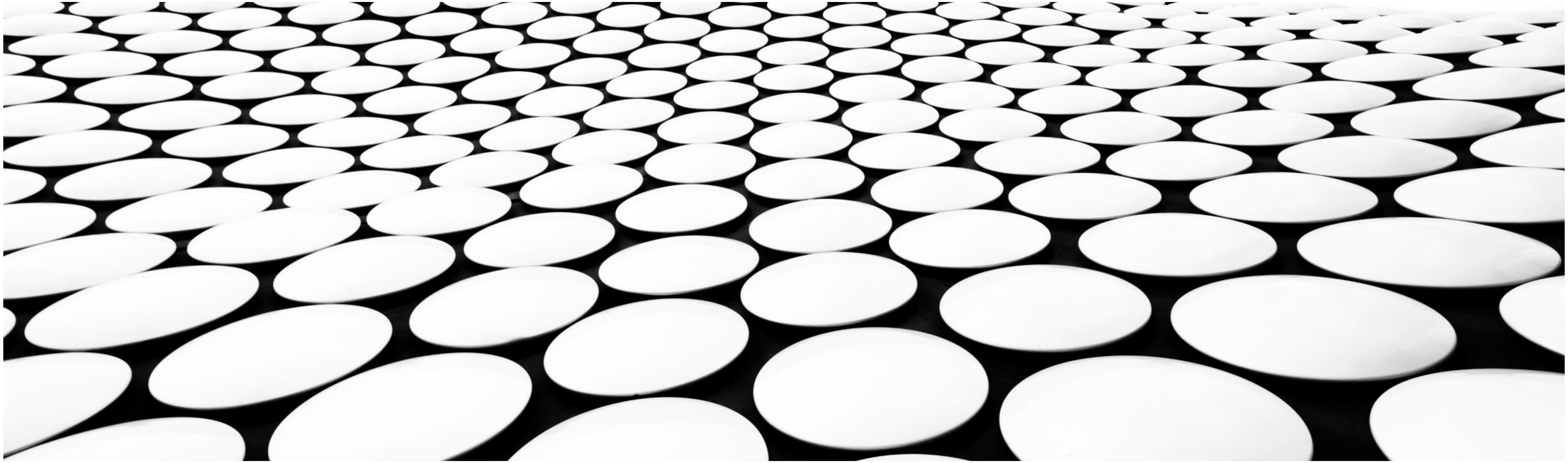
Resiliency in distributed systems (part 1 of 2)

COEN-317: Distributed Systems

Robert Bruce
Department of Computer Science and Engineering
Santa Clara University



Data replication: advantages and disadvantages

Advantages:

"Improving the reliability of the distributed system" [1]

"Improving performance" [1]

Disadvantages:

The update process creates temporary inconsistencies between the data replicas [1]

[1] p. 420, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Maintaining consistency in data replication

CONIT:

- A portmeanteau of "**CON**sistency un**IT**" [1]
- Represents a fundamental data item or record [1]

To measure effectiveness of data replication techniques:

- We evaluate the number of effective CONIT units that have been consistently replicated [2]

[1] p. 360, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.
[2] p. 361, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Consistency versus coherence

Consistency model:

"Describes what can be expected with respect to that set when multiple processes concurrently operate on that data" [1].

Coherence model:

"Describe what can be expected to hold true for only a single data item" [1].

[1] p. 372, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Implementing consistency protocols

- Remote-write protocol
  - Synonymous with primary-backup protocol [2]
- Local-write protocol
- Replicated-write protocol
- Quorum-based protocol
- Cache-coherence protocol

