AU : Dec-15, Marks 8

3.10 Two Phase Locking

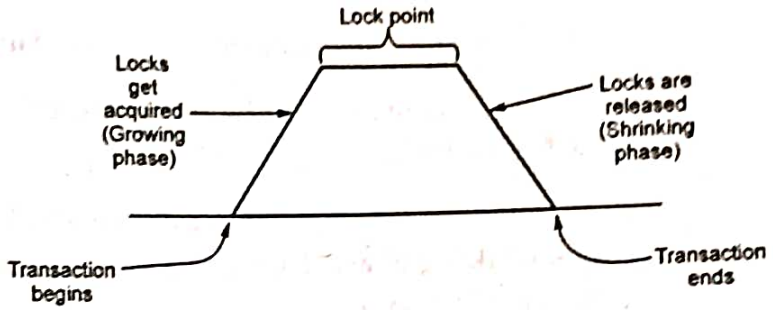
AU : May-14, 18, Dec.-16, Marks 7

- The two phase locking is a protocol in which there are two phases :

i) **Growing phase (Locking phase) :** It is a phase in which the transaction may obtain locks but does not release any lock.

ii) **Shrinking phase (Unlocking phase) :** It is a phase in which the transaction may release the locks but does not obtain any new lock.

- **Lock Point : The last lock position or first unlock position is called lock point. For example**



Lock(A)
Lock(B)
Lock(C)
....
...
...
Unlock(A)
Unlock(B)
Unlock(C)

Lock Point

For example –
Consider following transactions

T1	T2
Lock-X(A)	Lock-S(B)
Read(A)	Read(B)
A=A-50	Unlock-S(B)
Write(A)	
Lock-X(B)	
Unlock-X(A)	
B=B+100	Lock-S(A)
Write(B)	Read(A)
Unlock-X(B)	Unlock-S(A)

The important rule for being a two phase locking is - **All Lock operations precede all the unlock operations.**

In above transactions T1 is in two phase locking mode but transaction T2 is not in two phase locking. Because in T2, the Shared lock is acquired by data item B, then data item B is read and then the lock is released. Again the lock is acquired by data item A , then the data item A is read and the lock is then reloased. Thus we get lock-unlock-lock-unlock sequence. Clearly this is not possible in two phase locking.

Example 3.10.1 *Prove that two phase locking guarantees serializability.*

Solution:

- Serializability is mainly an issue of handling write operation. Because any inconsistency may only be created by **write** operation.
- Multiple reads on a database item can happen parallely.
- 2-Phase locking protocol restricts this unwanted read/write by applying **exclusive lock.**
- Moreover, when there is an **exclusive lock** on an item it will **only be released in shrinking phase.** Due to this restriction there is no chance of getting any inconsistent state.

The serializability using two phase locking can be understood with the help of following example

Consider two transactions

T_1	T_2
R(A)	
	R(A)
R(B)	
W(B)	

Step 1 : Now we will **apply two phase locking.** That means we will **apply locks in growing** and **shrinking** phase

T_1	T_2
Lock-S(A)	
R(A)	
	Lock-S(A)
	R(A)
Lock-X(B)	
R(B)	
W(B)	
Unlock-X(B)	
	Unlock-S(A)

Note that above schedule is serializable as it prevents interference between two transactions.

The serializability order can be obtained based on the **lock point**. The lock point is either last lock operation position or first unlock position in the transaction.

The last lock position is in T_1, then it is in T_2. Hence the serializability will be $T_1 \rightarrow T_2$ based on lock points. Hence The **serializability sequence** can be **R1(A);R2(A);R1(B);W1(B)**

Limitations of Two Phase Locking Protocol

The two phase locking protocol leads to two problems – deadlock and cascading roll back.

(1) Deadlock : The deadlock problem can not be solved by two phase locking. Deadlock is a situation in which when two or more transactions have got a lock and waiting for another locks currently held by one of the other transactions.

For example

T1	T2
Lock-X(A)	Lock-X(B)
Read(A)	Read(B)
A=A-50	B=B+100
Write(A)	Write(B)
Delayed, wait for T2 to release Lock on B	Delayed, wait for T1 to release Lock on A

(2) Cascading Rollback : Cascading rollback is a situation in which a single transaction failure leads to a series of transaction rollback. For example –