e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-13, Issue-2 (April 2023)
https://doi.org/10.31033/ijemr.13.2.19

A Study of Distributed Database Management System

Shikha Mahajan[1], Dr. Leena Jain[2] and Rupali[3]
[1]Assistant Professor, Global Group of Institute, Amritsar, Punjab, INDIA
[2]HOD, Global Group of Institute, Amritsar, Punjab, INDIA
[3]MCA Student, Global Group of Institute, Amritsar, Punjab, INDIA

[1]Corresponding Author: shikhamahajan46@gmail.com

Received: 13-03-2023 **Revised:** 01-04-2023 **Accepted:** 21-04-2023

ABSTRACT

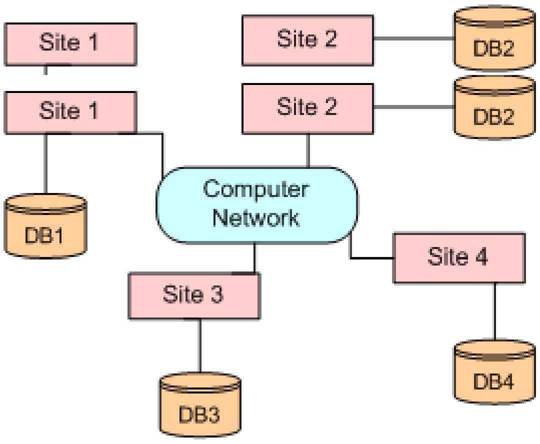
A Database is a collection of data describing the activities of one or more related organizations with a specific well defined structure and purpose. A Database is controlled by Database Management System (DBMS) by maintaining and utilizing large collections of data. Distributed computer applications built from off the shelf hardware and software are increasingly common at networked computers communicate and coordinate their activity only by-passing messages.[2] Distributed Database System the database is stored/spread physically across computers or sites in different locations that are connected by some form of data communication network. They may be spread over WAN or LAN. The computers may be of different types such as IBM Mainframes, VAXs, SUN workstation, PCs etc managed by different operating systems and each fragment of the data base may be managed by a different DBMS such as Oracle, Ingress, and Microsoft SOL server. This paper presents an overview of Distributed Database System[1]

Keywords-- Distributed Database Management System, Distributed Databases Architecture, Data Fragmentation, Complete Replication, Types of Distributed Database Systems

I. INTRODUCTION

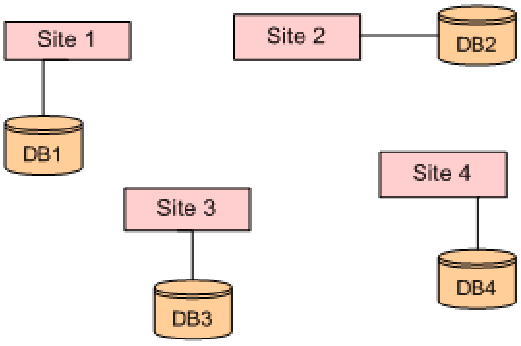
Information storage has been a challenging endeavor throughout human history and existed even before modern computer systems. The last three decades are marked by rapid growth of computer technology. This has raised the needs to evolve new techniques to manage huge amounts of data. Today, mostly centralized databases are used to store and manage data. They carry the advantages of high degree of security, concurrency and backup and recovery control. However, they also have disadvantages of high communication costs (when the client is far away and communication is very frequent), unavailability in case of system failure and a single source bottlenecks These issues raise the need of distribution of databases over various systems or locations. But the main motivation behind the concept of distributed databases is the efficient management of huge amount of data with increased availability and reduced communication cost. Research conducted in 1991 for distributed databases predicted a huge shift from traditional databases to distributed databases in the coming arena primarily due organizational needs to manage huge amounts of data. According to like, many applications in the future will be distributed due to the development in technology, and therefore the databases will also be distributed[7].