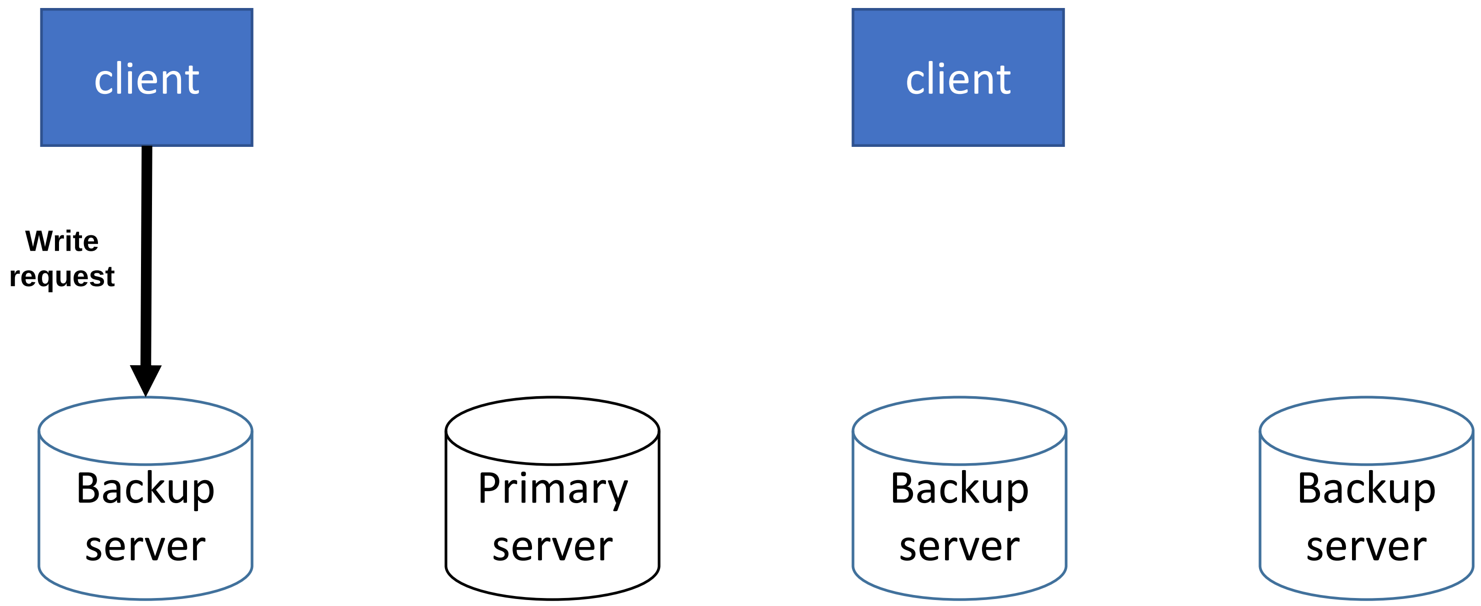
[1] p. 372, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.
[2] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol

Remote-write protocol (also called primary-backup protocol) [1]

