Fundamentals of Distributed Database System

[1]Sagar Bajpayee, [2]Swapnil Shukla

Computer Science & Engineering,
Dronacharya College of Engineering, Greater Noida, India

***Abstract*— A Distributed Database System is a system in which portions of the database are stored on multiple computers within a network. The purpose of this paper to present an introduction to Distributed Database System. We discuss distributed database design, types of distributed database, fundamentals of transaction management and concurrency control. This paper highlights the basic concept of Distributed database system.**

***Index Terms*— Distributed Database, Fragmentation, Replication, Concurrency control, Transaction management.**

I. INTRODUCTION

A distributed database is a single logical database that is spread physically across computers in multiple locations that are connected by data communication links. Distributed database is a kind of virtual database whose component parts are physically stored in a number of distinct real databases at a number of distinct locations. The users at any location can access data at anywhere in the network as if the data were all stored at the user's own location. A distributed database management system is the software that manages the Distributed Databases, and provides an mechanism to access it that makes this distribution transparent to the user. The objective of a distributed database management system (DDBMS) is to control the management of a distributed database (DDB) in such a way that it appears to the user as a centralized database. The idea is that instead of having one, centralized database, we are going to spread the data out among the cities on the distributed network, each of which has its own computer and data storage facilities. All of this distributed data is still considered to be a single logical database. When a person or process anywhere on the distributed network queries the database, it is not necessary to know where on the network the data being sought is located. The user just issues the query, and the result is returned.

II. TYPE OF DISTRIBUTED DATABASE

Homogeneous distributed database

i. All sites have identical software
ii. Are aware of each other and agree to cooperate in processing user requests
iii. Fairly easy to design and manage.

Heterogeneous distributed database

i. Different sites may use different schemas and products
ii. The sites may not be aware of one another.
iii. They may provide only limited facilities for cooperation in transaction processing.

III. DISTRIBUTED DATA STORAGE

Distributed Database Systems are needed for the applications where data and its access are imherently distributed and to increase the availability during failures. The ffine examples are nternational air-line reservations, financial institutions and automated manufacturing. The methodology for designing Distributed Systems is same as that used for centralized darabase. However some additional factors have been considered for a distributed database.

Data Replication

If relation r is replicated, a copy of relation r is stored in two or more sites. In the most extreme case, we have full replication, in which a copy is stored in every site in the system. There are a number of advantages and disadvantages to replication.

i. **Availability--** If one of the sites containing relation r fails, then the relation r can be found in another site.Thus, the system can continue to process queries involving r, despite the failure of one site.
ii. **Increased parallelism**--In the case where the majority of accesses to the relation r result in only the reading of the relation, then several sites can process queries involving r in parallel. The more replicas of r there are, the greater the chance that the needed data will be found in the site where the transaction is executing. Hence, data replication minimizes movement of data between sites.
iii. **Increased overhead on update**--The system must ensure that all replicas of a relation are consistent; otherwise, erroneous computations may result.Thus, whenever r is updated, the update must be propagated to all sites containing replicas. The result is increased overhead. For example, in a banking system, where account information is replicated in various sites, it is necessary to ensure that the balance in a particular account agrees in all sites.

Data Fragmentation

In Distributed Databases, we need to define the logical unit of Database Distribution and allocation. 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. Three Types of Data Fragmentation are:

i. **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.

ii. **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 constructed if needed.

iii. **Hybrid fragmentation**-- Hybrid Fragmentation comprises the combination of characteristics of both Horizontal and Vertical Fragmentation. Each fragment can be specified by a SELECT-PROJECT combination of operations. In this case the original table can be reconstructed be applying union and natural join operations in the appropriate order

Transparency

The user of a distributed database system should not be required to know either where the data are physically located or how the data can be accessed at the specific local site. This characteristic, called data transparency, can take several forms:

i. **Fragmentation transparency**-Users are not required to know how a relation has been fragmented.

ii. **Replication transparency**. Users view each data object as logically unique. The distributed system may replicate an object to increase either system performance or data availability. Users do not have to be concerned with what data objects have been replicated, or where replicas have been placed.

iii. **Location transparency**. Users are not required to know the physical location of the data. The distributed database system should be able to find any data as long as the data identifier is supplied by the user transaction.

IV. FUNDAMENTALS OF TRANSACTION MANAGEMENT

A transaction is a unit of program execution that accesses and possibly updates various data items.

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.

i. **Atomicity** –Either all operations of the transaction are reflected properly in the database, or none are.

ii. **Consistency**-Execution of a transaction in isolation (that is, with no other transaction executing concurrently) preserves the consistency of the database.

iii. **Isolation**-Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions Ti and Tj, it appears to Ti that either Tj finished execution before Ti started or Tj started execution after Ti finished. Thus, each transaction is unaware of other transactions executing concurrently in the system.

iv. **Durability**-After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

V. CONCLUSION

In this paper we have discuss about the distributed database system that is considered to be more reliable than centralized database system. We describe the distributed database design, types of distributed database and fundamentals of transaction management.

REFERENCES

[1] Database System Concepts [Hardcover] Abraham Silberschatz (Author), Henry Korth (Author), S. Sudarshan (Author) McGraw-Hill Companies 2006 .

[2] Gupta V.K., Sheetlani Jitendra, Gupta Dhiraj and Shukla Brahma Datta, Concurrency control and Security issues in Distributed database system, Vol. 1(2),70-73, August (2012).

[3] Mandeep Kaur and Harpreet Kaur,Concurrency Control on Distributed Database System,Volume 3, Issue 7, July 2013.

[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.

[5] Distributed Database: http://wps.pearsoned.co.uk/wps/media/objects/10977/11240737/Web%2