Figure 1: Distributed database System[8]



e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-13, Issue-2 (April 2023)
https://doi.org/10.31033/ijemr.13.2.19

Figure 2: Decentralized database System[8]



Distributed database management system (DDBMS)In a DDS, database applications running at any of the system's sites should be able to operate onany of the database fragments transparently as if the data come from a single database managed by one DBMS. The software that manages a distributeddatabase in such a way is called DDBMS. The notion of distributed database is different from that of decentralized database. The latter does not imply sharing of data by a communication network. The former implies a collection of sites connected together with some kind of network and where each site has a database in its own right, but the sites work together as if data was stored at only one site.[8]

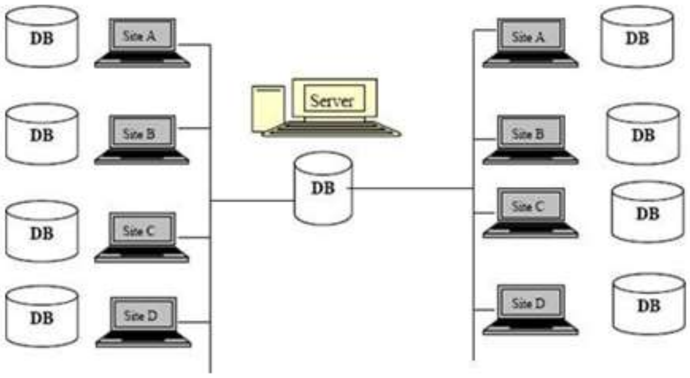
II. DISTRIBUTED DATABASES ARCHITECTURE

"A distributed database is a collection of multiple, logically interrelated databases distributed over a computer network" It may also be a single database divided into chunks and distributed over several locations. The database is scattered over various locations which provides local access to data and thus reduces communication costs and increases availability. Most of today‟s business applications have shifted from traditional processing to online processing. This has also changed the database needs of the applications.

Today, the role of databases to organize voluminous data has increased compared to previous era. Large companies need to distribute their data for many reasons for being economic and competitive [6]. However, the main motivation behind the concept of data distribution is the efficient management of huge amounts of data with increased availability and reduced communication cost. So, it has become a very attractive solution for areas like: online banking, e-commerce merchant, HR departments, telecommunication industry and air line ticketing etc.

Generally, distributed database is the collection of databases distributed across different locations or sites over a network as illustrated in Figure 4. Similarly, it may also be a single database, divided into chunks and distributed over various sites.[9]

Figure 3: Distributed Database System and Architecture[9]



Each site has a certain amount of data that it needs frequently and it can getthe rest from some other site. Distributed databases are very useful when availability and fast response time is needed. They increase performance andreduce communication costs.

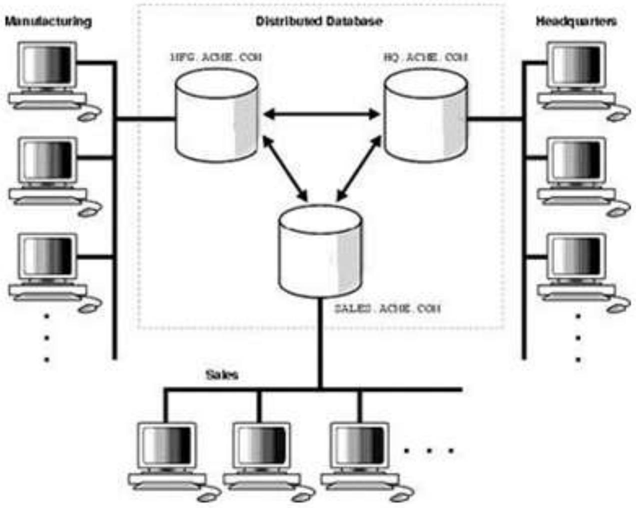
Distributed database design: The methodology used for the logical design of a centralized database applies to the design of the distributed one as well.

However, for a distributed database three additional factors have to be considered.

- **Data Fragmentation:** Before we decide how to distribute the data we must determine the logical units of distribution. The database may be broken up into logical units called fragments which will be stored at different sites. The simplest logical units are the tables themselves.

- **Horizontal fragmentation:** A horizontal fragment of a table is a subset of rows in it. So horizontal fragmentation divides a table 'horizontally' by selecting the relevant rows and these fragments can be assigned to different sides in the distributed system (for ex. Euston Road branch gets the fragment where myTable.branch ='Euston Road').
- **Vertical fragmentation:** A vertical fragment of a table keeps only certain attributes of it. It divides a table vertically by columns. It is necessary to include the primary key of the table in each vertical fragment so that the full table can be reconstructed if needed.
- **Mixed fragmentation:** In a mixed fragmentation each fragment can be specified by a SELECT-PROJECT.[9]

Figure 4: Distributed Database system Architecture[9]



Combination of operations: In this case the original table can be reconstructed be applying union and natural join operations in the appropriate order.

- **Data Replication:** A copy of each fragment can be maintained at several sites. Data replication is the design process of deciding which fragments will be replicated.
- **Data Allocation:** Each fragment has to be allocated to one or more sites, where it'll be stored.

