Solutions for Concurrency Control in DBMS : A Survey

M Faheem Feroz : International Islamic University Islamabad, Pakistan

July 11, 2018

Abstract

In Database Management System (DBMS), several algorithms have been proposed for inconsistency issue.Various researches took a look at the the concurrency control algorithms created for Data Base Management Systems and changing them with different environments. This paper provides few Concurrency Control methods and algorithms and shows weaknesses and good points of those.

1 Introduction

We all know that DBMS has become an essential part of almost all applications whether they are mobile applications or web applications. It is necessary to check that transactions made by databases of these applications are safe and all the data of users is also secured. Databases basically stores all the data of these applications in a kind of store, so when user need any data that application fetches the data from that store. Many techniques/methods are there to ensure the safety of data.In field of IT and computer science, concurrency control makes sure that accurate results for concurrent operations are obtained while obtaining very quickly. If transactions are executed in such a way that they don't overlap than nothing of these methods are needed. However if two transactions are executed at same time ,than there comes the role of concurrency control techniques, so that both of the transactions execute properly and changes made by those transactions in DB are done safely and securely.
As this (Concurrency control) [1] is an important part of DB, so in this paper we will take a brief look on all the techniques and methods used to ensure that all the concurrent transactions are secure. Many solutions are presented by researchers in this field and are being used but we will consider only those which are used mostly by DBMS.
For analysis of methods or techniques of concurrency control in DBs, there are few metrics [2] given such as:
Accuracy : It is a percentage at which correctness of data is obtained or achieved.
Serializability : It makes sure that the schedule for the concurrent execution of the transactions gets unchanging results and transaction can be accomplished in the same sequence as their request was sent.
Deadlock : It occurs when two transactions working on same data wait for each other to unlock data they both need.
Storage : Memory and RAM requirements for database storage.
This paper contains the following sections. In the first section I will try to explain the inside details of concurrency controls methods or techniques. In the next section I will discuss their advantages and disadvantages. At the end, Conclusion will be given.

2 Related Work

Now in this related work [3] part I will discuss the some work that has been done in this field. I have selected the following techniques : (1) Two-Phase Locking Protocol , (2) Multi version Schemes , (3) Time stamp-Based Protocols, (4) Lock-Based Protocols.

2.1 Two Phase Locking Control

This protocol is known as 2 Phase locking because it has two principal phases.The first phase is growing phase and second is shrinking phase. This Protocol ensures serializability by putting some limitations on transaction because Any other transaction can not get new locks till it had released a previous lock, this limitation or check is called two phase locking.The first phase, the growing phase; in which a transaction gets all the locks it needs and the second phase, shrinking phase; where the process set the locks free it got in first phase [4]. If a process can not get all the locks during the first phase due to any reason, then it is has to let go all of these locks, and has to wait, and then start over to get the locks again.This protocol make sure that conflict–serializable schedules. As described in [5], [6], [7].
This protocol may work better if any information about the transactions or the databases is not given to you. I have studied two types of 2-phase locking protocol: strict two-phase locking and rigorous two-phase locking.

Strict two-phase Locking

In this protocol any transaction does not lets go of any of its write lock before it commits or aborts. Any other transaction cannot acquire the locked item before ongoing transaction has committed. Transaction should keep all of its acquired locks till it commits or aborts and no rollback occurs. Read lock of transaction can be set free only after transaction completes but write lock should not b let go before termination of specific transaction.

Rigorous two-phase Locking

In this all locks (read and write) are kept up to the termination of a specific transaction.Main disadvantage of these protocols is starvation which happens when a transaction cannot work for an unknown span of time while other transactions going smoothly.

2.2 Multi-version concurrency control

In this protocol, when the item is updated it stores the old values of a data item, there are many kinds of data item given for transaction for write operation and correct type is kept for read operation. when a transaction issues write operation, it writes a new version and old version is kept. A big disadvantage of multi version concurrency techniques is that more and more storage will b needed for all kinds or versions of data.

