

A Review: Transaction Processing Management System

Kritika Bhatnagar

kritika360@gmail.com

MCA Scholar

Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan

Abstract

Transaction is enterprise applications often require concurrent access to distributed data shared amongst more components, to perform operation on data. A transaction processing system is a real time processing system for business transactions involving the collection, modification and retrieval of all transaction data.

Key Words: Transaction, Processing, Management, State, ACID, Properties

1. Introduction

A Transaction includes one or more statements that modified data on one or more two different sites of distributed database. In this paper, we discuss transaction processing and their problems that arise due to distribution of data. We also discuss process of transaction and states of transaction. The transaction in enterprise applications often require concurrent access to distributed data shared amongst more components, to perform operation on data. The concept of a transaction and a transaction processing service simplifies construction of such enterprise level distributed applications while maintaining integrity of data in a unit of work. Such applications should maintain integrity of data that means distributed access to a single resource of data, and access to distributed resources from a single application component. In such cases, it may be required that a group of operations on resources treat as one unit of work. In a unit of work, all the operations should either succeed or fail and recover together [1]. Transaction can be defined as a unit of work. In a traditional way, a transaction is an agreement between a customer and a vendor to exchange an asset for payment. This is the definition in a business sense. In a database point of view, it can be understand as a unit of work performed within a database management system against a database, and treated reliable way independent of other transactions [3].

2. Transaction Processing System

A transaction processing system is a real time processing system for business transactions involving the collection, modification and retrieval of all transaction data. ATM (Automatic teller machines) is an example of transaction processing. A transaction processing system are often opposite with a batch processing system, where many requests are executed all at one time. TPS requires the interaction with a user, whereas a batch processing system does not require interaction with a user. In transaction processing there is no delay and the results of transaction are immediately available, whereas a batch processing error can occur. TPS characteristics are performance, reliability and consistency, data must be readily accessible in a data warehouse, backup procedures and the recovery process must be deal in place with system failure, computer viruses, and human failure and software applications [5].

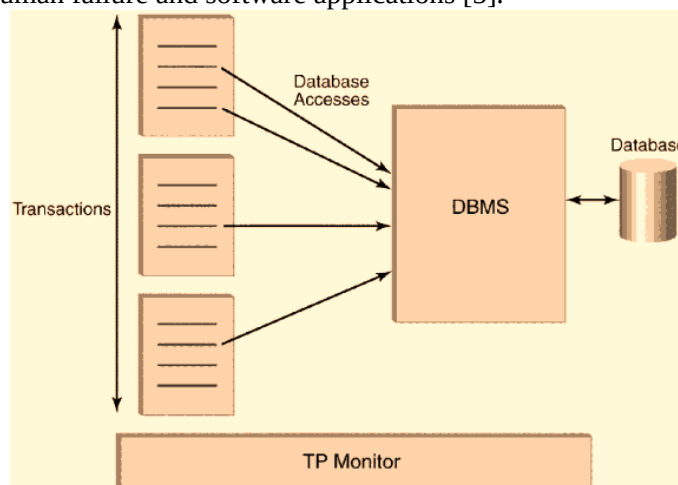


Figure 1: Transaction Processing System

A web browser or a graphical user interface application is often “Front end”. The DBMS is a “Back end” component of the transaction processing system. There may be multiple databases being accessed and multiple transactions in process at any given time [2].

3. Transaction Process in Account

Let’s take an example of a simple transaction. A bank employee transfers Rs.5000 from A’s account to B’s account. This is very simple and small transaction [4].

A’s Account

Open Account (A)

Old Balance = A. balance

New Balance = Old Balance - 5000

A. balance = New Balance

Close Account (A)

B’s Account

Open Account (B)

Old Balance = B. balance

New Balance = Old Balance + 5000

B. balance = New Balance

Close Account (B)

Transaction management is one of the most important requirements for enterprise application development. Most of the large enterprise applications in the domains of banking, finance and electronic commerce believe on transaction processing for delivering their business functionality [1]. Transaction processing is a type of computer science in which the computer responds quickly to user request. Transaction processing is related to any real-time business transaction. It is a process that performed by a transaction processing system (TPS).

4. Process of Transaction

The transaction is executed as a series of read and writes operation of database object.

There are two types of operation

A. Read operation (x): The read a database object, it is first brought into main memory from disk, and then its value is copied into a program variable x. To simplify the notation we assumed our variable name is x.

B. Write operation(x): The write a database object, an in-memory copy of the object is first modified and then written to disk. Writes the value of program variable x into the database [6].

