CAT-3

2127200801094
Thanigaivelan A
INT-"B"

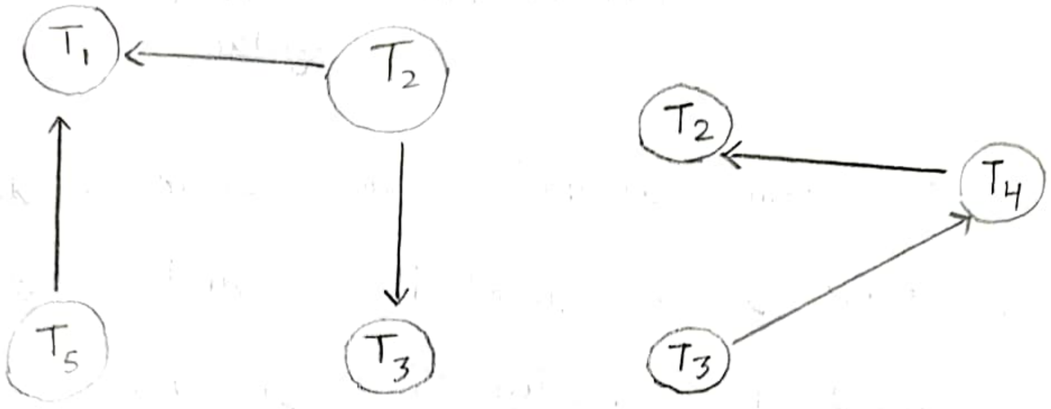
IT18305 - Database management system

8 (b) The deadlock - prevention and deadlock - detection algorithms can be used in a distributed system, provided that modification are made.

For example, we can use the tree protocol by defining a global tree among the system data items. similarly, the timestamp - ordering approach could be directly applied to a distributed environment.

Deadlock prevention may result in unnecessary waiting and rollback furthermore, certain deadlock -prevention technique may requires more site to be involved in the execution of a transaction than would otherwise be the case.

If we allow deadlocks to occur and reply on deadlock detection, the main problem in a distributed system is deciding how to maintain the wait for graphs.



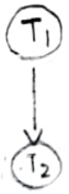
common techniques for dealing with this issue require that each site keep a local wait-for graph. The node of the graph correspond to all the transactions (local as well as nonlocal) that are currently either holding or requesting any of the items local to that site.

Example: depicts a system consisting of two sites each maintaining its local wait-for graph. note the transactions T_2 and T_3 appear in both graphs, indicating that the transactions have requested items as both sites.

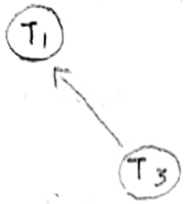
when a transaction T_i on site S_1 needs a resource in site S_2, it sends a request message to site S_2. If the resource is held by transaction

T_j, the system insert an edges $T_i \rightarrow T_j$ in the local wait-for graph of site S_2.

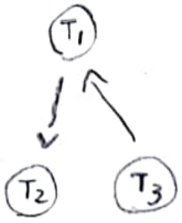
In the centralized deadlock detection approach the system constructs and maintains a global wait-for graph in a single site: the deadlock-detection coordinator. since there is communication delay in the system, we must distinguish between two types of wait-for graphs. The real graph describes the real but unknown state of the system at any instance in time, as would be seen by an omniscient observer.



S_1



S_2



coordinator

The global wait- for graph can be reconstructed or updated under the conditions.

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

* periodically, when a number of changes have occurred in a local wait for graph.

* whenever the coordinator needs to invoke the cycle- detection algorithm.

False cycles exist in the global wait for graph. As an illustration, consider a snapshot of the system represented by the local wait for graph suppose that T_2 redease the resource that is holding in site S_1, resulting in the delection of the edge $T_1 \rightarrow T_2$ in S_1. Transaction T_2 then request a resource held by T_3 at site S_2, resulting in the addition of the edges $T_2 \rightarrow T_3$ 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, then coordinator may discover the false cycle $T_1 \rightarrow T_2 \rightarrow T_3$ after the insert.

