Consistency and Replication

Distributed Systems

By: Sushant Bramhacharya

Replication

- Caching of the resources from web servers in browsers, proxies and secondary servers
- Mechanism of maintaining multiple copies of data at multiple nodes.

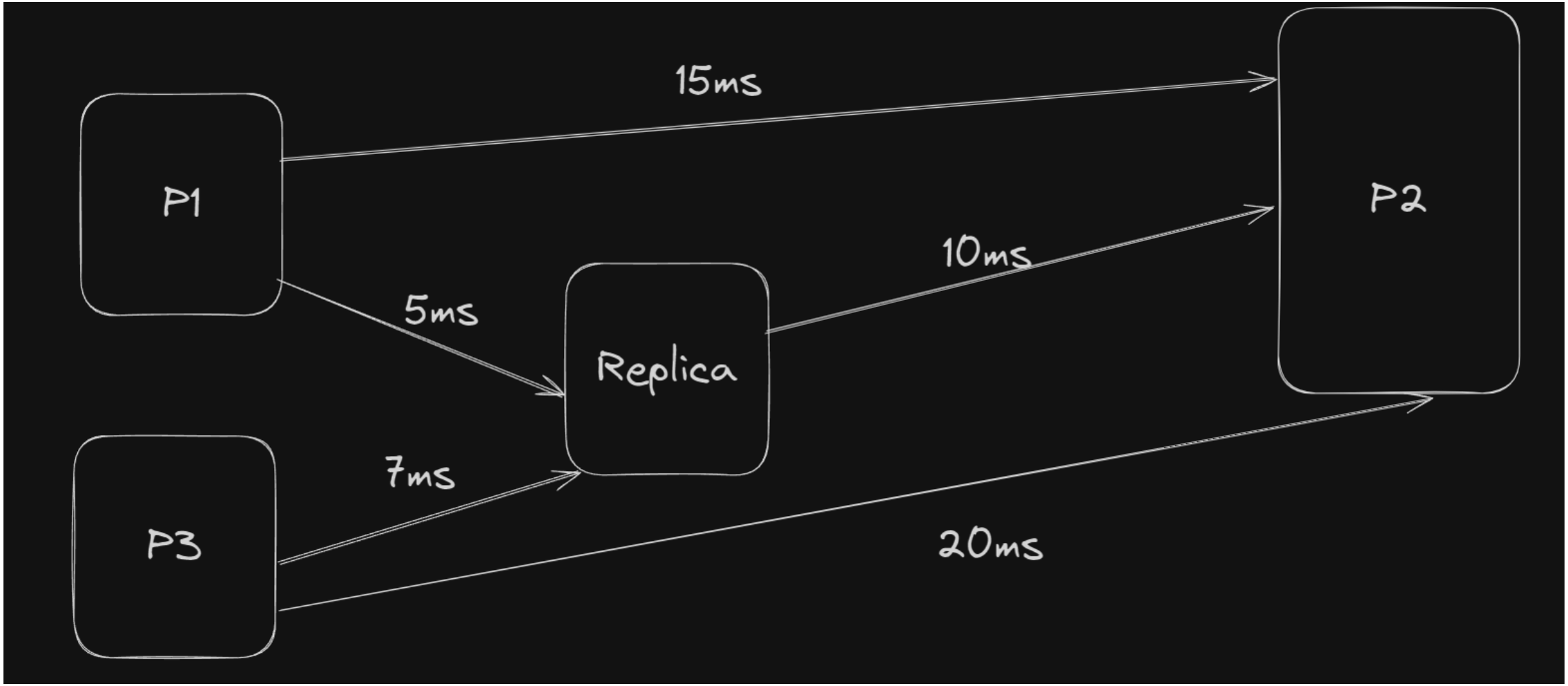
Reasons for Replications

- Performance Enhancements
- Increased Availability
- Fault tolerance

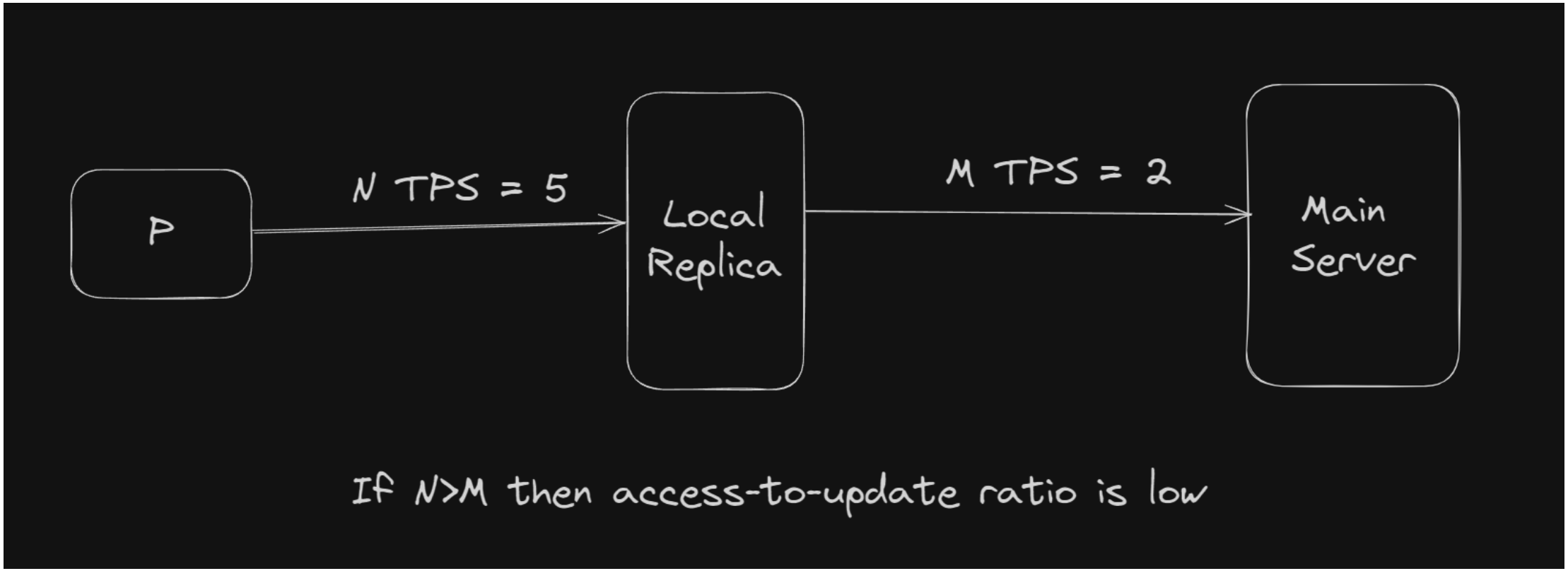
Replication as Scaling Techniques

- Scalability issues generally appear in the form of performance problems.
- Placing copies of data close to the processes using them can improve performance through reduction of access time and thus solve scalability problems.
