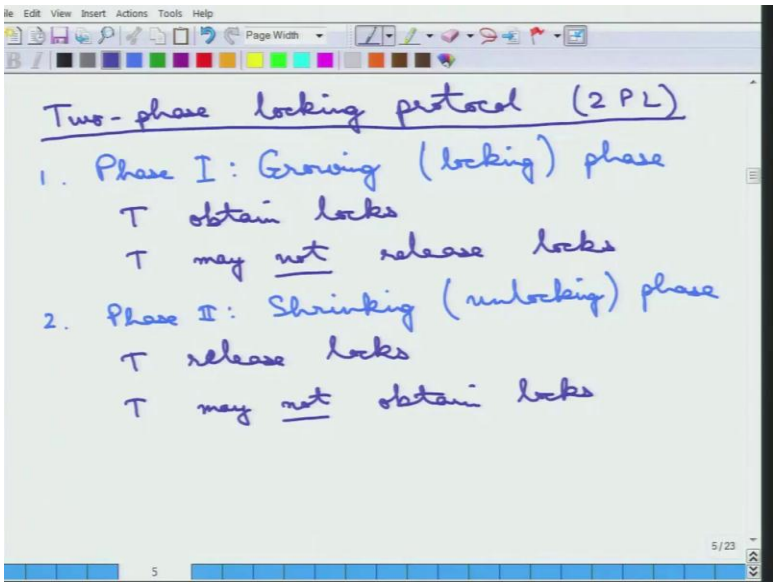
Fundamentals of Database Systems
Prof. Arnab Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 39
Schedules: Concurrency Control: Two-phase Locking Protocol

(Refer Slide Time: 00:12)



We are going to study a locking protocols and the first locking protocol, that we will study is called the Two Phase Locking Protocol or the 2 PL. The 2 PL is a more common term probably and this is the two phase locking protocols. So, there is why is it called two phase, we will see. There is, there are essentially two phase, in the phase 1, the phase 1 is also called the growing phase, growing or the locking phase. In this growing phase or phase 1, a transaction may only obtain locks.

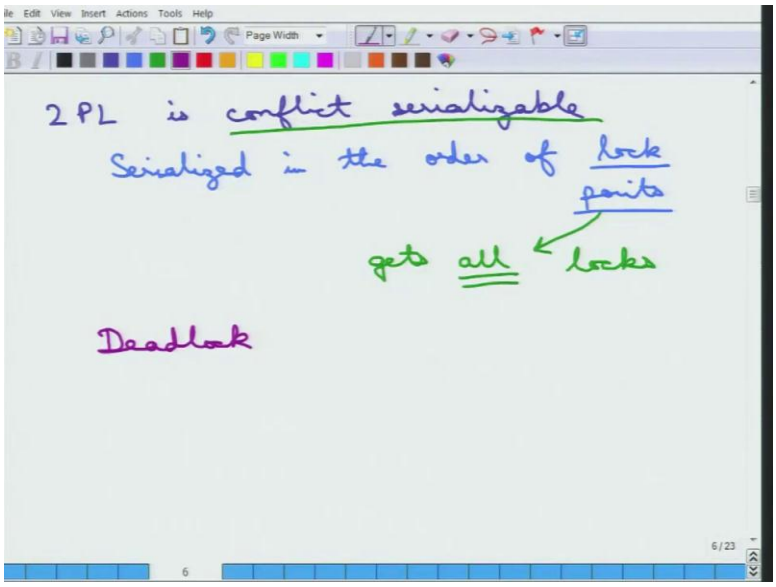
It may not release, it may not release locks, so, it can only get locks, it cannot release locks, so that is the growing phase and the complementary is the phase 2, which is the shrinking phase or the unlocking phase, whereas probably you can already guess, what is going to happen here is that, a transaction may only release locks, it may not obtain locks. So, these are the important thing, so what is essentially happening is that there are two distinct phases.

In the first phase, all the transactions can simply request for locks, it can only request and it cannot release any of the locks. In a second phase, the transactions can only release the

locks, but it cannot get any more locks. So, very simply the way to think of this is that, in the first phase all the transactions is to request for all the locks without releasing any one of them, because once it starts releasing the second phase, it cannot obtain anything.

