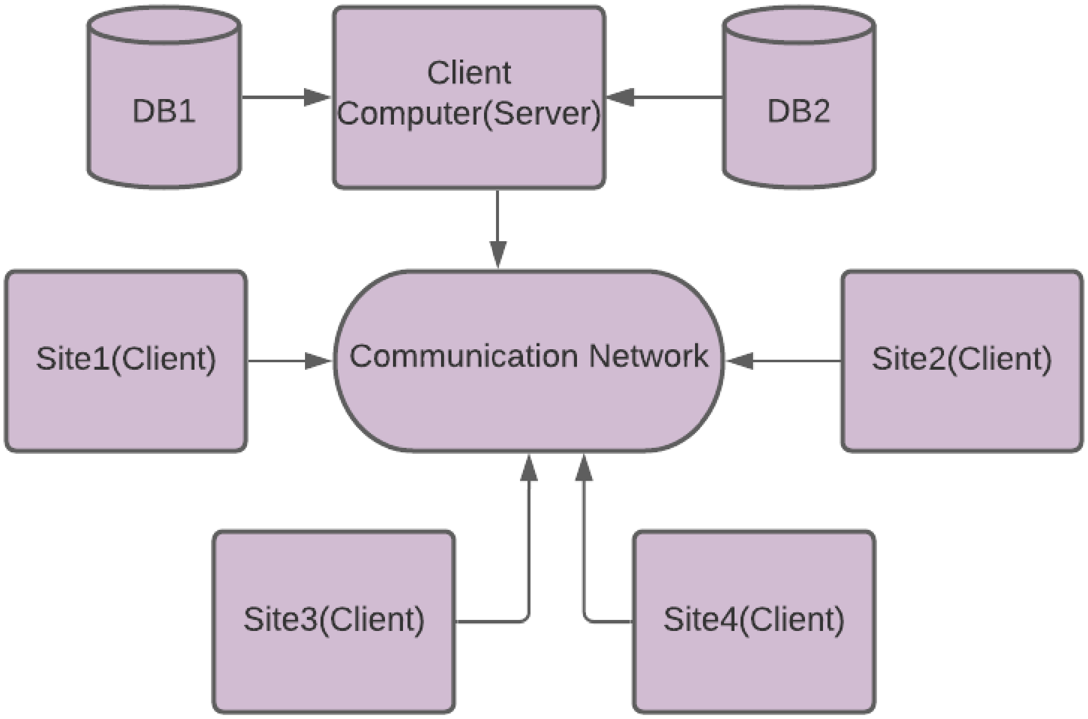
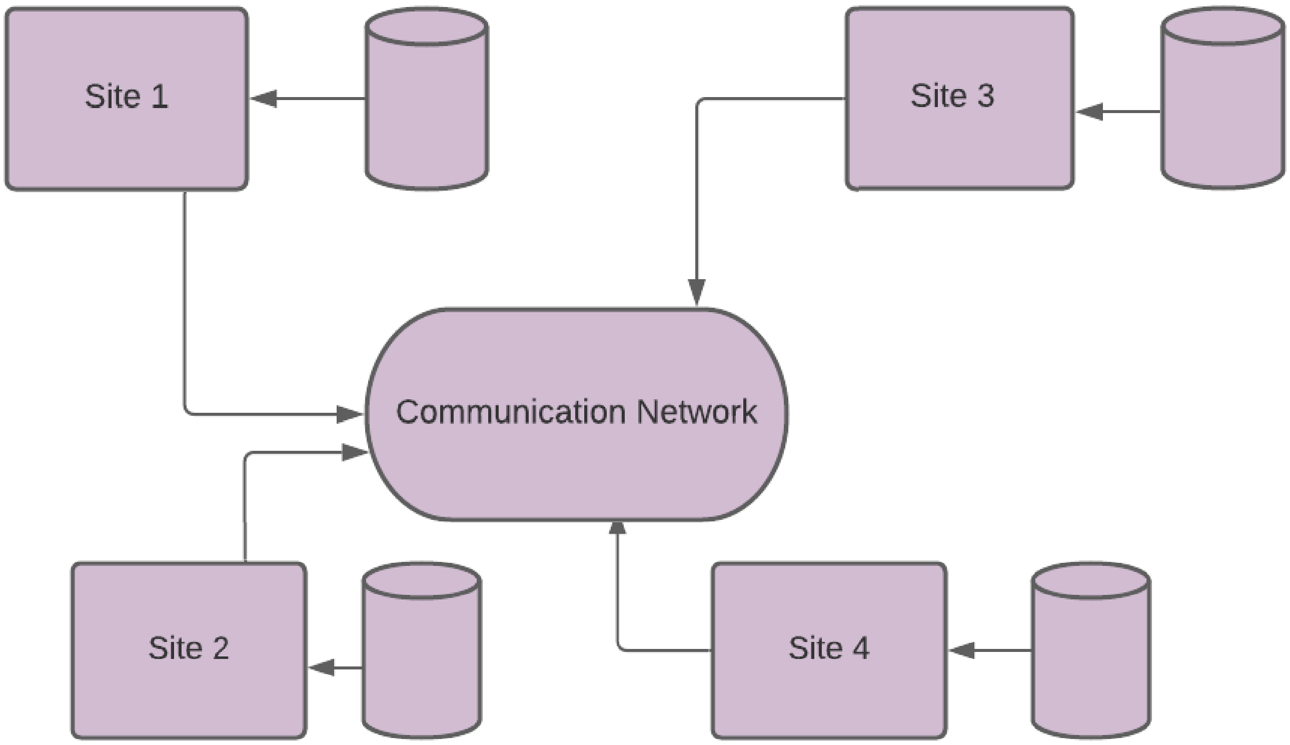
Distributed Client/Server Database system

Client/server is developed to deal with various computing environments that have a large number of computers and servers connected together via a network. In this architecture, a Client is a user machine which provides the user interface and local processing capabilities. When any client requires additional functionality like database access, it can connect to Server that is capable of providing the functionality needed by the client. Basically Server is a machine that provides services to the Client i.e user machine.

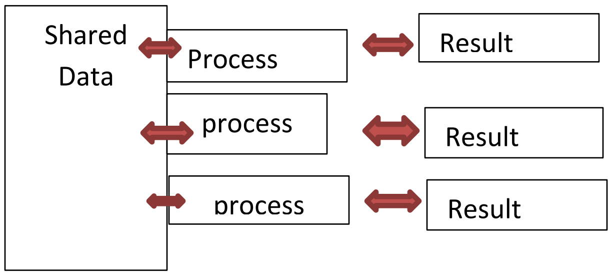


In Distributed DBMS, data is distributed over the geographical site. Each site is a complete database system site on its end but the different sites have to work together because if any user wants to access the data it can easily access data anywhere in the network as the data is stored at the user's own computer.



Data Distribution:

Distributed data and distributed processing are terms used widely in the word of Client/Server computing. The differences in these two can be easily understood by the two figures 1.01 and 1.02.



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.

Data can be accessed and manipulated by the end user at any time in many ways. Data accessibility increases because end users are able to access data directly and easily, usually by pointing and clicking in their GUI-based system. End user can manipulate data in several ways, depending on their information needs.

DISTRIBUTED DBMS

Client/Server database is commonly known for having distributed database capabilities. But is not necessarily able to fulfil the entire required Client/Server characteristics that are in need of a particular system. Client/Server architecture refers to the way in which computers interact to form a system.

The DBMS must provide distributed database transparency features like:

- Distribution transparency.
- Transaction transparency.
- Failure transparency.
- Performance transparency.
- Heterogeneity transparency