- Less Load to main server

Replication as Scaling Techniques



Problems in Replication as Scaling



Problems in Replication as Scaling

- We are now faced with a dilemma.
- On the one hand, scalability problems can be alleviated by applying replication and caching, leading to improved performance.
- On the other hand, to keep all copies consistent generally requires global synchronization, which is inherently costly in terms of performance.

Data Centric Consistency Models

- A consistency model is essentially a contract between processes and the data store.

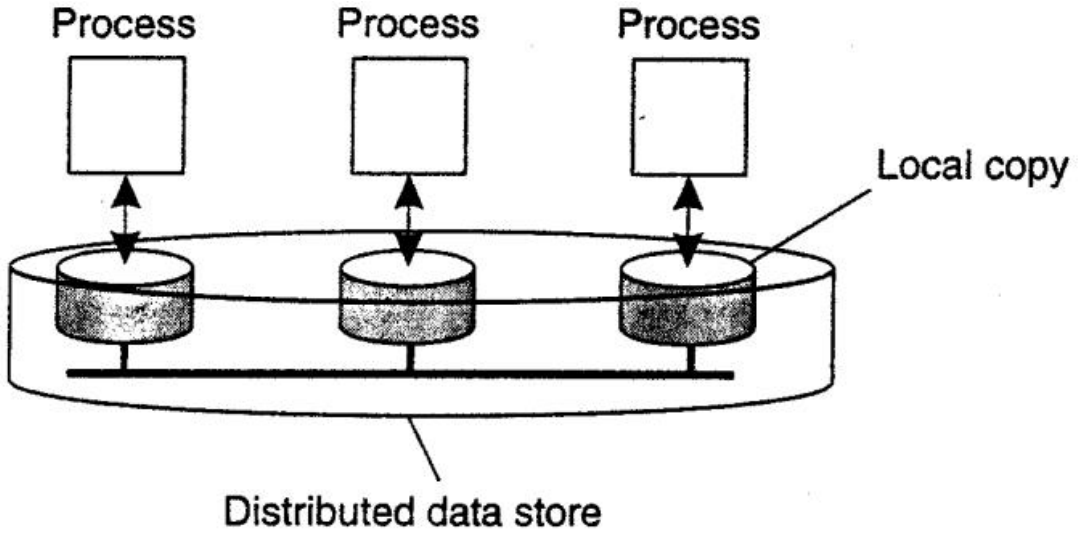
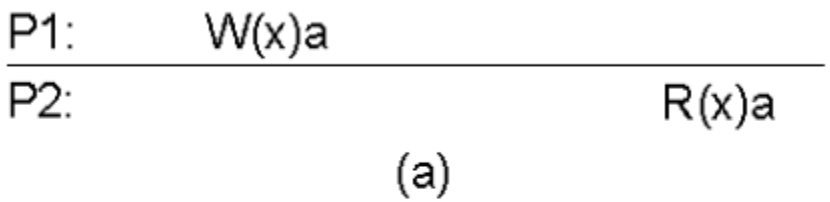


