Data Replicas In Fragmentation In Distributed Computing

Reena Hooda
Asst Professor, Dept of CSE
Indira Gandhi University Meerpur (Rewari)- Haryana -122502, India.

Abstract- *The present paper contributed towards the implications of keeping replicas in distributed computing. In distributed computing, data is fragmented in to different parts and stored at different locations, so that can be managed and traffic can be reduced with minimal security requirements. However, the problem of failure is still there; if the site fails on which data fragment is stored. Second problem of handling increased users is still there that can leads to deadlock. Therefore, the role of retaining replicas of data fragments to handle certain issue in fragmentation is vital. In the current paper, different schemes of replicas and certain issues in maintain replicas are also discussed.*

Keywords- Fragments, Concurrency, Deadlock, Data Allocation, Traffic Load.

I. INTRODUCTION

Data fragmentation is the division of the original data into units (subsets) and these subset or partitions are known as fragments. The 3 basic schemes of fragmentation are vertical fragmentation, horizontal fragmentation and hybrid fragmentation. Vertical fragmentation involves the column wise partitioning of a table through the "projection", each fragments at least must contain 2 columns as it is mandatory to include the primary key column in each of the fragment so that the original table (databases) can be reconstructed without the loss of information. The horizontal fragmentation involves the division of a table (databases) on the bases of rows (tuples) based on the condition given in "select" statement. The horizontal fragmentation subdivided into primary horizontal fragmentation and derived horizontal fragmentation. Primary horizontal fragmentation is done on the primary (master) table whereas in derived horizontal fragmentation, the fragmentation is done on the dependent data table through the foreign key references to the primary table. The hybrid fragmentation includes both the vertical as well as horizontal fragmentation. [1] [3] Fragments must be logically connected through unions and outer join [3].Elementary tasks of distributed data bases management system for query processing in distributed computing containing different data fragments is shown in figure 1.

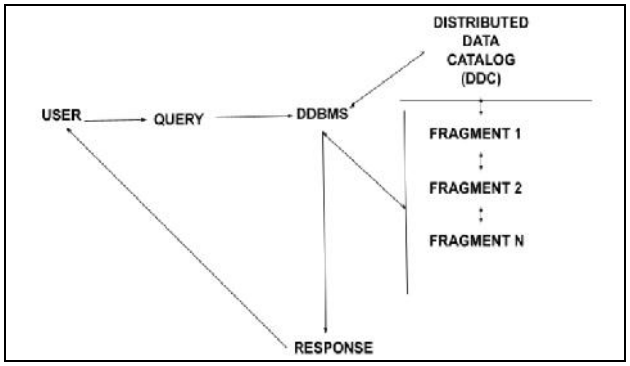

Figure 1: Elements of Data Fragmentation in Distributed Databases & Query Processing

As the fragments are stored on different sites or nodes [6], so any failure in the network whether it is site or link; may crash the system. Therefore, for the continuous availability of the data, copies of data fragments are maintained. [7]

If a given fragment is represented by many copies and stored on different sites which are connected to each other, these copies are considered as replicas. [2] These copies or replicas stored on different locations say computer systems or servers make continuous availability of data. [7]

A. Advantages of Replicas

Different advantages of creating replicas are as under:

1) Better performance: as data operations can be done locally on the sites near to user. [1] [2]
2) Availability of the data: Even if some of the sites or fragmentation fails, the data can be still available & accessed from the replicas. [1] [2] [3] [4]
3) Faster Execution: Due to parallel processing of data as replicas of same data are stored on different sites, parallel execution of transaction makes the processing faster. [6]
4) Reduced network Traffic: more the replicas, more chance to get the replicas locally near to user, therefore, network traffic is reduced and problem of congestion, network failure will not arise. [6]

5) Maintaining copies on different sites, adding more sites to the distributed system is less expensive than the centralized updations& maintaining standards. [9]
6) Distributed replicas like the distributed fragments resulted in less hits on a single replica therefore the problem of deadlock is reduced. [9]

B. Data Allocation Schemes in Fragmentation

Three kinds of schemes can be adopted in distributed computing given as under [4]:

1) Centralized: Where the whole database is stored centrally at a single place.
2) Partitioning/ fragmented: Data is stored on two or more sites.
3) Replicated: Data fragments are replicated/copied and stored on other sites.