2.3 Time stamp based protocols

This protocol tries to keep information about the order of entry of implementation. Locking algorithms are completely pushed aside for this protocol and this algorithm is carried out using timestamps.A distinctive value that is given to transaction when it starts is known as time stamp. There is a write time stamp and read time stamp for every data item. Write operation is carried out successfully by write time stamp and it is the biggest time stamp while read operation is carried out successfully by read time stamp and it is also the largest of the time stamp.Serializability order is determined by using time stamp and have concurrent execution as well. You have to make sure serializability by letting older transaction to get data before the higher time stamp or newer transaction. This is very important in this.Time stamp protocol always manages that transactions's read and write time stamp for every data member every time a process tries to gain access to any data. We use WTS for Write time stamp and RTS is used for Read time stamp.
Now lets talk about the advantage of this protocol,that is any problem related to appearance of deadlock is solved by this protocol easily, because there are two values against every item in database in this protocol , one for the last read and other one for the last update commit in the database. As a result memory needs are expanded and processing overhead of database is enhanced.

2.4 Lock-Based Protocols

In this protocol a specific lock is assigned to current transaction so that it can use any data item freely as long as it needs.In this lock is kept for only one transaction for any data item and it stops others to get that data item in that period of time when that specific data item is locked for another transaction. Second transaction can get the data only after the lock from data is released by the first transaction. After releasing lock, data item becomes available for any other transaction.
There are two modes in which data can can be locked, one is exclusive mode and second is shared mode. We denote exclusive mode with X and shared mode with S.

Exclusive-mode lock is granted to that transaction that can read as well as write from data item X. On the other hand shared mode lock is granted to those transaction that can read data bus cannot write on items S.Transaction will go further after the request is accepted. after request is granted.Shared locks (S) on an item can be taken by unlimited number of transactions. while no other transaction can use exclusive lock on that item if a transaction already kept a lock on same data item.After the lock is released by one transaction then it is granted to others.
Disadvantage of this protocol is that the schedule may produce deadlock.

3 Comparison

3.1 Performance

For read only optimistic protocols are good while locking protocols are good for update-intensive applications.This is because overheads of locking of read-only transactions are not needed in this and can give better results.Time stamp can give better results if some available information about the transactions or the database can be used for increasing concurrency then time stamp can give better better results.
In a locking approach, having them wait at certain points, while in an optimistic approach backing them up controls the transactions. In multi-version scheme a read operation is never cast out or excluded, while large parts of the database stays on secondary storage.

3.2 Seializability

Locking makes sure serializability for all kinds of transactions: Read only or Update-intensive which can work concurrently with a given transaction. It is good for update-intensive applications because it is much risk less.
The time-stamp-ordering protocol makes sure conflict serializability. This happens due to conflicting operations are carried out in time-stamp order.
Every transaction can read the same item again and again and there will be no conflict.
The Optimistic Concurrency Control operates on this supposition that conflicts occurs very rarely among the transaction .It does not need locking mechanism at all.When the validation is done then the transaction is executed.
The multi-version two-phase locking protocol tries to join the benefits of multi-version concurrency control with the benefits of two-phase locking for giving serializable schedules.

4 Conclusion

In this paper, balanced study between the different forms of lock algorithm, time stamp ordering and optimistic concurrency control algorithms which have been used recently in distributed, mobile databases have been done based on few factors like reduced blocking, consistency, load balancing, efficiency, security etc. Depending upon application's needs and resources free for any system, acceptance of a specific variant to be used can be chosen for a specific environment.

References

[1] Wikipedia.org

[2] Transactions and Concurrency Control in DBMS by Thien Si Li in linkedin.com

[3] academia.stackexchange.com (How to write related work) [4] www.geeksforgeeks.org

[5] Samuel Kaspi and Sitalakshmi Venkatraman,"Performance Analysis of Concurrency Control Mechanisms for OLTP Databases", International Journal of Information and Education Technology, Vol. 4, No. 4, August 2014

[6] Md. Anisur Rahman," An Efficient Concurrency Control Technique for Mobile Database Environment", Global Journal of Computer Science and Technology, Vol 13,No. 2,2013.

[7] D. Lomet et Al.," Multi-version Concurrency via Timestamp Range Conflict Management", IEEE 28th International Conference of Data Engineering (ICDE) ,1- 5 April 2012