On Distributed Database Security Aspects

Zakaria Suliman Zubi
Computer Science Department,
Faculty of Science,
Al-Tahadi University,
P.O Box 727
Sirt, Libya
{zszubi@yahoo.com}

Abstract-- This paper will examine the underlying features of the distributed database architecture .Learning the task of distributed database management system will lead us to a successful design. The design will improve scalability, accessibility and flexibility while accessing various types of data. Developing a successful distributed database system requires to address the importance of security issues that may arise and possibly compromise the access control and the integrity of the system. We propose some solutions for some security aspects such as multilevel access control, confidentiality, reliability, integrity and recovery that pertain to a distributed database system.

Keywords-- Distributed database security, distributed database, distributed database management system, distributed database retrieval problems, discretionary security distributed database, query processing.

1. INTRODUCTION

"A distributed [1] database is a collection of databases which are distributed and then stored on multiple computers within a network". A *distributed database* is also a set of databases stored on multiple computers that typically appears to applications as a single database. "Consequently [14], an application can simultaneously access and modify the data in several databases in a network ". A database [2], link connection allows local users to access data on a remote database ". For this connection to occur, each database in the distributed system must have a unique global database name in the network domain. The global database name uniquely identifies a database server in a distributed system. Which mean users have access to the database at their location so that they can access the data relevant to their tasks without interfering with the work of others?

A distributed database management system (DDBMS) provides a central database resident on a server that contains database objects. A centralized DDBMS manages the database as if it were all stored on the same computer. The DDBMS also synchronizes all the data periodically and, in cases where multiple users must access the same data and it ensures that updates and deletes performed on the data at one location will be automatically reflected in the data stored elsewhere.

In a distributed database system, the database is stored on several computers. The computers in a distributed system communicate with one another through various communication media, such as high-speed networks or telephone lines. They do not share main memory or disks. "The computers [14] in a distributed system may vary in size & function, ranging from workstations up to mainframe systems ".

The main difference between centralized & distributed databases is that the distributed databases are typically geographically separated, are separately administered, & have slower interconnection. Also in distributed databases we differentiate between local & global transactions. A local transaction is one that accesses data only from sites where the transaction originated. A global transaction, on the other hand, is one that either accesses data in a site different from the one at which the transaction was initiated, or accessed data in several different sites.

In this paper, we will review all the security features of databases in a general form and distributed databases in particular. We will also investigate the security problems found in both models. Moreover, we will evaluate the security problems unique to each system. Finally, comparing the relative merits of each model with respect to security will be applied as well.

2. DATABASE SYSTEM CONCEPTS

One of the technology terms that most people have become accustomed to hearing either at work or while surfing the internet is the database. The database used to be an extremely technical term, however with the rise of computer systems and information technology throughout our culture, the database has become a household term.

On a more personal level, your personal computer can have its own database management system. You might have spreadsheets that contain mountains of data. Any time you fill up a spreadsheet with data and run queries to find and analyze data in different ways, you are accessing a database management system. The question is how do you view the data that is the result of a query? The answer is by looking at a report. Most database management systems have a reporting function that is the last step in the data manipulation process. After all, collating the data without looking at it won't get you very far.

One of the main functions of the database management system is doing the heavy lifting for you. In other words, you don't necessarily have to know exactly where all that data is in the system; as long as the database management system knows

978-1-4244-3757-3/09/$25.00 ©2009 IEEE

where it all is, it can deliver a report for you to peruse. This might not seem to matter if you're thinking of just your computer; but throw in a mainframe that contains reams and reams of data, and we're talking about a huge amount of information that can be stored any number of places within the mainframe system. The result is the same, though: a report that you can read, analyze, and act on.

This functionality also extends to a multi-user database. Such a database management system under this scenario would allow you as one user to operate all functions within the database without having to know what other users are accessing the same database? The user interacts with the database management system in order to utilize the database and transform data into information. Furthermore, a" database [2], offers many advantages compared to a simple file system with regard to speed, accuracy, and accessibility such as: shared access, minimal redundancy, data consistency, data integrity, and controlled access ". All of these aspects are enforced by a database management system. Among these things let's review some of the many different types of databases.

