International Journal of
Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2020.33814

Volume 6, Issue 5
May - 2020

Reliable Loan Information System Using Eager Replication

Mar Mar Soe[1] and Yin Yin Cho[2]

[1]Faculty of Computer Science, [2]Faculty of Information Science

University of Computer Studies (Loikaw), University of Computer Studies (Meiktila)

Loikaw, Kayah State, Meiktila, Mandalay Division

Myanmar

ABSTRACT

Replication is the process of sharing information so to ensure consistency between redundant resources, such as software or hardware components. One of the challenges in database replication is to introduce replication without severely affecting performance. Because of this difficulty, current database products use lazy replication, which is very efficient but can compromise consistency. As an alternative, eager replication guarantees consistency but most existing protocols have a prohibitive cost. In the distributed systems community, software-based replication is seen as a cost-effective way to increase availability. Depending on the application, data to be replicated may be stored in a file or a file collection, a database or a database table, or an object stored in a file or in a database. This system is applied eager replication method for loan databases in loan organizations. This system can support reliable replicated data for the loan system between replicated databases.

Key Words: *Data replication, Lazy and Eager replication, Primary Copy, Linear Interaction.*

1. INTRODUCTION

Replication is the process of copying and maintaining database objects, such as tables, in multiple databases that make up a distributed database system. Replication means that the same data is available at multiple locations. Data replication is a key factor to achieve scalability and fault tolerance in databases as it maintains several clones of data objects. Data replication is a process in which duplicated data objects are created and maintained in a distributed database systems scattered on different locations. This system can support reliable replicated data for loan system between replicated databases. Database replication can be used on many database management systems, usually with a master/slave relationship between the original and the copies. When several replicas of a database are present, they can divide the workload of incoming requests between each other. Replication uses distributed database technology to share data between multiple sites, but a replicated database and a distributed database are not the same [8]. Data is replicated for performance and availability. In a distributed database, data is available at many locations, but a particular table resides at only one location. Replication means that the same data is available at multiple locations.

Replication technique provide access to the local database objects instead of remote server which minimizes network traffic and make sure fast access to the data. In case of system failure, data replication is much more needed because it still works due to the availability of the secondary copy of data and user can still continue to query or update the remaining locations [7]. A replication service is needed in order to make sure data consistency across the disparate environments. This system can be studied eager replication methods and techniques, that provide high availability and low database latency. The objectives are to achieve fast data access and load distribution, to reduce cost and time consuming for manual recording information, to get computerized loan database system, and to know replication of databases is a good way to get a well working database system.

This paper is organized into five sections. Section 1 includes introduction and objectives of the system. Section 2 describes theoretical background that includes: Database Replication Technique, Replicated data, Replication Objects**,** Linear Interaction, Primary Copy and Update Everywhere. Section 3 discusses Eager Replication, Eager Update Everywhere. Section 4 present System design, implementation results and program code for loan system. Also this section includes the implementation process with figures. Finally, Section 5 presented the conclusions that include advantages and further extension item.

2. DATABASE REPLICATION TECHNIQUES

To get a reliable and fast system with distributed databases, a number of different techniques can be used. The main areas to take into account when choosing the technique is performance, consistency and redundancy. When handling with a replicated database system, a number of actions are done during an execution of a transaction [2]. Five phases where presented to describe the different steps in the system. The steps are:

- Request: The first step of the transaction, where the client sends its request to a node in the system
- Server Coordination: Synchronization between the different nodes to decide on the execution of the transaction
- Execution: The execution phase of the transaction
- Agreement Coordination: The phase where agreement is done between the nodes to ensure that all nodes have received the same results of the update
- Response: Responding on the outcome of the transaction to the client

Replication can be performed in either synchronous or asynchronous mode. Synchronous replication guarantees continuous data integrity during the replication process with no extra risk of data loss. However, the impact on the performance of the application can be significant and is highly dependent on the distance between the sites. Asynchronous replication overcomes the performance limitations of synchronous replication, but some data loss must be accepted [6]. Generally, replication methods can be broken down using three different components: server architecture, server interaction, and transaction termination. Transaction termination can be further broken down into voting and non-voting termination [5].

2.1 Replicated Data

Replicated data are becoming more and more of interest lately. The use of data replication has many advantages including the increased read availability (many operations can be handled locally, reducing communication costs and delays) and reliability (if one site is down, or has lost some of its data, the data is likely to be available at another node) but makes the data updating more complicated. While data copying can provide users with local and much quicker access to data, the problem is to provide these copies to users so that the overall systems operate with the same integrity and management capacity that is available within a centralized model [1, 4].

