

## Fragmentation Issues in Distributed Database

**D.S.Hiremath<sup>a</sup>, S.B.Kishor<sup>b</sup>**

<sup>a</sup>Research Scholar Gondwana University, Gadchiroli, MS, India

<sup>b</sup>Head of Department of Computer Science, S.P.College, Chandrapur, MS, India

---

### Abstract

Distributed database play and important role in the field of database. The main objective the the distributed database is to maintain the reliability and avaiability of data. As all of us know the the data is asset of any organization such banking sector, marketing etc. In this paper the researcher study varios Fragmentation issues while executing the Query .Design of distributed database is one the most important step in implementing the Distributed database. If designing of distributed database is not proper then their related problems are occurs such query processing , deadlock etc. So in this paper the researcher is studying about design process of distributed database and what are the different issues are arises while design the distributed database.Fragmentation, Replication and allocation of data is very important with proper precautions and directions otherwise that affect on the performance of Distributed Database. This paper introduced the fragmentation, replication and allocation strategies and related issues with Distributed database.

**KEYWORDS** :- Distributed Database, Fragmentation, Replication, Query Processing, Deadlock, Allocation.

---

### I. Introduction

*In* todays growing technology distributed database is play an vital role in the management of database over the computer network. In any organisation reliability and avaiability of data is very important. Distributed database is a collections of number of nodes which have own database and share the data among various other nodes via computer network. Distributed database system also distributed database management system, distributed database and network connections. The main goal of Distributed Database management system is to control distributed database and apperas as centralised database to users.

The Distributed database is classified in Homogeneous and Heterogeneous distributed database. Homogeneous distributed database is collections of similar database, software and hardware configurations. While Heterogeneous distributed database is collections of different databases, software and hardware configurations. The different nodes are connected and communicates with each other through computer network. The following fig shows the typical example of distributed database.

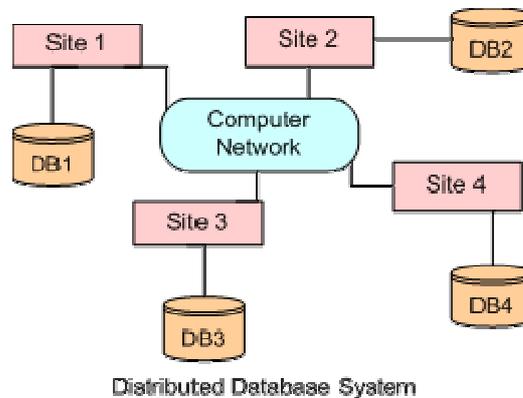


Fig. 1- Distributed database Environment

## II. Literature Review

### *Design Issues in Distributed Database*

#### 1) Distributed Database Design

One of the main questions that is being addressed is how database and the applications that run against it should be placed across the sites. There are two basic alternatives to placing data: partitioned (or no-replicated) and replicated. In the partitioned scheme the database is divided into a number of disjoint partitions each of which is placed at different site. Replicated designs can be either fully replicated (also called fully duplicated) where entire database is stored at each site, or partially replicated (or partially duplicated) where each partition of the database is stored at more than one site, but not at all the sites[4]. The two fundamental design issues are fragmentation, the separation of the database into partitions called fragments, and distribution, the optimum distribution of fragments

#### 2) Distributed Directory Management

A directory contains information (such as descriptions and locations) about data items in the database. A directory may be global to the entire DDBS or local to each site; it can be centralized at one site or distributed over several sites; there can be a single copy or multiple copies [6].

#### 3) Distributed Query Processing

Query processing deals with designing algorithms that analyze queries and convert them into a series of data manipulation operations. The problem is how to decide on a strategy for executing each query over the network in the most cost-effective way, however cost is defined [1]. The factors to be considered are the distribution of data, communication cost, and lack of sufficient locally-available information

#### 4) Distributed Concurrency Control

Concurrency control involves the synchronization of access to the distributed database, such that the integrity of the database is maintained. It is, without any doubt, one of the most extensively studied problems in the DDBS field. The concurrency control problem in a distributed context is somewhat different than in a centralized framework [1]. One not only has to worry about the integrity of a single database, but also about the consistency of multiple copies of the database

#### 5) Reliability of Distributed DBMS

It is important that mechanisms be provided to ensure the consistency of the database as well as to detect failures and recover from them. The implication for DDBSs is that when a failure occurs and various sites become either inoperable or inaccessible, the databases at the operational sites remain consistent and up to date. Furthermore, when the computer system or network recovers from the failure, the DDBSs should be able to recover and bring the databases at the failed sites up-to-date. This may be especially difficult in the case of network partitioning, where the sites are divided into two or more groups with no communication among them [4].

#### 6) Replication

If the distributed database is (partially or fully) replicated, it is necessary to implement protocols that ensure the consistency of the replicas, i.e. copies of the same data item have the same value. These protocols can be eager in that they force the updates to be applied to all the replicas before the transactions completes, or they may be lazy so that the transactions updates one copy (called the master) from which updates are propagated to the others after the transaction completes[2].

### III. Material and Methods

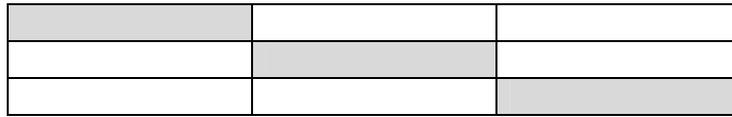
In Distributed database distribution of design is major part of designing. In designing of distributed database Fragmentation and Replication of database should be in proper way so Reliability and performance maintained.

