Database Systems Model: Distributed

Woko, Ovunda
Research Scholar, School of Post Graduate Studies,
Department of Computer Science,
Faculty of Natural and Applied Sciences,
Ignatius Ajuru University of Education.

Asagba, Prince Oghenekaro
Professor, Department of Computer Science,
Faculty of Science, University of Port Harcourt,
Choba, Port Harcourt

Abstract
A Database is a collection of structured data describing the activities of one or more related organizations with a specific defined purpose. Databases are controlled by Database Management System by maintaining and utilizing large collections of data. A Distributed Database is a collection of multiple, logically organized databases distributed over a Computer Network. It is also a collection of databases that can be stored at different computer network sites. This work presents an overview of Distributed Database System: explaining why we distribute, what we distribute, and how we distribute. This paper classified distributed database objects, stated various strategies for distribution and proposed a standard mathematical model for understanding, interpreting and implementing the distributed database design method.

Keywords: *Database, Distributed Database, Database Management System, Mathematical Model*

1.0 ITRODUCTION

Some couple of years ago, numerous organizations migrated from a paradigm of data processing in which each application defined and maintained its own data (Traditional File Processing) as shown in Figure1 to one in which the data are defined and administered centrally (Database Processing) as shown in Figure 2 (Özsu, M. T. & Valduriez, P., 2011).

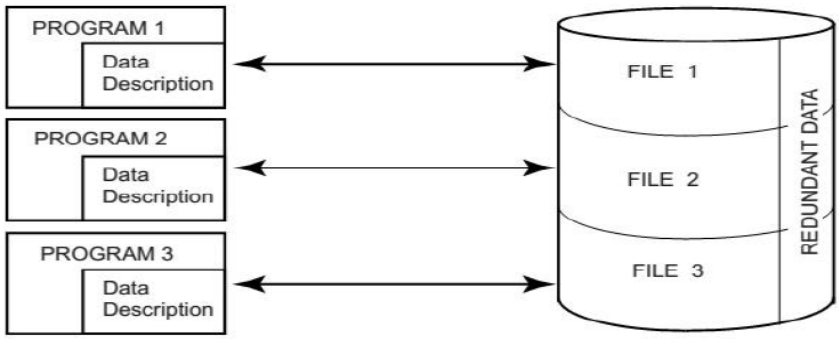
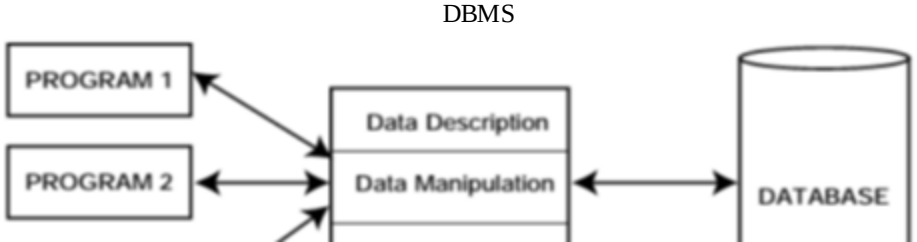


Figure 1: Traditional File Processing



Centralized Databases Systems (CDBS) were used for daily transactions in diverse domains of activities: booking, library, banking, commerce, manufacturing, etc. Even nowadays, a handful of organizations still adopt CDBS approach. However, there are performance, maintenance, cost of data communication, scalability, and other limitations associated with centralized database system during query processing as end-users from different sites query a single host. Hence, these issues and advancement in computer networks motivated the design and implementation of efficient Distributed Database Systems (DDBS) otherwise known as Decentralized Database Systems (DDS).

Distributed Database System is a model derived from the combination of two entirely opposed approaches to data processing: Databases and their Networking. This approach implements different strategies like Data replication, Data fragmentation and Data allocation (Özsu, M. T. & Valduriez, P., 2011; Shareef M. I. & Rawi A.W., 2011). A Distributed Database is a set of more than one database interconnected and propagated physically across various locations (sites) which communicate, via a computer network (Kaur K. & Singh H., 2016 ; Tomar, P., 2014). Furthermore, Singh, I. and Singh, S. (2015) proposed a practical and explanatory definition that "A Distributed Database is a collection of multiple, logically interrelated databases distributed over a Computer Network". He added that "sometimes Distributed Database System is used to refer jointly to the Distributed Database and the Distributed Database Management System". In this approach, processing logic or elements, functions, data, and control are distributed in a multiply location of a computer network (Tomar, P., 2014) as shown in figure 3. However, the object of the distribution remains the data in the database.

