CAT - III

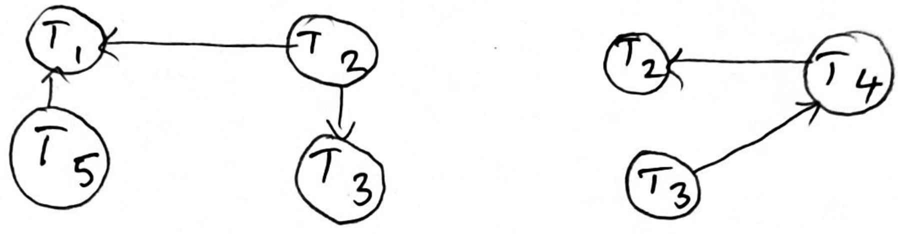
Name : G. Navina
Branch : IT - B
Regno : 212720080l054
Subcode : IT18305
Subject name : Database system

Part - C

8b) → The deadlock - prevention and deadlock-detection algorithm can be used in distrubed system, provided that medification are made.

→ Deadlock prevention may result in unnecessary waiting and rollback. If we allow deadlocks to occur and rely on deadlock detection the main problem in a disturbed system is deciding how to maintain wait-for-graph.

→ Common technique for dealing with this ~~to~~ issue require the each site keep a local wait for graph the nodes of the graph correspond to all the transaction that are currently either holding or requesting any of the items local to that site



Eg:- The above system consisting of 2 sites each maintaining its local wait-for graph.

Note that transaction T_2 and T_3 appear in both graphs, indicating that the transaction have ~~requested~~ ~~T_2~~ c requested items at both sites

→ These local wait for graph are constructed in the usual manner for local transaction and dataitems. When Transaction T_i on site S_1 needs a resource in site S_2 it send a request message to site S_2. If the resource is ~~both~~ held by transaction T_j. then system inserts an edge $T_i \rightarrow T_j$ in the local wait for graph site S_2 .

-> Each wait-for graph is acyclic ; nevertheless a deadlock exist in the system because the union of the local wait-for graphs contains a cycle .

-> In the centralized deadlock detection approach the system constructs and maintains a global wait for graph in a single site, the deadlock detection coordinator. Since there is communication delay in the system we must distinguish between the 2 types of wait for graph .

-> The global wait-for graph can be reconstructed or updated under these conditions :-

* whenever a new edge is inserted in or removed from one of the local wait for graph
* Periodically when a number of changes have occured in a local wait for graph .

~~&~~ whenever the coordinator needs to invoke the cycle-detection algorithm.

→ When the coordinator invokes the deadlock detection algorithm, it searches its global graph. If it finds a cycle, it selects a victim to be rolled back. The coordinator must notify all the sites that a particular transaction has been selected as victim and sites, in turn roll back.

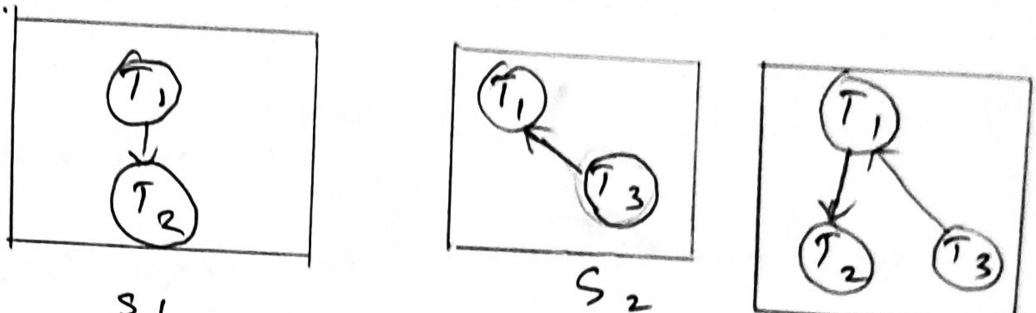
→ False cycles :-

→ It exist in the global wait for graph. Suppose that ~~Sat~~ T_2 releases the resource that it is holding in site S_1 resulting in the deletion of edge $T_1 \rightarrow T_2$ in S_1.

→ Transaction T_2 then requests a resource held by T_3 at the ~~ser~~ site S_2 resulting in the addition of edge $T_2 \rightarrow T_3$ in S_2

→ If the insert $T_2 \rightarrow T_3$ message from S_2 arrives before the remove $T_1 \rightarrow T_2$ message from S_1

the coordinator may discover the false cycle $T_1 \rightarrow T_2 \rightarrow T_3$ after the insert. Deadlock recovery may be initiated although no deadlock has occured.



S1 S2 coordinator

Part-B

6(b) Challenges in maintaining Data consistency :-

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

Some potential cause of data discrepancy :-

→ Migration error :

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

→ Lift & shift workload to cloud.

→ Difference in source and target ie Difference in source and target database configuration.

→ Instantiation error, before migration or replication
→ ~~configuration error~~ can begin the target database will need to be instantiated with correct schema and constraints, failure to do so will lead to source and target being out of sync

→ Configuration error improper and unintended configuration of replication products can cause discrepancies.

→ Gaps in replication

→ Replication latency

→ User error, application error.

Requirements for managing data consistency :-

→ High speed low impact data comparisons

→ support for heterogeneous database

→ capability for handling large data volumes

→ Minimal intrusive

→ Suppor for live database with constantly changing data

→ Flexible options for managing data comparisons.

