Distributed Database: Fundamentals & Design Techniques

Mrs. Sinha Nitya[1] Ms. Patil Sampada Dattatrey[2]
[1]Lecturer [2]ME Scholar
[1,2]Department of Computer Engineering
[1]Thakur Polytechnic, Kandivali, India [2]SLRTCE, Mira Road, India

Abstract— This paper reviews the coverage of concurrency management in a distributed information. Concurrency management is a vital part for correctness in any system wherever two or more transactions will access same source of data. This paper gives review of distributed database design. We tend to describe a fundamentals of concurrency control.
Key words: Distributed Database

I. INTRODUCTION

In today's world of universal dependence on data systems in the rising one would like for secure, reliable and accessible data. In today's business atmosphere, the requirement for distributed databases and client/ server applications is additionally increasing. A distributed information may be a single logical information that will reveal physically across computers in multiple locations. That specific area unit connected by digital communication links. Distributed information may be a quite virtual information whose part components area unit physically keep in a very range of distinct real databases at variety of distinct locations. The users at any location will access knowledge at any place within the network as if the info were all keep at the user's own location. A distributed direction system is that the package that manages the Distributed Databases, and provides an access mechanism that creates this distribution clear to the user. The target of a distributed direction system is to manage the management of a distributed information in such the simplest way that it seems to the user as a centralized information [1].

II. DISTRIBUTED DATABASE DESIGN

Distributed info Systems area unit required for the applications wherever knowledge and its accesses area unit inherently distributed and to extend the provision throughout failures. The prime examples area unit the systems for international air-line reservations, monetary establishments, and automatic producing. The methodology for planning Distributed Systems is same as that used for the Centralized databases. However, some extra factors are thought-about for a Distributed Database: [1]

A. Data Fragmentation

In Distributed Databases, we'd like to outline the logical unit of info Distribution and allocation. The info is also choppy into logical units known as fragments which is able to be hold on at completely different sites. The only logical units art he tables themselves. Three kinds of Fragmentation are:

1) Horizontal Fragmentation.
2) Vertical Fragmentation.
3) Hybrid Fragmentation.

1) Horizontal Fragmentation:

Horizontal fragmentation groups the tuples of a table in agreement to values of one or more fields. Horizontal fragmentation should also confirm to the rule of reconstructiveness i.e., Fragmentation should be done in a way so that the original table can be reconstructed from the fragments whenever required. Each horizontal fragment must have all columns of the original base table.

2) Vertical Fragmentation:

In vertical fragmentation, the fields or columns of a table are grouped into fragments. In order to maintain reconstructiveness, each fragment ought to comprise of the primary key field(s) of the table. Vertical fragmentation can be used to enforce privacy of data.

3) Hybrid Fragmentation:

In hybrid fragmentation, a combination of horizontal and vertical fragmentation techniques are used. This is the most flexible fragmentation technique since it generates fragments with minimal extraneous information. However, rebuilding of the original table is every so often a costly task.

Hybrid fragmentation can be done in two alternative ways −

- Initially, generate a set of horizontal fragments; then generate vertical fragments from one or more of the horizontal fragments.
- At the start, make a set of vertical fragments; after that generate horizontal fragments from one or more of the vertical fragments.

B. Data Replication

Data replication is the process of storing separate copies of the database at two or more sites. It is a popular fault tolerance technique of distributed databases [1].

1) Advantages of Data Replication:

- Reliability – When a failure of any site occurs, the database system continues to work since a copy is available at another site(s).
- Reduction in Network Load − as local copies of data are available, query processing can be done with reduced network usage, particularly during prime hours. Data updating can be done at non-prime hours.
- Quicker Response − Accessibility of local copies of data ensures swift query processing and thus faster response time.
- Simpler Transactions − Transactions involve fewer number of joins of tables positioned at different sites and marginal coordination across the network. Thus, they become simpler in nature.

2) Disadvantages of Data Replication:

- Increased Storage Requirements − Keeping multiple copies of data is connected with greater than before storage costs. The storage space necessary is in multiples of the storage required for a centralized system.
- Increased Cost and Complexity of Data Updating− Each time a data item is updated, the update needs to be

reflected in all the copies of the data at the different sites. This involves complex synchronization techniques and protocols.

– Undesirable Application – Database coupling − If complex update mechanisms are not used, removing data inconsistency requires complex co-ordination at application level. This results in undesirable application – database coupling.

3) Some commonly used replication techniques are

– Snapshot replication
– Near-real-time replication
– Pull replication

III. FUNDAMENTALS OF CONCURRENCY CONTROL

Transaction: A transaction consists of a series of operations performed on a database. The important issue in transaction management is that if a database was in a consistent state prior to the initiation of a transaction, then the database should return to a consistent state after the transaction is completed [2].

Properties of Transaction: A Transaction has four properties that lead to the consistency and reliability of a distributed database. These are Atomicity, Consistency, Isolation, and Durability.

1) Atomicity: This refers to the fact that a transaction is treated as a unit of operation. It dictates that either all the actions related to a transaction are completed or none of them is carried out.
2) Consistency: The consistency of a transaction is its correctness. In other words, a transaction is a correct program that maps one consistent database state into another.
3) Isolation: According to this property, each transaction should see a consistent database at all times. Consequently, no other transaction can read or modify data that is being modified by another transaction.
4) Durability: This property ensures that once a transaction occurs, its results are permanent and cannot be removed from the database. This means that whatever happens after the COMMIT of a transaction, whether it is a system crash or aborts of other transactions, the results already committed are not modified or undone [3].

Concurrency Control: In distributed database systems, database is stereotypically used by many users. These systems generally allow multiple transactions to run concurrently i.e. at the same time. Concurrency control is the activity of coordinating concurrent accesses to a database in a multiuser database management system (DBMS). Concurrency control allows users to access a database in a multi-programmed method while maintaining the illusion that each user is executing alone on a dedicated system. The main technical difficulty in attaining this goal is to prevent database updates performed by one user from interfering with database retrievals and updates performed by another. When the transactions are updating data concurrently, it may lead to several problems with the consistency of the data [1].

IV. CONCLUSION

In this paper we have discuss about the distributed database system that is considered to be more reliable than centralized database system. It is really important for database to have the ACID properties to perform.

REFERENCES

[1] Arun Kumar Yadav & Ajay Agarwal, An Approach for Concurrency Control in Distributed Database System, Vol. 1, No. 1, pp. 137-141, January-June (2010)
[2] Navathe Elmasri, Database Concepts, Pearson Education, V edition (2008)
[3] Fundamentals of DBMS, Lakhanpal Publisher, III edition (2008)
[4] Swati Gupta, Kuntal Saroha, Bhawna, Fundamental Research of Distributed Database, IJCSMS International Journal of Computer Science and Management Studies, Vol. 11, Issue 02, Aug 2011