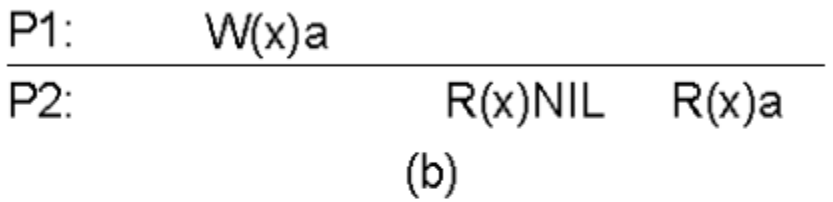
Figure 7·1. The general organization of a logical data store, physically distributed and replicated across multiple processes.

Strict Consistency

- Any read on x should return value on x which is recently written on X.



(a)



(b)

Behavior of two processes, operating on the same data item.

a) A strictly consistent store.
b) A store that is not strictly consistent.

Sequential Consistency (Weaker)

- All processes should access the data in sequential manner.
- The results should be same as if (R/W) is executed in same sequential manner.

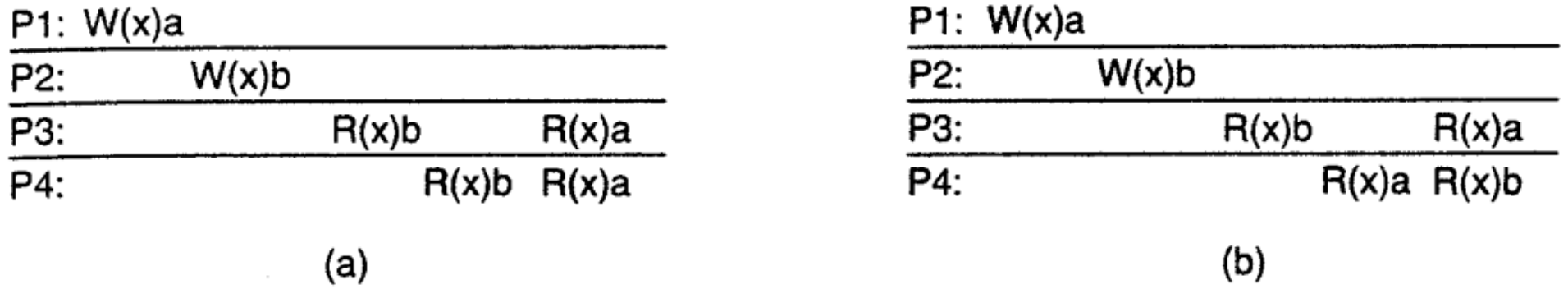


Figure 7-5. (a) A sequentially consistent data store. (b) A data store that is not sequentially consistent..

Casual Consistency (More Weaker)

- Same order for causally related operations.
- Concurrent Operations need not to be in same order.

```
A = A + 1; // First two events are causally related,
B = A * 5; // because B reads A after A was written.
C = C * 3; // This is a concurrent statement.
```

Casual Consistency (More Weaker)

