11/2/22

DATABASE SYSTEMS

CAT-3 EXAM.

SWETHA·S
IT-'B'
DBMS
IT18305
III SEM
2127200801092

PART-C

16 marks:-

8 b) **DEADLOCK HANDLING:-**

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 distributed system is deciding how to maintain the wait-for graph.

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 : write (X) write (Y)	T_2 : write (X) 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 co-ordinator.

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

    * Constructed graph:- Approximation generated by the controller during the execution of its algorithm.

- The global wait-for graph can be constructed from:-

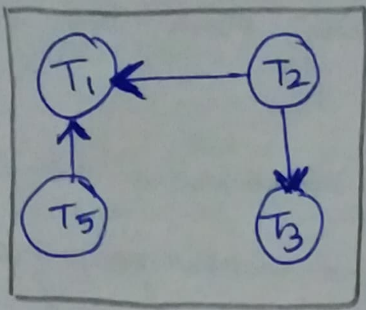
    * a new edge is inserted in or removed from one of the local wait-for graphs

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

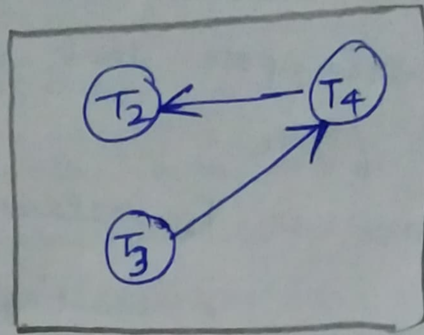
    * the co-ordinator needs to invoke cycle-detection

- If the co-ordinator finds a cycle, it selects a victim and notifies all sites. The sites roll back the victim transaction.

LOCAL AND GLOBAL WAIT-FOR GRAPHS:-

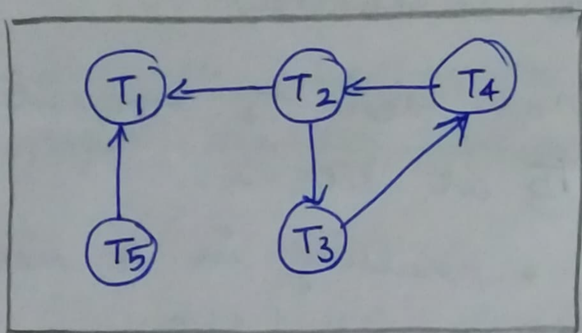


site S1



site S2

LOCAL

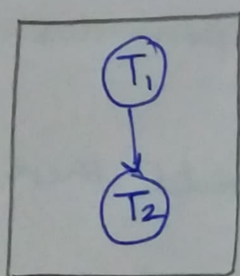


Global

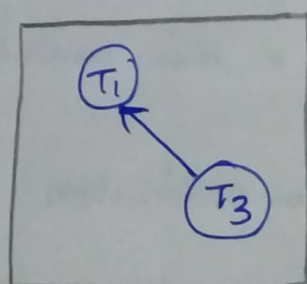
Example:-

WAIT-FOR GRAPHS FOR FALSE CYCLES:-

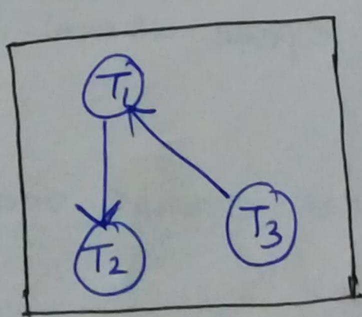
Initial state:-



S1



S2



Co-ordinator

FALSE CYCLES:-

*Suppose that starting from the state shown in figure :-

1. T_2 releases resources at S_1

   - resulting in a message remove $T_1 \to T_2$ message from the Transaction Manager at site S_1 (to the co-ordinator)

2. And then T_2 requests a resource held by T_3 at site S_2.

   - resulting in a message insert $T_2 \to T_3$ from S_2 to the co-ordinator

* Suppose further that the insert message reaches before the <u>delete</u> message

   - this can happen due to network delays

*The co-ordinator would then find a false cycle

$$T_1 \to T_2 \to T_3 \to T_1.$$

*The false cycle above <u>never existed</u> in reality.

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

Distributed deadlocks:-

* Unnecessary rollbacks may result

→ When deadlock has indeed occured 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 a 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

→ Complicated.//

PART-B.

8 marks:-

6b) CHALLENGES IN MAINTAINING DATA CONSISTENCY:-

Data discrepancy occurs when the data in the target database derivates from the source database. The extent to which the data depends on various factors, some of which may be intended and others unintended. Some of the potential causes for data discrepancy are described below:-

Migration errors:-

Different kinds of migration tools are employed to facilitate the initial load of the target databases before replication can begin.

Differences in configuration for handling data by the migration tools and replication products can result in data ~~dispen~~ discrepancies.

Lift and shift workload to cloud:-

Since, the world is moving towards. clouds, the lift and shift of database workload from on-premises to cloud is the need of today's IT world. Oracle Golden gate helps moving the work load, the data-consistency access on-peremises and cloud data centers and data compliance assurance is the top challenge in hybrid cloud scenario.

Differences in ~~Source~~ source and Target:-

Differences in source and target database configuration, for example: different encodings, locales, endianness or database versions can cause subtle discrepancies to happen during migration.

Instantiation errors:-

Before migration or replication can begin, the target database will need to be instantiated with the correct schema and constraints.

