SANJIV.T.S
IT-B
2008101075

DBMS CAT-3 :-

8)b). Deadlock Handling :

Consider the following two transactions and history, with item X and transaction T_1 at site 1, and item Y and transaction T_2 at site 2 :

T_1 :	T_2
write(X)	write(X)
write(Y)	write(Y)

X-lock on X	
write(X)	
	X-lock on Y
	write(Y)
	wait for X-lock on X.
wait for X-lock on Y	

Result : deadlock which cannot be detected locally at either site.

Deadlock Detection :

In the centralized deadlock-detection approach, a global wait for graph is constructed and maintained in a single site ; the deadlock-detection coordinator.

① Real graph : Real, but unknown, state of the system.

② constructed graph : Approximation generated by the controller during the execution of its algorithm.

⊙ The global wait-for graph can be constructed when:

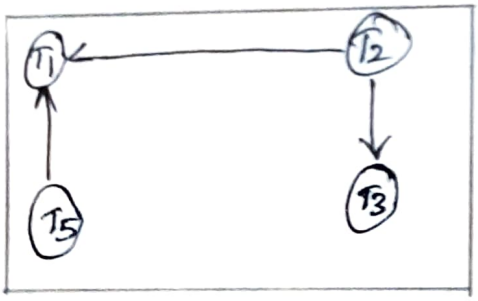
→ a new edge is inserted in or removed from one of the local wait-for graphs.

→ A number of changes have occurred in a local wait-for graph.

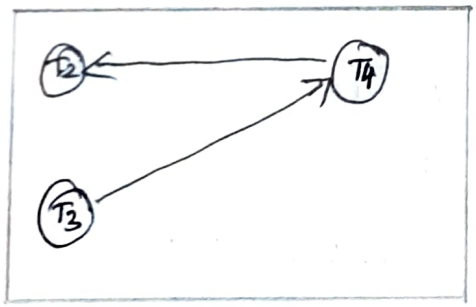
→ The coordinator needs to invoke cycle-detection.

⊙ If the coordinator finds a cycle, it selects a victim and notifies all sites. The sites roll back the victim transaction.

→. Local and Global Wait-for Graphs :

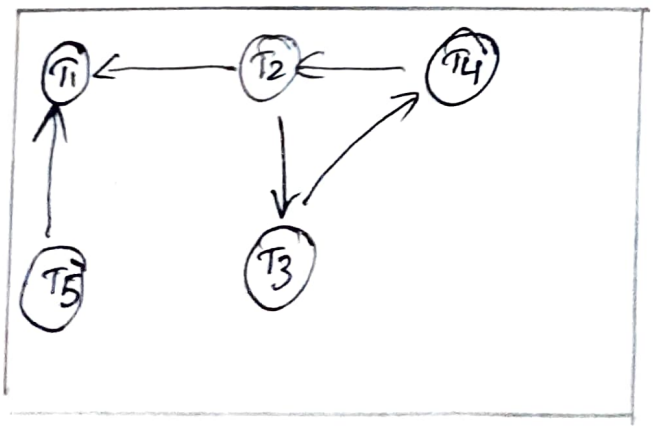


site S1



LOCAL

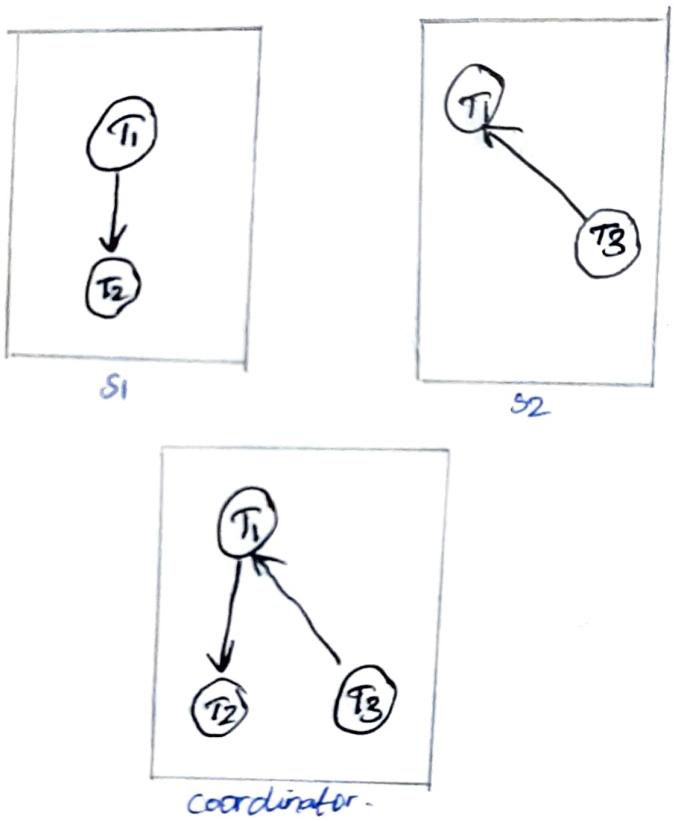
Site S2



Global.

Example wait-for Graph for False cycles :-

Initial state :



→. False Cycles :-

⊙ Suppose that starting from the state shown in figure.

→. T_2 releases resources at S_1

resulting in a message remove $T_1 \rightarrow T_2$ message from the Transaction Manager at site S_1 to the coordinator.

2. And then T_2 requests a resource held by T_3 at the site S_2.
resulting in a message insert $T_2 \rightarrow T_3$ from S_2 to the coordinator.

3. Suppose further that the insert message reaches before the delete message.
   ⊙ this can happen due to network delays.