2.2 Linear Interaction

Linear interaction protocols handle each replicated operation on an individual basis, which means that communication overhead can reach unacceptable levels in an environment where hundreds of Update, Delete and Insert (UDI) transactions occur every second. Even when replicated operations are bundled in a single transaction, each operation results in communication between master and slave. In a situation where only a few nodes are replicated the communication overhead of linear interaction is negligible, but in a highly scaled environment replica lag results. Replica lag is the lag time between a delegate server and its slaves. High replica lag means low data availability which can be critical in financial institutions where data such as exchange rates are replicated [2, 5].

2.3 Non-voting termination

Transaction Termination of this type implies that sites can decide on their own whether to commit or abort a transaction. Nonvoting techniques require replicas to behave deterministically. This, however, is not as restrictive as it may appear at first glance since the determinism only affects transactions that are serialised with respect to each other. Transactions or operations that do not conflict can be executed in different orders at different sites [2, 5].

2.4 Replication Objects

A **replication object** is a database object existing on multiple servers in a distributed database system. In a replication environment, any updates made to a replication object at one site are applied to the copies at all other sites. Advanced replication enables you to replicate the following types of objects:

- Tables, Indexes
- Views and Object Views
- Packages and Package Bodies
- Procedures and Functions

DOI: 10.31695/IJASRE.2020.33814

- User-Defined Types and Type Bodies
- Triggers, Synonyms
- Index-types
- User-Defined Operators

Regarding tables, replication supports advanced features such as partitioned tables, index-organized tables, tables containing columns that are based on user-defined types, and object tables [2, 8].

3. EAGER REPLICATION METHOD

Eager replication keeps all replicas exactly synchronized at all nodes by updating all the replicas as part of one atomic transaction. Eager replication gives serializable execution, there are no concurrency anomalies. But, eager replication reduces update performance and increases transaction response times because extra updates and messages are added to the transaction. Eager replication is not an option for mobile applications where most nodes are normally disconnected. Mobile applications require lazy replication algorithms that asynchronously propagate replica updates to other nodes after the updating transaction commits [3]. Simple eager replication systems prohibit updates if any node is disconnected. For high availability, eager replication systems allow updates among members of the quorum or cluster. Even if all the nodes are connected all the time, updates may fail due to deadlocks. Eager systems have fewer-longer transactions. Eager replication can be separated into two sub categories. The first is primary copy that contains a primary database node with one or many backup nodes. The second technique is update everywhere where transactions can be done in all nodes in parallel.

3.1 Eager Primary Copy

In an eager Primary Copy system, no updates are done in the master system until all secondary databases also has received the update [1, 6]. Figure 1 shows an example of an eager Primary Copy system where all update transactions are sent directly to the primary database. The client sends its read operation to a secondary database. When an update operation is executed by the client, it is sent to the primary database. When the primary database receives an update transaction, the transaction is first performed, but not committed, in the primary database. The update is sent to all secondary databases that perform the update, either operation by operation with linear interaction or as one single message with constant interaction.

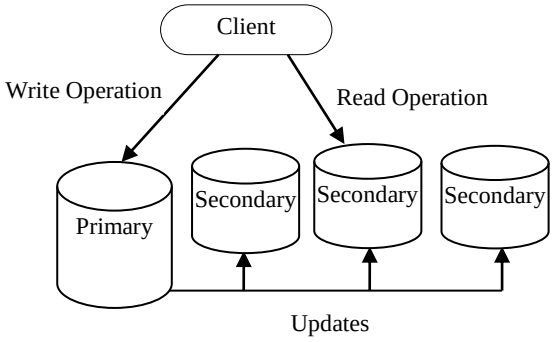

Figure 1. Eager Primary CopY

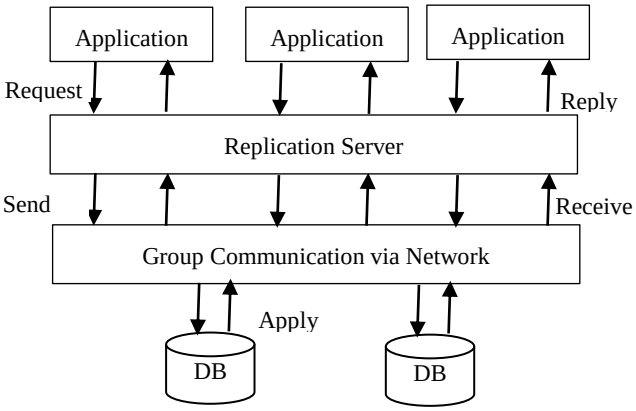

Figure 2. Eager Update Everywhere

3.2 Eager Update Everywhere

There are mainly two different approaches to achieve consistency with the update everywhere strategy. The first is to use locking to ensure that no simultaneous transaction deals with the same data. The second approach is to use an atomic broadcast in the system that decides the total order in which conflicting transactions should be dealt with [1, 6].

