Name: Grishma Raj Khanal.
THA -076- BCT -016

2069 Chaitra:

Q.n.7 Compare passive replication with active replication approach. Also discuss with a technique that makes the distributed system service highly available. [2+4]

⇒ The mechanism of maintaining multiple copies of data at multiple nodes is called replication. A replication service is correct if it keeps responding despite faults and clients cannot tell the difference between a service provided by replication and one with a single copy of the data.

	Passive ~~Primary~~ replication	Active Replication
1.	There is at ~~ea~~ any one time a single primary replica manager and one secondary replica manager.	The replica managers are state machines that play equivalent roles and are organized as a group.
2.	The primary executes the operations and sends copies to the backups.	All execute the operation in equivalent roles.
3.	If the primary fails, one of the backup is promoted to be primary.	If any replica manager crashes then this need has no impact on performance, since remaining replica will respond.

Fault-tolerant services send updates in an eager fashion- all correct replica manager receive updates as soon as possible. This may be unacceptable for high availability services.

A few systems that provide high availabity servie are:

i) Gossip

ii) Bayou

iii) Coda

Gossip and Bayou both allow clients to make updates to local replicas while partitioned. In each system, replica managers exchange updates while partioned. Gossip provides its highest availabity at the expense of relaxed casual consistency. Bayou provides stronger eventual consistency guarentees, employing automatic conflict detection, and technique of operational transformation to resolve conflicts.

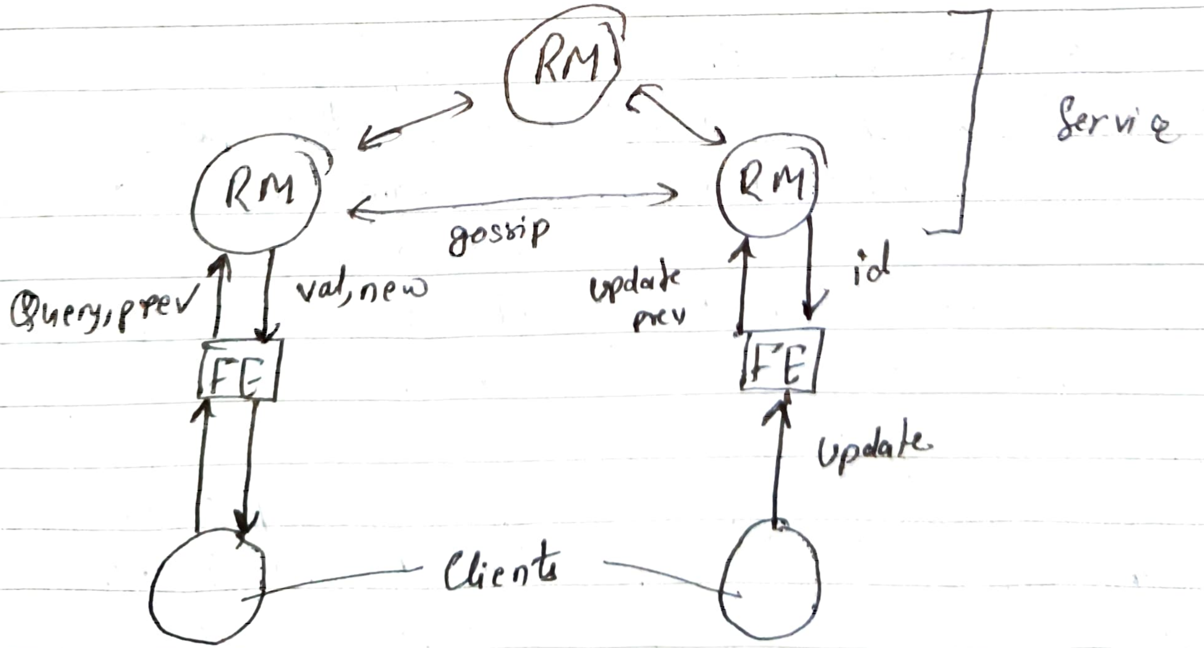

Fig: Query and update operation in Gossip Architecture.

Q no 8

What do you mean by distributed deadlock? Explain two-phase commit ~~protol~~ protocol of handling distributed transaction.

[2+5]

⇒ The situation in which two transactions are waiting and each is dependent on the other. In Distributed deadlock the following conflicts may occur:

a. Conflicts between two transactions on the same site.

b. Conflicts between two transactions on the different sites.

Additionally, transaction location and transaction control issues may occur.

The two-phase commit protocol is due to Gray without loss of generality, consider a distributed transactions involving ~~two~~ participants each running on different machine. The protocol consists of two phases, each consisting of two steps.

Voting phase:

→ The coordinator sends a VOTE-REQUEST message to all participants

→ When participant receives a vote-request message, it return either VOTE_COMMIT or VOTE-ABORT message.

Completion phase:

→ The coordinator collects all votes from the participants. If all participants have voted to commit, then so will the coordinator. It will send GLOBAL_COMMIT to all else sends GLOBAL_ABORT to abort transaction.