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

Configuration error:-

Improper and unintended configuration of replication products cause discrepancies. This type of discrepancy doesn't show up in the replication logs, since from the replication products perspective, it is performing as configured. This may prevent QA tests from detecting the issues as well

User errors:-

Often target databases are created to offload query processing from the source database. This enables rich operational reporting without impacting the applications on the source database.

The challenges are addressed by "CAP theorem".

CAP THEOREM:-

We need to know the three properties of a system :-

* Consistency
* Availability
* Partitions.

*CAP theorem says that "You can have atmost two of these three properties of any system".

* Very large systems will partition at some point.

* Choose any one of consistency or availability.

* Traditional database choose consistency

* Many web applications choose availability.

* Except for specific parts such as Order processing.

Latency is another factor.

Many applications choose to serve potentially stable data to reduce latency.

PART-C

7a 16 marks:—

7a) * Torrent protocol enables de-centralization of its resources by making use of peer-to-peer network.

* A small torrent file is created to represent a file or 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 any peers which may be sharing pieces, the client connects to the trackers named in the torrent file; acheiving a similar result through the use of distributed hash tables.

The technology to ensure fault-tolerance is Fault-tolerant services by Replicated

State machines :-

* Key requirement : Make a service fault tolerant.

Example : Torrent , lock manager , etc.

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

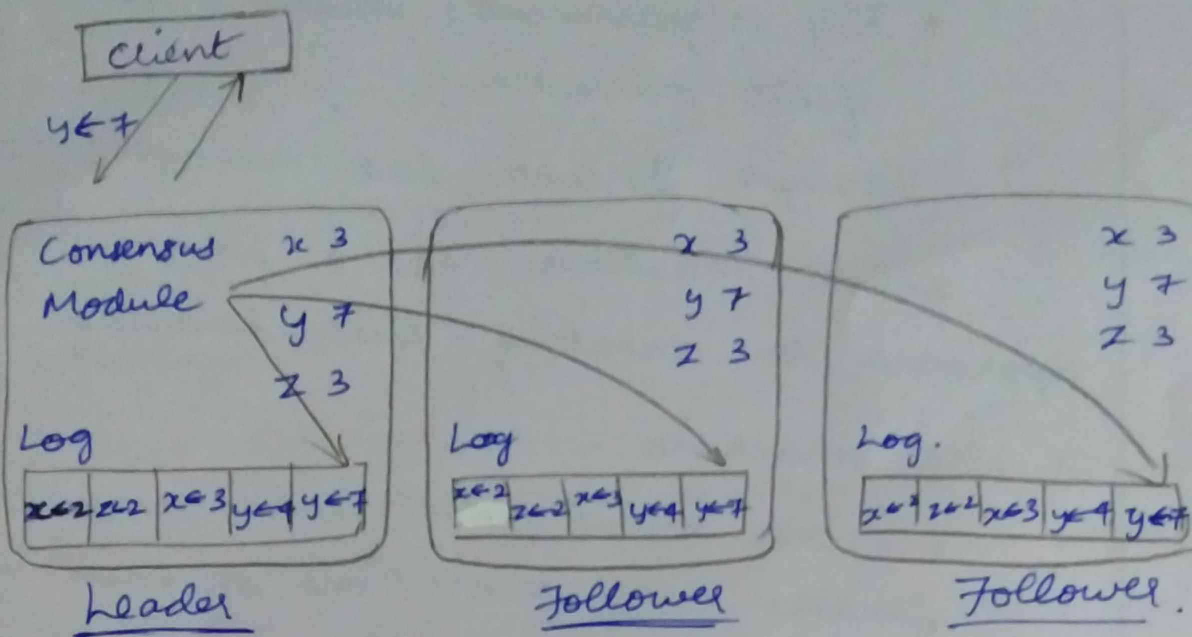
A state machine :-

- has a stored state and received 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 processes only committed inputs!
- Even if some of the nodes fail, state and output can be obtained from other nodes

Replicated State Machine :-



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
- Result of operation can be sent from any replica.
  - → Usually gets executed only when log record is committed in replicated log
  - → Usually sent from leader, which knows which part of a log is committed

Example :- Fault-tolerant lock manager.

- State : lock table
- Operations : 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.

PART-A.

2 marks :-

3. Advantages :-

1. Data Retrieval :-

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

2 Editing :-

• It is easy to edit any information stored in computers in form of files.

• Special application programs or editing software can be used for this purpose.

Disadvantages :-

1. Data Redundancy :-

It is possible ~~to~~ that the same information may be duplicated in different files, this leads to data redundancy which results in memory wastage.

2. Data Inconsistency :-

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

1. For the two disk mirrored case, we assume A disk and B disk. In order to lose data, A and B need to be failed at the same time.

If A is already failed and within 1,00,000 hours, B disk will fail, then data will be lost.

The other case is B is already failed and within 1,00,000 hours, A will fail and then data will be lost.

For the first case, A disk is failed for 100 hours every 1,00,000 hours, so in order to make B to fail, it will need

$$\frac{1,00,000^2}{100} \text{ hrs}$$

Because the other case, the time is reduced to $\frac{1,00,000^2}{(2 \times 100)}$

5. SQL server supports storing and querying of ~~geop~~ geospatial data, that is, location data referrenced to the earth.

Common models of these data are the planar and geodetic co-ordinate systems.

The main distinction between these two systems is that the latter takes into account the curvature of the earth.

SQL Server supports geometry and geography which correspond to the planar and geodetic models

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.

In multi-node database systems, hash tables are commonly used to distribute rows amongst nodes, reducing networks 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 and password_reuse_internal system variables