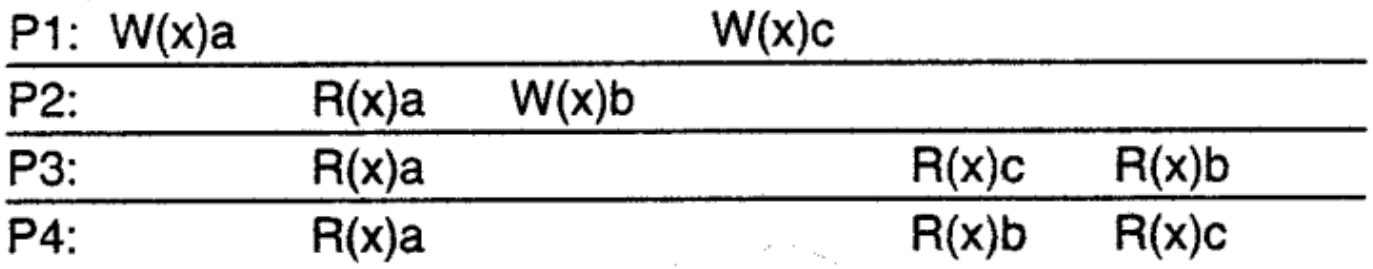


Figure 7-8. This sequence is allowed with a causally-consistent store, but not with a sequentially consistent store.

Casual Consistency (More Weaker)

P1:	W(x)a			W(x)c	
P2:		R(x)a	W(x)b		
P3:		R(x)a		R(x)c	R(x)b
P4:		R(x)a		R(x)b	R(x)c

Figure 7-8. This sequence is allowed with a causally-consistent store, but not with a sequentially consistent store.

P1: W(x)a				
P2:	R(x)a	W(x)b		
P3:			R(x)b	R(x)a
P4:			R(x)a	R(x)b

(a)

P1: W(x)a				
P2:		W(x)b		
P3:			R(x)b	R(x)a
P4:			R(x)a	R(x)b

(b)

Figure 7-9. (a) A violation of a causally-consistent store. (b) A correct sequence of events in a causally-consistent store.

Client Centric Consistency Models

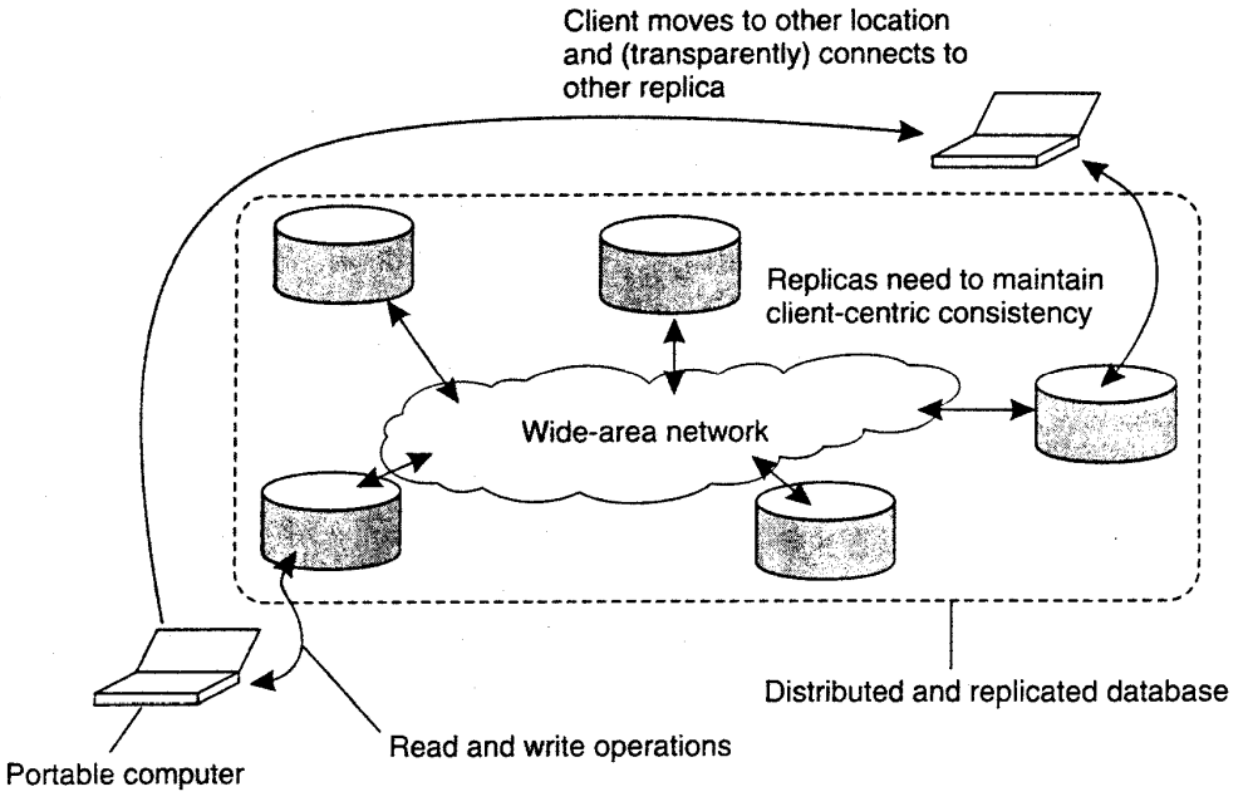


Figure '-11. The principle of a mobile user accessing different replicas of a distributed database.