→ Each participant voted for commit waits for GLOBAL-COMMIT, if it gets it locally commits the transaction otherwise it is localled aborted.

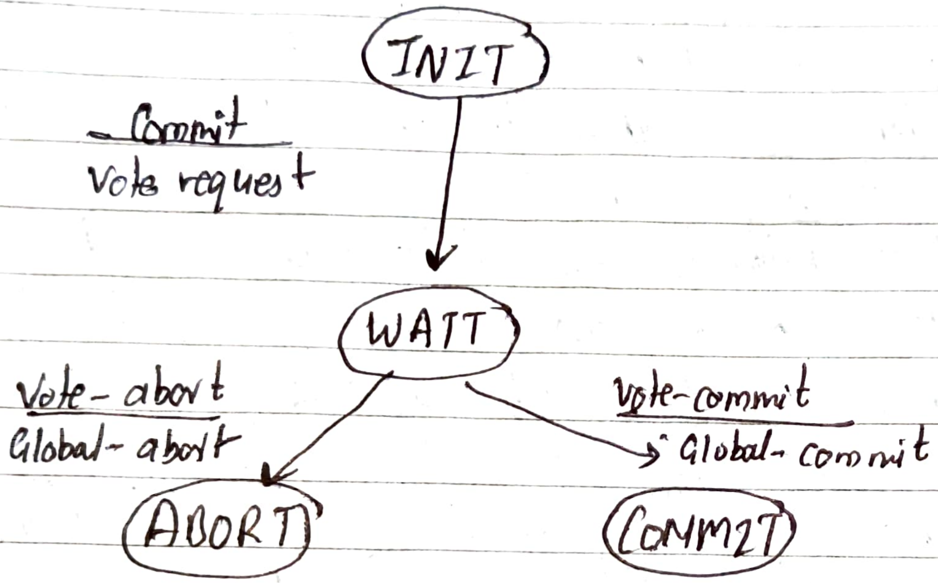

Fig: Coordinator

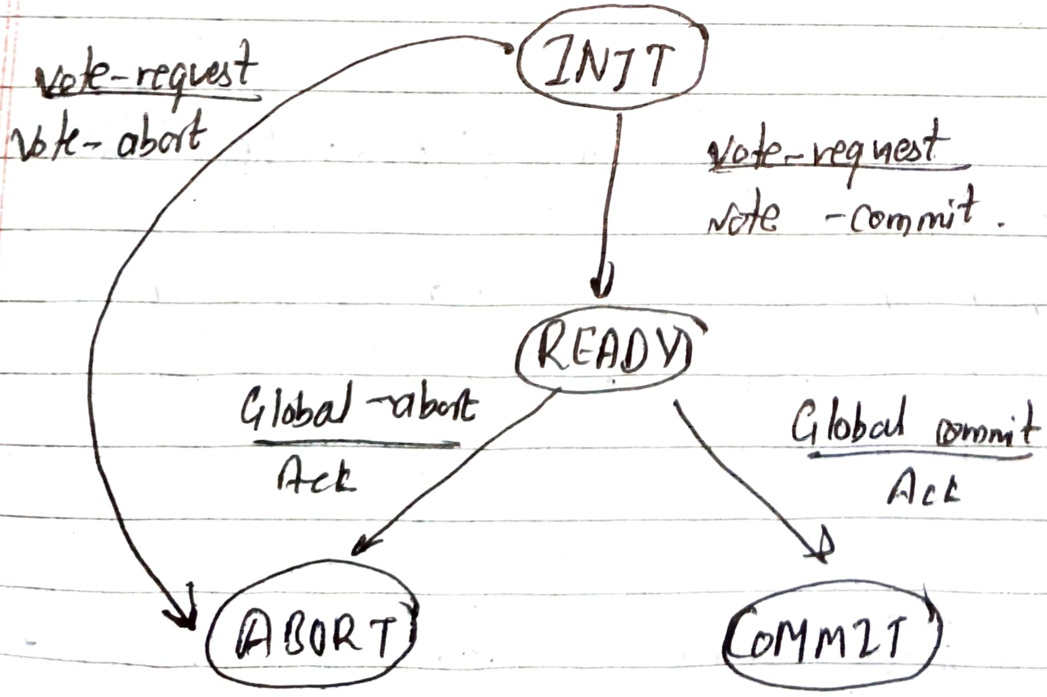

Fig- Participant

Date
Page

Qno 9 What are flat and nested transactions? Describe methods for concurrency control in distributed system. [3+4]

⇒ Those transactions that access objects managed by multiple serves are called distributed transactions.

Flat transaction:

In flat transactions, a client makes requests to more than one server. It completes each of its requests before going on to next one, sequentially.

