# Pull-based Log-transfer Replication System

**Su Hlaing Phyo**
*University of Computer Studies, Yangon*
*suhlaingphyo59@gmail.com*

## Abstract

*Replication is important in distributed environment. Replication provides users with their own local copies of data. These local, updatable data copies can support increased localized processing, reduced network traffic and easy scalability. However, the major disadvantage of replication is that when a given replicated object is updated, all copies of that object must be updated: the update propagation problem. The pull-based log-transfer replication system allows replica contents to become stale but in a controlled way. This system will become more efficient and available than traditional replication systems that keep all the replicas consistent, especially when the network and computers are unreliable. Therefore, this system makes to reduce the update propagation in the pull-based single master optimistic replication system by using the log transfer approach.*

## 1. Introduction

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases, managed by a variety of different DBMS, running on a variety of different machines, supported by a variety of different operating systems and connected together by a variety of different communication network where the term transparently means that the application operates from a logical point of view as if the data were all managed by a single DBMS running on a single machine. A distributed database system consists of a collection of sites, connected together via communication network [1]. Replication is important in distributed environment. Replication provides users with their own local copies of data. These local updatable data copies can support increase localize processing, reduced network traffic, easy scalability and cheaper approaches for distributed, non-stop processing [2]. The objectives of the proposed system are to implement replication system by using pull-based log-transfer and single-master method and to reduce the network traffic. This system is also tends to solve the update propagation problem in replication system and to minimize the number of update exchanged between replicas. This paper is organized as follows: Section-2 presents the background theory, Section-3 presents the proposed system, Section-4 presents the system implementation, Section-5 presents the related work and Section-6 presents the conclusion.

## 2. Background Theory

### 2.1 Single-master and Multi-master System

Single-master systems designate one replica as the master that stores the authoritative copy of the object. All updates are accepted first on the master and are then propagated to other replicas, or slaves. Multi-master systems let any replica issue and update at any time, and they exchange and merge updates among replicas.

The main advantage of single master system is the simplicity. Because updates are accepted only at one place, single-master systems can detect and report update conflicts to users immediately, making the system less confusing to users. They are simpler algorithmically as well, because updates flow only one-way, from the master to the slaves.

The multi-master system are move available than single-master systems. However, the disadvantage of multi-master system is lost update problem: they may lose the effects of some updates, because update conflicts are detected after the updates are accepted by replicas and the users who issued them have long logged out from the system [6].

### 2.3 Pull-based and Push-based approach

Pull-based approach makes each replica responsible for polling other replicas and downloading new updates, whereas push-based approach makes a replica responsible for delivering the update to other replicas. Pull-based systems never send the same update to same replica twice,

because the set of updates to be received is determined precisely through polling. Each replica only needs to keep track only of its own state in pull-based systems, because the state of other replica is obtained through polling.

### 2.2 Log-transfer and Contents-transfer System

A change to an object is expressed either by the new object contents or by its (often semantic) description ("log"). System that exchanges contents is called contents-transfer systems, whereas systems that exchange log are called log-transfer systems. Contents-transfer system transfers the entire database contents to other replicas, whereas a log-transfer only a description of the update. When an object is updated at a replica, a content-transfer system would transfer the entire database contents to other replicas, whereas a log-transfer system would transfer only a description of the update. Log-transfer owns several advantages over contents-transfer. First, log-transfer can handle update conflicts more flexibly, especially in multi-master systems. Second, log-transfer reduces both the computational and the networking overhead, especially when the object size is large and update is small.

## 3. Proposed Pull-based Log-transfer Replication System

The traditional replication systems that keep all the replicas consistent, therefore update propagation problem arise. Pull-based Log-transfer replication system allows replica contents to become stale and work well over slow or unreliable network links [5]. They can propagate update among replicas without blocking access to any replica. In this replication system, single master system is used to accept and apply changes. For sending the updated data to replicas, pull-based system is used. This system uses the log-transfer method to inform the changes to all replicas. This system implements the banking system.

### 3.1 Pull-based single master system

Single master systems designate one replica as the master that is responsible for accepting and applying changes. Other replicas, or slaves, receive changes from the master. Each replica stores a timestamp that show the last time its contents were modified. A slave replica obtains the master's timestamp periodically and downloads the master replica's contents only if its timestamp is older than the master's.

