Transactions- Distributed Database Systems: Issues and Challenges

Idowu, S. A1, Maitanmi S. O2
1Computer Science Department, Babcock University, Ilisan Remo, Ogun State, Nigeria
2Computer Science Department, Babcock University, 
1saidowu07@gmail.com, 2maitanmi@yahoo.com

Abstract
Implementation of distributed database were hindered with lots of challenges perused in past researches, few of such include unreliable network technology, high cost of Computers, and insecurity among users. However, as networks have become more reliable and the cost of computers has drastically dropped because of the multitudes involved in the manufacturing and supply of computers. Users are better informed on the benefits involved in the use of distributed database systems. Therefore, this research paper is tailored towards mentioning the challenges encountered in distributed database transaction and proffered likely solutions. Few of the mentioned challenges are; problems of distribution of resources, search and updating of resources.

Keywords: Distributed database, transaction, sharing, planning, and task sharing.

1.0 INTRODUCTION
Despite the amount of Computers available in the world today, most of these systems have suffered greatly on how to communicate by sharing resources. However, recent innovation in distributed databases (DDBs) made it possible for all corporate to be accessible through a single resources which is called distributed database or decentralized database.

Distributed database allow a decentralized scheme for data management where files are spread through a collection of autonomous nodes that communicate with one another through a common language.

Distributed database (DDB) is a collection of multiple, logically interrelated databases distributed over a computer network [1]. A distributed database management system (DDBMS) is the software that manages the DDB and consists of a collection of sites, each of which maintain a local database system and provides access mechanisms that make the distribution transparent to the user and that have their data distributed and replicated over several locations as shown in figure 1; unlike the centralized database system (CDBS), where one copy of the data is stored [5]. Data may be replicated over a network using horizontal and vertical fragmentation similar to projection and selection operations in Structured Query Language (SQL). Both types of databases share the same problems of access control and transaction management, such as user concurrent access control and deadlock detection and resolution. Access control and transaction management in DDBS require different rules to monitor data retrieval and update to distributed and replicated databases. In distributed database, users use two-phase and three-phase commit technique to maintain a consistent state for the databases. The objective of these paper present issues involved in distributed database systems and the proposed possible solutions.

Figure 1: Distributed Database Architecture. Source: [1]

1.1 TYPES 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. Each site surrenders part of its autonomy in terms of right to change schemas or software
iv. Appears to user as a single system

**Heterogeneous distributed database**
i. Different sites may use different schemas and software

- Difference in schema is a major problem for query processing
  1. Difference in software is a major problem for transaction processing
  2. Sites may not be aware of each other and may provide
  3. only limited facilities for cooperation in transaction processing.

The research of [2] unveiled the following characteristics:

**Characteristics of Distributed Database**
i. Collection of logically-related shared data.
ii. Data split into fragments.
iii. Fragments may be replicated.
iv. Fragments/replicas allocated to sites.
v. Sites linked by a communications network.
vi. Data at each site is under control of a DBMS.
vii. DBMSs handle local applications autonomously.

**Advantages of Decentralized Database**
i. The purpose of decentralize database is to make all the data that is available to a company or corporation conveniently available to individual users.
ii. The availability of this data would further facilitate the local management of daily tasks while providing a basis at the corporate level for planning future strategies.

**Operation of Distributed Database**
i. Hundreds of individual processors are decentralized around the world
ii. Only the data used on daily basis are kept on a local node.
iii. Other useful information in the database is accessible remotely.
iv. Naturally placing data next to its most frequent users spends response time in retrieving this data.
v. The autonomy of nodes in a distributed database allows each organisational entity to manage its information in its own way which in turns makes the database configuration modular and flexible.

### 2.0 PROBLEMS OF DISTRIBUTED DATABASE

#### 2.1 Distribution of resources problem

This has to do with equal distribution of resources across all servers. There are two approaches to solving this problem in a distributed database environment; decentralize by function, or decentralize by location [3].

(a) **Decentralize by function**: this works well with data that will be accessed repeatedly by the same users. Example includes putting manufacturing materials list at the appropriate manufacturing plants and customer information at sales locations.

(b) **Decentralize by location**: occurs when you partition customer information on a node per region or other geographical based entity within an organisation as seen in figure 2.

**Practical demonstration of decentralization**

![Diagram of decentralized communication](https://example.com/diagram.png)

Figure 2. Demonstration of decentralization of communication regionally.

Each telephone company independently implements the common protocols by the international phone network, e.g for dialing and billing as shown in figure 2, the dialing and billing of customer vary from one country to another. The only centralize function is the architecture of these protocols. It is worthy to note that design and architecture are typically centralize within each company; while operation and control are delegated to the operating countries which in turn delegate operation and maintenance to individuals responsible for maintaining their own hardware and software[4].

#### 2.2 SEARCHING
Another notable problem of distributed database results from lack of adequate knowledge of the entire database. For example, locating where file x, y, z was stored. The solution to the challenge is using the concepts of global, local, and semi global data.

i. Global data involves information that is common to all sites and shared by all sites.
ii. Local data is information that is strictly meant for an individual site although it is accessible to all sites.
iii. Semi global data is used in intermodal and intersite transactions.

Information is made available to the whole network by partitioning or replicating the data files. Partitioning involves splitting a data file into records and then distributing the records across networks while replication means duplicating records at more than one location/node. Partitioning only works with local data while global data are replicated.

Example
i. A typical example is found in bank transactions where servers are located in local vicinity where customers can transact which often increases the speed of transaction when traffics are eradicated.

Advantages of partitioning a network
i. It makes data available to all remote users
ii. It reduces traffics and increase speed of access

Advantages of replicating a network
i. Replication improves data availability
ii. Replication improves response time.

2.3 UPDATING

As discussed above, partitioning of data is most efficient when the data are kept current that is, updated regularly. The single copy of each data item makes updating an efficient process. However, nonlocal “read” operations are more expensive, making partitioning less efficient for data that is widely used but updated infrequently. Replicated data is most efficient when multiple reads of the data are expected, but updates are not as regular compared to partitioning process. The data is duplicated at nodes where high-volume reads are expected, producing high availability and good response time. When replicated data must be updated, however, an update to a record at one node should cause an identical update at all other nodes where that record resides. If any one replica is unavailable there could be problems [6]. A variety of schemes can be employed for updating Replicated data, even though the copy of the record may be temporarily unavailable at one or more of the nodes. One technique requires that a majority of the replicas be read and updated as part of each transaction, though the definition of “majority” varies with the application. This scheme has the advantage of tolerating some nodal unavailability, but it is not practical for either very small or very large networks. In a very small network of, say, two nodes, having either node unavailable prevents an update of a majority of the nodes. In larger networks, delays in completing the update transaction are proportional to network size: As the network grows, transactions will take longer to complete [7].

7.0 CONCLUSIONS

The paper has treated the proposed challenges of distributed database system. However, new challenges show up daily in transactions. Therefore there may be need for constant reviewers of these challenges.

REFERENCES


www.sciencepublication.org