5. State of Transaction

These are the various states through which a transaction passes through in its lifetime. In this now, we will discuss all the states that cause transition from one state to another. These are the following six states which are explained below:

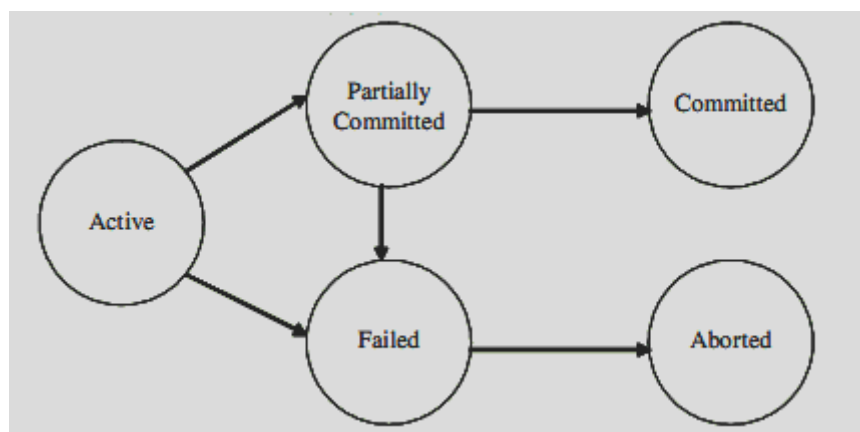


Figure 2: States of Transaction

A. Active: This state is the initial state of every transaction. In this state, transaction is being started or in other words we can say that transaction has just started execution.

B. Partially committed: When a transaction executed its final operation or initial statement has been executed properly, then it is said to be a partially committed.

- C. Failed:** In this state when the normal execution can no longer proceed. A transaction is in failed state, when any of the checks made by the database recovery system fails.
- D. Aborted:** If any of the transaction is failed and the transaction has reached a failed state. The temporary values are no longer required, and then the transaction is set to ROLLBACK. That means restoring a database to a previous state by canceling a specific transaction set. It is performed automatically by database system or manually by the user. Transaction in this state are aborted. For example if the failed transaction has withdrawn Rs. 500/- from Account A, then the ROLLBACK operation should add Rs 500/- to account A.
- E. Committed:** After the successful completion and if no failure occur then the transaction reaches the COMMIT POINT. It is said to be committed. All the temporary values are written to the stable storage and all its effects are now permanently established on the database system [4], [7].

6. ACID Properties

Now we discuss about ACID properties in distributed database system. ACID (Atomicity, Consistency, Isolation and Durability) properties are an important concept for database. In distributed database transaction ACID properties are widely used and very important for distributed database transaction. There are many models presented to support distributed database system, but they failed. These are all four properties to give reliable services. Distributed database system is controlled by a Central Database Management System that is dispersed over computer network. Replication and duplication these are two processes that performed on distributed database. It must be assured that the database should be consistent and stable. Replication that means, it is the frequent electronic copying from a database in one computer or server to a database in another so that all user share the same information. The main purpose of replication is eliminating data ambiguity or inconsistency. ACID properties are the solution of this problem. In transaction management it gives the reliability of transaction over distributed database [8]. The ACID properties allow safe sharing of data. ACID are developed for business oriented applications. Without these ACID properties, everyday occurrences such using computer system to buy products would be difficult. For ex, suppose that more than one person trying to buy the same size and color of shirts at the same time- a regular occurrence. The ACID properties make it possible for the merchant to keep these shirts purchasing transactions from overlapping each other [8],[9].

7. Types of ACID Properties

- A. Atomicity:** In Atomicity concept, database follows the all or nothing rule. This property determines that a transaction can either be a single operation, or a sequence of operation respectively sub-transactions [10]. Atomicity concept said either all of the rules within the transaction will be reflected in the database or none of them will be reflected. This states should be defined either before of the transaction or after the execution or failure of the transaction [4]. A transaction is executes all its actions in one step or not executes any actions at all. Atomicity states that every transaction should be reach the commit point and abort point. If the transaction successfully completes it is called **COMMIT**. If the transaction does not successfully completes it is called **ABORT**.

