Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 5, May 2016, pg.599 – 602

Concurrency Control in DBMS- A Review

Sukhdev Singh Ghuman

Assistant Professor, SBDSM Khalsa College

Domeli(Kapurthala), Punjab (India)

ghumanggg@gmail.com

Abstract: *Concurrent operations on database are very much important to get the benefit from it, but concurrent operations also cause some problems which need to be addressed. A transaction is a unit of work which may be doing multiple updates to data. All the transaction execution should follow ACID properties. Many concurrency control mechanisms can used to control the unwarranted side effects of simultaneous transactions. These include optimistic, pessimistic and semi-optimistic.*

Keywords: *Transaction, Concurrent, DBMS, Roll Back, Correctness.*

I. Introduction

Database systems are essential for many applications, ranging from space station operations to automatic teller machines. A database state represents the values of the database objects that represent some real-world entity. The database state is changed by the execution of a user transaction. Individual transactions running in isolation are assumed to be correct. When multiple users access multiple database objects residing on multiple sites in a distributed database system, the problem of concurrency control arises [3].

Concurrency control is mechanism applied in concurrent operations of database so that all the operation complete successfully without interfering in the results of each other. It ensures that the correct result of the concurrent operations is generated.Concurrency control provides rules, methods, design methodologies, and theories to maintain the consistency of components operating concurrently while interacting with other components.It ensures the consistency and correctness of the whole system without performance reduction [1]. Concurrency Control makes sure that individual

users see consistent state of database even though operations on behalf of many users may be interleaved by the database system[2]. A transaction is a unit of work which may be doing multiple updates to data. All the transaction execution should follow ACID properties.

II. ACID Rules

The most basic rules which each transaction should follow to prevent concurrency problems are as listed below [2]:-

- **Atomicity** –It means that either the transaction should complete itself or it should not start at all. It should appear atomic means indivisible.
- **Consistency** - Every transaction must leave the database in a consistent state.It should maintain the integrity rules applicable to the database. A transaction must transform a database from one consistent state to another consistent state.
- **Isolation** - Transactions cannot interfere with each other. The effects of an incomplete transaction are not even visible to another transaction. Providing isolation is the main goal of concurrency control.
- **Durability** - Effects of successful transactions must persist through crashes. The effect of the committed transactions must survive failures.

III. Concurrency Control Mechanism

The main categories of concurrency control mechanisms are [1]:

- **Optimistic** – It checks the transaction at the end to see whether the transaction meets the isolation and other integrity rules, without blocking any of its operations and then abort a transaction to prevent the violation if the rules are violated upon its commit. An aborted transaction is immediately restarted and re-executed, which incurs an obvious overhead.
- **Pessimistic** - Block an operation of a transaction, if it may cause violation of the rules, until the possibility of violation disappears. Blocking operations is typically involved with performance reduction.
- **Semi-optimistic** - Block operations in some situations, if they may cause violation of some rules, and do not block in other situations while delaying rules checking to transaction's end, as done with optimistic.

IV. Goals of Concurrency Control

The main Goals of Concurrency control mechanism are as given below:-

- **Correctness**
- The first and foremost goal of every system is to operate correctly. Each transaction's integrity rules should be obeyed while transactions are running concurrently, and thus the integrity of the entire transactional system is maintained. Correctness must reduce the performance, it should be achieved with as good performance as possible.
- **Serializability**
- The second major major goal of most concurrency control mechanisms is generating schedules with the *Serializability* property. Without serializability many problems occur in DBMS updation. Serializability is considered the highest level of isolation among database transactions, and the major correctness criterion for concurrent transactions
- **Recoverability**
- The Recoverability property of schedules is very important for maintaining correctness in cases of aborted transactions. Recoverability means that no committed transaction in a schedule has read data written by an aborted transaction. Such data disappear from the database upon the aborting a transaction. Reading such data violates the consistency rule of ACID.
- **Recover**
- Almost all systems fail at one time or at another. The handling *recovery* from failure is very important. The properties of the generated schedules, which prepared as per the concurrency control mechanism, may have an impact on the effectiveness and efficiency of recovery.
- **Replication**
- Replication is the process of copying data at more than one location so that its availability is high. Database objects are replicated. The Updates of replicas of a same database object need to be kept synchronized. This may affect the way concurrency control is done.

V. Approaches to design concurrency Control Algorithms

There are basically three generic approaches that can be used to design concurrency control algorithms. These are as given below:-

- **Wait:** If two transactions conflict, conflicting actions of one transaction must wait until the actions of the other transactions are completed.

- **Timestamp:** The order in which transactions are executed is selected based on a time stamp. Each transaction is assigned a unique timestamp by the system and conflicting actions of two transactions are processed in timestamp order. The time stamp may be assigned in the beginning, middle or end of the execution. Version-based approaches assign time stamps to database objects.
- **Rollback:** If two transactions conflict, some actions of a transaction are undone or rolled back or else one of the transactions is restarted. This approach is also called optimistic because it is expected that conflicts are such that only a few transactions would rollback.

VI. Conclusions

Concurrency Control is a problem that arises when multiple processes are involved in any part of the system. It is the performance requirement which has influenced the work in concurrency control. In summary, I believe that the basic principles of concurrency control are well understood. Various goals of concurrency control are explained in detail and approaches to concurrency control are discussed in detail. Different approaches to concurrency control algorithms like wait, time stamp and roll back are important. The use of serializability as a correctness criterion is popular among researchers [3].

References:

[1] https://en.wikipedia.org/wiki/Concurrency_control

[2] “Concurrency Control and Recovery “, Michael J. Franklin, University of Mary Land.

[3] “Concurrency Control in Database Systems “,Bharat Bhargava, Fellow, IEEE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 11, NO. 1, JANUARY/FEBRUARY 1999