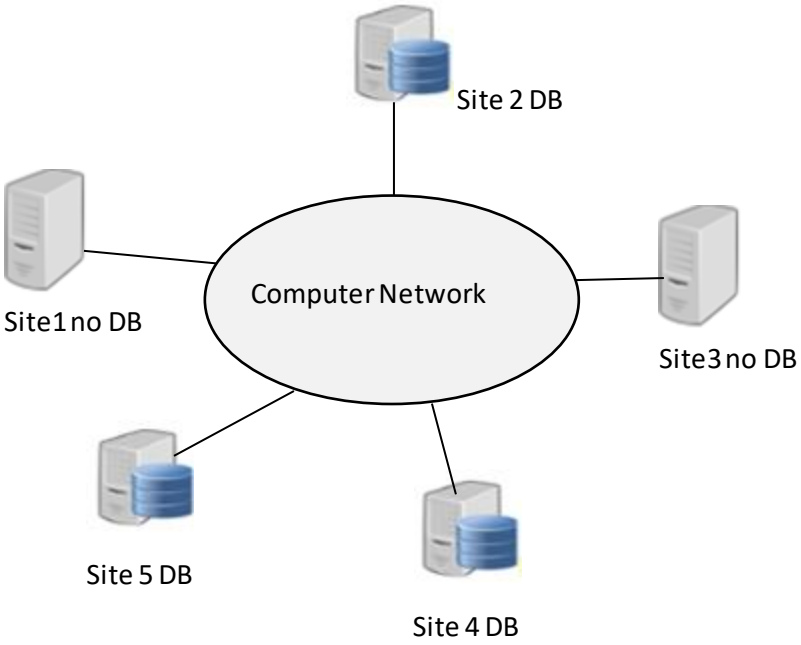


Figure 3: Distributed Database System Architecture

When designing a Distributed Database, it is required that it be fully accommodated or fragmented on various sites in a computer network. Going by this approach in the computer network, there must be at least two sites hosting the database and not certainly each site. The main aim of a Distributed Database System is to appear as a centralized system to end-users (Tomar, P. 2014). All the
Administrative activities of Distributed Database are piloted by the Distributed Database Management Systems (DDBMS). The DDBM is a software that manages the distribution of the Distributed Database to each site on the network, maintaining its database schema globally and locally. The DDBM provides the capability for fragmentation, replication and allocation of data on several sites as shown in Figure 1, which is different from the Centralized Database System (CDBS), where only a replica of the Database is stored as shown in Figure 2 Singh, I. and Singh, S. (2015). In a Centralized Database System (CDBS), the Database is managed by one computer system on a central site (Site 2) and all query transactions from other sites are directed to the central site as shown in Figure 4.

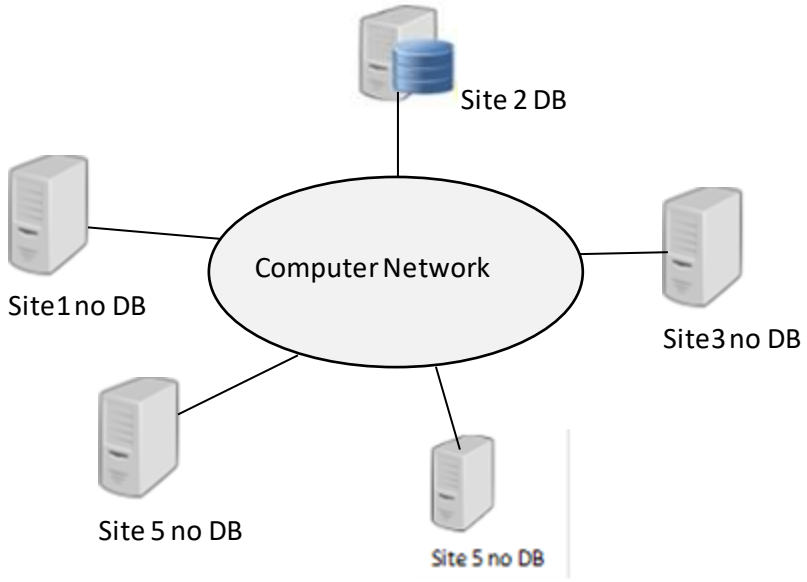


Figure 4. Centralized Database System Architecture

It is important to know that the most crucial objective of the database technology now is integration, and not centralization. It is also important to know the concept of integration and centralization are distinct (Özsu, M. T. & Valduriez, P., 2011) because integration can be achieved without centralization. Therefore, Distributed Database System is an integrated Database Technology distributed by Computer Networks as shown in Figure 5

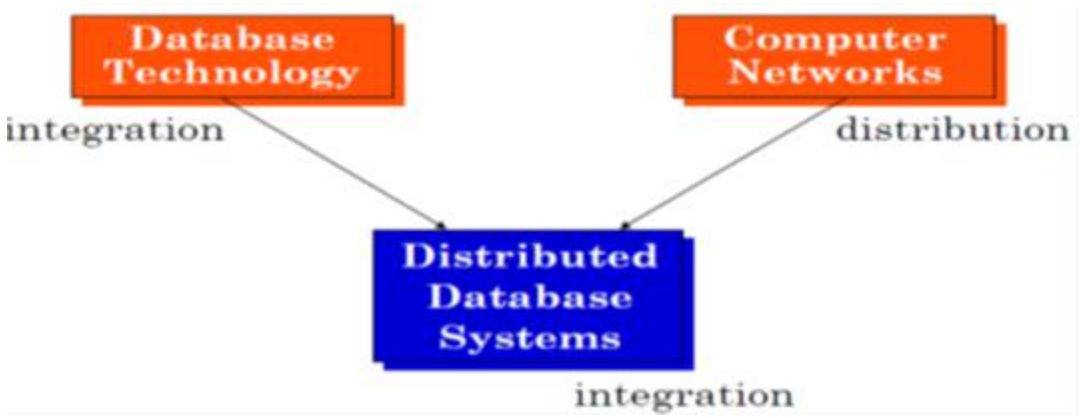


Figure 5: Integration ≠ Centralization (Özsu, M. T. & Valduriez, P., 2011)

