

Serializability & Concurrency Control Protocols

RAMNA SATTAR



Serializability

Serializability:

A serializable schedule of 'n' transactions is a parallel schedule which is equivalent to a serial schedule comprising of the same 'n' transactions.

A serializable schedule contains the correctness of serial schedule while ascertaining better CPU utilization of parallel schedule.

The serializability is straight forward. Local transaction is independent of each other; each concerns local data. In case of global transactions local sub transactions will be treated as different transactions.

The function of the concurrency controller is to generate serializable schedule.



Serializability

Equivalence of Schedules:

Equivalence of two schedules can be of the following types:

➤ **Result equivalence :**

Two schedules producing identical results are said to be result equivalent.

➤ **View equivalence :**

Two schedules that perform similar action in a similar manner are said to be view equivalent.

➤ **Conflict equivalence :**

Two schedules are said to be conflict equivalent if both contain the same set of transactions and has the same order of conflicting pairs of operations.



Concurrency Control Protocols

Locking Based Concurrency Control Protocols:

Locking-based concurrency control protocols use the concept of locking data items. A **lock** is a variable associated with a data item that determines whether read/write operations can be performed on that data item. Generally, a lock compatibility matrix is used which states whether a data item can be locked by two transactions at the same time.

Locking-based concurrency control systems can use either one-phase or two-phase locking protocols.



Concurrency Control Protocols

Timestamp Concurrency Control Algorithms:

Timestamp-based concurrency control algorithms use a transaction's timestamp to coordinate concurrent access to a data item to ensure serializability. A timestamp is a unique identifier given by DBMS to a transaction that represents the transaction's start time.

Some of timestamp based concurrency control algorithms are:

- **Basic timestamp ordering algorithm.**
- **Conservative timestamp ordering algorithm.**
- **Multiversion algorithm based upon timestamp ordering.**



Concurrency Control in Distributed Systems

Distributed Timestamp Concurrency Control

In a centralized system, timestamp of any transaction is determined by the physical clock reading. But, in a distributed system, any site's local physical/logical clock readings cannot be used as global timestamps, since they are not globally unique. So, a timestamp comprises of a combination of site ID and that site's clock reading.

For implementing timestamp ordering algorithms, each site has a scheduler that maintains a separate queue for each transaction manager. During transaction, a transaction manager sends a lock request to the site's scheduler. The scheduler puts the request to the corresponding queue in increasing timestamp order. Requests are processed from the front of the queues in the order of their timestamps, i.e. the oldest first



Concurrency Control in Distributed Systems

Distributed Two-phase Locking Algorithm

The basic principle of distributed two-phase locking is same as the basic two-phase locking protocol. However, in a distributed system there are sites designated as lock managers. A lock manager controls lock acquisition requests from transaction monitors. In order to enforce co-ordination between the lock managers in various sites, at least one site is given the authority to see all transactions and detect lock conflicts.

Depending upon the number of sites who can detect lock conflicts, distributed two-phase locking approaches can be of three types :

- **Centralized two-phase locking**
- **Primary copy two-phase locking**
- **Distributed two-phase locking .**



Concurrency Control in Distributed Systems

Distributed Two-phase Locking Algorithm

Centralized two-phase locking – In this approach, one site is designated as the central lock manager. All the sites in the environment know the location of the central lock manager and obtain lock from it during transactions.

Primary copy two-phase locking – In this approach, a number of sites are designated as lock control centers. Each of these sites has the responsibility of managing a defined set of locks. All the sites know which lock control center is responsible for managing lock of which data table/fragment item.

Distributed two-phase locking – In this approach, there are a number of lock managers, where each lock manager controls locks of data items stored at its local site. The location of the lock manager is based upon data distribution and replication.



THANK YOU

ANY QUERY???