A deadlock has indeed occured and a victim has been picked, while one of the transactions was aborted for reasons unrelated to the deadlock. For example, suppose that site S_1 decides to abort T_2. At the same time, the cordinator has discovered a cycle and has picked T_3 as a victim. Both T_2 and T_3 are now rolled back, although only T_2 needed to be rolled back.

part - B

6 (b)

Executive overview

The requirement for high data availability and the need to access data or near 24/7/365 without performance degradation and service interruption has created the need for having redundant distributed copies of the data. However in today's complex IT environment, where data increases in light speed, maintaining data consistency across distributed copies of data is challenging and the possibility of data discrepancy is an unfortunate reality.

oracle goldengate veridata provides an easy-to-use yet powerful solution for identifying out of synch data before it negatively impacts the business. Deployed together with the oracle Groldengate real-time data replication product or separately, oracle goldengate veridata ensures data consistency is maintained across databases.

challenges in maintaining Data consistency

before we discuss the requirements for the solution that helps manage data consistency across database, we need to understand the common causes of data discrepancies in an enterprise

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 intended and others unintended

Some of the potential causes of data Discrepancy are described in the following sections.

migration error

Different kind of migration tools are empl
to faciliate the initial load of the target databa
before replication can begin Differences in configurato
for handling data by the migration tools and replica
products can result in data discrepancies.

For example. a migration tool may use "?" and the replication product may use "Null" when the value of the column is known.

lift & shift worload to cloud

since the world is moving towards cloud, the lift & shift of database workload from on premises to cloud is the need of todays IT world. oracle goldengate helps moving the work load, the data consistency across on-premises and cloud data

Difference in source and Target:

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

Instantiation errors:-

Before migration of replication can begin the target database will need to be instantiated with the correct schema and constraints. Failure to do so will result in the same and target being out of sync.

Configuration Errors:-

Improper and unintended configuration of replication products can cause discrepancies. This type of discrepancy doesn't show up in the replication logs, since from the replication product's perpective it is forming as configured.

Replication latency:-

With asynchronous replication, there will be a short lag between changes to the service database and delivery to those changes to the target. Failure to meet the maximum latency requirement, however can potentially violate service level agreement levels or data compliance requirements.

user errors:.

often target databases are treated to offload query processing from the source database. This enable rich operational reporting without impacting the application running on the source database.

Requirements for managing Data consistency:

* high speed, low impact data comparisons
* support for heterogeneous databases
* capability for handling large data volumes flexible options for managing data comparison
* minimally intrusive
* support for live databases with constantly changing data
* comparison of only changed data in continous replication.
* comparison of huge table through automated and manual partioning.

* Data comparison reports for auditing purp

* Data security.

* Easy to use, understand, Configure, deploy and diagnose.

7 a)