The modelling, designing, and implementation of the Distributed Database System is a daunting task (Singh, I. & Singh, S., 2015; Bhuyar P. R., Gawande, A. D., & Deshmukh A. B., 2012**).** The design of the Distributed Database System involves the global conceptual schema, which is added to local schemas, based on the three-level architecture (Physical, Conceptual, and External) of the DBMS in all sites. The establishment of a computer network across sites of a distributed system is an additional complex problem of design (Katembo K. E., Shri K., & Ruchi A., 2019). This daunting task for distributed database can be mathematically model to establish a bases for its design approaches.

Models describe our beliefs about how the world functions. So, mathematical models formulate and express real world activities and abstract ideas using mathematical well-

defined rules and symbols. In research works, more often than not, mathematical framework or models form the bases for communicating real world ideals. Mathematical models play vital roles for design of many concepts. Therefore, a standard working model has to express the working concept of distributed database.

In this work, we focus on why do we distribute, what is distributed, and how do we distribute (model). We propose a mathematical model for distributed database design for easy expression and interpretation of the integration of global conceptual schema and the local schemas since distributed database system can only be achieved by integration especially with heterogeneous databases.

2.0 REVIEW OF RELATED LITERATURE

Özsu, M. T. and Valduriez, P. (2011) designed the distributed database approach. The design process phases include the company situation analysis, the problems definition and constraints, the objectives definition, and the scope design and boundaries. The paper classified distributed database design into Top-down and bottom-up design or approach. But these approaches lack standard mathematical frame or model for easy interpretation and implementation.

Singh, I. and Singh, S. (2015) stated that the Bottom-up design process requires the following steps: the selection of a mutual prototype to describe the global schema of the database; the conversion of all local schemas into a mutual data model; and the unification of local patterns to arrive at a mutual global schema. However, there is not standard mathematical frame or model for unification of local mutually exclusive database to arrive at a mutual global schema that can interpret all these.

Shareef M. I., and Rawi A. W. (2011) expressed distributed database model is a model that its goal is to break the relation, to allocate and to replicate the fragment in different sites of the distributed system with local optimization on each site. This model is shallow and only focused on distributed database management system but not on distributed database.

Tomar, P. and Megha (2014) presented an overview of Distributed Database System along with their advantages and disadvantages. This paper also provides various aspects like replication, fragmentation and various problems that can be faced in distributed database systems.

Kumar, N., Bilgaiyan, S., & Sagnika, S. (2013) explained how the cost of implementing multiple transparencies interact and how to reduce operating system and communication stack.

Hiremath, D.S. & Kishore, S.B. (2016) emphasized on distributed database problem areas and approaches.

3.0 DISCUSSION

This section discusses why we distribute, what we distribute and how distribution of database is achieved, proposing a mathematical model for unification of local mutually exclusive databases to arrive at a mutual global schema for easy interpretation and implementation of distributed database design approach.

3.1 WHY DO WE DISTRIBUTE?

This is a fundamental question because many of the current applications of computer technology are inherently distributed. Applications such as web-based applications, e-commerce business over the Internet, multimedia applications, and manufacturing control systems are all examples of distributed applications. From a more global view, however, it

can be identified that the fundamental reason behind distributed processing is to be better able to cope with the challenges of huge data management problems that we face today, by using a variation of the well-known divide-and-conquer rule (Michel A. et al., 2016). Almeida F. and Calistru, C. (2012) explained that data warehouse operational processes normally compose a labour intensive workflow and constitute an integral part of the back-stage of data warehouse architectures, where the collection, extraction, cleaning, transformation, and transport of data takes place, in order to populate the warehouse. Tanenbaum, S. A. and Steen V. M. (2016) stated goals distributed database: resource sharing, making distribution transparent, being open, and being scalable are the four important goals for distributed database.

i. **Support for resource sharing:** The significant goal of a distributed system is to make it easy for systems, applications and users (people) to access and share remote resources. Resources can be virtually anything: processing logic or elements, functions, data, and control. Common examples are peripherals, storage facilities, data, files, services, and networks, etc. The reason for sharing of resource is because of limited resources. It is cost effective to share a single high – end storage facility on a network than to deploy the storage facility in each of the systems in the network.