Monotonic Reads

- If a process reads the value of a data item x, any successive read operation on x by that process will always return that same value or a more recent value.

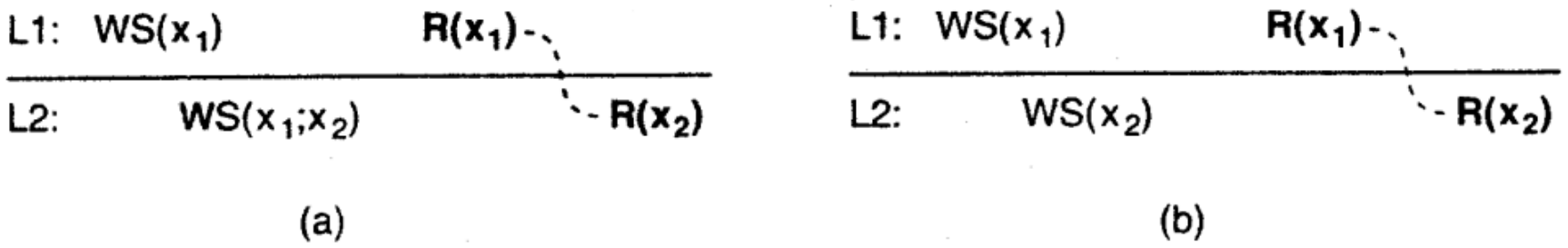


Figure 7-12. The read operations performed by a single process *P* at two different local copies of the same data store. (a) A monotonic-read consistent datu store. (b) A data store that does not provide monotonic reads.

Monotonic Writes

- A write operation by a process on a data item x is completed before any successive write operation on X by the same process.

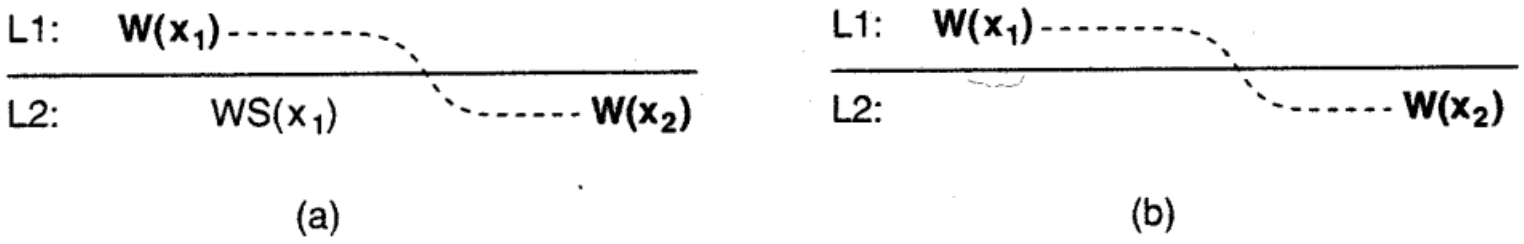


Figure 7-13. The write operations performed by a single process *P* at two different local copies of the same data store. (a) A monotonic-write consistent data store. (b) A data store that does not provide monotonic-write consistency.

Read Your Writes

- The effect of a write operation by a process on data item x will always be seen by a successive read operation on x by the same process.

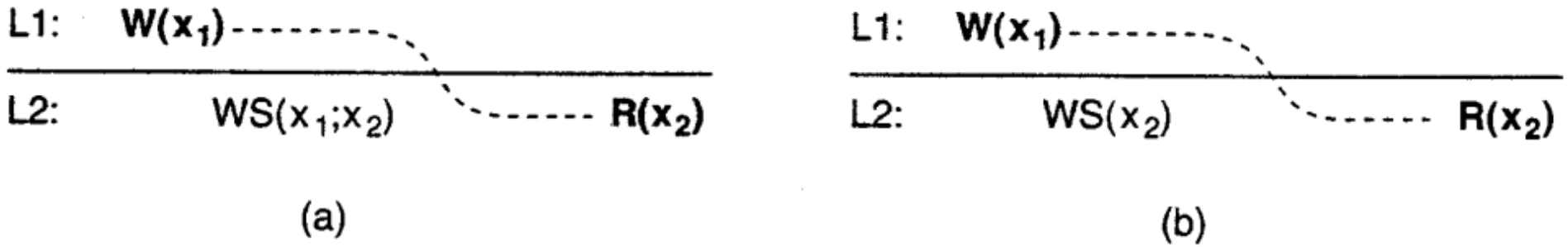


Figure 7-14. (a) A data store that provides read-your-writes consistency. (b) A data store that does not.