The vertical columns are known as the attributes. "Data that is stored on two or more tables establishes a "link" between [3], the tables based on one or more field values common in both tables." A relational database uses a standard user and application program interface called Structure Query Language (SQL). This program language uses statements to access and retrieve queries from the database. Relational databases are the most commonly used due to the reasonable ease of creating and accessing information as well as extending new data categories.

When dealing with intricate data or complex relationships, "object databases [13], are more commonly used "."Object databases in [13], contrast to relational databases store objects rather than data such as integers, strings, or real numbers". Each object consists of attributes, which define the characteristics of an object. Objects also contain methods that define the behavior of an object (also known as procedures and functions). When storing data in an object database there are two main types of methods, one technique labels each object with a unique ID. Every unique ID is defined in a subclass of its own base class, where inheritance is used to determine attributes. A second method is utilizing virtual memory mapping for object storage and management. Advantages of object databases with regard to relational databases allow more concurrency control, a decrease in paging, and easy navigation. However, there are "some disadvantages [9],of object databases compared to relational databases such as: less effective with simple data and relationships, slow access speed, and the fact that relational databases provide suitable standards oppose to those for object database systems".

On the other hand, network databases alleviate some of the problem incorporated with hierarchical databases such as data redundancy. "The network model [7, 10, 14], represents the data in the form of a network of records and sets which are related to each other, forming a network of links."

3. DISTRIBUTED DATABASES DESIGN

"The developments [14] ,in computer networking technology and database systems technology resulted in the development of distributed databases in the mid 1970s". It was felt that many applications would be distributed in the future and therefore the databases had to be distributed also. Although man y definitions of a distributed database system have been given, there is no standard definition. A distributed database system includes a distributed database management system (DDBMS), a distributed database and a network for interconnection. The DDBMS manages the distributed database. A distributed database is data that is distributed across multiple databases.

In brief term "distributed database [5] is a collection of databases that can be stored at different computer network sites". Each database may involve "different [6],database management systems and different architectures that distribute the execution of transactions". "The objective [10], 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".

A development of a distributed database structure for centered databases offers scalability and flexibility, allowing participating centers to maintain ownership of their own data, without introducing duplication and data integrity issues.

Moreover, Distributed database system functions include distributed query management, distributed transaction processing, distributed metadata management and enforcing security and integrity across the multiple nodes.

"The centralized [7], database system is one of the many objectives of a distributed database system". This system will be accomplished by using the following transparencies: Location Transparency, Performance Transparency, Copy Transparency, Transaction Transparency, Transaction Transparency, Fragment Transparency, Schema Change Transparency, and Local DBMS Transparency. These eight transparencies are believed to incorporate the desired functions of a distributed database system.

The design of responsive distributed database systems is a key concern for information systems. In high bandwidth networks, latency and local processing are the most significant factors in query and update response time. Parallel processing can be used to minimize their effects, particularly if it is considered at design time. It is the

978-1-4244-3757-3/09/$25.00 ©2009 IEEE

judicious replication and placement of data within a network that enable parallelism to be effectively used. Distributed database design can thus be seen as an optimization problem requiring solutions to various interrelated problems: data fragmentation, data allocation, and local optimization.

Meanwhile, a successful distributed database could include free object naming. “Free object naming means that it allows different users the ability to access the same object with different names, or different objects with the same internal name.” Thus, giving the user complete freedom in naming the objects while sharing data without naming conflicts.

Concurrency control (CC) is another issue among database systems. It permits users to access a distributed database in a multi-programmed fashion while preserving the illusion that each user is executing alone on a dedicated system. For this, CC mechanisms are required that interleave the execution of a set of transactions under certain consistency constrains while maximizing concurrency. Two main categories of CC mechanisms are: Optimistic concurrency - Delay the synchronization for transactions until the operations are actually performed. Conflicts are less likely but won't be known until they happen, making rollback operations more expensive. Pessimistic - The potentially concurrent executions of transactions are synchronized early in their execution life cycle. Blocking is thus more likely but will be known earlier avoiding costly rollbacks.