ii. **Making distribution transparent:** Hiding the processes and resources physically distributed across multiple computers, perhaps separated by large distances is crucial. Distributed systems tries to make the distribution of processes and resources invisible to end users and applications. This is called transparency. There are different transparencies require (Tanenbaum, S. A. & Steen V. M., 2016):

- Access - differences in data representation and how an object is accessed must be hidden
- Migration – How objects move to another location must be hidden to end user
- Replication – How an object is replicated must be hidden to end user
- Location - Where an object is located must be hidden to end user
- Failure - Failure and recovery of an object must be hidden to end user
- Relocation - Hide that an object may be moved to another location while in use
- Concurrency- Hide that an object may be shared by several independent users

iii. **Being open:** Distributed systems involve a lot of integrated components in the networks. An important fact is that the components must be flexibly used. If components are not user friendly, then distribution is not opened for use. At user end, the users must be able to use the system with little or no supervision. Distributed systems must support Interoperability, composability, and extensibility.

iv. **Being scalable:** To distribute resources wide world, scalable design must a serious goal, that is being able to accommodate more processes and resources in the future. Scalability can be measured along at least three different dimensions (Neuman, B., 1994):

- Size scalability - A system can be scalable with respect to its size, meaning that we can easily add more users and resources to the system without any noticeable loss of performance.
- Geographical scalability - A geographically scalable system is one in which the users and resources may lie far apart, but the fact that communication delays may be significant is hardly noticed.

- Administrative scalability - An administratively scalable system is one that can still be easily managed even if it spans many independent administrative organizations.

3.2 WHAT IS DISTRIBUTED?

In distributed database system, the following objects are distributed: Processing logic or elements, Functions, Data, and Controls are distributed (Özsu, M.T. & Valduriez, P., 2001).

a. Processing logic or elements: These are numbers of autonomous processing elements (not necessarily homogeneous) that are interconnected by a computer network, that cooperate in performing their assigned tasks. The "processing element" are computing devices that can execute a program on its own (Özsu, M.T. & Valduriez, P., 2011). Interconnection of these processing logic or processing elements computing devices and fundamental across in distributed database system,

b. Function: Functions that ensure that authorized users perform correct operations on the database, contributing to the maintenance of database integrity. These functions are distributed across to enable perform correct operation.

c. Data: The major reason for distribution is data in a base. Data stored must be made available for use through distribution technology. Replication of logical data item and physical data items become necessary (Özsu, M.T. & Valduriez, P., 2011). There are five categories of distributed data namely: replicated data, horizontally fragmented data, vertically fragmented data, reorganized data, and separate-schema data (Borysowich C., 2007).

d. Control: The definition of the rules for controlling data manipulation is part of the administration of the database, a function generally is performed by a database administrator. An important requirement of a centralized or a distributed DBMS is the ability to support semantic data control - data and access control using high-level semantics (Özsu, M.T. & Valduriez, P., 2011). According to Özsu, M.T. and Valduriez, P. (2011), semantic data control typically includes view management, security control, and semantic integrity control. Rules for semantic data control must be stored in a catalog, the management of a distributed directory (also called a catalog) and be distributed.

3.2.1 Classification of Distributed Database System Objects

Therefore mentioned and explained distributed database system objects are classified into two: Object of the distribution and Distributed Objects for Distribution. Object of the distribution is the database object (relations) distributed and managed by Distributed Database Management System, while the Distributed Objects for Distribution guarantee effective distribution of database even if there are also distributed in a way. Distributed Database System Objects as shown in Figure 6

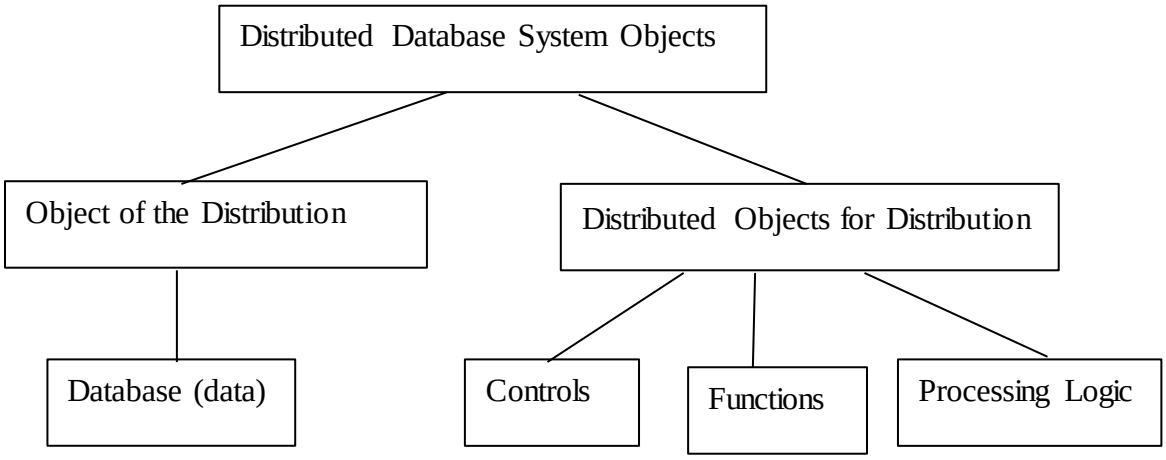


Figure 6: Distributed Database System Object

3.3 HOW DO WE DISTRIBUTED?

3.3.1 Classification of Database Systems

Database management systems can be classified based on several measures, such as the data model, user numbers and database distribution. However, DBMS is classified into Centralized database systems and Distributed Database System.