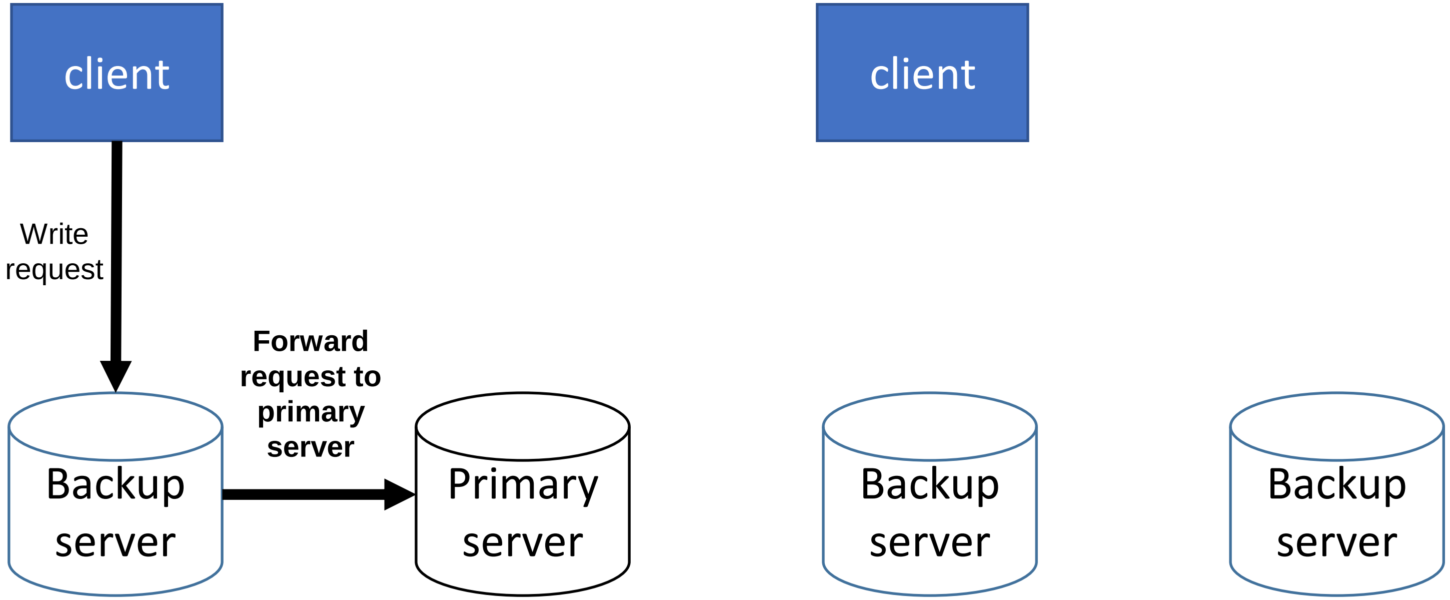
[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol



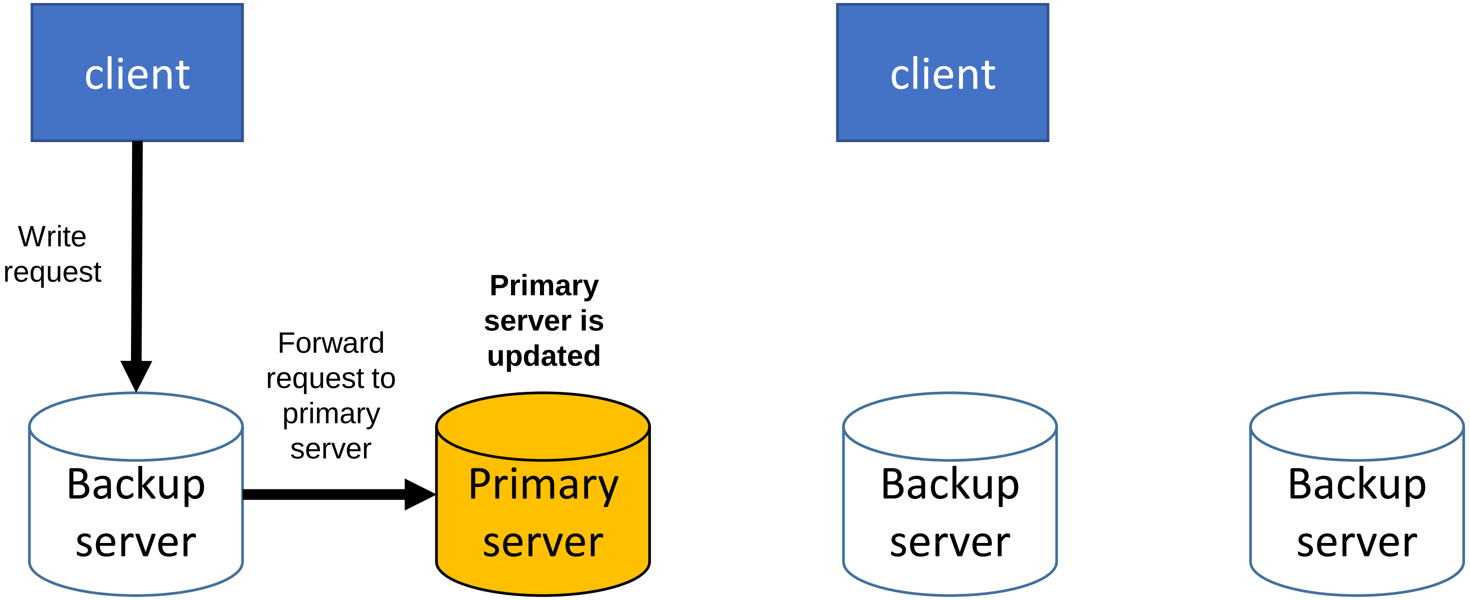
[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol



