An Impression of Transparency in Distributed Database Management System: A Review

[1]P.Banupriya and [2]M.Natarajan,
[1]M.Phil Research Scholar and [2]Assistant Professor
Department of Computer Science, Thanthai Hans Roever College, Perambalur, India

Abstract: The declining cost of computer hardware and the increasing data processing needs of geographically dispersed organizations have led to substantial interest in distributed data management. 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 by maintaining and utilizing large collections of data. A Distributed System is the one in which hardware and software components at networked computers communicate and coordinate their activity only by passing messages. This paper presents an overview of the transparencies of Distributed databases system.

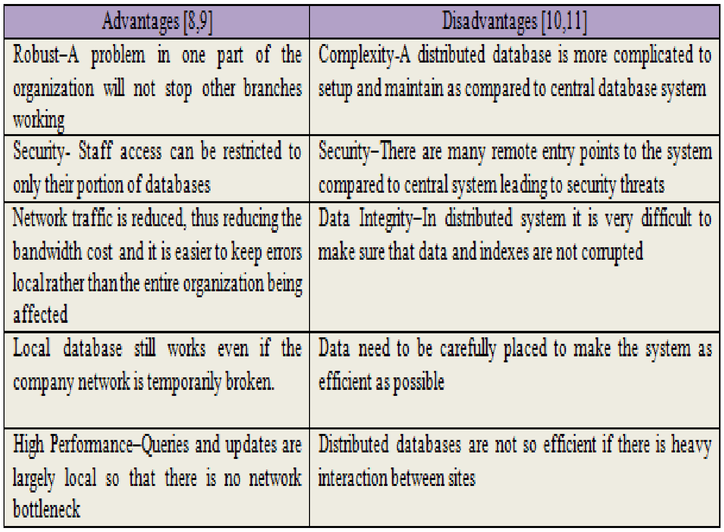
Keywords: *Distributed Database Management System, Access Transparency, Location Transparency, Concurrency Transparency, Replication Transparency, Failure Transparency.*

I. INTRODUCTION

A distributed database is a collection of multiple, logically interrelated databases distributed over a computer network [1]. 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 [2]. 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 [3]. Large companies need to distribute their data for many reasons e.g. for being economic and competitive [4]. 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. [5]

Table 1: Advantages and Disadvantages of Distributed Databases



Advantages [8,9]	Disadvantages [10,11]
Robust-A problem in one part of the organization will not stop other branches working	Complexity-A distributed database is more complicated to setup and maintain as compared to central database system
Security- Staff access can be restricted to only their portion of databases	Security–There are many remote entry points to the system compared to central system leading to security threats
Network traffic is reduced, thus reducing the bandwidth cost and it is easier to keep errors local rather than the entire organization being affected	Data Integrity–In distributed system it is very difficult to make sure that data and indexes are not corrupted
Local database still works even if the company network is temporarily broken.	Data need to be carefully placed to make the system as efficient as possible
High Performance–Queries and updates are largely local so that there is no network bottleneck	Distributed databases are not so efficient if there is heavy interaction between sites

A Database is systematically organized or structured repository of indexed information that allows easy retrieval, updating, analysis, and output of data. Each database may involve different database management systems and different architectures that distribute the execution of transactions [6]. A distributed database is a database in which storage devices are not all attached to a common processing unit such as the CPU. It may be stored in multiple computers, located in the same physical location; or may be dispersed over a network of interconnected computers. A distributed database system consists of loosely-coupled sites that share no physical components [7].

II. TYPES OF DISTRIBUTED DATABASE SYSTEMS

Distributed Database Systems are broadly classified into two types [9]:

A. Homogeneous Distributed System

In Homogenous distributed database system, the data is distributed but all servers run the same Database Management System (DBMS) software.