a. Centralized database systems

In a centralized database system, the Database Management System (DBMS) and database are stored at a single site that is used by several other systems too. This is illustrated in Figure 2. All the sites or workstations or nodes or terminals have access to the database at the site (central computer).

b. Distributed Database System

In a distributed database system, the Database Management System (DBMS) and database are stored at more than one site that is used by several other sites.

Distributed Database management system is a system that allocate a set of fragments F = {F_1, F_2, ..., Fm} resources across computer network nodes of a distributed environment comprising sites S = {S_1, S_2, ..., Sn} on which a set of query Q = {$Q1$, $Q2$, ..., Qp} is running. Distributed database model is a model that its goal is to break the relation, to allocate and to replicate the fragment in different sites of the distributed system with local optimization on each site (Shareef M. I., Rawi A. W., 2011). In a distributed database system, the actual database and the DBMS software are distributed from various sites that are connected by a computer network, as shown in Figure 3.

3.3.2 CLASSIFICATION OF DISTRIBUTED DATABASE SYSTEM

Distributed databases can be generally classified into homogeneous and heterogeneous distributed database. Homogeneous is further classified into Autonomous and Non-Autonomous, while Heterogeneous is classified into Federated and Multi-database as shown in the Figure 6

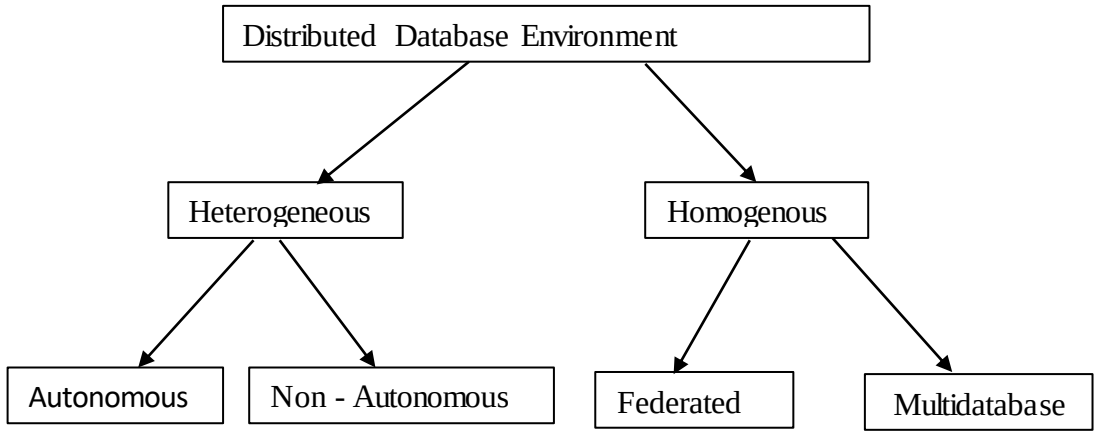


Figure 7: Classification of Distributed Database Environment (source: https://www.tutorialspoint.com)

3.3.2.1 Homogeneous Distributed Database System

Homogeneous distributed database systems use the same DBMS software from multiple site. Data exchange between these various sites can be handled easily. The environment of a homogeneous is typically defined by the following features:

i. Data are distributed across all the nodes.
ii. The same DBMS is used at each location.
iii. All data are managed by the distributed DBMS (so there are no exclusively Local data).
iv. Operating system may vary

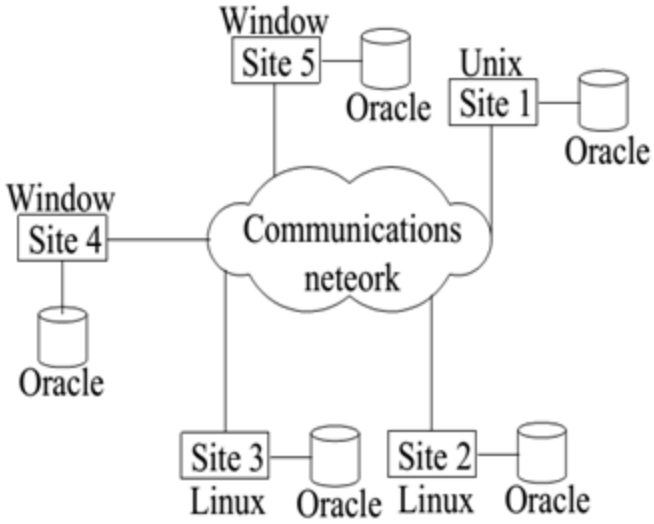


Figure 8: Homogeneous distributed database system (source: google)

Data exchange or access policy between these various sites gives rise to two types of homogeneous distributed database: Autonomous and Non-autonomous

i. Autonomous

Each database is independent and functions on its own at each of the sites. All the sites are integrated by a controlling application and use message passing to share data updates. That is, access to databases is done by a controlling application and a message passing to share data updates.

ii. Non-autonomous

Data is distributed across the homogeneous nodes and a central or master DBMS co-ordinates data updates across the sites.