4. SYSTEM DESIGN AND IMPLEMENTATION

This system is developed by using ASP.net. It can duplicate the SQL database to the client. By using this system, the user can get the Loan from Loan Bank in efficient way. The design and implementation of the system as well as the structure of the program are described in this section. The system has the following components:

1. Login- The system allows the user to enter user name and password for insert, update, and retrieve data.
2. Data Entry- The system allows the user to insert data to get loan.

DOI: 10.31695/IJASRE.2020.33814

3. Update- The system allows the user to pay monthly interest and return loan.
4. Replicate- The system replicates the updated data to any site.
5. Retrieve – The system show the information about loan to the user.

Replication provides fast, local access to shared data because it balances activity over multiple sites. Some users can access one server while other users access different servers, thereby reducing the load at all servers. Also, users can access data from the replication site that has the lowest access cost, which is typically the site that is geographically closest to them.

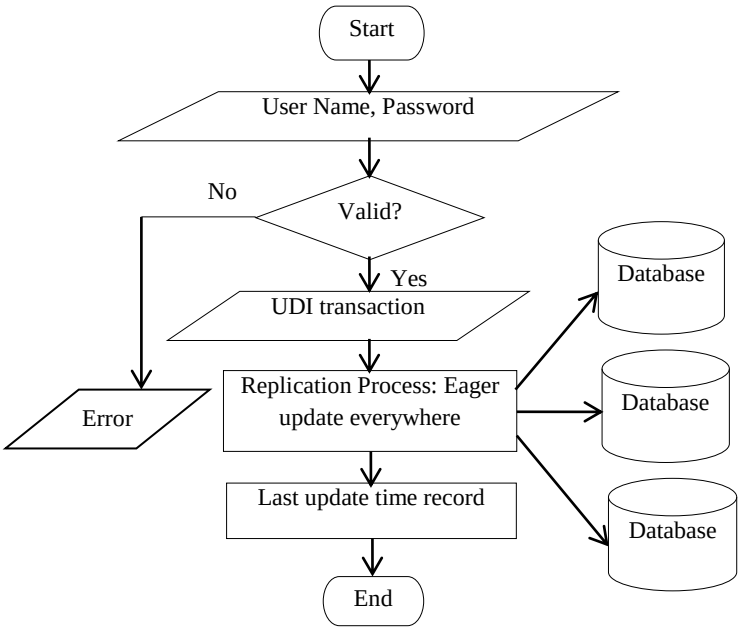

Figure 3. System flow diagram for Update Everywhere process

4.1 Data Table and Replication Code

This system contains 4 data tables. Borrower table, Recommender table, Loan table and Register table. Users who gain access to the Database Tables are stored in the Register table with their password. Before data entry to the database table, the system confirms the username and password.

```
Void RequestFromClient (String trans, String sqlStatement)
  { //(1) Check read or update transaction
     if (trans == read-transaction)  { (2) Answered directly from the secondary DB
            Query DB (sqlStatement); }
      else if (trans == update-transaction) {(3) Save the operation into Queue
              myqueue = new Queue();
              myqueue-Enqueue (sqlStatement);
          (4) Send to the primary DB (sqlStatement);
          (5) Copy of the primary DB is propagated to the secondary DB
            do replication ();        }
    else Message Box.show (“Error”);
  }
  Void do-replication ()
  {copyDB (PrimaryDB,SecondaryDB); }
```


