| | |
|---|---|
| IT18305- Database Systems<br><br>CAT-III | Name: A. Vijayalakshmi<br>class: IT-B<br>Reg no: 212720080l102 |

PART-C

8.)b.) Deadlock handling:-

Consider the following two transactions and history, with item x & transaction $T_1$ at site1, & item Y and transaction $T_2$ at site 2.

| $T_1$ : Write (x) Write (Y) | $T_2$ : Write (x) Write (Y) |
|---|---|
| x-lock on x<br>Write (x) | |
| | x-lock on Y<br>Write (Y)<br>Wait for x-lock on x |
| Wait for x-lock on Y | |

Result: Deadlock which cannot be detected locally at either side.

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

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

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

The Global wait-for graph can be constructed when

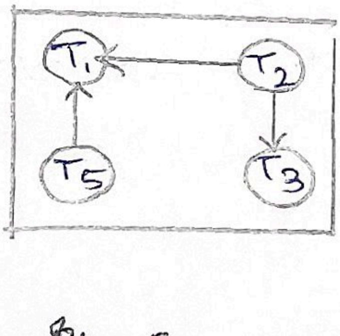
* a new edge is inserted or removed from one of the local wait-for graphs

* a number of changes have occured in a local wait-for graph

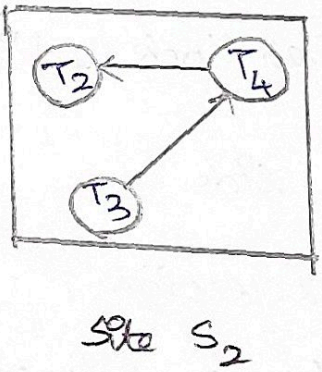
* The coordinator needs to invoke cycle-detection.

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

Local and Global wait-for Graphs

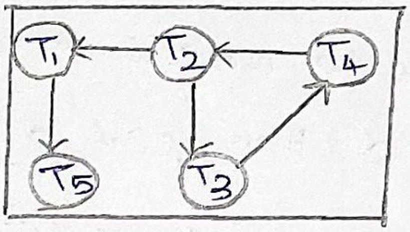



Site $S_1$




Site $S_2$

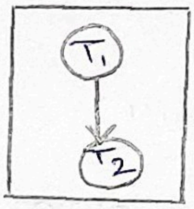
Local



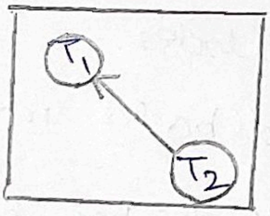
Global.

Example wait-for graph for false cycles :-

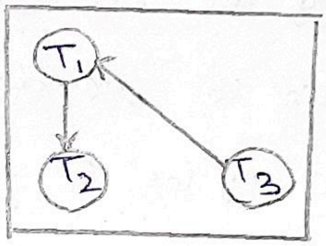
Initial state :-



$S_1$



$S_2$



coordinator

False cycles :-

* $T_2$ releases resourses at $S_1$

⇒ resulting in message remove $T_1 \rightarrow T_2$ message from the transaction manager at site $S_1$ to coordinator)

* And $T_2$ requests a resource held by $T_3$ at site $S_2$

⇒ resulting in a message insert $T_2 \rightarrow T_3$ from $S_2$ to coordinator

* Suppose further that insert message reaches before

the delete message

⇒ This happen due to network delays

* The coordinator would then find a false cycle.

* The false cycle above never existed in reality.

* false-cycles cannot occur if two-phase locking is used.

## Distributed Deadlocks:-

- Unnecessary rollbacks may result:

⇒ When deadlock has indeed occurred & a victim has been picked & one of the transaction was aborted for reasons unrelated to deadlock.

⇒ Due to false cycles in global-wait-for cycle graph; however likelihood of false cycles is low.

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

⇒ Expensive & not used in practice.

---

PART-B

6.) b) Before we discuss the requirements for sol that helps manage data consistency across databases, we need to understand common causes of data discrepancies in an Enterprice.

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

Some of the Potential causes of data discrepancy are described below.

Mitigation 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 discrepancies

Insantiation Errors:-

* Before the migration or replication can begin, the target database will need to be insantiated with the correct schema & constraints.

* Failure to do so will result in the source and target being out of sync.

Configuration Errors:-

* Improper and unintended configuration of replication products can result in discrepancy.

* It doesn't show replication logs properly.

User errors:-

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

* This enables rich operational reporting without impacting the application running on source database

Requirements for maintaining Data consistency:-

* High speed, low impact data comparison

* Minimally intrusive.

* Capability to identify data inconsistency

* Zero downtime of source & target systems

* Data security.

* Support to heterogenous databases

Oracle GoldenGate Veridata:-

* It is an easy to use, high performance & minimally invasive product that helps manage data consistency across an enterprice.

* It can be used in multiple senarios to ensure data consistency, including comparing database with large data volumes or comparing databases with constantly changing data

PART-C

7.a.) Fault-Tolerant Services using Replicated State machines

⇒ Key requirement : make a service fault tolerant

- eg : lock manager, key-value storage system, ...

⇒ State machines are powerful approach to creating such services.

⇒ A state machine

- Has a stored state, and received inputs
- makes a state transitions on each input, & may output some results.