3.3.2.2 Heterogeneous Distributed Database Systems

In a heterogeneous distributed database system, different sites may use different database model, DBMS software and operating systems, but access to data policy can be different - a single conceptual schema (global) access policy or multi conceptual schema access policy. The Figure 10 depicts heterogeneous distributed database systems.

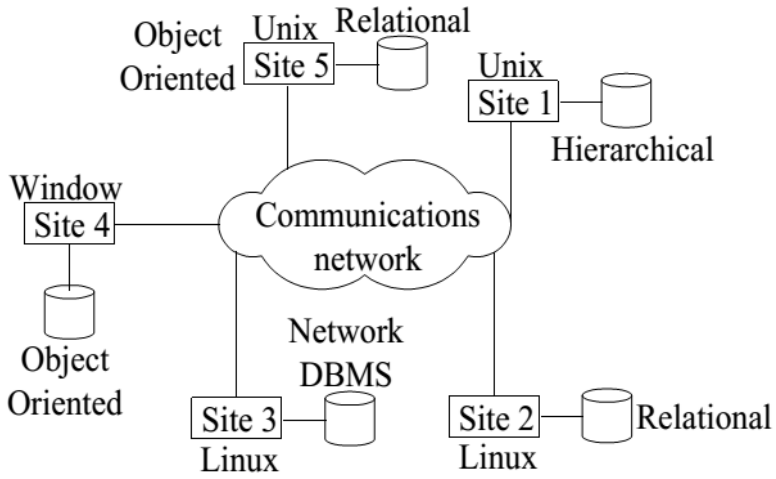


Figure 9: Depicts Heterogeneous Distributed Database Systems (Source:google)

The data access policy gives rise to two types of heterogeneous distributed database

i. Federated: Here each site may run different database system but the data access is managed through a single conceptual schema. This implies that the degree of local autonomy is minimum. Each site must adhere to a centralized access policy. There may be a global schema.

Federated distributed Database Management Systems has the following Issues:

- Differences in data models: Relational, Objected oriented, hierarchical, network, etc.
- Differences in constraints: Each site may have their own data accessing and processing constraints.
- Differences in query language: Some site may use SQL, some may use SQL-89, some may use SQL-92, and so on.

ii. Multidatabase: There is no one conceptual global schema. For data access, a schema is constructed dynamically as needed by the application software.

3.3.3 Distributed Database Management System (DDMS) Features

Since a distributed database management system (DDBMS) is a centralized software system that manages a distributed database in a manner as if it were all stored in a single location, it has following features:

i. It is used to create, retrieve, update and delete distributed databases.
ii. It synchronizes the database periodically and provides access mechanisms by the virtue of which the distribution becomes transparent to the users.
iii. It ensures that the data modified at any site is universally updated.
iv. It is used in application areas where large volumes of data are processed and accessed by numerous users simultaneously.
v. It is designed for heterogeneous database platforms.
vi. It maintains confidentiality and data integrity of the databases.

3.3.4 Factors for Organization's Choice for DDBMS

The following factors encourage organizations to migrate to DDBMS

ii. Physical distributed nature of organizational units: Most organizations in the nowadays are subdivided into multiple units that are physically distributed over the globe. So, the overall database of the organization becomes distributed as each unit requires its own set of local data.
iii. Need for sharing of data: Various organizational units regularly need to communicate with each other and share their data and resources. This demands

common databases or replicated databases that should be used in a synchronized approach.

iv. Support for both OLTP and OLAP: Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP) work upon diversified systems which may have common data. Distributed database systems aid both these processing by providing synchronized data.

v. Database recovery: One of the common techniques used in DDBMS is replication of data across different sites. Replication of data repeatedly helps in data recovery if database in any site is damaged. Users can access data from other sites while the damaged site is being restored.

vi. Support for multiple application software: Quite number of organizations use a variety of application software each with its specific database support. DDBMS provides a uniform functionality for using the same data among dissimilar platforms.

3.3.5 Advantages of Distributed Database System (DDBS)

i. It increases reliability and availability. The problems experienced in one branch or site of the organization can not affect other branches in the same manner

ii. It supports smooth transactions due to replication of the database.

iii. It supports hardware, operating system, network, fragmentation, DBMS, replication and location independence.

iv. Its distributed query processing improves performance.

v. It supports distributed transaction management

vi. Performance of system cannot be affected by single-site failure.

