Log Based Conquency Control

- In log based conquency control conflicting actions of differnt transcations in which the logs are opting and the log protocol extends this ordering on actions to transcations there by ensuring serializibility
- In optimistic control a time stamp ordering is imposed on transcations and validation checks that all conflicting actions occured in the same order timestamps can also be used in another way i.e each transcation can be assigned a timestamp at startup and we can ensure at execution time if action ai of transcation Ti conflicts with action aj of transcation Tj, ai occurs before aj if TS(Ti) < TS(Tj) if an action violates then the transcation is aborted and restarted

Multi Version Conquency Control

- This protocol represents yet another of using timestamps assigned at startup time to achieve serializibility the goal is to ensure that the transcation never has to wait to read a database object and the idea is to maintain sereval versions of each database object each with a write timestamp and let transcation Ti read the most recent version whose timestamp preceds TS(Ti)
- If transcation Ti wants to write an object we must ensure that the object has not already been read by some other transcation Tj such that TS(Ti) < TS(Tj) if the allowed Ti to write such an object it's change should be seen by Tj for serializibility but obousliy Tj which read the object at sometime in the past will not see Ti's change
- The drawbacks of this scheme are similar to those of timestamp conquency control and in addition their is the cost of maintaining versions on the other hand reads are never blocked which can be important for workloads dominatedd by transcations that only read values from the database