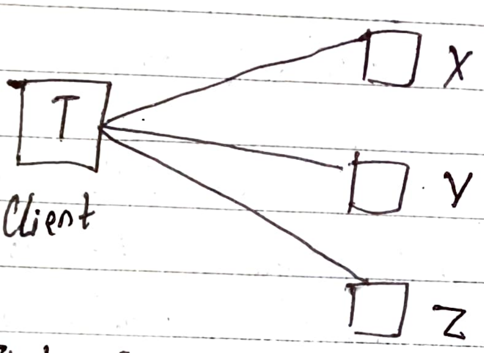

Fig: Flat Transaction

Nested Transaction:

In nested transactions, the top level opens sub transactions; each of which can further open sub transactions. Sub transactions at the same level runs concurrently.

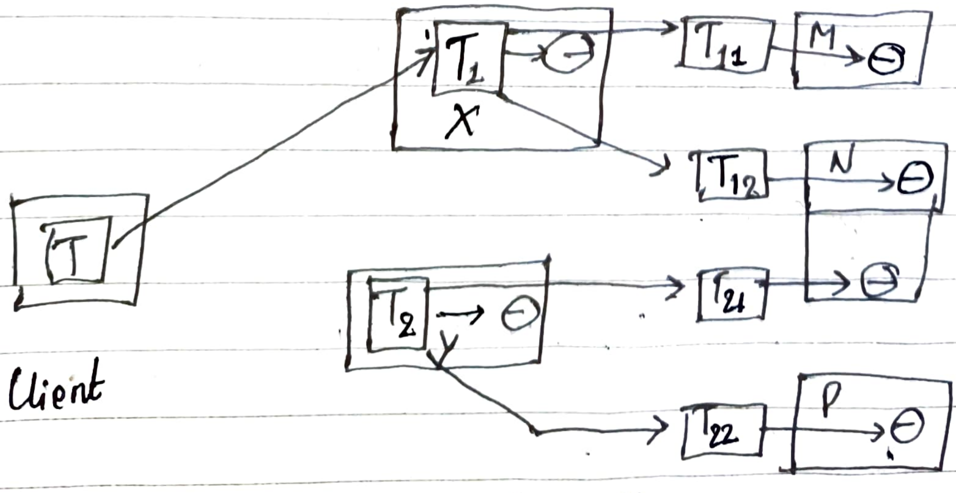

Fig: Nested Transaction.

Concurrency control is the proces of managining simultaneous execution of transactions in a shared system, to ensure the serializability of transactions.

Some concurrency control techniques are:

~~i) Locks~~
~~ii) Two-phase locking protocol~~
~~iii) ~~Deadlock~~ Timestamp ordering~~

i) Locks (Two-phase locking protocol)
ii) ~~Optimistic~~ Timestamp ordering.

i) <u>Locks;</u> Two-phase locking protocol

A transaction is said to follow two-phase locking protocol if locking and unlocking can be done in two phases. Under strict executing regimes, a transaction that needs to read or write an object must be delayed until other transactions that wrote the same object have comitted or aborted. The presence of the locks prevents other transactions reading or writing the objects.

Each transaction is executed in two phases:

- Growing phase: New locks on items can be acquired
- Shrinking phase: Existing locks, but no new lock can be aquired.

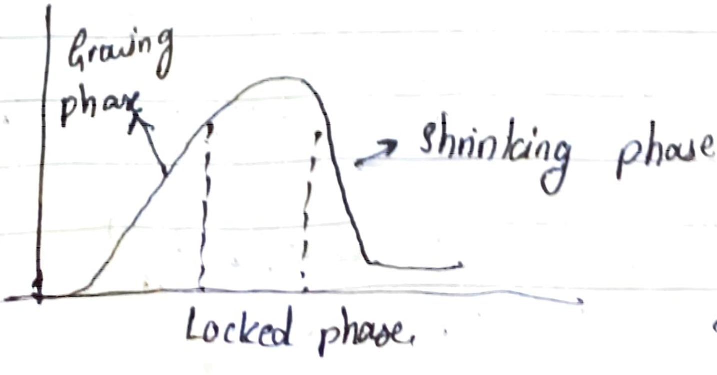

Fig: Two-phase lock

ii) Timestamp ordering:

On timestamp ordering, each operation in a transaction is validated when it is carried out. If the transaction cannot be validated, it is immediately aborted and can be restarted by client. Each transaction is assigned a unique Timestamp value when it starts. The timestamp defines its position in time sequence and ordered according to their timestamps.

A transaction's request to read or write an object is valid only if the transaction was last read and written by earlier transactions

Rule	T_c	T_i	
1.	Write	read	T_c must not write an object that has been read by any T_i where $T_i > T_c$.
2.	write	write	T_c must not write an object that has been written by T_i where $T_i > T_c$. This requires that $T_c >$ the write timestamp of committed object.
3.	read	write	This requires that $T_c >$ the write timestamp of committed object.

Comparison of concurrency control method:

→ Time stamps and locks use pessimistic approach

→ Time stamps are better for read only transactions.

→ lock is better for operational updates

→ Timestamps immediately aborts transactions.

→ locking makes transactions wait.