Another activity of CC is to "coordinating [8] ,concurrent accesses to a database in a multi-user database management system (DBMS).” There are a number of methods that provide "concurrency control [7], such as: Two phase locking, Time stamping, Multi-version timestamp, and optimistic non-locking mechanisms. Some methods provide better concurrency control than others depending on the system".

4. THE FEATURES OF DISTRIBUTED DATABASE SYSTEM

In this section we will examine the most common features of the distributed database system. One of the main features in a DDBMS is the Database Manager. “A Database Manager [1], is software responsible for processing a segment of the distributed database". A Distributed Database Management System is defined as the software which governs a Distributed Database System. It supplies the user with the illusion of using a centralized database.

Another main component is the User Request Interface, known some times as a customer user interface, which is usually a client program that acts as an interface to the distributed Transaction Manager. A customizable user interface is provided for entering requested parameters related to a database query. The customized parameter user interface provides parameter entry dialogs/windows in correlation to a data view (e.g., form or report) that is produced according to a database query. The parameters entered may provide for modification of the data view. also, the manager of the database may structure data views of a database to automatically include prompts for parameters before results are returned by the database. "These prompts [12] , may be customized by the manager and may be provided according to dialogs such as pop-ups, pull-down menus, fly-outs, or a variety of other user interface components".

“A Distributed Transaction [10] , Manager is a program that translates requests from the user and converts them into actionable requests for the database managers, which are typically distributed. A distributed database system is made of both the distributed transaction manager and the database manager". The components of a DDBMS are shown in the "fig. 1" below:

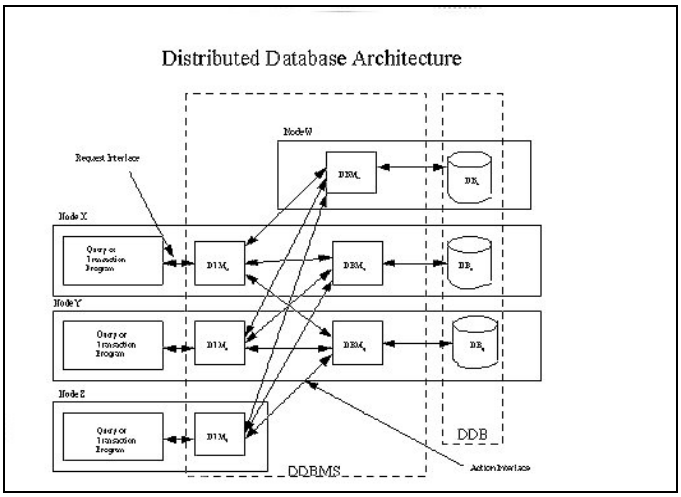

Figure 1. Shows the architecture of DDBMS

5. DISTRIBUTED QUERY PROCESSING

Distributed query processing in a large scale distributed system; it is often difficult to find an optimal plan for a distributed query: distributed systems can become very large, involving thousands of heterogeneous sites. As new databases are added/ removed to/from the system, it becomes harder for a query processor to maintain accurate statistics of the participating relations stored at the different sites and of the selectivity's of the related query operations. Also, as the workload at the various interacting processing servers and the transmission speeds of the links between them fluctuate at runtime, there is the need of distributed query engines that dynamically adapt to large-

978-1-4244-3757-3/09/$25.00 ©2009 IEEE

scale distributed environments. The query will be request usually from the user or the client host from a proper user interface.

The user request interface is merely a client program running on one end system that requests and receives a service from a server program running on another end system. Due to the fact that the client and the server run on separate computers, by definition the client/server programs are considered distributed applications. There are two types of client/server applications. One particular client/server application is an implementation of a protocol standard defined in an RFC (request for comments).

