ADVANCED OPERATING SYSTEMS

MINU MARIA JOSEPH

Module 5 [7 Hours]

Database Systems: Requirements of a Database Operating System, Problem of Concurrency Control, Serializability, Basic Synchronization Primitives for Concurrency Control- Lock Based Algorithms-Static Locking, Two-Phase Locking (2PL), Time Stamp Based Algorithms- Basic Timestamp Ordering Algorithm, Thomas Write Rule (TWR), Multiversion Timestamp Ordering Algorithm, Conservative Timestamp Ordering Algorithm, Optimistic Algorithms.

Self-Study: Computer security and database security.

OPTIMISTIC ALGORITHMS

- Optimistic concurrency control algorithms are based on the assumption that conflicts among transactions are rare.
- In optimistic algorithms, no synchronization is performed when a transaction is executed, but at the end of the transaction's execution, a check is performed to determine if the transaction has conflicted with any other concurrently running transaction.
- In case of a conflict, the transaction is aborted, otherwise it is committed.
- When conflicts among transactions are rare, very few transactions need to be rolled back.
- Thus, transaction roll-backs can be effectively used as a concurrency control mechanism rather than locking.

Kung-Robinson Algorithm

- In their technique, a transaction always executes concurrently with other transactions without any synchronization check, but before its writes are written in the database , it is validated.
- In the validation phase, it is determined whether actions of the transaction have conflicted with those of any other transaction.
- If found in conflict, then the tentative writes of the transaction are discarded and the transaction is restarted. The basic algorithm is as follows:

Algorithm

- The execution of a transaction is divided into three phases: read phase, validation phase, and write phase.
- In the read phase, appropriate data objects are read, the intended computation of the transaction is done, and writes are made on a temporary storage.
- In the validation phase, it is checked if the writes made by the transaction violate the consistency of the database.
- If the check passes, then in the write phase, all the writes of the transaction are made to the database.

THE VALIDATION PHASE.

- In the validation phase of a transaction T, it is checked if a transaction exists that has its write phase after the beginning of the read phase of T, but before the validation phase of T, and which has its writeset intersected by the readset of T.
- If there exists such a transaction, a conflict occurs and T is restarted.
- Formally, each transaction is assigned a unique (monotonically increasing) sequence number after it passes the validation check and before its write phase starts.

Let t_s be the highest sequence number at the start of T and t_f be the highest sequence number at the beginning of its validation phase.

After the read phase of transaction T, the following algorithm is executed which consists of the validation phase and a possible write phase of T:

- <valid: = true;

  for t:= t_s + 1 to t_f do

  if (writeset[t] $\cap$readset[T] $\neq \phi$) then

  valid: = false;

  if valid then {write phase; increment counter,

  assign T a sequence number} >

- A read-only transaction does not have a write phase, but it still has to be vali¬ dated using the above validation algorithm.
- The optimistic approach is suitable only in environments where conflicts are unlikely to occur, as in a query dominant system.
- Schlageter proposed an improvement to the Kung-Robinson method wherein a read transaction always proceeds without validation check and thus without the risk of restarts ,
- In the Kung-Robinson method, read transactions are treated in the same way as update transactions, and thus, are subject to a validation check with the risk of restart