Fault- tolerant services using replicated state machine

- Key requirement: make a service fault tolerant

  E.g: lock manager, key-value system

- state machines are a powerful approach to creating such services

- A state machine

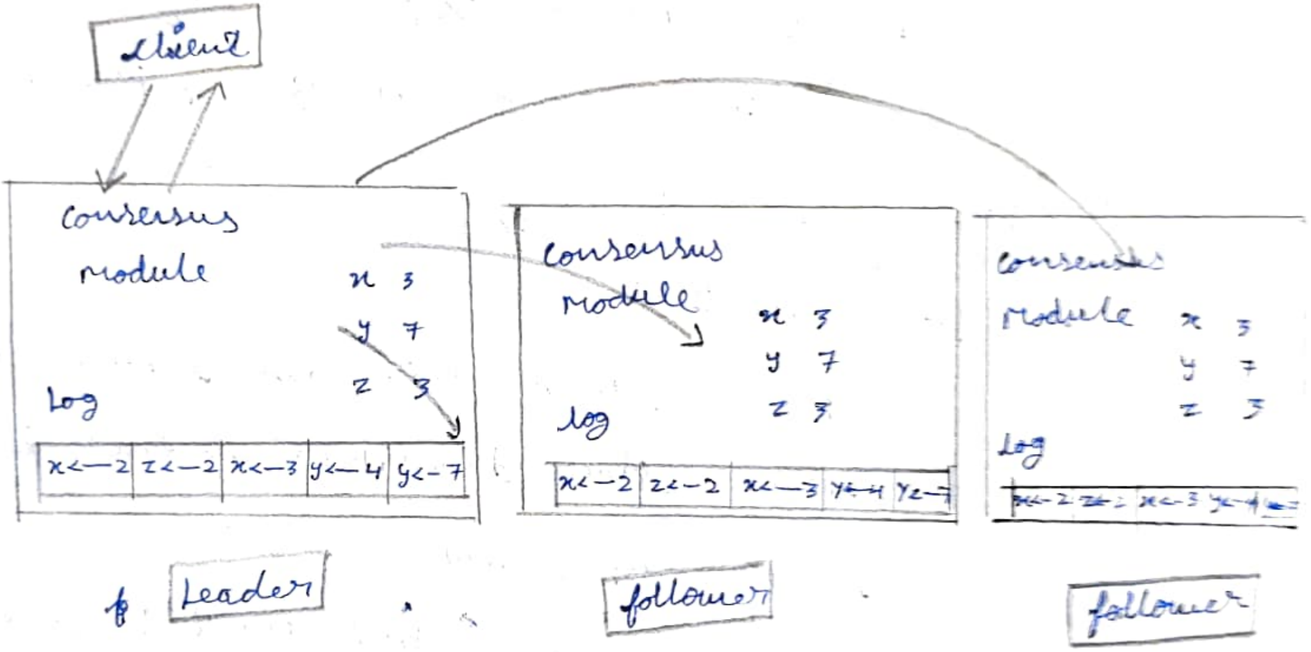
  - Has a stored state, and receives input
  - makes state transitions on each input and may output some results
  - Transition and output must be deterministic

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

  - All replicas must get exactly the same 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

* Replicated state machine based on replicated log
* Example commands assign values to variables



Leader declares log record committed after it is replicated at a majority of nodes, update of state machine at each replica happens only after log record has been committed.

uses of replicated state machines

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

- input can specify operations with paramete

- But operation must be deterministic

- Result of operation can be sent from any replica

creds executed only when log records is committed in replicated log.

usually sent from leader, which knows which part of log is committed

Example: fault-tolerant lock manager

* state: lock table
* operations: lock requests and lock releas
* output: grant, or rollback requests on deadlock

* centralized implementation is made fault tolerant by simply running it on a replicated state machine.

* Fault tolerent 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

note: even get() operations need to be processed via log.

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

Data is partitioned and each partition is replicated across multiple nodes.

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

operations initiated at leader and replicated to other nodes.

part-A

3. Aduantages of storing multiple relations in a single file:

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

Disaduantages of storing multiple relation in a single file

*Increases the size and complexity of the DBM

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 100,000 hours B disk will fail, then data will be lost. The other case is B is already failed and within 100,000 hours A will fail and then data will be lost.

For the first case, A disk is failed for 100 hours every 100,000 hours. So in order to make B to fa it will need 100,000^2/100 hours. Because the other Case, the time is reduced to 100,000^2/(2* 100).

3. Advantage

* Data retrieval: Computer-based system provide enhanced data retrieval techniques to retrieve data stored in files in easy and efficient way.

* Editing:

It is a easy to edit any information stored in Computers in form of files. specific application programs or editing software can be used for this purpose.

Disadvantage

* Data redundancy: It is possible that the same information may be duplicated in different files. This Leads to data redundancy results in memory wastage.

* Data inconsistency: Because of data redundan it is possible that data may not be in consistent sta

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

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

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