The Objective of Fragmentation aims to improve reliability, performance, communication cost and security. Following information is used to decide fragmentation:-

- i) Frequency of Queries
- ii) Sites (nodes)
- iii) Where Query is run
- iv) Types of access of data

#### Types of Fragmentation:-

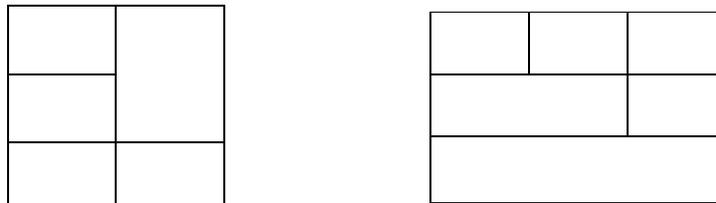
- 1) **Horizontal Fragmentation:** - It is division of relation along with rows (tuples).
- 2) **Vertical Fragmentation:** - It is division of relation along with columns (attributes).
- 3) **Mixed Fragmentation:-** It is Combination of Horizontal and Vertical Fragmentation.



**Fig. 1: Horizontal Fragmentation**



**Fig. 2 : Vertical Fragmentation**



**Fig. 3: Mixed Fragmentation**

**Consider Example of PROJ Relation**

**Horizontal fragmentation of PROJ relation**

PROJ1: projects with budgets less than 200, 000

PROJ2: projects with budgets greater than or equal to 200, 000

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PROJ<sub>1</sub>

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop	135000	New York

PROJ<sub>2</sub>

PNO	PNAME	BUDGET	LOC
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

### Vertical Fragmentation of PROJ Relation

PROJ1:- Information of about Project Budgets

PROJ2: Information about Project Names and Location

PROJ

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York
P4	Maintenance	310000	Paris
P5	CAD/CAM	500000	Boston

PROJ<sub>1</sub>

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000
P5	500000

PROJ<sub>2</sub>

PNO	PNAME	LOC
P1	Instrumentation	Montreal
P2	Database Develop.	New York
P3	CAD/CAM	New York
P4	Maintenance	Paris
P5	CAD/CAM	Boston

### IV. Analysis and Result

After Studying example of Horizontal and Vertical Fragmentation, we perform simple query on horizontal fragmentation. Consider the following query

“Find the name and budget of projects given their PNO”.

The Query is issued at three two sites and Fragmentation based on LOC Using set of Predicates.

{LOC= 'Montreal', LOC=' New York}

$$PROJ_1 = \sigma_{LOC='Montreal'}(PROJ)$$

PNO	PNAME	BUDGET	LOC
P1	Instrumentation	150000	Montreal

$$PROJ_2 = \sigma_{LOC='NewYork'}(PROJ)$$

PNO	PNAME	BUDGET	LOC
P2	Database Develop.	135000	New York
P3	CAD/CAM	250000	New York

If accessed is only according to the Location above set of predicate is complete i.e., each tuple of each fragment PROJ<sub>i</sub> has the same probability of being accessed.

If there is second Query to access only those project tuples where the budget is less than 200000, the set of predicate is not complete. In that case we have to add budget in our query.

After analysis of Fragmented relation it is found that it is very complicated to decide how should and how much should we fragment, how to allocate fragments and test the correctness of fragmentation.

## V. Conclusion

There are various advantage and disadvantages of horizontal and vertical fragmentation. The Researcher found that while making fragmentation of relation it is necessary to focus on various information regarding fragmentation such frequency of queries, sites, where query is run, how much we should fragment. So proper fragmentation is done and distributed database is to maintain reliability and availability. The Research also found that using Vertical fragmentation distributed database give opportunity to many applications to run on only one fragment.

## VI Future Scope

This work purely working on Design issues of vertical and horizontal fragmentation issues. The fragmentation and replication Facing various issues in distributed database such deadlock, concurrency, storage space. Distributed database provided tremendous research area with respect to fragmentation and replication.

## VII. References

1. Andrew S Tanenbaum, "Distributed systems principles and paradigms", PHI
2. Ceri, S Pelagatti, G, "Distributed Databases: Principles and Systems"
3. DBMS – Complete Practical Approach by Sharad Maheshwari, Ruchin Jain
4. George Coulouris, "Distributed System", ean Dollimore Publication.
5. M Tamer Ozsu,"Principle of Distributed Database System" PHI
6. <http://www.ijeset.com/media/11N5-IJES0202505>
7. Saeed K.Rahimi and Frank S.Haug,"A Distributed Database Management System a practical approach", Wiley Publication.
8. Stefano Ceri , "Distributed database principals systems by", TMH
9. William Buchanan , "Distributed systems and networks",TMH
10. Wesley W. Chu, "Distributed Systems: Vol. II: Distributed Data Base Systems
11. P. R. Bhuyar, A.D.Gawande, A.B.Deshmukh (2012). Horizontal Fragmentation Technique in Distributed Database. International Journal of Scientific and Research Publication 2(5).pp.1-6.
12. Y.F. Huang, J.H. Chen (2001). Fragment Allocation in Distributed Database. Journal of Information Science and Engineering 17, pp.491-506