Data Recovery of iBanking using Deferred Update Recovery

Swe Zin Aye, Dr. Ei Phyu Zaw
University of Computer Studies, Yangon
swezinaye552@gmail.com, zaw.eiphyu@gmail.com

Abstract

Transaction Processing System (TPS) plays a main role to handle data transaction in the business world. In TPS, there are three main functions: data loss prevention, data locking and deadlock detection and resolution. If a failure affects the operation of a database system, it is important to recover the database and return to normal condition as quickly as possible, to prevent the data loss. Recovery process from data loss prevention protects the database and users from unnecessary problems. The main process of recovery is that restoring the database to a correct state after some failure has rendered the current state inconsistent. This system intends to prevent data loss during process the online banking services or the internet banking system (iBanking). In this system, deferred update recovery is used to prevent data loss in iBanking.

Keywords: *Transaction processing system (TPS), Deferred Update Recovery, iBanking*

1. Introduction

Nowadays, internet banking system also known as iBanking and it also provides banking system to get more developed. iBanking has evolved from helping customers meet their basic banking needs and their sophisticated needs. In recent years, the number of iBanking users has been increasing exponentially. As customers get more and more educated, getting insight about iBanking has become a primary area of concern for all leading banks. iBanking refers to a system allowing individual customers to perform banking activities from home or office or from other locations through the internet.

The internet or online banking through traditional banks enable customers to perform all routine transactions, such as account transfers, balance inquiries, bill payments and stop – payment requests, and also apply for loan and Credit card. The growing popularity of the internet has made banks use this as a channel for receiving instructions and also delivering the products and services to the customers. As the iBanking service also operate on electrical appliance devices, they can be out of power supply and can have the network problems. So, this system intends to recover from these issues. If the banks can address these issues which are identified in this study and motivate the customers, then the number of internet banking users would increase.

When the user made a payment transaction, the amount must be both withdrawn from one account and added to the other in this transaction. During this transaction processing, if an error occurred or the transaction is canceled, data consistency is very important.

In this condition, this system provides original data is restored depend on checkpoint and so, the data of both payment and receive is not changed in this transaction. To have the data consistency, the recovery process run as the main part by recovering that transaction from a backup and then redo work to commit. When the current state is inconsistent during iBanking's transaction period, this system intends to prevent data loss by restoring the database to a correct state using **deferred update recovery**.

In deferred update, records in the log file are discarded on roll back and are never applied directly to the database. It is only necessary to redo the last update from the log during recovery. This method's main benefit is that any transactions need never be undone, as a transaction does not record its changes in the database until it reaches its commit point. And then the old values of data do not have to be present in the log-records. So, it requires less log storage space and takes less time to recover in the case of a system failure.

As a result, data which accurate and available when needed may improve the customer's satisfaction.

2. Related Work

Transaction Processing System (TPS) is the data life-line for a certain business organization

because it is the source of data for information systems such as MIS (Management Information System) and DSS (Decision Support Systems) [13]. TPS can be used for the various organizations, where data plays the highest priority and which are evolving in many areas such as airline reservation, banking, hotel, tourism, industry, shopping mall and health care. This system is fast and reliable enough to handle the hourly bank transactions [5].

Transaction Management System (TMS) provided access to their transmission system in order to promote completion in wholesale power market [2]. TMS is envisioned to automate and integrate the market interface and to coordinate security processes. This system supported the management of transmission and ancillary service is fast and reliable enough to handle the hourly market transactions. Transactions are indeed the unit of recovery [1]. Business organizations usually use recovery methods for data security and reliability [12].

In the development of computer science, the banking system provides simplification of using the system interacts effectively to the user request. This banking system needs to remind users a message to redo or undo when abort transactions exist until that transaction commits and it will be more effective for customers. Preventing Data loss in Bank Transactions system supports different tasks by imposing a set of rules and guidelines that specify how to prevent data loss using forward and backward recovery. It will also reduce operation time and may satisfy customer's need by using Write Ahead Logging Protocol [10].