II. DATA REPLICA IN DATA FRAGMENTATION

Data replica is not actually a strategy; it simply used to back up the data that can be recovered in case of failure or damage.Though, replicas can also be used to minimize the load on a single fragment. The process of replica is started with data fragmentation, data is first divided and then after the fragmentation, the fragments are backed up on different sites as copies or replica of the original data fragments. [4]
To maintain consistency, it is necessary to update or modify the same data at all the locations carefully. [4]. The usability of replica can be clarified with an example, for instance, if the data is divided horizontally into three parts based on the department (dept) one is" MCA", second is "accounts" and third is "MBA" so the data in F1 F2 and F3 as shown in figure 2.

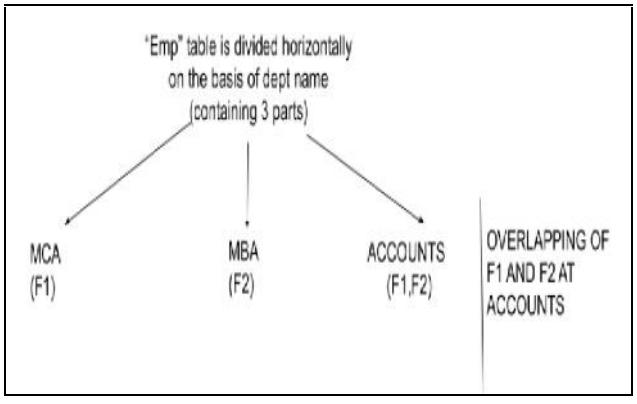

Figure 2: Data Duplicacyin Horizontal Fragmentation.

As both department data MCA and MBA is required by the accounts branch. To make the salary statement, information of both the department is also required at account branch. A replica of F1 and F2 is also contained in accounts branch, so if there is any change in F1 and F2 that must be reflected at account branch's replica F1 and F2 to maintain the consistency, accuracy and reliability of the data. [4] The cost of the replicas is due to its mutual consistency rule and storage requirements of copies and increased time in data transactions due to the update operations in all the replicas. [4] Query handling by DDBMS with replicas is shown in figure 3:

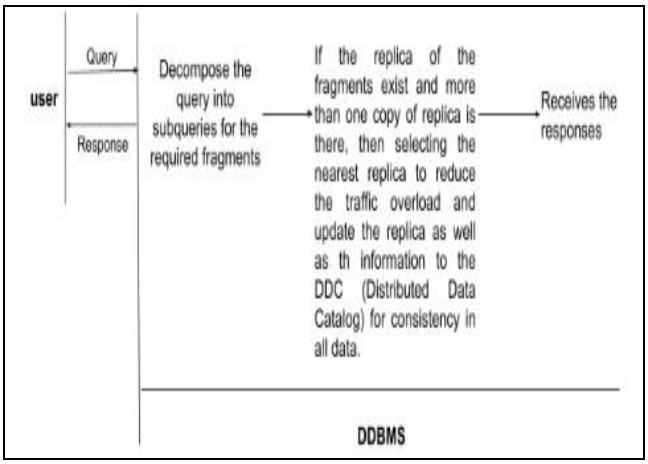

Figure 3: Shows DBMS tasks in case of Replicas.

Three Types of approaches can be there to create replicas [4]:

1) Fully replicated: In a fully replicated scheme, all the data fragments are copied on different sites.
2) Partially replicated: In partially replicated scheme, only few fragments are replicated on different sites.
3) No replica**:**in this scheme, any copy of the data fragments is maintained.

Factors Affecting the Replica

The key factors that can affect a replica strategy are as under [4]:

1) High bandwidth
2) High transaction time
3) Storage or memory requirements
4) Management of consistency
5) Frequency of updations in fragments as frequently accessed data is vulnerable to changes.
6) Selecting the right replica in fragmentation to answer a query.
7) Method to update all replicas after changes.

III. CONCURRENCY CONTROLLED IN DISTRIBUTED COMPUTING

When there are many replicas or copies are kept for the data items then maintaining the consistency in distributed databases is a big issue thus concurrency control is must. Another issue emerged when a site is failed on which the replica is stored then the issue of recovery is in a consistent way is the concern. [8] Failure can be a site failure or the network failure for example if some of the communication links fail or the link is not working then the connection to some sites also failed as a result [8]. Apart from these matters, the problem of distributed commit protocol is also a big issue that occurs when accessing data from multiple fragments or replicas stored on different sites and any of the sites crashed in a commit process. Distributed deadlock is another issue in distributed data access. To handle such issues some of the methods may be applied. For instance, when many copies/replicas are maintained, one of them is treated as a primary copy that is stored on a particular site and is called a distinguished copy. All the lock or unlock requests are needed to be sent to this copy on that particular site only [8], this site is also known as primary site. The several methods of handling this distinguished copy; one is the primary site method where the primary/distinguished copies (replica) are stored on the same single site. This site acts as coordinator for all the data items for which the primary copies are maintained as shown in figure 4. This technique is simple as an extension of centralized locking method [8]. However, as all the copies are kept at the same site, the problem of loading and storing can be occurred. Coordinating all stored information and maintaining distinguished copy on a single location and making it dependent on a single site will decrease the reliability of the system if such sites stop working. [8]