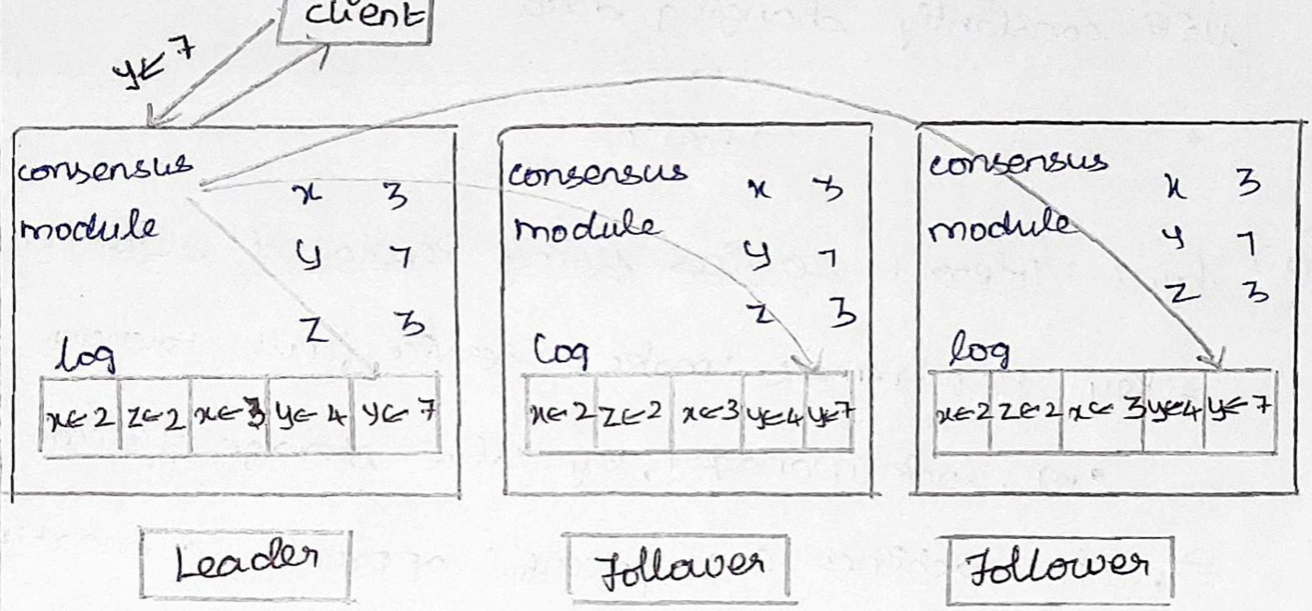
* Transaction & output must be deterministic.

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

- All replicas must get exactly same inputs
- Even if some of the nodes fails, state & output can be obtained from other nodes.

## Replicated State machine:-

* Replicated state machine based on replicated log
* Eg : commands assign values to Variables.




## Uses of Replicated state machines.

- Replicated state machines can be used to implement wide variety of services
  - ⇒ Inputs can specify operations with parameters
  - ⇒ But operations must be deterministic
  - ⇒ Results of operation can be sent from any replica.
    - * Gets executed only when log record is commited in replicated log.
    - * Usually sent from leader, which knows which part of log is committed.

Ex: Fault - tolerant lock manager

* State : lock table

* Operations : lock requests & lock releases

* Output : grant or roll back req on deadlock

* Centralized implementation is made fault tolerant by simply running it on a replicated state machines

Fault tolerant Key-Value store

⇒ State : Key - Value storage state

⇒ Operations: get () & put () are first logged.

* Operations executed when the log record is in commited state

* Note: even get () operations needs to be processed via log.

⇒ Google spanner uses replicated state machine to implement Key - Value store

* Data is partitioned, & each partition is replicated across multiple nodes

* Replicas of partition form a paxos group with one node as leader

* Operations initiated at leader & replicated to other nodes.

## PART-A

1.) For the two disk mirrored case, we assume A disk and B disc. In order to lose data, A & B need to be failed at the same time. If A is already failed & within 100,000 hrs B disk will fail, then data will be lost. The other case is B is already failed & within 100,000 hrs A will fail and then data will be lost.

For the first case, A disk is failed for 100 hrs every 100,000 hrs so in order to make B to fail, it will need 100,000^2 / 100 hrs. Because the other case, the time is reduced to 100,000^2/(2×100)

3.) Advantage:-

i) Data retrieval:

* computer-based systems provide enhanced data retrieval techniques to retrieve data stored in files in easy and efficient way.

ii) Editing:

* It is easy to edit any information stored in

computers in form of files

*Specific application prgs or editing software can be used for this purpose.

Disadvantages:-

(i) Data Redundancy :

It is possible that same information may be duplicated in diff files. This leads to data redundancy results in memory wastage

ii) Data inconsistency :-

Because of data redundancy, it is possible that data may not be in consistent state.

2.) Database indexing. Hash tables may also be used as disk-based data structures & database indices (such as in dbm) although B-trees are more popular in these applications. In multi-node database systems, hash tables are commonly used to distribute rows among nodes, reducing network traffic for hash joins.

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

5.) Map database management systems are software programs designed to efficiently store and recall spatial information. They are widely used in localization and navigation, especially in automotive applications. Moreover, they are playing an increasingly important role in the emerging areas of Location-based services, active safety functions & advanced driver assistance systems. Common to these functions is requirement for an on-board map database that contains information describing road network.

When designed well, a map database enables the rapid indexing & lookup of large amount of geographic data.