This type of application forces the client and server programs to abide by certain rules dictated by the RFC. Another sort of client/server application is a proprietary client/server application. “In this case [7], the client and server programs do not necessarily conform to any existing RFC”.

In developing a proprietary client/server application the developer must decide whether to run the application over TCP or UDP. TCP connection oriented and ensures reliable byte-stream channel. However "UDP is [4] , connectionless and forwards independent packets of data and does not guarantee delivery".

6. SECURITY ASPECTS IN DISTRIBUTED DATABASE

A proximately all of the early work on secure databases was on discretionary security. But the most important issues in security are authentication, identification and enforcing appropriate access controls. For example, the mechanisms for identifying and authenticating the user, or if a simple password mechanisms suffice? With respect to access control rules, "languages [7], such as SQL have incorporated GRANT and REVOKE statements to grant and revoke access to users". For many applications, simple GRANT and REVOKE statements are not sufficient. There may be more complex authorizations based on database content. Negative authorizations may also be needed. Access to data based on the roles of the user is also being investigated. "Numerous papers [7], have been published on discretionary security in databases ". "Fig. 2", shows a simple security model.

DBMS have many of the same security requirements as operating systems, but there are significant differences since the former are particularly susceptible to the threat of improper disclosure, modification of information and also denial of service. Some of "the most [4], important security requirements for database management systems are: Multi-Level Access Control: Confidentiality. Reliability, Integrity, Recovery".

"Security in distributed [11], database systems has focused on multilevel security". Specifically approaches based on distributed data and centralized control architectures were proposed. Prototypes based on these approaches were also developed during the late 1980s and early 1990s. Notable among these approaches are the efforts by Unisys Corporation and the Naval Research Laboratory.

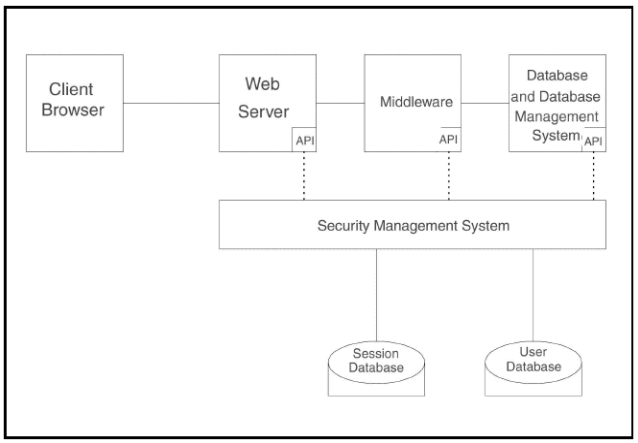

Figure 2: A Simple Security Model

7. EMERGING SECURITY USED IN DISTRIBUTED SYSTEM TOOLS

Security impact in most of the distributed database tools became an emerging technology that has evolved in some way from distributed databases and discussions. These include data warehouses and data mining systems, collaborative computing systems, distributed object systems and the web. First, let us consider data warehousing systems. The major issues here are ensuring that security is maintained in building a data warehouse from the backend database systems and also enforcing appropriate access control techniques when retrieving the data from the warehouse. For example, security policies of the different data sources that form the warehouse have to be integrated to form a policy for the warehouse. This is not a straightforward task, as one has to maintain security rules during the transformations. For example, one cannot give access to an entity in the warehouse, while the same person cannot have access to that entity in the data source. Next, the Warehouse security policy has to be enforced. In addition, the warehouse has to be audited.

Finally, the retrieval problem also becomes an issue here. For example, the warehouse may store average salaries. A user can access average salaries and then deduce the individual salaries in the data sources, which may be sensitive and therefore, the inference problem could become an issue for the warehouse. To date, little work

978-1-4244-3757-3/09/$25.00 ©2009 IEEE

has been reported on security for data warehouses as well as the retrieval problem for the warehouse. This is an area that needs much research intention.