Workflow management systems (WFMS) coordinate execution of logically related multiple tasks in an organization. Such coordination is achieved through dependencies that are specified between the tasks of a workflow. Often time preventive measures are not enough and a workflow may be subjected to malicious attacks. Traditional workflow recovery mechanisms do not address how to recover from malicious attacks. Database survivability techniques do not work for workflow because tasks in a workflow have dependencies that are not present in traditional transaction processing systems. In this paper, an algorithm that shows how can assess and repair the effects of damage caused by malicious tasks. This algorithm focuses not only on restoring the consistency of data items by removing the effects of malicious tasks but also takes appropriate actions to ensure the satisfaction of task dependencies among all the committed tasks [14].

Dewitt et al [3] suggest copressing the log size by writing only new values to disk but require the presence of stable memory large enough to hold the write-ahead log for active transactions. In absence of such storage, they flush log records in batches (group commit).

Levy and Silberschatz [4] describe an incremental recovery algorithm for main memory databases, which does not require recovery to be performed in a quiescent state, allowing transaction processing in parallel. This is achieved by recovering database pages individually.

3. Background Theory

Transaction Processing System (TPS) is a type of information system that collects, stores, modifies and retrieves the data transactions of an enterprise. TPS may fail for many reasons such as system failure, human errors, hardware failure, software application errors or natural disasters. As it is not possible to prevent all failures, TPS must be able to detect errors when they occur and cope with failures. TPS will go through a recovery of the database which may involve system log, checkpoint and recovery manager. A system log contains start and end of each transactions, any updates and records all the essential data for each transaction. The purpose of checkpoint is to provide a shapshot of the data within the database. A checkpoint that identifies the state of the database at a point in time. If a checkpoint interrupted and a recovery is required, then the database system must start recovering from a previous successful checkpoint. A recovery manager that restores the database to a correct condition which allows transaction processing to be restarted.

3.1. Transaction Processing System

Transaction Processing System (TPS) is an information processing system for business transactions involving the collection, modification and retrieval of all transaction data. TPS can be used for the various organizations, where data plays the highest priority. TPS is fast and reliable enough to handle the hourly bank transactions. There are three features of TPS. They are rapid response, inflexibility and storing and retrieving.

Rapid response: Fast performance with a rapid response time is critical. Businesses cannot afford to

have customers waiting for a TPS to respond. The turnaround time from the input of the transaction to the production for the output must be a few seconds or less.

Inflexibility: A TPS wants every transaction to be processed in the same way regardless of the user, the customer or the time for day. If a TPS were flexible, there would be too many opportunities for non-standard operations.

Storing and retrieving: Storing and retrieving information from a TPS must be efficient and effective. The data system must be well designed for its backup and recovery procedures.

3.2. Data Recovery Problem

Data updates made by a Database Management System (DBMS) are not automatically written to disk. Therefore there may be some delay between the commit and the actual disk writing (i.e., regarding the changes as permanent and worthy of being made to disk). If there is a system failure during this delay, the system must still be able to ensure that these updates reach the disk copy of the database. Conversely, data changes that may ultimately prove to be incorrect. Ensuring that only the results of complete transactions are committed to disk is an important task, which if inadequately controlled by the DBMS may lead to problems, such as the generation of an inconsistent database.

The different causes of failure such as transaction errors and system crashes will introduce some of the techniques that can be used to recover from transaction failures. Such failure will introduce some concepts that are used by recovery processes such as the system log, checkpoints, and commit points. Recovery from transaction failures usually means that the database is restored to some state from the past so that a correct state – close to the time of failure – can be reconstructed from that past state.

To do this, the system must keep information about changes to data items during transaction execution outside the database. This information is typically kept in the system log. It is important to note that a transaction may fail at any point, e.g., when data is being written to a buffer or when a log is being written to disk. All recovery mechanisms must be able to cope with the unpredictable nature of transaction failure.