3.3.6 Disadvantages of Distributed Database System (DDBS)

According to Tomar, P. and Megha (2014) the following are the various disadvantages of distributed databases.

i. Complexity-A distributed database is more complicated to setup and maintain as compared to central database system.

ii. Security–There are many remote entry points to the system compared to central system leading to security threats.

iii. Data Integrity–In distributed system it is very difficult to make sure that data and indexes are not corrupted.

iv. In distributed database systems, data need to be carefully placed to make the system as efficient as possible.

v. Distributed databases are not so efficient if there is heavy interaction between sites.

vi. Failures: Several types of failures may occur in distributed database systems like, transaction failure, site failure, media failure, and communication failure.

vii. Economics: Increased complexity and a more extensive infrastructure means extra labour costs.

3.3.6 Distributed Databases Design Methods

Designing Distributed Databases involve two main ingenuities: the top-down method and the bottom-up method (Özsu, M. T. & Valduriez, P., 2011; Singh, I. & Singh, S., 2015; Hiremath D., S. & Kishor S., B., 2016). These methods deliver very different techniques in the design process. The top down method is much more suitable when designing homogeneous strongly cohesive Distributed Databases, while the bottom-up method is more suitable for heterogeneous or multidatabases (Katembo K. E., Shri K., & Ruchi A. , 2019).

a. Top-down method

More often than not, top-down method is used when the Distributed database is implemented from start as shown in the Figure 3. The design process starts from the analysis of requirements. The design process phases include the company situation analysis, the problems definition and constraints, the objectives definition, and the scope design and boundaries (Özsu, M. T. & Valduriez, P; Katembo K. E., Shri K. & Ruchi A., 2019; Gadicha, A. B. et, al., 2012).

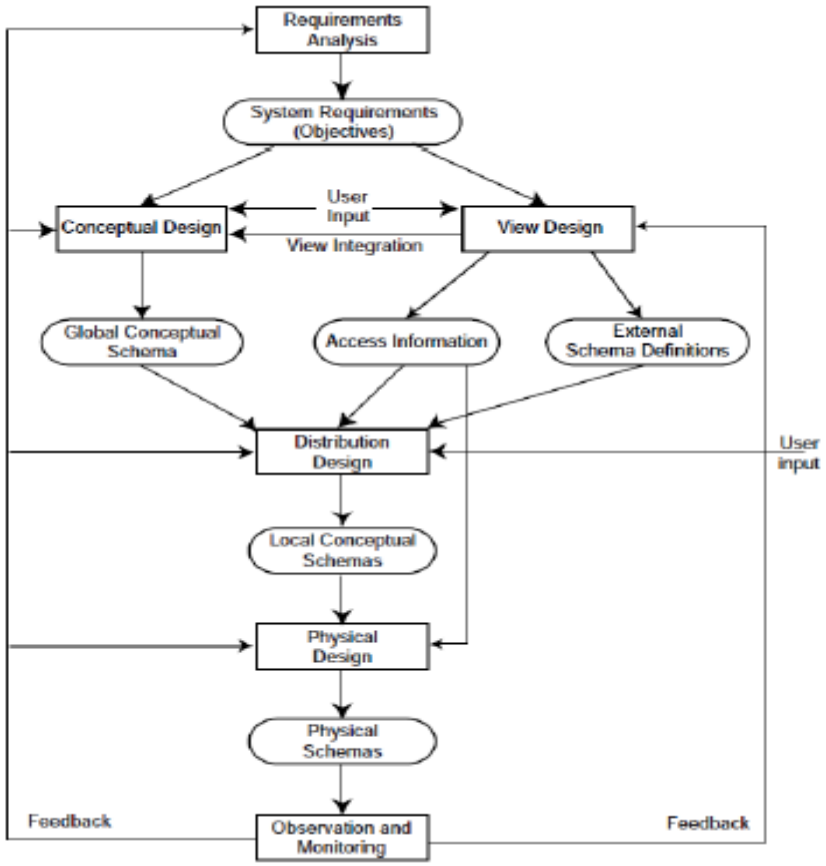


Figure 11: Top-Down Design Method (Özsu, M. T. & Valduriez, P. 2011)

- The conceptual modelling and view design are two next level tasks concerned. The conceptual modelling formalizes and standardizes the entity relationships while focusing on data requirements. Its modelling process controls the types of entities and the relationships among them and then the entity analysis advances to determine the entities and their attributes.
- The View design provides interface for the end users. The functional analysis connected determines the fundamental functions involved in associating the modelling.
- The View integration is the activity that defines the conceptual model which supports existing applications as well as future applications (Özsu, M. T. & Valduriez, P. 2011).

b. Bottom-up method

This method is used when Distributed Database already exists and requires scalability to other features or another Database have to be integrated into the existing environment (Hiremath D., S. & Kishor S. B., 2016). This method provides integration capability of several existing local schemas into a global conceptual schema in already developing distributed system. The bottom-up method is adopted when combining several existing databases to develop a distributed system because it is based on the integration of several

existing local schemas into a single global schema. This capability integrates more than one existing heterogeneous databases to build a distributed database system. This is also called ascending order method. Therefore, (Özsu, M. T. & Valduriez, P. , 2011; Singh, I. & Singh, S., 2015 ; Gadicha, A. B. et, al. , 2012) states that the Bottom-up design process requires the following steps:

- The selection of a mutual prototype to describe the global schema of the database;
- The conversion of all local schemas into a mutual data model;
- The unification of local patterns to arrive at a mutual global schema

3.3.7 Proposed Model for Bottom-up Design Method for Distributed Database

We propose a mathematical model for distributed database design (Özsu, M. T. & Valduriez, P. , 2011; Singh, I. & Singh, S., 2015 ; Gadicha, A. B. et, al. , 2012) for easy expression and interpretation of the integration of global conceptual schema and the local schemas since distributed database system can only be achieved by integration especially with heterogeneous databases.

Let's consider three heterogeneous local database schema to be integrated into a single global distributed database schema:

Let O represent an Oracle db, M = Microsoft SQLServr db,

S = MySQL db.

Let db = Database

L = db (local database)

Let f(G) = function of a global distributed database schema

f(L) = function of a local database schema function

Thus: The Integrated global database of heterogeneous databases of Oracle, Microsoft SQLServer and My SQL can be represented as follow:

$$f(G) = \sum_{\substack{i=1,j=1,k=1 \\ i\in O,\ j\in M,\ k\in S}}^{N} f(L_{ijk}), \text{ where } L = db \text{ and } \{x: \forall\ O, M, S\}$$

Nothing that i,j,k are mutually exclusive.

The design of the Distributed Database System integrating global conceptual schema and local schemas base on the three-level architecture of the DBMS in all sites; the complex design for establishment of a computer network across sites of a distributed system; and the modelling and implementation of all these make Distributed Database System a daunting task (Singh, I. & Singh, S., 2015; Bhuyar P. R., Gawande A. D., & Deshmukh A. B., 2012). The Distributed Database system provides the capability for fragmentation, replication and allocation of data on several sites with help of efficient query join operators and optimization. Data fragmented, replicated and allocated are strategies used to distribute to different sites. For the sake of space, details of distribution strategies shall be the focus of our next work.

CONCLUSION

In a distributed database system, from a more global view, however, it can be identified that the fundamental reason behind distributed processing is to be better able to cope with the challenges of huge data management problems that we face today, by using a variation of the well-known divide-and-conquer. Processing logic or processing elements, controls, and functions are the distributed objects for distribution of relation objects, while the relations

object remain the main object of distribution via the computer networks. Data fragmented, replicated and allocated are strategies used to distribute to different sites. Our proposed model becomes a mathematical model for understanding, interpreting and implementing the complex distributed database design method for integration of mutually exclusive local schemas and mutual global conceptual schema in the heterogeneous model.

REFERENCES

Almeida F. and Calistru, C. (2012). The main challenges and issues of big data management. *International Journal of Research Studies in Computing* , 2(1). 11-20

Bhuyar P. R., Gawande, A. D., and Deshmukh A. B. (2012). Horizontal fragmentation technique in distributed database. *International Journal of Scientific and Research Publications*. 2(5).1-7.

Gadicha A.B, Alvi AS, Gadicha VB, Zaki SM (2012). Top-Down Approach Process Built on Conceptual Design to Physical Design Using LIS, GCS Schema. *In,ternational Journal of Engineering Sciences & Emerging Technologies*.

Hiremath D., S. and Kishor S., B. (2016). Distributed Database Problem areas and Approaches. *Journal of Computer Engineering: National Conference on Recent Trends in Computer Science and Information Technology*, 2278-8727.

Katembo K. E., Shri K. and Ruchi A. (2019). A Systematic Review on Distributed Databases

Kaur K., Singh H., "Distributed database system on web server: A Review". *International Journal of Computer Techniques*, 3, pp. 12-16,

Michel A. et al. (2016). Big Data Management Challenges, Approaches, Tools and their limitations.
Retrieved 2021 from https://www.researchgate.net/publication/295134268_Big_Data_Management_Challenges_Approaches_Tools_and_their_limitations.

Özsu M. T, Valduriez P. (2011). Principles of distributed database systems. *Springer Science & Business Media* .

Özsu, M. T., & Valduriez, P. (2011). Introduction Principles of Distributed Database Systems. *Third Edition, 1–40*. doi:10.1007/978-1-4419-8834-8_1

Özsu, M.T. & Valduriez, P. (2001). Principles of Distributed Database Systems. *fourth Edition.*

Shareef M. I. , Rawi A.W.(2011). The Customized Database Fragmentation Technique in Distributed Database Systems.

Singh, I. and Singh, S. (2015). Distributed Database Systems: Principles, Algorithms and Systems, New-Delhi, India: *Khanna Book Publishing, Co.(P) Ltd*.

Tomar, P.(2014). An overview of distributed databases. *International Journal of Information and Computation Technology.*