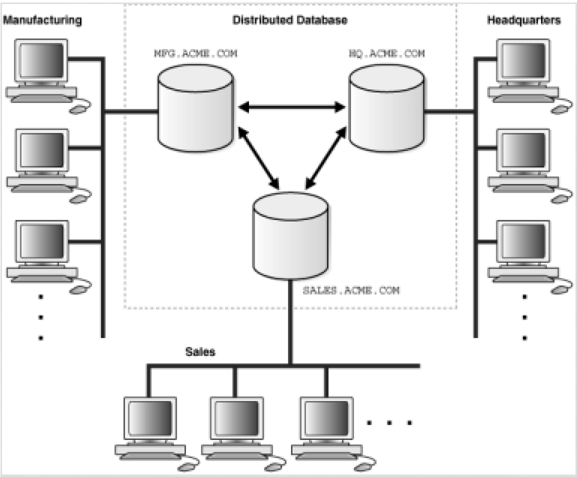


Figure 1: Homogeneous DBs Can Communicate Directly With Each Other

B. 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.

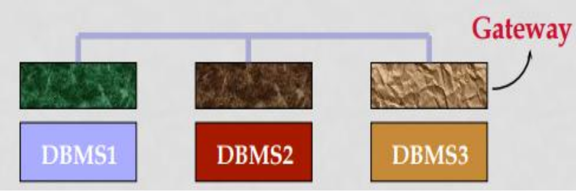


Figure 2: Heterogeneous DBs Communicate Through Gateway Interfaces

III. FEATURES OF DISTRIBUTED DATABASE SYSTEMS

As we have seen distributed system is a collection of autonomous systems, which are connected together by means of a local area network. Even though distributed systems are found in many applications designing them is a difficult task, as many issues have to be considered during its implementation. Any ideal distributed system should have all the features discussed below. But it may not be possible to incorporate all of them, so depending on the application requirements the features required are considered [13]. The following issues have to be taken care while designing,

- Heterogeneity
- Openness
- Security
- Scalability
- Fault Tolerance
- Concurrency
- Transparency

IV. TRANSPARENCY

The distributed systems should be perceived as a single entity by the users or the application programmers rather than as a collection of autonomous systems, which are cooperating. The users should be unaware of where the services are located and also the transferring from a local machine to a remote one should also be transparent.

One of the major objectives of Distributed database system is providing the appearance of centralized system to end user. The eight transparencies are believed to incorporate the desired functions of a distributed database system [12]. Such an image is accomplished by using the following transparencies:

- Access Transparency
- Location Transparency
- Concurrency Transparency
- Replication Transparency
- Failure Transparency
- Migration Transparency
- Performance Transparency
- Scaling Transparency

Other objective of distributed database is free object naming. Free object naming is basically allowing different users to access the same object with different names, or different objects with the same internal name. This will provide complete freedom to name the objects while sharing data without naming conflicts. Another objective of distributed system is Concurrency control. Concurrency control is the activity of coordinating concurrent accesses to a database in a multi-user database management system (DBMS). The following figure shows the sample transparency custom,

Transparency ask query:

```
SELECT ENAME,SAL FROM EMP,ASG,PAY
        WHERE DUR > 12
        AND EMP.ENO = ASG.ENO
        AND PAY.TITLE = EMP.TITLE
```

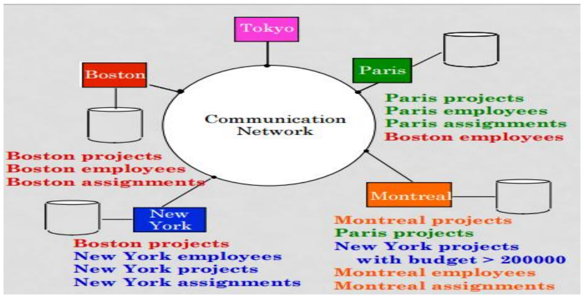


Figure 3: Example of Transparency

A. Access Transparency

Clients should be unaware of the distribution of the files. The files could be present on a totally different set of servers which are physically distant apart and a single set of operations should be provided to access these remote as well as the local files. Applications written for the local file should be able to be executed even for the remote files. The examples illustrating this property are the File system in Network File System (NFS), SQL queries, and Navigation of the web.