So, essentially all the transactions first request for all the locks that it will need to complete the transactions. So, the locks on all the items that it will need, that is the growing phase or the locking phase and in the second phase, the unlocking phase it will just keep on releasing this things, it will not ask for any more locks. So, there are the two phase locking protocol.

(Refer Slide Time: 02:40)



Now, it can be shown, couple of things can be done is that this two phase locking protocol, 2 PL can be shown to be conflict serializable, this is very, very interesting. So, the two phase locking protocol is conflict serializable. Now, there can be proof of it, but intuitively it can be said that, that the two phase locking protocol, the schedules this is serialized, so this can be serialized. This is equivalent to a serial schedule, where the order of the transaction in the serial thing is serialized in the order of something called a lock points.

So, what is the lock point? The order of lock points, so a lock point is the time when a transaction gets all the locks. So, that is the thing, so this is called the lock point of a transaction. So, what I am trying to say is the lock point is the following is that... Suppose, transaction one wants the lock on item A and item B. So, once it gets item A, it

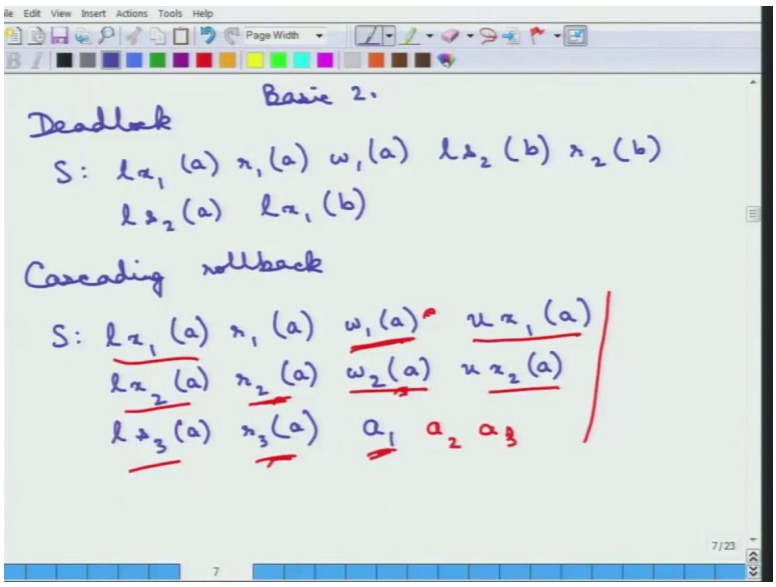
still not got hold of the lock point, once it is gets also the lock for item B, it will enter, it will then attain what is called the lock point.

Now, do note that in the two phase locking protocol, transaction one after it has got the lock of A, it will not release A, it will only, because it has not got ((Refer Time: 04:00)). First, try to get all the locks, which means it will get the lock of B before it will start releasing all the point. And after it has got the locks of all the items that it wants that is why it is called the lock point; that means, the transaction then holds the locks for everything that it needs.

So, that is the lock point of a transaction, though it is scheduled, then it can be serialized in the order of the lock points. So, whichever transaction gets into the lock point first, is the first transaction in the equivalent serial schedule of a two phase schedule. So, here an two phase schedule is conflict serializable, that can be proved, so this is the important part. However, conflict serializable protocols may still suffer from the problem of dead lock, because a transaction one wants the locks and as we saw the example actually; that is actually in a transact two phase locking protocol.

Because, none of the transaction has released any locks, so it was just for trying to get it and it can of course, enter into the deadlock. Because, transaction one gets the lock of A and it wants B, while transaction two has got the lock of B and it wants A. So, this is the basic two phase locking protocol and it may simply suffer from the problem of deadlock.