Data mining causes serious security problems. For example, consider a user who has the ability to apply data mining tools. This user can pose various queries and infer a sensitive hypothesis. That is, the retrieval problem occurs via data mining. There are various ways to handle this problem. Given a database and a particular data-mining tool, one can apply the tool to see if sensitive information can be deduced from legitimately obtained unclassified information. If so, then there is a retrieve problem. There are some issues with this approach. One is that we are applying only one tool. In reality, the user may have several tools available to him or to her. Furthermore, it is impossible to cover all of the ways that the retrieval problem could occur. Another solution to the retrieval problem is to build a retrieval controller that can detect the motives of the user and prevent the retrieval problem from occurring. Such a retrieval controller lies between the data-mining tool and the data source or database, possibly managed by a DBMS.

Data mining systems are being extended to function in a distributed environment. These systems are called distributed data mining systems. Security problems may be exacerbated in distributed data mining systems. This area has received very little attention. Other emerging technologies that have evolved in some way from distributed databases are collaborative computing systems, distributed object management systems and the web. Much of the work on securing distributed databases can be applied to securing collaborative computing systems. With respect to distributed object systems security, there is a lot of work by the Object Management Group's Security Special Interest Group.

More recently, there has been much work on securing the web as well. The main issue here is ensuring that the databases, the operating systems, the applications, the web servers, the clients and the network are not only secure, but are also securely integrated.

8. CONCLUSION

In this paper we introduce distributed database and all common aspects related to distributed database such as database system concepts, features and design of distributed database and distributed query processing as well.

We also indicate some related security issues including multilevel security in distributed database systems. The security aspects lead us to investigate the distributed data and centralized control, distributed data and distributed control and some retrieval problems in distributed databases in since of accessing control and integrity. Moreover, we describe the most common mechanisms of discretionary security and stated the emerging security used in distributed system tools.

Finally we believe that as more and more distributed database tools, the impact of secure distributed database systems on these tools will be a significant requirement.

REFERENCES

[1] Bell, David and Jane Grisom, Distributed Database Systems. Workinham, England: Addison Wesley, 1992.

[2] Charles P. Pfleeger and Shari Lawrence Pfleeger, Security in Computing, Prentice Hall Professional Technical Reference, Upper Saddle River, New Jersey, 2003.

[3] Haigh, J. T. et al., "The LDV Secure Relational DBMS Model," In Database Security, IV: Status andProspects, S. Jajodia and C.E. Landwehr eds., pp. 265-269, North Holland: Elsevier, 1991.

[4] İlker Köse, GYTE, Veri ve Ağ Güvenliği, Distributed Database Security, Spring 2002.

[5] James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, Pearson Education, Inc, New York, 2003.

[6] Paul Lothian and Peter Wenham, Database Security in Web Environment, 2001.

[7] Pfleeger, Charles P., (1989) Security in Computing. New Jersey: Prentice Hall. 1989.

[8] Simon Wiseman, DERA, Database Security: Retrospective and Way Forward, 2001.

[9] Stefano Ceri, Giuseppe Pelagatti: Distributed Databases: Principles and Systems. McGraw-Hill Book Company 1984, ISBN 0-07-010829-3.

[10] Thuraisingham B., Security for Distributed Database Systems, Computers & Security, 2000.

[11] Thuraisingham, Bhavani and William Ford, "Security Constraint Processing In A Multilevel Secure Distributed Database Management System," IEEE Transactions on Knowledge and Data Engineering, v7 n2, pp. 274-293, April 1995.

[12] "Components of a Distributed Database System," http://www.fi/~hhyotyni/latex/Final/node44.html, October 24, 2008.

[13] "Object Oriented Databases," http://www.comptechdoc.org/independent/database/basicdb/dataobject.html,October 25, 2008.

[14] "Network Databases," http://wwwdb.web.cern.ch/wwwdb/aboutdbs/classification/network.html, October 25, 2008.

978-1-4244-3757-3/09/$25.00 ©2009 IEEE