Concurrency Control

By,

Harshil T. Kanakia

Concurrency Control

The mechanism through which the system controls the interaction among the concurrent transactions.

Concurrency control techniques

- Lock based protocols
- Time stamp based protocols
- Validation based protocols

Lock based protocols

To allow a transaction to access a data item only if it is currently holding a lock on that data item

Locks

- Shared :- Transaction can read but can not write.

- Exclusive:- Transaction can read as well as write.

Lock Compatibilty Matrix

	S	**X**
S	true	false
X	false	false

Example of Transaction which uses locks

T1
lock-x(B);
read(B);
B := B - 50;
Write(B);
unlock(B);
lock-x(A);
read(A);
A := A + 50;
Write(A);
unlock(A);

Example of Transaction which uses locks

T2
lock-s(A);
read(A);
unlock(A);
lock-s(B);
read(B);
unlock(B);
display (A+B);

How Concurrency control manager comes into picture?

T1	T2	CCM
lock-x(B);		grant-x(B,T1)
read(B);		
B := B - 50;		
Write(B);		
unlock(B);		
	lock-s(A);	grant-s(A,T2)
	read(A);	
	unlock(A);	
	lock-s(B);	grant-s(B,T2)
	read(B);	
	unlock(B);	
	display (A+B);	

Two phase locking protocol

- It ensures serializibilty.
- It requires that each transaction issue lock and unlock request in 2 phases.

a) Growing

b) Shrinking

Example of Transaction which follows 2 phase locking protocol

T3
lock-x(B);
read(B);
B := B - 50;
write(B);
lock-x(A);
read(A);
A := A + 50;
write (A);
write(A);
unlock(A);
unlock(B);

Example of Transaction which does not follow 2 phase locking protocol

T4
lock-x(A);
read(A);
A : = A+10;
write(A);
unlock(A);
lock-s(A);
read(A);
unlock(A);

Upgrade and Downgrade

- Upgrade :- Conversion from shared to exclusive mode. (Only in Growing phase)
- Downgrade:- Conversion from Exclusive to shared mode. (Only in shrinking phase)

Example

T5
lock-s(A);
read(A);
lock-s(B);
read(B);
upgrade(A);
A := A + B;
Write(A);
unlock(A);
unlock(B);

Versions of 2 phase locking protocols

- Strict 2 phase locking protocol

- Rigorous 2 phase locking protocol

Strict 2 phase locking protocol

This protocol not only requires two-phase locking but also all exclusive-locks should be held until the transaction commits or aborts.

Rigorous 2 phase locking protocol

This protocol requires that all the share and exclusive locks to be held until the transaction commits.

Disadvantage of 2 phase locking protocol

Deadlock

Example of Deadlock

T1	T2
lock-x(B);	
read(B);	
B := B - 50;	
wirte(B);	
	lock-s(A);
	read(A);
	lock-s(B);
lock-x(A);	

Solution to Deadlock

Rollback one of the two transaction.

Timestamp based protocol

- Timestamp
- Timestamp ordering protocol
- Thomas write rule

Timestamp

- Every transaction has a timestamp associated with it, and the ordering is determined by the age of the transaction.
- A transaction created at 0002 clock time would be older than all other transactions that come after it. For example, any transaction 'y' entering the system at 0004 is two seconds younger and the priority would be given to the older one.

Timestamp

- It is denoted by TS(Ti).
- Methods:

1) **System Clock:**

the value of a clock when transaction enters the system.

2) **Logical counter:**

incremental value when transaction enters the system.

Timestamp

- To implement this, we associate with each data item Q two timestamp values:

a) W-timestamp(Q) :- largest timestamp of any transaction that excuted Write(Q) successfully.

b) R-timestamp(Q) :- largest timestamp of any transaction that excuted Read(Q) successfully.

Timestamp ordering protocol

Case 1:- Suppose that transaction Ti issues Read(Q)

i) If TS(Ti) < W-Timestamp(Q) then Ti needs to read a value of Q that was already overwritten. Hence, Read is rejected and Ti is rolled back.

Timestamp ordering protocol

Case 1:- Suppose that transaction Ti issues Read(Q)

ii) If TS(Ti) >= W-Timestamp(Q) then read operation is executed and

R-Timestamp(Q) is set to maximum of

R-timestamp(Q) and TS(Ti).

Timestamp ordering protocol

- Case 2:- Suppose that transaction Ti issues Write(Q)

i) If TS(Ti) < R-Timestamp(Q) then it is rejected and Ti is rolled back.

ii) If TS(Ti) < W-Timestamp(Q) then Ti is attempting to write an obselete value of Q. Hence system reject this write operation and Ti is rolled back.

Timestamp ordering protocol

- Case 2:- Suppose that transaction Ti issues Write(Q)

iii) Otherwise, system executes the Write operation and sets W-timestamp(Q) to TS(Ti).

Thomas write rule

Instead of making Ti rolled back, the 'write' operation itself is ignored.

Validation based protocols

- It is also called as Optimistic Concurrency Control Technique.
- In this technique, no checking is done while the transaction is been executed. Until the transaction end is reached updates in the transaction are not applied directly to the database. All updates are applied to local copies of data items kept for transaction.
- At the end of transaction execution, while execution of transaction, a validation phase checks whether any of transaction updates violate serializability.
- If there is no violation of serializability the transaction is committed and the database is updated; or else, the transaction is updated and then restarted.

3 Phases of Validation based Protocol

- **Read Phase:** Values of committed data items from the database can be read by a transaction. Updates are only applied to local data versions.
- **Validation Phase:** Checking is performed to make sure that there is no violation of serializability when the transaction updates are applied to database.
- **Write Phase:** On the success of validation phase, the transaction updates are applied to the database, otherwise, the updates are discarded and the transaction is slowed down.

End