Writes Follow Reads

- A write operation by a process on a data item x following a previous read operation on x by the same process is guaranteed to take place on the same or a more recent value of x that was read.

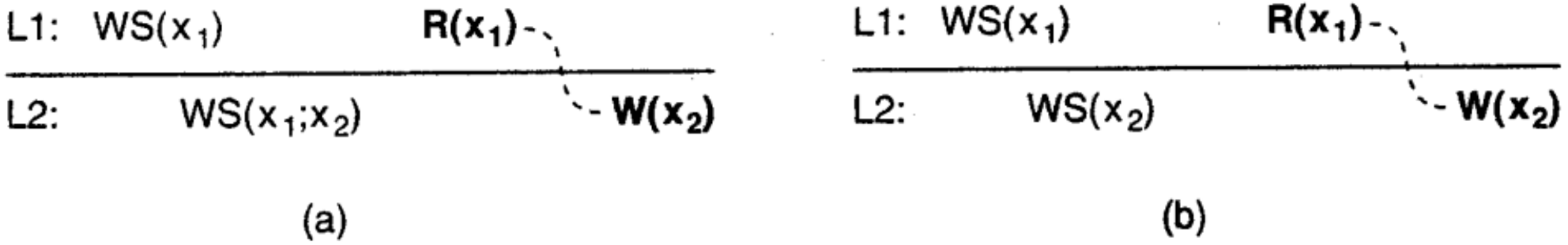


Figure 7-15. (a) A writes-follow-reads consistent data store. (b) A data store that does not provide writes-follow-reads consistency.

Replica Management

- Replica Placement
  - Somewhere where no of users are high
  - Near to users
  - Multiple Replica’s for high users