4. The coordinator would then find a false cycle.
$$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_1$$

5. The false cycle above never existed in reality.

6. False cycles cannot occur if two-phase locking is used.

Distributed Deadlocks:

→ Unnecessary rollbacks may result
⊙ When deadlock has indeed occurred and a victim has been picked, and meanwhile one of the transactions was aborted for reasons unrelated to the deadlock.

⊙ Due to false cycles in the global wait-for graph, however, likelihood of false cycles is low.

→ In the distributed deadlock-detection approach, sites exchange wait-for information and check for deadlocks

⊙ Expensive and not used in practice.

7)a). Torrent protocl enables decentralization of its resources by making use of peer to peer network

A small torrent file is created to represent a file or a folder to be shared.

users can download the required files using a unique magnet link associated to each file on torrent.

→ In order to learn the Internet location of the peer which may be sharing pieces, the client connects to the trackers named in the torrent file, achieve a similar result through the use of distributed hash tables. The technology to ensure fault tolerance is fault-Tolerat services using Replicated State Machines.

⊙ Key requirement make a service fault tolerant
Eg: torrent, lock manager, etc.

⊙ State machines are a powerful approach to creating such services.

→ A state machine:

⊙ Has a stored state and receive inputs

⊙ Makes state transitions on each input and may output some results.

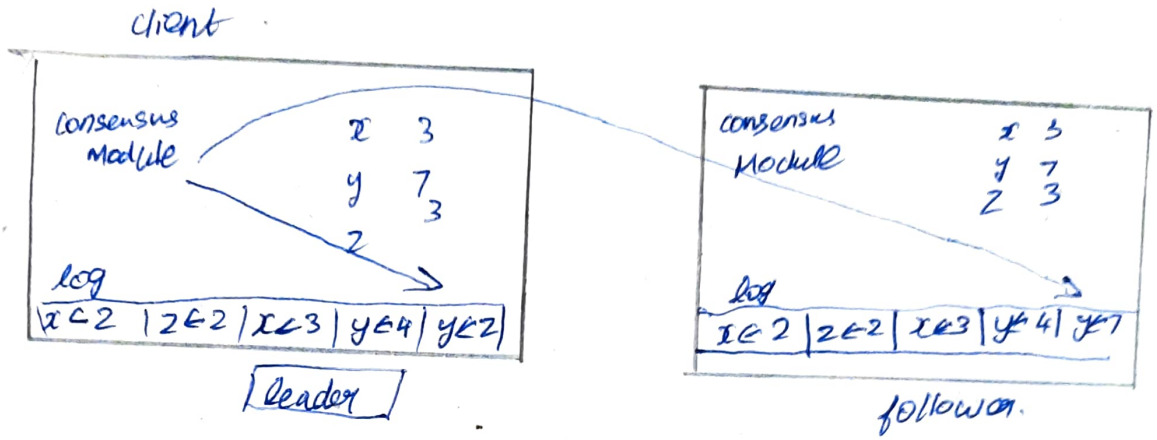
⊙ Transitions and outputs must be deterministic.

→ A replicated state machine is a state machine that is replicated on multiple nodes

All replicas must get exactly the same inputs.

Replicated log, state machine processor only committed inputs.

Even if some nodes fail, state and output can be obtained from other nodes.



uses of Replicated State Machines :

- Inputs can specify operations with parameters,
- But operations must be deterministic.
- Result of operation can be sent from any replica
- gets executed only when log record is committed in replicated log.
- usually sent from leader, which knows which part of log it is committed.

Eg: Fault tolerant key-value store :

- State : key-value storage state
- operations : get() and put() are first logged.
- operations executed when the log record is in committed state
- Even get() operations need to be processed via log.

Part – B :-

6)b). Challenges in Maintaining Data consistency :-

*) Data discrepancy occurs when the data in the target database deviates from the source database.

The extent to which the data deviates depends on various factors, some of which may be included and others unintended.

*). Even using products that replicates data reliably there remain potential causes of data discrepancy.

If the goal of target databases is to be strictly consistent with the source database, then it will need to put processes and policy in place to ensure this outcome.

*) Some of the potential causes.

Migration Errors :-

Different kinds of migration tools are employed to facilitate the initial load of the target database before replication

can begin. Difference in configuration for handling data by the migration tools and replication products can result in data discrepencies.

Lift and Shift workload to cloud:

Since the world is moving towards cloud, the lift and shift of database workload from on-premises to cloud is the need of today's IT world.

Difference in Source and Target:

Different including, locales, endianness or database versions can cause subtle discrepencies to happen during migration and replication.

Configuration errors:

Improper and misconfiguration of replication products can cause discrepencies. This type of discrepency doesn't show up in the replication logs, since from the replication products perspective it is forming as configured.

User error:

Often target databases are treated to offload query processing from the source database.

Requirements for managing Data consistency:

a) High speed, low impact data comparisons.

b) Support for heterogeneous databases.

c) Capability for handling large datas.

d) Minimally intrusive.

e) Data Security.

4) MySQL enables restrictions to be placed on reuse of previous passwords. To establish password-reuse policy globally, use the password history and password_reuse_internal system variables.

2) Database indexing. Hash tables may also be used as disk-based data structures and database indices although B-trees are more popular in these applications, hash tables are commonly used to reduce network traffic.

3). Advantages:

1. Data Retreivel
2. Editing

Dis Advantages.

1. Data Redundancy
2. Data inconsistency.

1). The mean time is $100000^2/(2*100)$.

5). Map database system is used because it is designed efficiently to store and recall spacial information.