Part-c

7(a) Fault - Tolerant services using replication state machines

→ Key requirement to make a service fault tolerant

eg :- lock manager, key value storage system.

→ State machines are a powerful approach to creating such service

→ A state machines

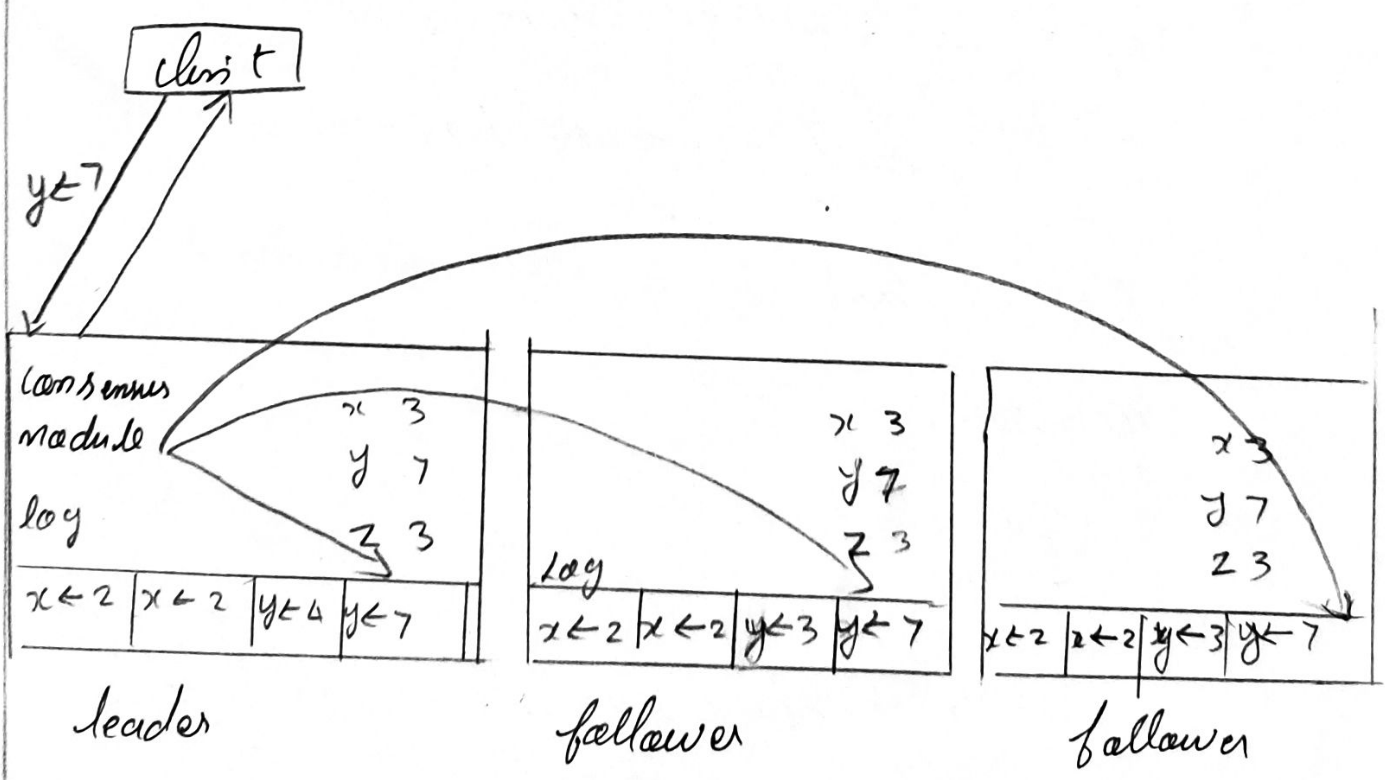
* has a stored state, and receives inputs
* Makes state transitions on each input and may output some results.

* Transitions and output must be deterministic

-> A replicated state machine is a state machine that is replicated on multiple nodes.

* All replicas must get exactly the same input
* Replicated log: state machine processes only committed inputs.
* Even if some of the nodes fail state and output can be obtained from other nodes.

Replicated state machine :-



→ Replicated state machine based on replicated log.

→ Example command assign values to variables

Use of Replicated state machines:-

→ ~~Inputs can sp~~

→ Replicated state machines can be used to implement wide variety of services.

* Input can specify operations with parameter
* ~~B~~ Operations must be deterministic
* Result of operation can be sent from any replica.

example :- Fault - tolerant lock manager

* State : lock table
* operation : lock requests and lock releases
* output : grant, or rollback requests on deadlock
* Centralized implementation is made fault tolerant by simply running it on a replicated state machine.

→ Fault tolerant key value store

* State : key - value storage state

→ operation : get () and put () are first logged

→ Google spanner uses replicated state machine to implement key-value store.

Part - A

3] Advantages of storing multiple relation in single file

* complex structures can be implemented through the DBMS, thus increasing performance

~~Dis~~ Disadvantages :-

* Increases the size and complexity of the DBMS.

5] Map database management system are software programs designed to efficiently store and recall spatial information. Used in localization and ag ~~automotive~~ localization especiall in automotive applications.

1] Mean time to failure of disk A is 1,00,000 hour time to repair is 10 hours. gives mean time to data loss of 500×10^6 hours or 57000 years for a mirrored pairs of disks.