DOI: 10.31695/IJASRE.2020.33814

4.2 Implementation Process

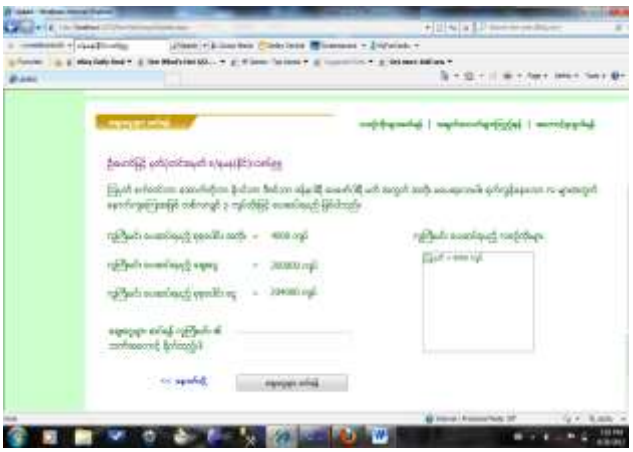

Figure 4. Updated data replication process

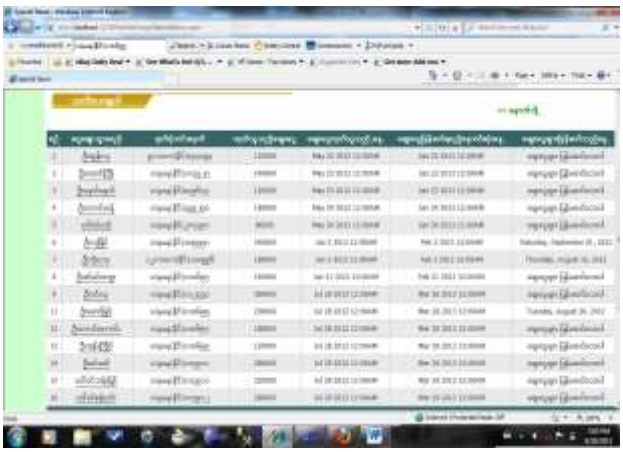

Figure 5. Monthly interest notice list

This system can give the data entry, update and retrieval with easy and less time. In loan system, there are 24 townships, and about 9 quarters in each township of Magwe Division. All of the user in this region can borrow the loan and automatically process with particular officer using this system. Each user can borrow fifty thousand Kyats per each with two precents interest per month. The total amount will return within one year. If the user wants to borrow the loan, the user entry the information as shown in following Figure to get the loan. If the officer wants to return total loan, officer will fill name, NRC and code for township. This system will display the total loan, and interest amount. The officer must type bank account and click interest button. Deleting process was done and it can be replication as shown in figure 4. This system can apply for monthly interest process in each quarter. This process is update data and this transaction is replicated to other database. If the user from particular's township and quarters wants to know the information, the total information will display. If the person did not give monthly loan, this system will list and calculate the extra return loan in figure 5.

5. CONCLUSION

Replication can be used to distribute data over multiple regional locations. Then, applications can access various regional servers instead of accessing one central server. This configuration can reduce network load dramatically. Data replication consists of maintaining multiple copies of data, called replicas, on separate computers. It is an important enabling technology for distributed services. Replication improves availability by allowing access to the data even when some of the replicas are unavailable. A good way to increase the performance in a database system is to use replication techniques so that the database system can be separated into different servers. When several replicas of a database are present, they can divide the workload of incoming requests between each other. This system can provide the computerized loan database that is replicated in different site. So particular quarter and township officers can access reliable data quickly and can support easy to use this system. Using web application with Myanmar language, users and officers can understand with easily. Synchronous replication does not provide protection against data corruption and loss of data due to human errors. Applications such as call centers and Internet systems require data on multiple servers to be synchronized in a continuous, nearly instantaneous manner to ensure that the service provided is available and equivalent at all times. Here, data consistency is more important than site autonomy. Using lazy replication, data consistency cannot support. If the data consistency can get for reliability, eager replication should use. This system can extend the performance comparison result of lazy and eager replication method.

REFERENCES

1. Breitbart, Y., Komondoor, R., Rastogi, R., Seshadri, S., and Silberschatz, A. 1999. Update propagation protocols for replicated databases. In Proceedings of the 1999 ACM SIGMOD international Conference on Management of Data (Philadelphia, Pennsylvania, United States, May 31 - June 03, 1999). SIGMOD '99.
2. Fredrik Nilsson, and Patrik Olsson, A survey on reliable communication and replication techniques for distributed databases, School of Innovation, Design and Engineering, School of Innovation, Design and Engineering, 2007.
3. Jim Gray, Pat Helland, Patrick O'Neil, Dennis Shasha, The Dangers of Replication and a Solution, ACM SIGMOD Conference at Montreal, pp. 173-182, 1996
4. M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso., Understanding replication in databases and distributed systems. In Proceedings of 20th International Conference on Distributed Computing Systems (ICDCS'2000), pages 264–274, Taipei, Taiwan, R.O.C., April 2000. IEEE Computer Society Los Alamitos California.

DOI: 10.31695/IJASRE.2020.33814

5. M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, and G. Alonso., Database replication techniques: a three parameter classification," Reliable Distributed Systems, 2000. SRDS-2000. Proceedings the 19th IEEE Symposium on, vol., no., pp. 206-215, 2000
6. Mohamed Osman Ali Hegazi, An Approach for Designing and Implementing Eager and lazy Data Replication, International Journal of Computer Applications (0975 – 8887) Volume 110 – No. 6, January 2015
7. Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in Proceedings of 16th ACM International Conference on Supercomputing (ICS'02), June 2002.
8. Sushant Goel, Rajkumar Buyya, Data Replication Strategies In Wide Area Distributed Systems, Grid Computing and Distributed Systems (GRIDS) Laboratory,Department of Computer Science and Software Engineering,The University of Melbourne, Australia.

DOI: 10.31695/IJASRE.2020.33814