Consistency Model and Replication Management.

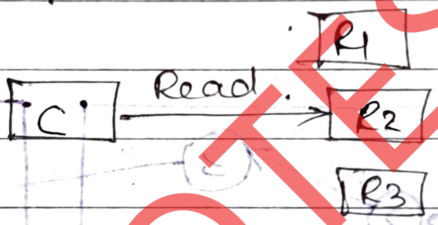
Replication :-

Process of creating the copy as a part of backup process to make the data ~~are~~ all the time available is known as REPLICATION.

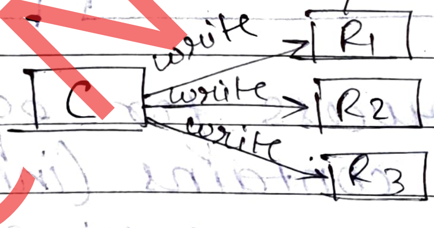
Replication Management Protocols :-

1.) Read any write all.

Client can read from any of the replicas.

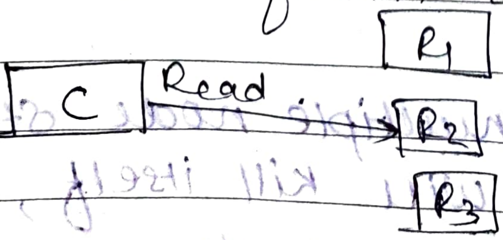


But, if client updates the data it should update all the replicas.

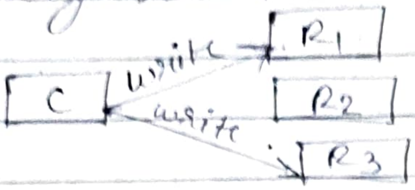


2.) Read any write available

Similarly, to read any write all client can read from any replica.

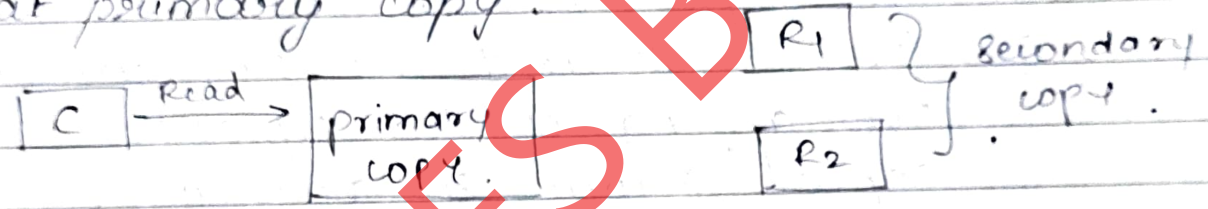


when client updates the data it will update all available copies and the one which is down. Eg: R2 will reboot and self updates from neighbour.

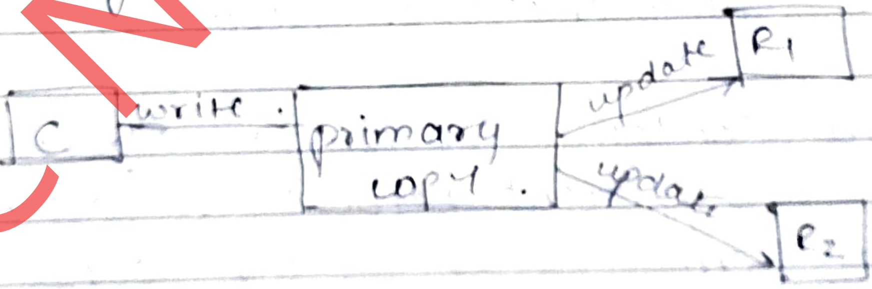


3) Primary Copy Protocol:

One of the copies would be elected as primary copy, client will read from that primary copy.



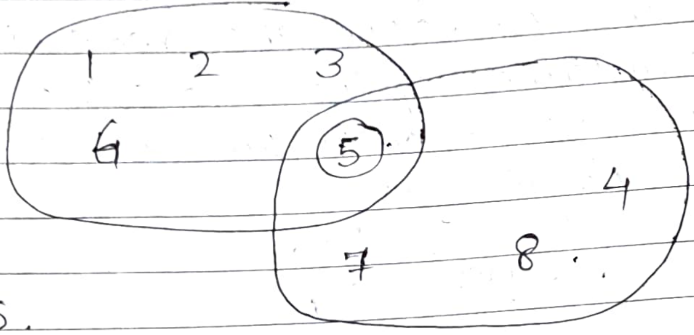
when client updates the data it will write to primary copy which will update the remaining secondary copy.



4) Quoram Based Protocol

This protocol works by creating Read Quorom 'r' and write Quorom 'w' made from all the copies 'n'. The Quorum is created in the following manner.

$[r + w > n]$



$r = 5.$

$w = 4$

$\therefore r + w > 8 \leftarrow n.$

Any client will read from any copies r. Quorom Eg. 1, 2, 3, 6, 5. and when client updates the data, it will update to all copies of w Quorom.

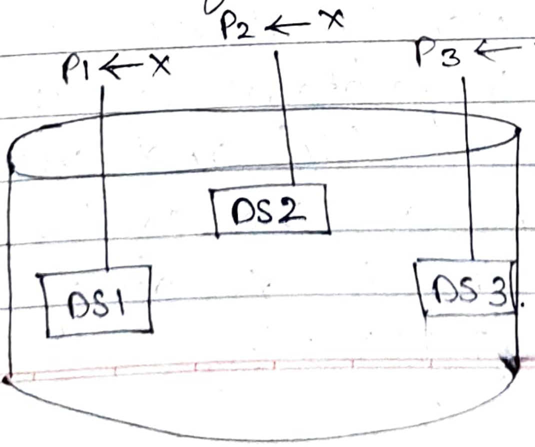
Eg: 4, 5, 7, 8 and the one which is overlapping would update the other copies from r quorom.