(Refer Slide Time: 05:14)



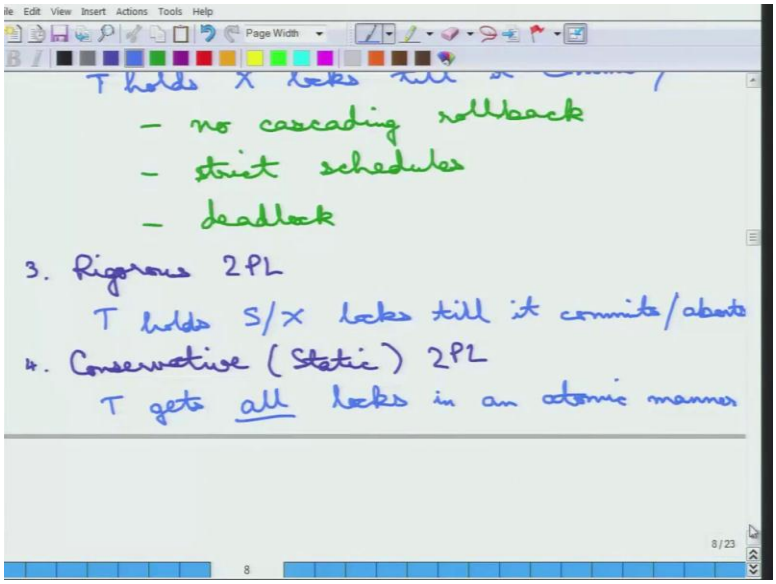
So, here is probably an example to, just to highlight the point deadlock, so and the schedule that it simplifies that it may suffer from the deadlock. So, this is a very simple example of how it can deadlock, so this point the system is dead lock. The two phase locking protocol can also suffer from the problem of cascading roll backs. If you, remember the cascading rollbacks is that when one transaction aborts, other transaction has to abort as a result of that and then, some other transaction may have to abort and so on and so forth, so that may also happen and here is the example for this.

So, let me first put on the example and on this point, what happens is that, transaction one aborts. So, if transaction one aborts, so everything has been going on fine, now transaction one has aborted, so; that means, this is all, this write operation is essentially invalid later. So, this should not go through; that means, this read is invalid later; that means, this write is also invalided later, so; that means, this read is also invalid later. So, essentially if a 1 happens, then transaction two must abort and transaction three must abort.

So, this is the cascading rollback, but note that this is in the two phase protocol, because for every transaction it gets the locks and then it releases it, it gets the locks and then it releases it, it gets the locks and then it releases it, it is getting all the locks that it wants. So, there is a, it is a simple way to see that this is still in two phase locking protocol, but it can suffer from this problem of cascading rollbacks.

So, it can suffer from both the... So, the basic two phase protocol, this is the basic two phase protocol that we have been studying. This can suffer from the problems of both deadlock and cascading rollbacks.

(Refer Slide Time: 06:48)



There are some variants of the two phase locking protocol. So, the first thing of course, we studied is the basic one. So, this is the basic, this is called the basic two phase locking protocol. This is the basic protocol that we have just studied, then there is something called a strict two phase locking protocol. So, a strict two phase locking protocol, the following thing happens. If the transaction must hold all it is exclusive locks till it commits.

So, it holds X locks till it commits or transaction must hold it is X locks, till it commits or abort. So, then let us see what this happens is that, so transaction holds the exclusive of... What it means is that, once it has got an exclusive lock for writing, essentially exclusive lock use for writing, it will hold on to it unless it commits or aborts. So, what is the effect; that means that suppose the transaction commits and then, it releases the locks; that means, that the lock that it has got did it job correctly of course, and the write operation is also successful.

So, anything that reads after this will be correct, the point is since it holds the lock, nobody, no other transaction can start reading that particular data item, unless it releases the locks and it has releases the lock only after it has committed; that means, only after it has make sure that the write that it has done is correct. Otherwise of course, it just aborts and then, releases it, but then no other transaction will have any effect on this, because it cannot read.

So, there is in no way any other transaction can read that particular data item, before its commits, before it releases the lock and by the time the transaction either committed or aborted, so it has essentially taken the decision. So, this means, that there is no cascading rollback in this, so there is there is no cascading rollback in this strict two phase locking protocol. Because, it cannot be it cannot, so the cascading rollback happens, because another transaction reads the particular data item that has been written, but not committed or aborting and that cannot happen, because it holds all the exclusive locks.

And it produces strict schedules, because nobody can even writes to the other data item, because it still holds the x locks before it commits or aborts. So, this case called it produce a strict schedule and that is why, you can see that why it is called a strict two phase locking protocol, because it produces the strict schedule. However, it must still dead lock, so the problem of dead lock is still remains and again it is easy to proof that it may dead lock and I inviting to think of a example to show that in a strict two phase locking protocol it may still dead lock fine.