3.3. System Log

The recovery manager overcomes many of the potential problems of transaction failure by a variety of techniques. Many of these are heavily dependent upon the existence of a special file known as a system log. It contains information about the start and end of each transaction and any updates, which occur in the transaction. The log keeps track of all transaction operations that affect the values of database items. This information may be needed to recover from transaction failure. The log is kept on disk. Thus, the majority of the log is not affected by failures except for a disk failure or catastrophic failure. The types of entries that are written to the log are described below.

In these entries, T refers to a unique transaction identifier that is generated automatically by the system and used to uniquely label each transaction.

- start_transaction (T): this log entry records that transaction T starts the execution.
- read_item (T, X): this log entry records that transaction T reads the value of database item X.
- write_item (T, X, old_value, new_value) : this log entry records that transaction T changes the value of the database item X from old_value to new_value. The old value is sometimes known as before image of X, and the new value is known as an after image of X.
- commit (T): this log entry records that transaction T has completed all accesses to the database successfully and its effect can be committed (recorded permanently) to the database.
- Abort (T): this records that transaction T has been aborted.

3.4. Checkpoints

In the event of failure, most recovery managers initiate procedures that involve redoing or undoing operations contained within the log. Clearly, not all operations need to be redone or undone as many transactions recorded on the log will have been successfully completed and the changes written permanently to disk.

The problem for the recovery manager is to determine which operations need to be considered and which can safely be ignored. This problem is

usually overcome by writing another kind of entry in the log, the checkpoint entry. The checkpoint is written into the log periodically and always involves the writing out to the database on disk the effect of all write operations of committed transactions.

Hence, all transactions that have their commit (T) entries in the log before a checkpoint entry will not require their write operations to be redone in case of a system crash. The recovery manager of a DBMS must decide at what intervals to take a checkpoint, the intervals are usually decided on the basis of the time elapsed, or the number of committed transactions since the last checkpoint. A checkpoint record usually contains additional information including a list of transaction active at the time of the checkpoint: many recovery methods (including the deferred and immediate update methods) need this information when a transaction is rolled back, as all transactions active at the time of the checkpoint and any subsequent ones may need to be redone.

4. System Overview

This iBanking system is developed for the model of internet banking system. It has two roles: admin role and user role. They provide for the deposit, withdraw and transfer between individual accounts. The responsibility of the admin is that can add new account and provide the deposit process of the user. Because of the user has not granted for depositing by themselves. (User requested deposit transactions are executed by the admin but withdraw and transfer process can made by themselves). All of the transactions are electronically carried out via the computers in the network environments. Therefore, the users can easily access anywhere for the banking processes via the internet. As the system is generated as a public website, the user no need to install to perform the banking functions. (Operation can make by the use of system URL address). This system is developed with C# processing language and SQL server.

4.1. Recovery Techniques Based on Deferred Update

The idea behind deferred update is to defer or postpone any actual updates to the database itself until the transaction completes its execution successfully and reaches its commit point. During transaction execution, the updates are recorded only in the write log in the transaction workspace. After the transaction reaches its commit point and the write log data is force-written to database. (ie. the updates are recorded in the database itself). If a transaction fails before reaching its commit point, there is no need to undo any operations, because the transaction has not affected the database in any way and all processing are made on write log.

The steps involved in the deferred update protocol are as follows:

When a transaction starts, any operation is performed on write log.

- write a log entry write item (T, x, old value, new value);

When a transaction is commit, the committed data from the write log is written to the database for updates. If the transaction aborts, ignore the write log records and do not write the changes to the database.
The database is never updated until after the transaction commits, and there is never a need to UNDO any operations. This technique is known as the NO-UNDO/REDO technique. REDO is needed in the case of system fails after the transaction commits but before all its changes are recorded in the database.

4.2. Advantages of Deferred Update

The major advantage with Deferred Update is that the recovery process is highly simplified. The other benefit of Deferred Update is that rollback never occurs since no transaction relies on another transaction until the commit point is reached. A transaction does not record any changes in the database on disk until after it commits because of transaction failure during transaction execution. A transaction will never read the value of an item that is written by an uncommitted transaction.