The second method is to take another site as backup of the primary site, reducing the problem of the primary site failure. All the locking and unlocking information is now stored on both the sites, primary as well as backup. So, if the primary site crashes, the backup site will act as a primary site. However, due to loading and storing the information on both sites, the process takes a lot of time before sending a response and therefore, replies process becomes slow. In case the coordinator site stopped and no backup site, it is necessary to abort all the transactions and restore the process again to restart it. If a backup site is there and that also fails, then choose another site as primary site that will act as coordinator and to maintain distinguished copy. [8]

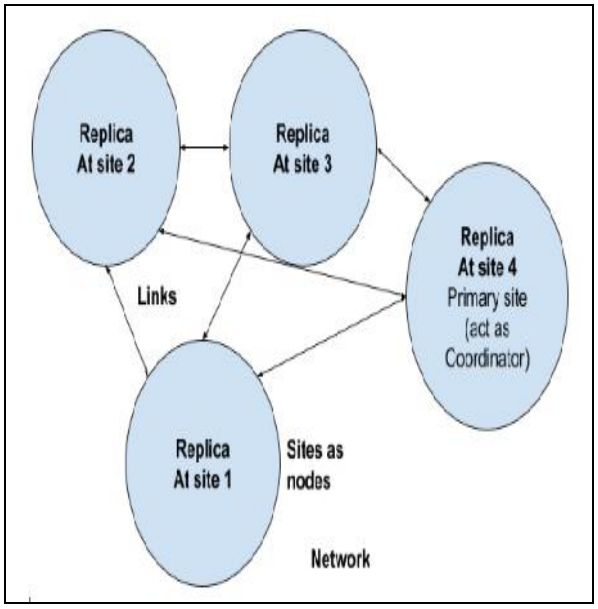

Figure 4: Shows the replicas in a network and coordinating site.

Issues in Replicas

The fragments are based on some criteria for instance, horizontal fragmentation is based on the conditions given under “where” clause in “select” statement like “select * from emp where salary > 50000”. If due to updations in the data, this condition is not satisfied then the tuples will be shifted to the other fragments. This will result in the updating of all the replicas accordingly and sending information to coordinator as well as to the distributed data catalogue (DDC). It is time consuming and costly but mandatory to maintain the consistency and accuracy in the data. This way the overload of the primary site acting as coordinator as well as the number of transactions increased that overall increases the traffic load and query response time thus may decrease the system performance. This scenario becomes worst, if any of the sites or communication links fail. [2] [4]This way, the overheads in replica updates, more disk space requirements makes it an expensive scheme too. [6] Maintaining standards for consistency in each replica, security levels at different locations and training to set up system & maintenance amplified the costs. [9]

IV. CONCLUSIONS

The current paper explained the different approaches of maintaining replicas and their control. The certain advantages and issues of the replicas are also discussed. The selection of a particular strategy is totally based on the size of data and number of users so that selection of replica scheme can match with the cost factors. Therefore, before selecting the

number and location of replicas the factors affecting replicas should also be considered which are also highlighted in this paper. To avoid the deadlock and loss of data availability of data, replicas are good options but the point of maintaining consistency after updations in data must be taken care off within time limits of response.

REFERENCES

[1] https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_design_strategies.htm
[2] https://www.cs.uct.ac.za/mit_notes/database/htmls/chp15.html
[3] https://www.tutorialcup.com/dbms/data-fragmentation.htm
[4] http://www.uobabylon.edu.iq/eprints/publication_4_2259_1575.pdf
[5] https://www.geeksforgeeks.org/data-replication-in-dbms/
[6] https://ecomputernotes.com/database-system/adv-database/data-replication
[7] https://www.tutorialride.com/distributed-databases/data-replication-in-distributed-system.htm
[8] https://www.brainkart.com/article/Overview-of-Concurrency-Control-and-Recovery-in-Distributed-Databases_11596/
[9] https://stevevincent.info/CIS2210_2018_12.htm