So, that is the strict two phase locking protocol, then there is something called a rigours two phase locking protocol. So, here what it happens is a transaction holds all locks till it commits, so it, so holds S X both locks till it commits or aborts. So, rigours two phase locking protocol it of course, strict, because it holds on to S X locks and it is, it also holds on to the shared lock, it does not may releases the shared locks till it aborts or commits.

So, this is again of course, can be serialized in the order of this. So, this can be serialized simply in the order of the commits, but this is the same property it is holds on to this no cascading roll to this on and all of these things. So, cascading rollback and strict schedules it me do and it dead locks also. So, that fourth one in this is called the conservative or sometimes called the static, two phase locking protocol that this is an important thing transaction gets.

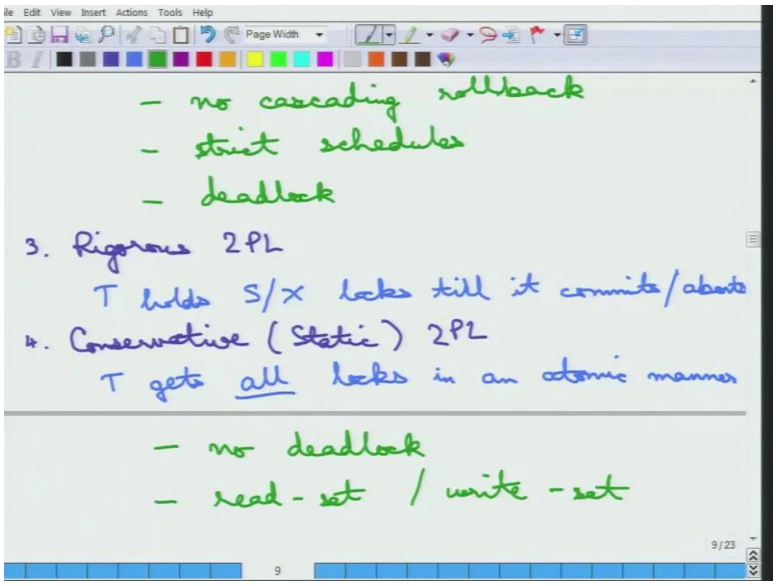
So, by the way rigours two phase locking protocol can also dead lock, because the simply, because one transaction ones another lock the transaction one wants a lock on B while transaction two wants a lock on A and transaction one holds the lock on A and transaction two holds the lock on B. So, nothing it, so only say is that it cannot it does

not release still it commit or abort, but does not say how to get it, so the dead lock may also happen.

In a conservative two phase protocol transaction gets all the locks in an atomic manner and now this requires a little bit more thought of what is this happen, so what does an atomic manner means. So, a transaction essentially at the beginning of did it declare as the intent of getting all the locks. So, it will say I want I exclusive lock on A and a shared lock on B and an exclusive lock on C and etcetera.

And it request all of those locks together, so both transaction one does it and transaction two does it and what the concurrency control protocol manager will do is that it either honours all the lock request together and it does not occur does not honouring it. So, what I am trying to say is that if a transaction one wants the locks A B and C, either it will get all the locks A, B and C or it will get none of the locks. So, its atomic that is why it is called all the locks in a atomic manner. So, it cannot happen that it will get the lock on A, but not a B and C and it will lock on A and B, but not on C, so and so forth.

(Refer Slide Time: 12:31)



So, very simply you can see that if this happens, then this cannot dead lock, what is the dead lock situation, the dead lock situation is, that there are two transaction holding locks that the other one wants. So, now, this cannot happen, because if transaction one wants the lock on B and holds on A it cannot happen it cannot want a lock on B while holding

A, because it either it has got both A and B or it has got none of them and the same thing for transaction, so, so there is no dead locks.

So, this is dead locks free and does not implementation detail are as a more detail of this things each transaction declares, what it called it is read set and write set and it gets the shared locks if it wants it can get shared locks and read set and the exclusive locks on write set. But, essential idea is that it either gets all the locks or it gets none of the locks, so that is why it cannot dead lock. So, that is the whole point of this conservative as static two phase locking protocol.

So, these are the four variance of the two phase locking protocol essentially just to summaries the two phase locking protocol at every transaction must first get all the locks and then, only it can stars releasing the locks.