Content Replication and Placement

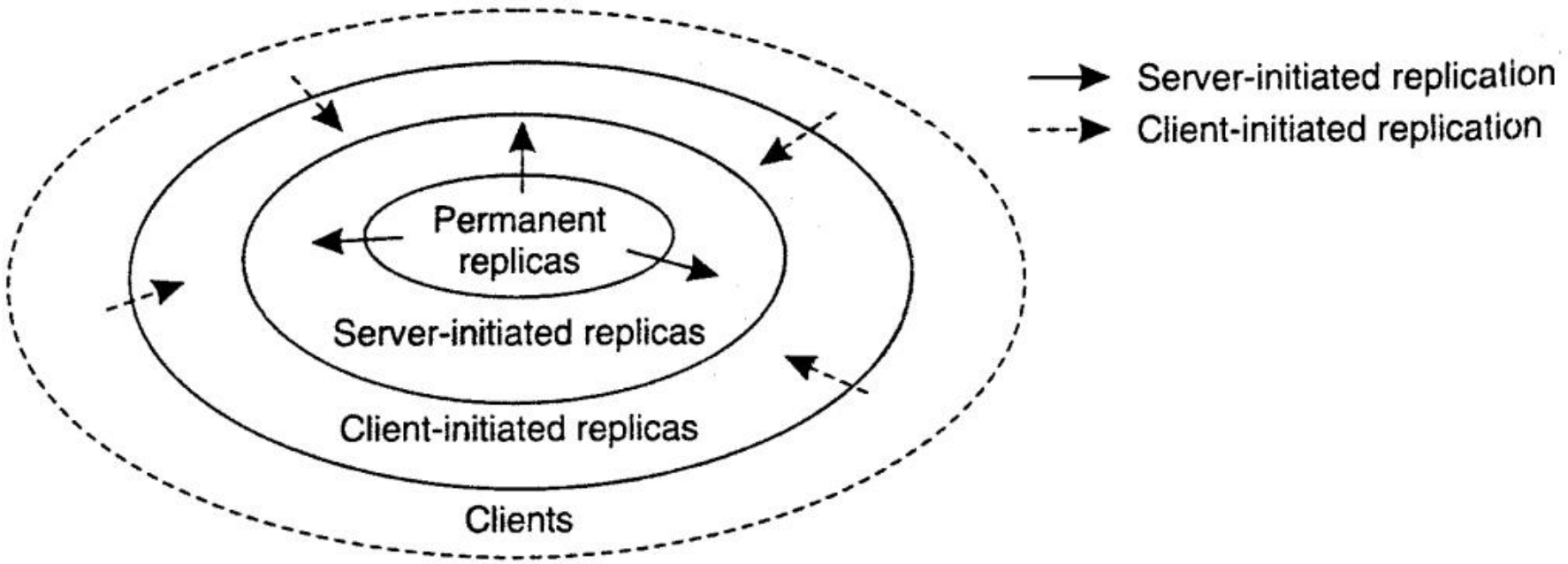


Figure 7-17. The logical organization of different kinds of copies of a data store into three concentric rings.

Content Distribution

- State versus Operations
  - Propagate only a notification of an update.
  - Transfer data from one copy to another.
  - Propagate the update operation to other copies.
- Pull vs Push protocols
  - Updates are propagated to other replicas without those replicas even asking for the updates, also called server-based protocols
  - a server or client requests another server to send it any updates it has at that moment. Pull-based protocols, also called client-based protocols