There are three strategies regarding the allocation of data:

- **Fragmented (or partitioned):** The database is partitioned into disjoint fragments, with each fragment assigned to one site (no replication). This is also called 'non-redundant allocation'.
- **Complete replication:** A complete copy of the database is maintained at each site (no fragmentation). Here, storage costs and communication costs for updates are most expensive. To overcome some of these problems, snapshots are sometimes used. A snapshot is a copy of the data at a given time. Copies are updated periodically.
- **Selective replication:** A combination of fragmentation and replication.[9]

III. PROBLEMS AND ISSUES OF DISTRIBUTED DATABASE DATABASES

Distributed database systems have advantages of high availability, good response time and reduced communication cost. However, there are some crucial and pivotal issues which must be taken into account. As compared to centralized databases the management of distributed databases is complex as it is scattered over several locations. It also raises many security issues. Data is prone to interception while communicating. Controlling widespread data is another issue as a single database administrator cannot control the overall distribution. To implement the concept of distributed databases special software products and tools are needed which are expensive to purchase and complex to operate. Improper distribution of data may also lead to poor response time which affects the overall performance of the system. Another main problem with distributed databases is to control concurrency. It becomes very difficult to control the concurrency when multiple users are accessing the same piece of data and there are many read and write requests at the same time.

Databases may be called a cornerstone as its ability to store, process and manage information is a key to any organization prevailing in any sector. The telecom sector is growing very fast as its number of users continues to increase. The telecom industry performs a variety of actions like customer relationship management, market analysis, the evaluation of call detail records, analysis of customer churn, complex billing system and personalized telecommunication services [5]. Performing these tasks not only increases customer satisfaction, but it also gives the companies a competitive edge.

Using centralized databases makes it problematic for large organizations to be competitive. So, an alternative database technology is needed to overcome this. This alternative is to distribute the database. So, it's important issues like backup and recovery, concurrency control, security, availability and its implementation need to be investigated. Along with these issues the main concern of this study is to distribute a data and check how it will affect the response time.[10]

IV. TYPES OF DISTRIBUTED DATABASE SYSTEMS

Distributed Database Systems are broadly classified into two types:

- Homogeneous Distributed System - In Homogenous distributed database system, the data is distributed but all servers run the same Database Management System(DBMS) software
- Heterogeneous Distributed System–In Heterogeneous distributed databases different sites run under the control of different DBMSs, These databases are connected somehow to enable access to data from multiple sites.[11]

V. ADVANTAGES OF DISTRIBUTED DATABASES

- Robust–A problem in one part of the organization will not stop other branches working.
- Security- Staff access can be restricted to only their portion of databases.
- Network traffic is reduced, thus reducing the bandwidth cost.
- Local database still works even if the company network is temporarily broken.
- High Performance–Queries and updates are largely local so that there is no network bottleneck.
- In distributed systems it is easier to keep errors local rather than the entire organization being affected.[8]

VI. DISADVANTAGES OF DISTRIBUTED DATABASES

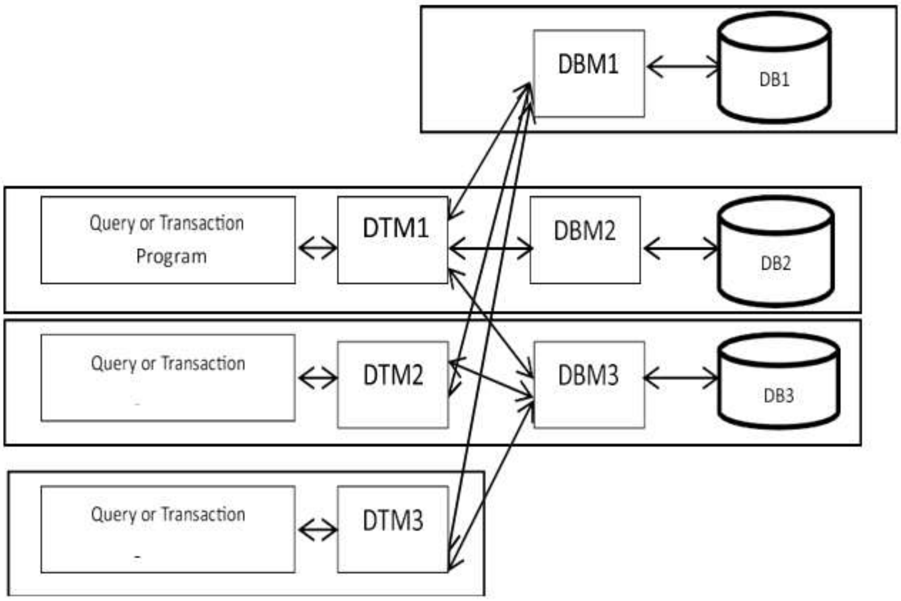
Following are the various disadvantages of distributed databases [9, 10]:

- Complexity-A distributed database is more complicated to setup and maintainas compared to central database system.
- Security–There are many remote entry points to the system compared to centralsystem leading to security threats.
- Data Integrity–In distributed system it is very difficult to make sure that dataand indexes are not corrupted.
- In distributed database systems, data need to be carefully placed to make the system as efficient as possible.
- Distributed databases are not so efficient if there is heavy interaction betweensites.[6]

VII COMPONENT OF DISTRIBUTED DATABASE SYSTEMS

- Distributed Database System consists of the various components (Fig. 3). Database manager is one of major component of Distributed Database systems. Database Manager is software responsible for handling a segment of the distributed database. User Request Interface is another important component of distributed database systems. It is usually a client program which acts as an interface to the Distributed Transaction Manager.
- Distributed Transaction Manager is a program that helps in translating the user requests and converting into format required by the database manager, which are typically distributed. A distributed database system is made of both the distributedtransaction manager and the database manager.[9]

Figure 5: Component Diagram of distributed Database[9]



VIII. PROBLEMS IN DISTRIBUTED DATABASE SYSTEMS

One of the major problems in distributed systems is deadlock. A deadlock is astate where a set of processes request resources that are held by other processes in the set and none of the process can be completed [14, 15, 16]. One process can request and acquire resources in any order without knowing the locks acquired by other processes. If the sequence of the allocations of resources to the processes is not controlled, deadlocks can occur. Hence we focus on deadlock detection and removal.[3]

Deadlock Detection

To detect deadlocks, in distributed systems, deadlock detection algorithm must be used. Each site maintains a local wait for graph. If there is any cycle in the graph, there is a deadlock in the system. Even though there is no cycle in the local wait for graph, there can be a deadlock. This is due to the global acquisition of resources. To find the global deadlocks, global wait for graph is maintained. This is known as centralized approach for deadlock detection.

The centralized approach to deadlock detection, while straightforward to implement, has two main drawbacks. First, the global coordinator becomes a performance bottleneck, as well as a single point of failure. Second, it is prone to detecting non-existing deadlocks, referred to as phantom deadlocks.[5]

Deadlock Recovery

A deadlock always involves a cycle of alternating process and resource nodes in the resource graph. The general approach for deadlock recovery is process termination. In this method, nodes and edges of the resource graph are eliminated. In Process Termination, the simplest algorithm is to terminate all processes involved in the deadlock. This approach is unnecessarily wasteful, since, in most cases, eliminating a single process is sufficient to break the deadlock. Thus, it is better to terminate processes one at a time, release their resources, and check at each step if the deadlock still persists. Before termination of process following parameters need to be checked:

a) The priority of the process:
b) The cost of restarting the process
c) The current state of the process[4]

IX. CONCLUSION

In the current scenario of the fast changing world, distribution of data became the necessity. Distribution of data has its own advantages and disadvantages. This paper presents a complete review on distributed databases. It is clear from the study that distribution of data involves the problem of deadlock. We need to find out the methods to data distribution and accessing which leads to minimization of deadlock and thus resulting in proper utilization of resources.

REFERENCES

[1] Distributed database design methodologies | IEEE Journals & Magazine | IEEE Xplore.
[2] www.webopedia.com/TERM/D/distributed_database.html.
[3] Distributed Database System – GeeksforGeeks.
[4] Database – Wikipedia.
[5] Distributed database design methodologies | IEEE Journals & Magazine | IEEE Xplore.

[6] (PDF) A systematic review on Distributed Databases Systems and their techniques (researchgate.net).

[7] Distributed Database Design: A Case Study – ScienceDirect.

[8] Bernstein, P.A., Hadzilacos, V. & Goodman, N. (1987). *Concurrency control and recovery in database systems*. Addison-Wesley, Reading,

[9] International Journal of Trend in Research and Development, Volume 2(5).

[10] International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 4, Number 2 (2014), pp. 207-214 © International Research Publications House http://www. irphouse.com /ijict.htm.

[11] Problem processing your request — ScienceDirect.

[12] Review on Distributed Database Systems by Mateo Santiago : SSRN.