4.3. System Flow of iBanking using Deferred Update

A valid user can only be allowed to enter the internet banking system as shown in Figure 1. When a transfer transaction is processed, system tries to change balance amounts of both sender and receiver sites. When transaction commits, both sender and receiver sites update the balance of original database in both sites. Then, system removes transaction backup of both sites. System recovers commit

transaction from transaction backup, and then redo transaction to commit.

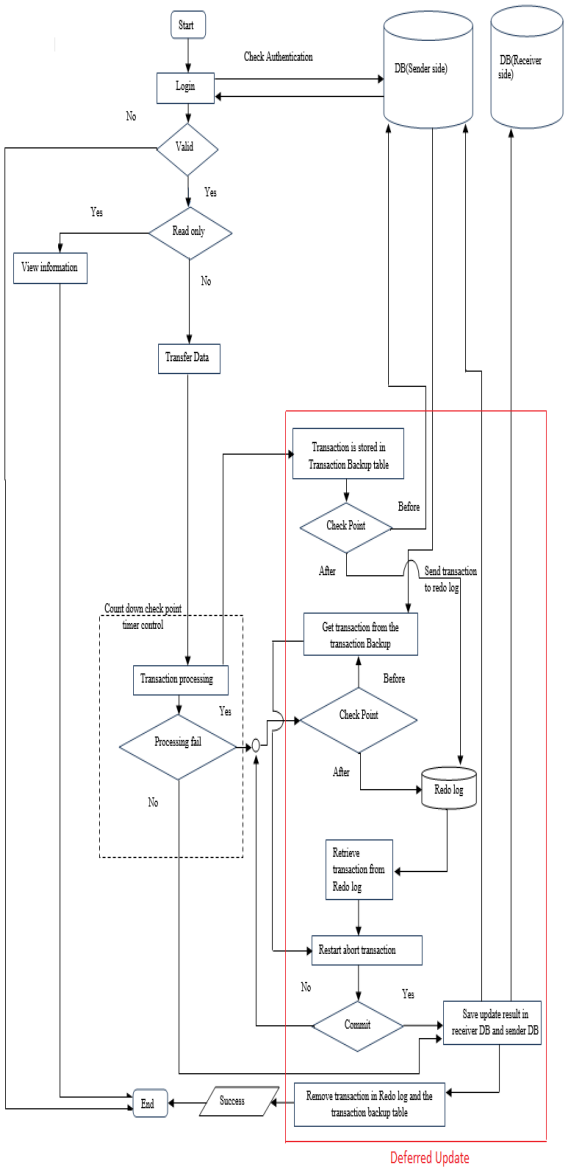

Figure 1. The System Flow

When that transaction commits, system updates balance of both sender and receiver sites and removes backup of both sites as shown in Figure 1. This system verifies how to prevent data loss in online system's transactions using **Deferred Update Recovery.**

4.4. The Database Design of the System

This system database contains four data table: User table, Transaction table, BackupTransaction table and RedoTransaction table as shown in Figure 2. User table is used for the customer data store and the Transaction table is used for the committed transactions history. RedoTransaction table and BackupTransaction table are mainly used for the control of deferred update recovery process.

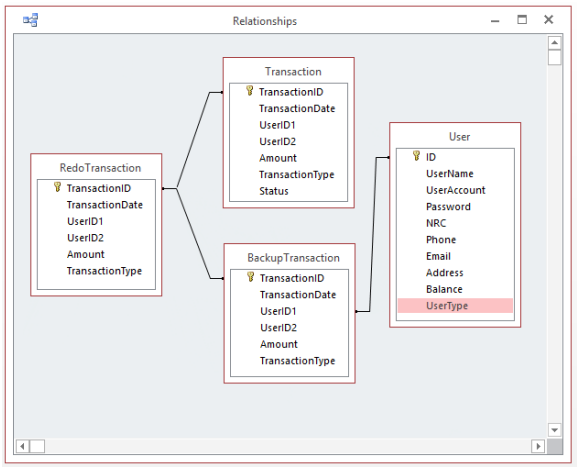

Figure 2. The Database Design

Transaction table

Transaction table consists of Transaction ID, Transaction Date, UserID1 (For sender), userID2 (For receiver), Amount, Transaction type, and Status as shown in Figure 2. Transaction table records commit transactions. So, Transaction table records transaction history in database.