Consistency Protocols

- Primary Based
  - Remote Write Protocol

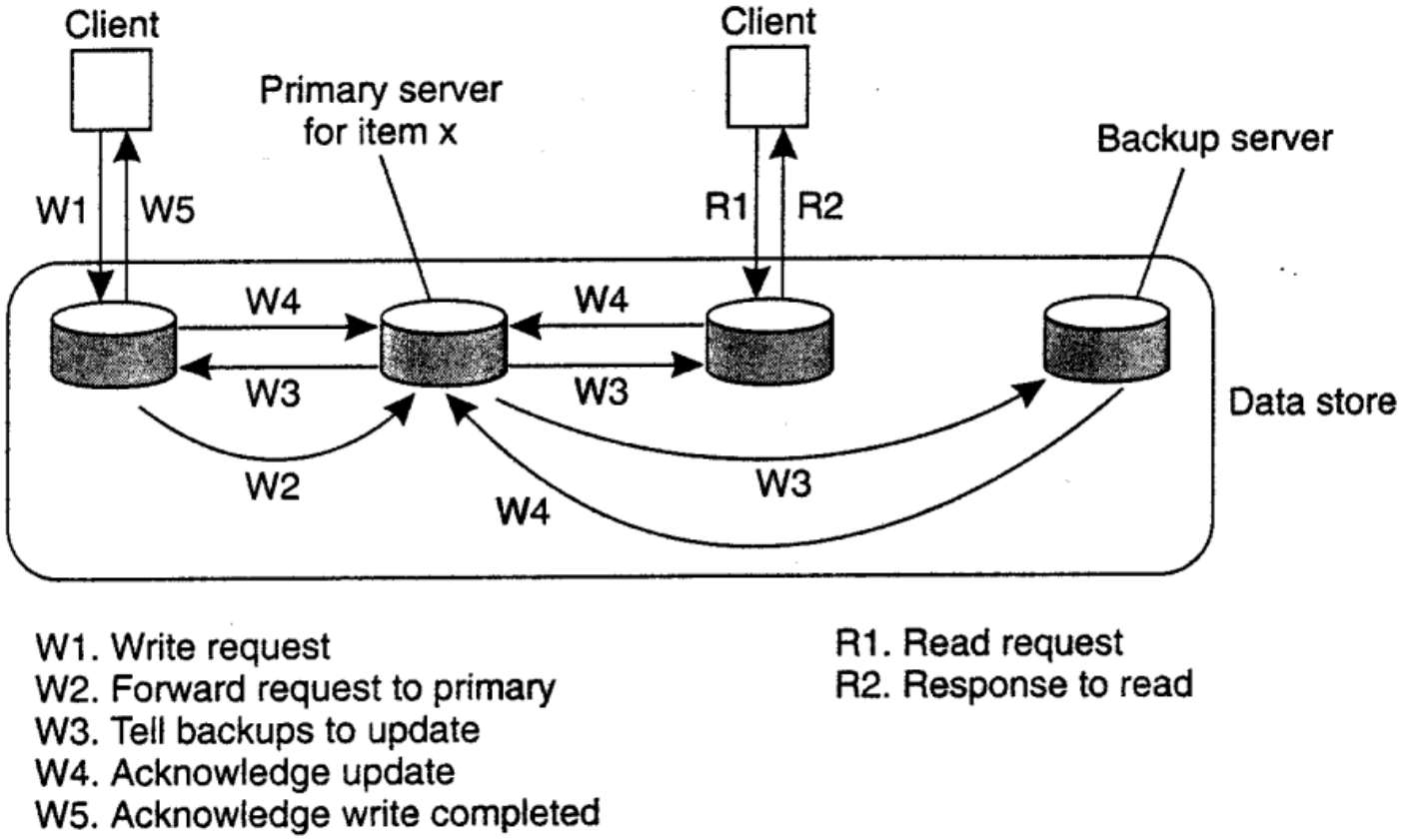
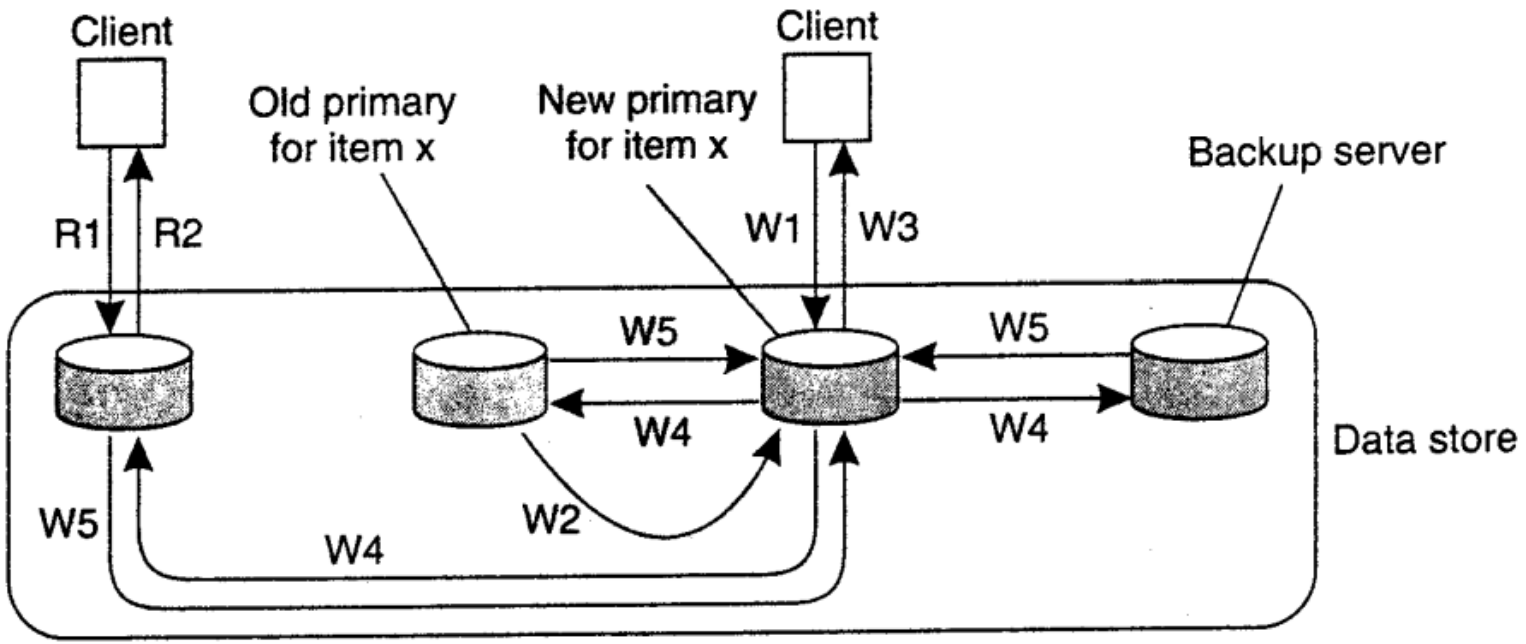


Figure 7-20. The principle of a primary-backup protocol..

Local Write Protocol



W1. Write request
W2. Move item x to new primary
W3. Acknowledge write completed
W4. Tell backups to update
W5. Acknowledge update

R1. Read request
R2. Response to read

Other Protocols

- Replicated-Write Protocols
  - Write operations can be carried out at multiple replicas instead of only one.
- Cache-Coherence Protocols
  - Caches form a special case of replication, in the sense that they are generally controlled by clients instead of servers.

Caching and Replication in Web

- Browsers are equipped with a simple caching facility.
  - Whenever document is fetched it stores in cache and next time it gets retrieved from cache.
- Web Proxies also caches result from requests send by local client and also serves to other clients if necessary.
- ISPs also Caches in their network to reduce network traffic and improve performance.
- Caches may not contain the requested information so that there is risk of latency.