### 3.2 Log-transfer method in proposed system

Log-transfer system transfer only a description of the update and contents-transfer system transfers the entire database contents. Therefore, the data volume to be sent in log transfer system is less than contents-transfer system. So, log-transfer system can reduce the total communication time than the contents-transfer system. The formula to compute the total communication time is:

**Total Communication time= (total access delay) + (total data volume / data rate)**

Access delay = delay time that takes after sending one message

Total communication time = total time that take to reach the data from one replica to another replica

Total data volume= the amount of data to be sent from one replica to another replica

Data rate= the amount of data to be sent per second

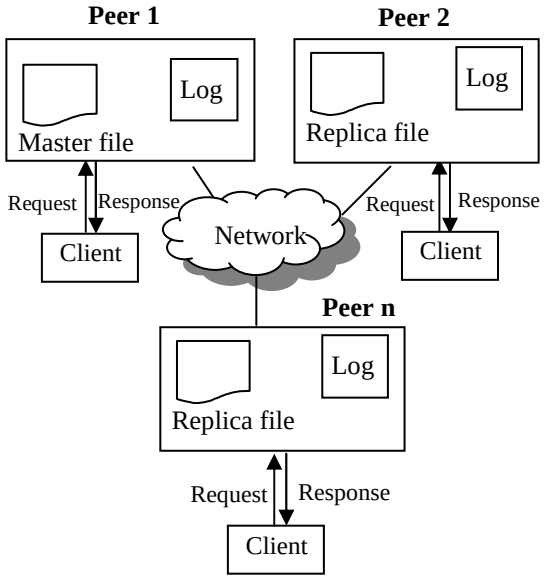


**Figure.1 System Overview**

In figure (1), each client can make request and response process at their local site because the file is replicated at each site.

### 3.3 Algorithms for replica and master site

**ALGORITHM FOR REPLICA SITE**

**1.** CLIENTS RECEIVE THE USER REQUEST.
**2**. CHECK THE UPDATE FILE.

**3**. CHECK MASTER'S LOG INFORMATION.
**4.** **IF** $VN_{SERVER}=VN_{CLIENT}$ **THEN**
**//VN is version number.**
**BEGIN**
**(I)** PROCESS THE REQUEST.
(II)SEND UPDATE INFORMATION TO SERVER.
**END**
**ELSE IF** $VN_{SERVER} > VN_{CLIENT}$ **THEN**
**BEGIN**
(I)PULL FROM SERVER.
(II) PROCESS THE REQUEST.
(III) SEND UPDATE COPY TO SERVER.
**END**
**END IF.**

**ALGORITHM FOR MASTER SITE**

**1**. RECEIVE CLIENT REQUEST.
**2**. CHECK THE UPDATE FILE.
**3.** **IF** NO MORE REQUEST **THEN**
**BEGIN**
(I) PROCESS THE REQUEST.
(II) SEND UPDATE VERSION (LOG) TO ALL REPLICAS.
**END**
**ELSE**
**BEGIN**
CHECK REQUESTS.
**IF** REQUESTS ARE QUERY THEN REQUESTS.
**IF** UPDATE REQUEST **THEN**
**BEGIN**
(I) SELECT THE LATEST UPDATE.
(II) PROCESS THE REQUEST.
(III) SEND UPDATE VERSION (LOG) TO ALL REPLICAS.
**END**
**END**
**END IF.**

### 3.4 Process Flow of the System

In this system, each replica stores a timestamp (ts) show the last time its contents were modified. When the user request enters to the replica site, compare the replica's timestamp and master's timestamp. If the replica's timestamp is older than the master's timestamp, the contents of master is downloaded to the replica site and perform the user request. If the replica's timestamp is not older than master's timestamp, perform the user request. After performed the user request, update the replica's timestamp and updated request are sent to the master site.