Reasons for Aborting

- i. System is crash
- ii. Resources are not available
- iii. If a site is down in distributed transaction
- iv. Integrity constraints is violated

Example of Atomicity: Let, x1 is an transaction that transfer Rs. 500 from Account A to Account B. this transaction can be define as- assume that just prior to execution of transaction x1 the values of account a and b are Rs 1000 and Rs 2000. Now imagine that during the execution of x1, A power failure has occurred that prevented the x1 to complete successfully. The point of failure may be after the completion write (A,a) and before write (B,b). That means the changes in A are performed but not in B. Thus the values of account A and remaining Rs. 500 and Rs 2000 respectively. We have lost Rs 500 as a result of failure. Now our database is inconsistent state and the reason is our transaction is completed partially and we save the changes of uncommitted transaction. To get the consistent state, database must be restored to its original values A to Rs. 1000 and B to Rs. 2000. This is the concept of Atomicity [7].

B. Consistency: In Consistency state, No violation of integrity constraints. We define consistency as the transition from one valid state to another and when executing transaction, never leaving behind an invalid state. Database is in a consistent state initially, when the transaction is complete then the transaction is consistent [10]. Consistency checking constraints are divided in two parts

Automatic and manual

Automatic → CHECK, ASSERTION is for static constraints and TRIGGER for dynamic constraints

Increases confidence in correctness

Decreases maintenance cost.

Manual perform check in Application code and Necessary checks are performed.

Example of consistency: To give better performance every database management system using CPU Time sharing to support the execution of multiple transactions at the same time. Concurrently executing transactions have to face with the problem of sharable resources that means resources that multiple transactions are trying to read and write both of the operation at the same time. For example, suppose that a record on which two transactions are trying to perform read and write operation at the same time. A transaction which deposits Rs 500/- to account A must deposit the same amount whether in conjunction with another transactions that may be trying to deposit some amount at the same time [7].

The consistency requirement is that the sum of A and B must be unchanged by the execution of transaction.

C. Isolation: when multiple transactions are executed simultaneously and trying to access the sharable resources at the same time then the system should create a sequence in their execution so that they should not create any ambiguity in the value stored at the sharable resources [7]. Isolation property means concurrent changes invisibles. Isolation property states that where more than one transaction are be executed concurrently and in parallel, that all the transaction will be carried out and executed as if it is the only transaction in the database system [4]. Isolation concept is that the data used during the execution of transaction cannot be used by a second transaction until the first transaction is completed.

Example of Isolation: Now consider another transaction x2, which has to display the sum of a/c A and B. then, its result should be 3000. Let's assume that both x1 and x2 perform simultaneously. The inconsistency of database is that it shows Rs 2500/- as sum of a/c A and B instead of Rs. 3000. The problem occur because second transaction concurrently running transaction x2 until the execution of first transaction x1. So Isolation property says that second transaction cannot be used during the execution of transaction until first one is completed.

D. Durability: In durability transaction should be durable by storing the data permanently. It is available in case of power failure and recovery system failure, crash etc [11]. Durability property means committed update persist. Durability concept says that once a transaction has been complete the changes it has made should be permanent. The effect of committed transaction should persist even after a crash. To hold all its updates even if the system is fails then the database should be durable. Once the transaction completes successfully all the updates, the system must guarantee that the result of its operation will never be lost even in case of failure. This state is Recovery based Database Management System.

Example of Durability:- If a transaction completes successfully but the due to some reason system fails before the data could be written on to the disk, then that data will be recover once the system springs returns into action [4].

8. Conclusion

In this paper we have reviewed the basic concept of advance transaction management. In this we show the basic concept of transaction management, its states, operation and mainly focus in ACID properties. The ACID properties are very important for database to perform Atomicity, Consistency, Isolation and Durability in transactions. It ensures all data business field and research to be a correct and valid state.

9. References

[1]. <http://www.subbu.org/articles/transaction/NutsAndBoltsOfTP.html>.

[2]. <http://jcsites.juniata.edu/faculty/rhodes/dbms/intro.htm>.

- [3]. Shiwei Yu, ACID properties in Distributed Databases.
- [4].http://www.tutorialspoint.com/dbms/dbms_transaction.htm.
- [5].<https://www.techopedia.com/definition/707/transaction-process-system-tps>.
- [6].<https://ecomputernotes.com/database-system/rdbms/transaction>.
- [7].<http://www.eazynotes.com/pages/database-maanagement-system/transaction-management.html>.
- [8].<http://www.uniassignment.com/essay-Samples/information-technology/acid-properties-in-distributed-database-transaction-information-technology-essay.php>.
- [9].http://www.service-architecture.com/articles/database/acid_properties.htm.
- [10].http://berb.github.io/diploma-thesis/community/061_challenge.htm.
- [11].<https://www.quora.com/What-is-the-use-of-ACID-property-in-database-manangement-system>.