B. Location Transparency

Clients should see a uniform file name space. Files or groups of files may be relocated without changing their pathnames. A location transparent name contains no information about the named object's physical location. This property is important to support the movement of the resources and the availability of services. The location and access transparencies together are sometimes referred as Network transparency. The examples are File system in NFS and the pages of the web.

C. Concurrency Transparency

Users and Applications should be able to access shared data or objects without interference between each other. This requires very complex mechanisms in a distributed system, since there exists true concurrency rather than the simulated concurrency of a central system. The shared objects are accessed simultaneously. The concurrency control and its implementation is a hard task. The examples are NFS, Automatic Teller machine (ATM) network.

D. Replication Transparency

This kind of transparency should be mainly incorporated for the distributed file systems, which replicate the data at two or more sites for more reliability. The client generally should not be aware that a replicated copy of the data exists. The clients should also expect operations to return only one set of values. The examples are Distributed DBMS and Mirroring of Web pages.

E. Failure Transparency

Enables the concealment of faults, allowing user and application programs to complete their tasks despite the failure of hardware or software components. Fault tolerance is provided by the mechanisms that relate to access transparency. The distributed systems are more prone to failures as any of the components may fail which may lead to degraded service or the total absence of that service. As the intricacies are hidden the distinction between a failed and a slow running process is difficult. Examples are Database Management Systems [14].

F. Migration Transparency

This transparency allows the user to be unaware of the movement of information or processes within a system without affecting the operations of the users and the applications that are running. This mechanism allows for the load balancing of any particular client, which might be overloaded. The systems that implement this transparency are NFS and Web pages.

G. Performance Transparency

Allows the system to be reconfigured to improve the performance as the load varies.

G. Scaling Transparency

A system should be able to grow without affecting application algorithms. Graceful growth and evolution is an important requirement for most enterprises. A system should also be capable of scaling down to small environments where required, and be space and/or time efficient as required. The best-distributed system example implementing this transparency is the World Wide Web.

CONCLUSION

Distributed system provides a transparent access to all its resources does not exist but the sub systems built on the distributed architectures do provide transparencies for particular resources like files, disks and memory. Thus a lot of research is being done in the field of distributed systems, which have good scope in the future and also the need for these kind systems would be high. In this paper, the different transparencies of distributed systems and how they are taken care of are discussed.

References

1. Özsu, M.T. And Valduriez, P. 1999. Principles of distributed database systems. Prentice Hall, Upper Saddle River, N.J.

2. Silberschatz , A., Stonebraker , M. And Ullman , J. 1996. Database Research : Achievements and Opportunities into the 21st century. In SIGMOND Record, 26-27.

3. Shanker , U., Misra , M. And K. Sarje , A. 2008. Distributed real time database systems: background and literature review. Distributed and Parallel Databases 23, 127-149.

4. Bernstein, P.A., Hadzilacos, V. And Goodman, N. 1987. Concurrency control and recovery in database systems. Addison-Wesley, Reading, Mas

5. Köse , I. Spring , 2002. Distributed Database Security.
6. www.businessdictionary.com/definition/database.html

7. en.wikipedia.org/wiki/Database

8. www.webopedia.com/TERM/D/distributed_database.htm

9. wps.pearsoned.co.uk/wps/media/objects/.../Chapter%2012_WEB.pdf

10. https://www.dlsweb.rmit.edu.au/toolbox/Database/.../dist_sys_def.htm

11. www.csc.liv.ac.uk/~dirk/Comp332/COMP332-DDB-notes.pdf

12. www.teach-ict.com/as_a2_ict.../A2.../distributed_database/

13. George Coulouris, Jean Dollimore, Tim Kindberg, “Distributed Systems Concepts and Design” 3rd edition, Addison-Wesley.

14. http://www.cs.ucl.ac.uk/staff/J.Crowcroft/ods/node18.html - SECTION0053000000000 0000000