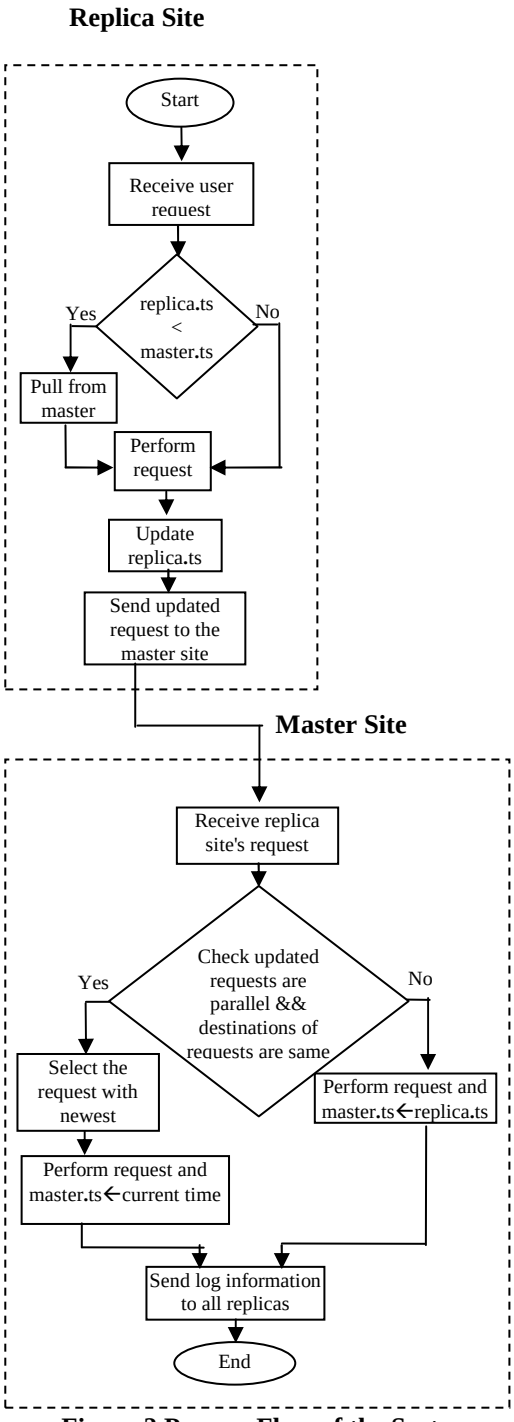


**Figure.2 Process Flow of the System**

After the replica site's request are received at the master site, if the updated requests are parallel and destination of requests are same, select the request with newest timestamp an perform the request and then the current time is assigned to the timestamp of master. Otherwise, perform the request and replica's timestamp is assigned to master's timestamp. Then send log information to all replicas. This process is shown in figure (2).

## 4. System Implementation

Proposed system presents replication protocol in distributed environment. Replicating data improve availability at the cost of maintaining consistency, since each site's view may be partial or stale. In this proposed system, replication model based on optimistic approach, since it can guarantee the successfulness of the transaction. In this system, pull-based log-transfer method is used to control the consistency.

There are three banks implanted: Yangon bank, Mandalay bank and Taunggyi bank. If the master bank is the Yangon bank then the Mandalay bank and Taunggyi bank are the replica banks. There is customer table, account table and master-log table at the master site. And there is customer table, account table and log table at the replica sites. The customer table contains Id, Name, NRC, Address, Reg_no and Create_date. The account table contains Id, Acc_no, Reg_no, Balance and Create_date. The master-log table contains Id, Acc_no, Amount, Type, Change_date and dbType. The log table contains Id, Acc_no, Change_date, dbType and Type. Customers can create bank accounts at the master bank. The data in Yangon bank is replicated at Mandalay bank and Taunggyi bank.

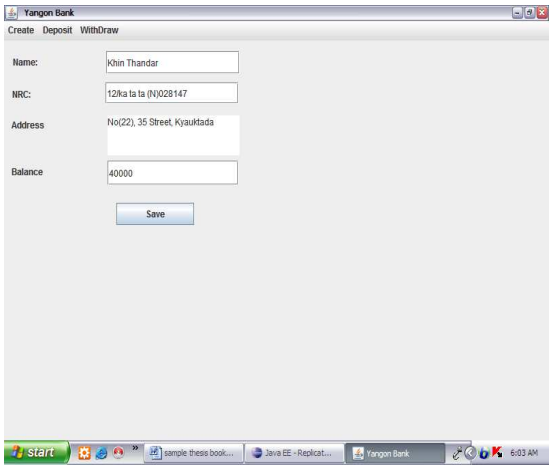


**Figure.3 Customer Entry Form**

Customer's data are filled into the system through the interface as shown in Figure.3. If a customer creates a new account at the master bank, the information about this customer will replicate at the client banks. This information contains customer name, NRC number, address, account number, balance and account created timestamp. Customers can deposit or withdraw at all banks.