BackupTransaction table

BackupTransaction table consists of TransactionID, TransactionDate, UserID1, UserID2, Amount and Transaction Type as shown in Figure 2. It can only record current transaction to recover.

User table

User table consists of User ID, UserName, UserAccount, Password, NRC, Phone, Address, Balance and UserType as shown in Figure 2. Any new customer can also be added to User table not only in sender bank but also in receiver bank. Valid customers can only login.

RedoTransaction table

Although, the "RedoTransaction" table contests are also same as "BackupTransaction" table, "RedoTransaction" table is used to recover the failure transaction which is failure after the check point.

5. Conclusions

This system intends to prevent data loss in online transactions of the bank. During the transaction, if an error occurred unfortunately, data consistency is very important and so the system need to recovery from the data loss and then redo work to commit. In this system, deferred update recovery is used to reduce operation time and may satisfy user's need. Because of the database is never updated until after the transaction commits, and there is never a need to UNDO any operations by the use of deferred updated recovery. Moreover, this system provides simplification and interacts effectively to the user request.

REFERENCES

[1] Amman, P., et. al., "Recovery from Malicious Transactions", IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 5, Sept. 2002.

[2] A.Vojdani, "Transaction Management System Architecture and Business Plan", Report, Electronic Power Research Institute, Jim Lamb, Dallas, Texas.

[3] C. Mohan, D. Haderle, B. Lindsay, H. Pirahesh, and P. Schwarz. Aries: A transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. ACM Trans. Database Syst , 1 7 : 94-1 62, March 1 992.

[4] E. Levy and A. Silberschatz. Incremental recovery in main memory database systems. iEEE Trans. on Knowl. and Data Eng. , 4: 529-540,

[5] Ghazi Alkhatib and Ronny S. Labban, "*Transaction Management in Distributed Database Systems*: the Case of Oracle's Two-Phase Commit", Senior Lecturer of MIS, Qatar College of Technology, Doha, Qatar and Computer & Communications Engineer; Consolidated Contractors International Company Athens, Greece;

[6] H .Garcia-Molina and K. Salem. Main memory database systems: An overview. iEEE Transactions on Knowledge and Data Engineering, 4: 509-5 16, 1 992.

[7] H. V. Jagadish, A. Silberschatz, and S .Sudarshan. Recovering from main-memory lapses. VLDB ' 9 3 , pages 3 9 1 -404.

[8] L .Gruenwald, J. Huang, M. H. Dunham, J.-L. Lin, and A. C. Peltier. Recovery in main memory databases, 1 996.

[9] M. H. Eich. Main memory database recovery. In Proceedings of i986 ACM Fall joint computer conference, ACM ' 86, pages 1 226- 1 232, 1 986.

[10] Swe Swe Win "Preventing Data loss in Bank Transactions using Write Ahead Logging Protocol" University of Computer Studies, Yangon.

[11] T. 1. Lehman and M. J. Carey. A concurrency control algorithm for memory-resident database systems. FOFO ' 89, pages 490-504, London, UK, 1989. Springer-Verlag.

[12] Tripathy, S. and B. Panda, "Post-Intrusion Recovery Using Data Dependency Approach", Proceedings of the 2001 IEEE Workshop on Information Assurance and Security.

[13] Vladimir Zwass, "*Foundations of Information System*", Fairleigh Dickinson University, McGraw-Hill Companies, Inc., International Editions 1998.

[14] Yajie Zhu,Tai Xin, and Indrakshi Ray, "Recovering from Malicious Attacks in Workflow Systems", Department of Computer Science, Colorado State University.