(*) CONSISTENCY MODEL.

The process of maintaining same value everywhere is known as consistency

1. Data-Centric Consistency Model.

Different process accessing same data item. Eg: x from different data stores should fetch some value of data item x.

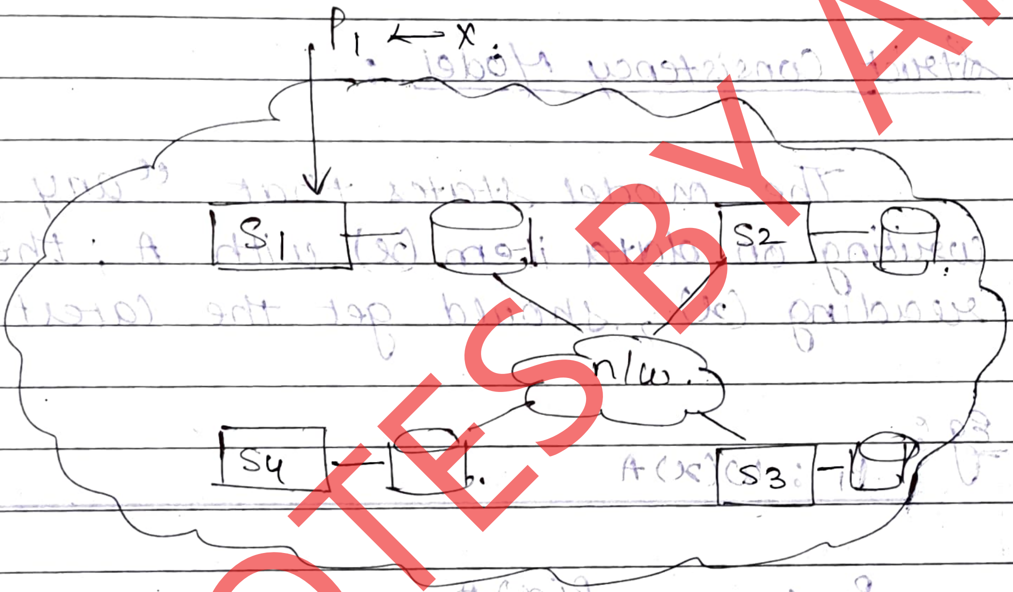


DS : Data store.

Database.

2.) Client-centric Consistency Model.

A single process updating a data item x at one location and trying to access same data item from different location should fetch same value of data item x.



Types of Data Centric Consistency Model

- Strict Consistency Model.
- Sequential consistency Model.
- Causal Consistency Model.
- Weak Consistency Model.
- Release Consistency Model.

1) Strict Consistency Model :

The model states that "any process writing on data item (x) with A, thereafter reading (x), should get the latest value".

Eg :

P1 : W(x)A

P2 : R(x)A

P3 : R(x)A

time →

P1 writes A on (x), henceforth ~~re~~ anyone reading (x) should get A.

Violation :

P1 : W(x)A

P2 : R(x) NIL

P3 : R(x)A

time →

Reason : It takes sometime to propagate the update.

ii) Sequential Consistency Model.

The model states that "any process writing A and B on data item (x). should give value of (x) in the same sequence to all processes."

P1 : W(x)A . W(x)B

P2 : R(x)A R(x)B

P3 : R(x)A . R(x)B time →

P1 writes A and B on (x), P2 (first) reads (x) and gets A and then B, so every other process should get A and B.

Violation :

P1 : W(x)A W(x)B

P2 : R(x)A . R(x)B

P3 : R(x)B . R(x)A

time →

Reason : i) Order of sending and receiving in the n/w may not be possible

iii) Causal Consistency Model.

The model states that "If any process reads data item (x) and gets A and then writes B on (x), then B is said to be causally dependent. and every process should always get A and then B."

Eg :. P1 : R(x)A W(x)B   W(x)C

P2 : R(x)A R(x)B R(x)C.

P3 : R(x)C . R(x)A R(x)B

time →

P1 reads (x) and then writes on (x) and further independently writes C on (x), then any process reading (x) should always get A and then B; C in any order.

Violation:

P1 : R(x)A. W(x)B   W(x)C.

P2 : R(x). B R(x). A. R(x) C

P3 : R(x)B. R(x) C R(x) A

time →

Reason : order of sending and receiving in a n/w may not be possible.

iv) Weak Consistency Model.

The model states that "when any process writes A on (x), then any other process before reading (x) should synchronize its data store using variable 'S'"

P1 : W(x) A

P2 : R(x) NIL

P3 : $\boxed{S}$ R(x) A

time →.

v) Release Consistency Model.

The model states that "any process can perform read or write operation by acquiring a lock on data item, ~~acquiring~~ operationg on it and then release the lock".

P1 : Acq(x) . W(x)A . Rel(x)

P2 : Acq(x) X kyuki already (x) locked hai.

P3 : Acq(x) . R(x)A . Rel(x) .

time →.

(*) Types of client centric.

- → Monotonic read.
- → Monotonic write.
- → Read your writes.
- → Write ~~your~~ follows Reads.

(*) Reliable client server group communication. {Refer pdf}.

(*) Process Resilience (Explain RPS failure with solution).