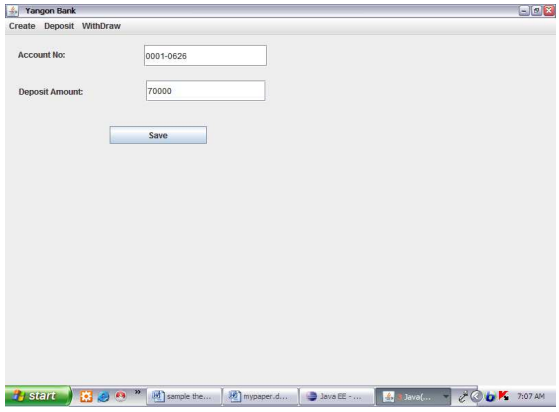


**Figure.4 Deposit Form**

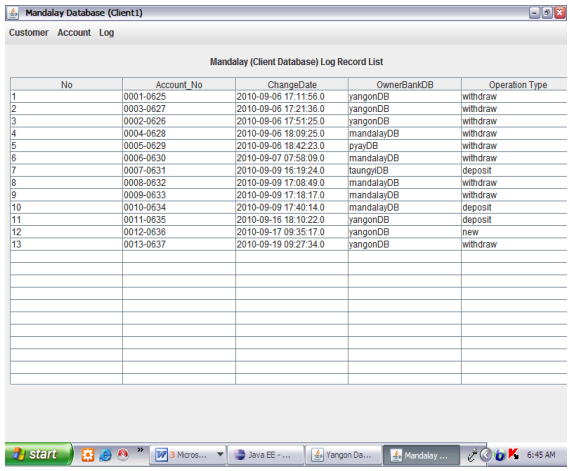


**Figure.5 Log Record Form at replica site**

If a customer deposits or withdraws at master bank (Yangon bank) through the interface as show in figure 4, the balance of this customer at the master bank will change and the log file will sent to the client banks. The log file contains customer's account number and changes timestamp. At the client bank, the changes at master bank are informed by receiving the log file as shown in figure 5. In the log file, the changes balance is not contain and so the balance at the client bank is not changed. Thus the balance at the client bank is not up-to-date.

Therefore, before a customer deposits or withdraws at client bank, compare the account created timestamp and the latest timestamp in the log file. If the latest timestamp in the log file is greater than account created timestamp, download the latest balance of this customer at the master bank and continue the deposit or withdraw process. If the latest timestamp in the log file is equal to the account created timestamp, will not need to download the latest balance of the master bank and make the deposit or withdraw process. After depositing or withdrawing at the client bank, the

content file is sent to the master bank, because the master bank is needed to store the up-to-date data. The content file contains customer's account number, changes balance and latest modified timestamp.

At the master bank, after receiving the content file from the client bank, update the old balance with the latest balance in the content file. And then send the log file to the other client bank. The log file contains customer's account number and the latest modified timestamp.

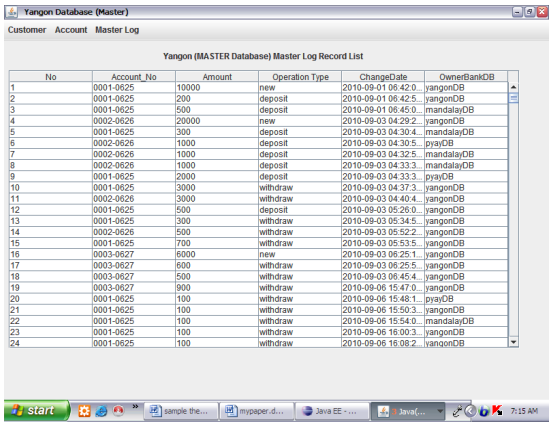


**Figure.6 Master-Log Record Form at master site**

All changes of each account number are stored in the master-log table at the master site as shown in figure 6.

### 4.1 Effectiveness of the System

Assume that all attributes in each field is 300 bits longs. For example, an attribute in field name 'Address' is 300 bits longs and so on.

- Communication assumptions:

  Data rate= 500 bits per second

  Access delay= 0.1 second

In traditional systems, if a customer deposits or withdraws at the master site, send the content information to all replicas. The content information contains customer's account number, changes balance and changes timestamp. The total communication time, the total time that take to send this information from master replica to other one replica, can be computed by the following formula. The formula to compute the total communication time is:

**Total Communication time= (total access delay) + (total data volume / data rate)**

Total access delay= 0.1second
Total data volume= 3*300= 900 bits
Data rate= 500 bits per second
Total communication time= 0.1+ (900/500)
= 1.9 seconds

In pull-based log-transfer system, if a customer deposits or withdraws at the master site, send the log information to all replicas. The log information only contains the customer's account number and changes timestamp.

Total access delay= 0.1second
Total data volume= 2*300= 600 bits
Data rate= 500 bits per second
Total communication time= 0.1+ (600/500)
= 1.3 seconds

For sending from master replica to one replica, using pull-based log-transfer system reduces the total communication time 0.6 seconds than using the traditional system. For sending from master replica to two replicas, 0.12 seconds will reduce. And then for sending from master replica to three replicas, 0.18 seconds will reduce and so on. Therefore, this system is more efficient when the number of replicas is large.

In this system, if any changes have occurred at the master site, send the log information to all replicas instead of content information. There is no need to send the changes balance and send only the changes timestamp to the replica site. When a customer deposits or withdraws at a replica site, if there is no latest modified data, the latest modified data are downloaded from the master site into this replica site by using pull-based method. There is no need to send this latest modified data to other replicas. Thus the update propagation will decrease.

## 5. Related work

The most important factor of the replication is to maintain the consistency among the replica. Therefore, computer scientist has been made research about the consistency maintaining approach. There has been an exploration of work (e.g., [5]). This works involves the combination of push and pull method to use in maintaining shared file consistency. This system employs both push and pull to achieve the advantages of both approaches. Push algorithm is server initiated approach and pull-algorithm is client initiated approach. Combining push with pull increases the consistency of the system. The earlier research guaranteed that copies would be consistent by assuring that a conflicting updates could not occur. Such approach decreases availability is desired. Further, there is increasing data that conflicting updates are extremely rare in many situations of interest. Therefore another work has been presented a practical set of algorithms for maintaining the

consistency of a replicated file system with an optimistic update policy. These algorithms permit a system which allows updates to an object so long as any copy is available; the algorithms then return the various copies to consistency at their first opportunity. These algorithms have been used to build the Ficus replicated file system [3]. Various object caching techniques between a client and a server have handled data consistency only. They have not handled object own consistency. Thus another work has been presented two techniques for object consistency maintenance [4]. The first technique makes it possible that the client confirms the object update time in the server based on RMI (Remote Method Invocation). The second is that the server broadcasts invalid message to the clients.

## 6. Conclusion

This system implements the pull-based log-transfer and single master approach for the propagating the updated data and that reduce the network traffic and update propagation time. In this system, to maintain the consistency among replicas, use the log-transfer method. If any changes have occurred at the master site, send the log information to all replicas instead of content information. Thus the amount of information to be sent on the network traffic will reduce. In order to reduce the update propagation, use the pull-based method. Pull-based method downloads the updated information only when needed to download. Therefore the update propagation will reduce. This system is useful when the number of replicas is large and when the replicated object is large and update is small.

## 7. References


[1] C.J.DATE, "An Introduction to DATABASE SYSTEMS", De Mont fort University, May 2000.

[2] Dr. George Schussel, "Replication, the Next Generation of Distributed Database Technology".

[3] R.G.Guy, G.J.Popek, T.W.Page, "Consistency Algorithms for Optimistic Replication", University of California,Los Angeles, CA 90024-1596.

[4] Seong Eun Chu, Jae Nam Kim, Dae Wook Kang, "RMI Object Consistency Maintenance Techniques at Distributed Computing", Chonnam National University, Kwangju Women's University.

[5] Si Thu Kyaw, "Cache Consistency Management in Distributed File System" , M.C.Sc Student of UCSY,2009.

[6] Yasushi Saito, "Consistency Management in Optimistic Replication Algorithms", Microsoft Research, Cambridge, UK, 2001.