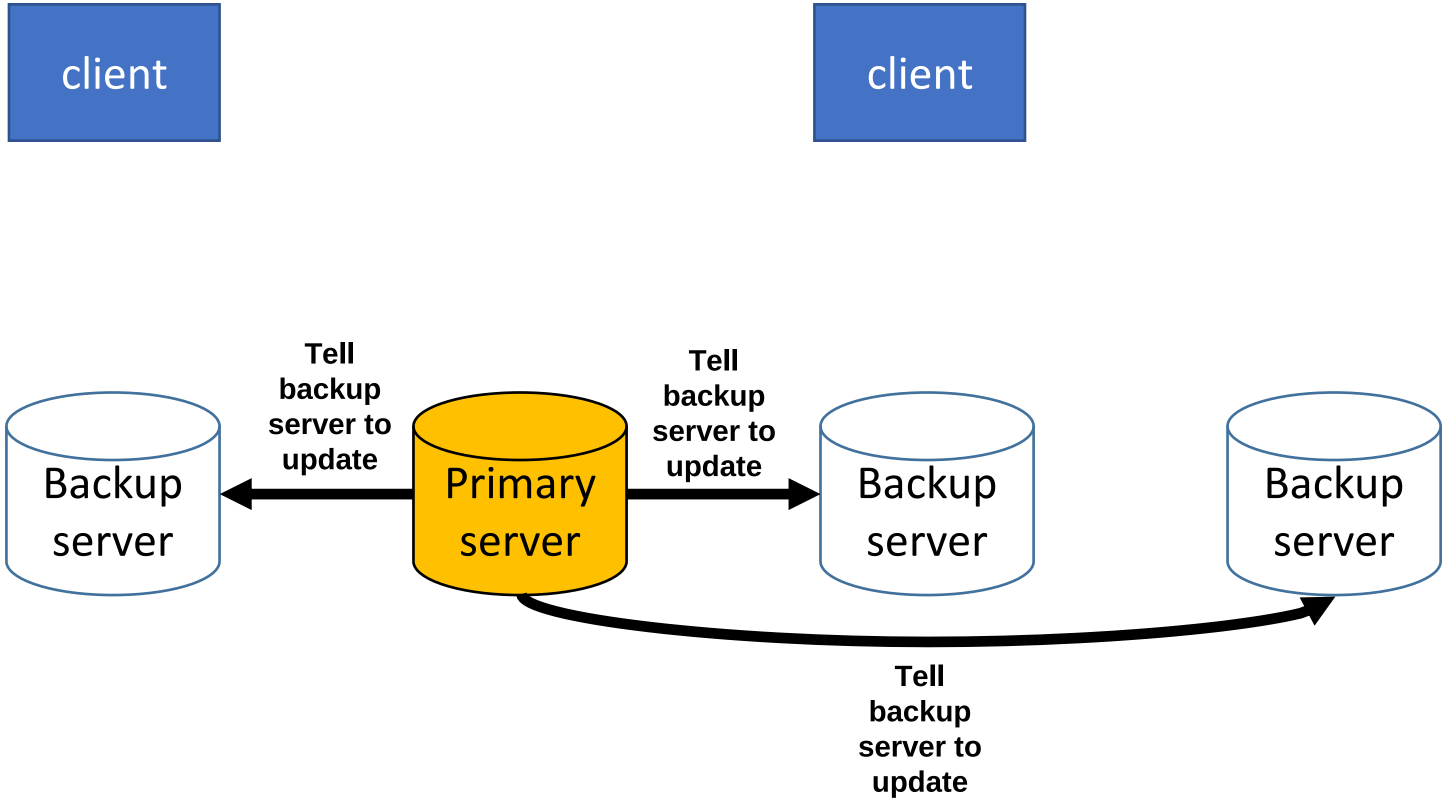
[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol



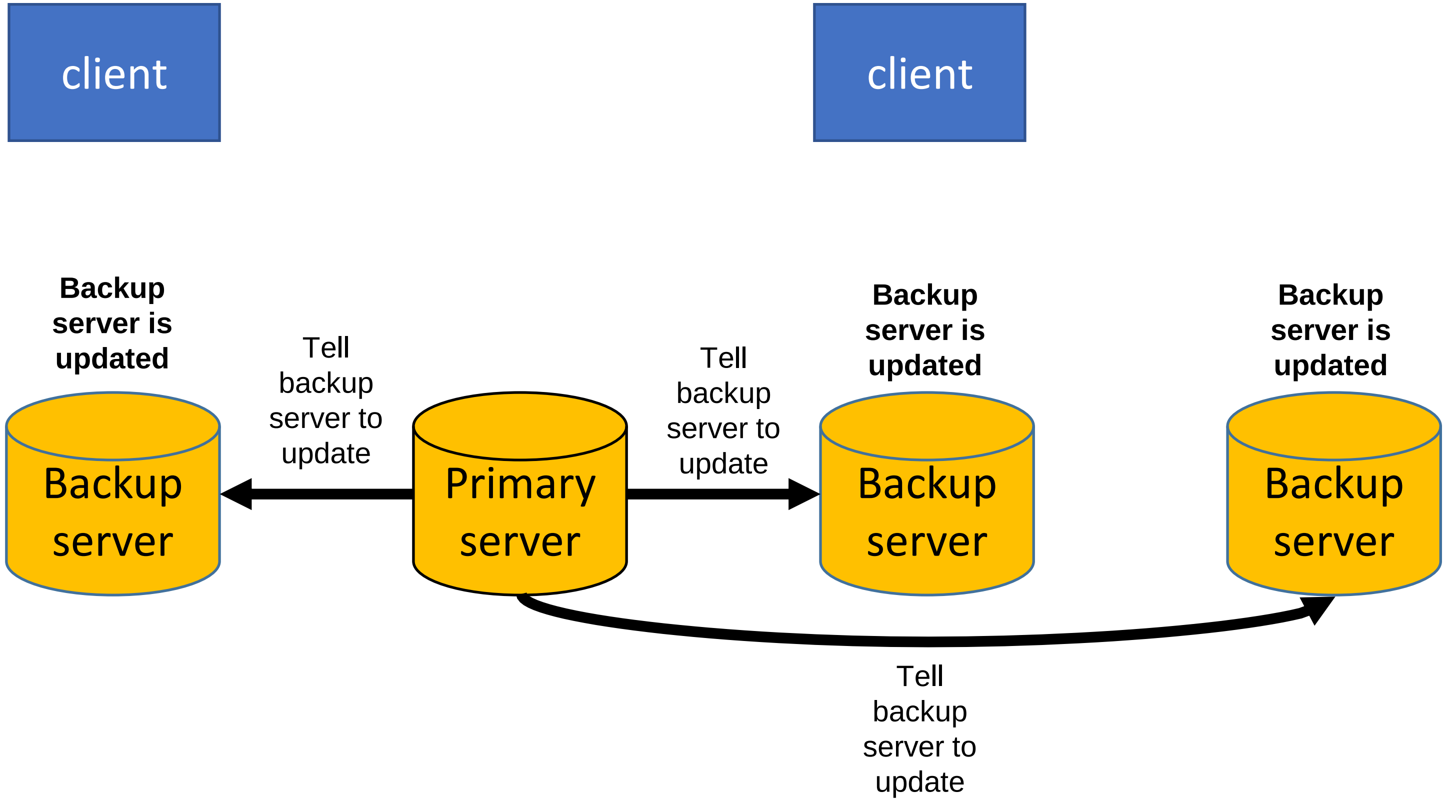
[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol



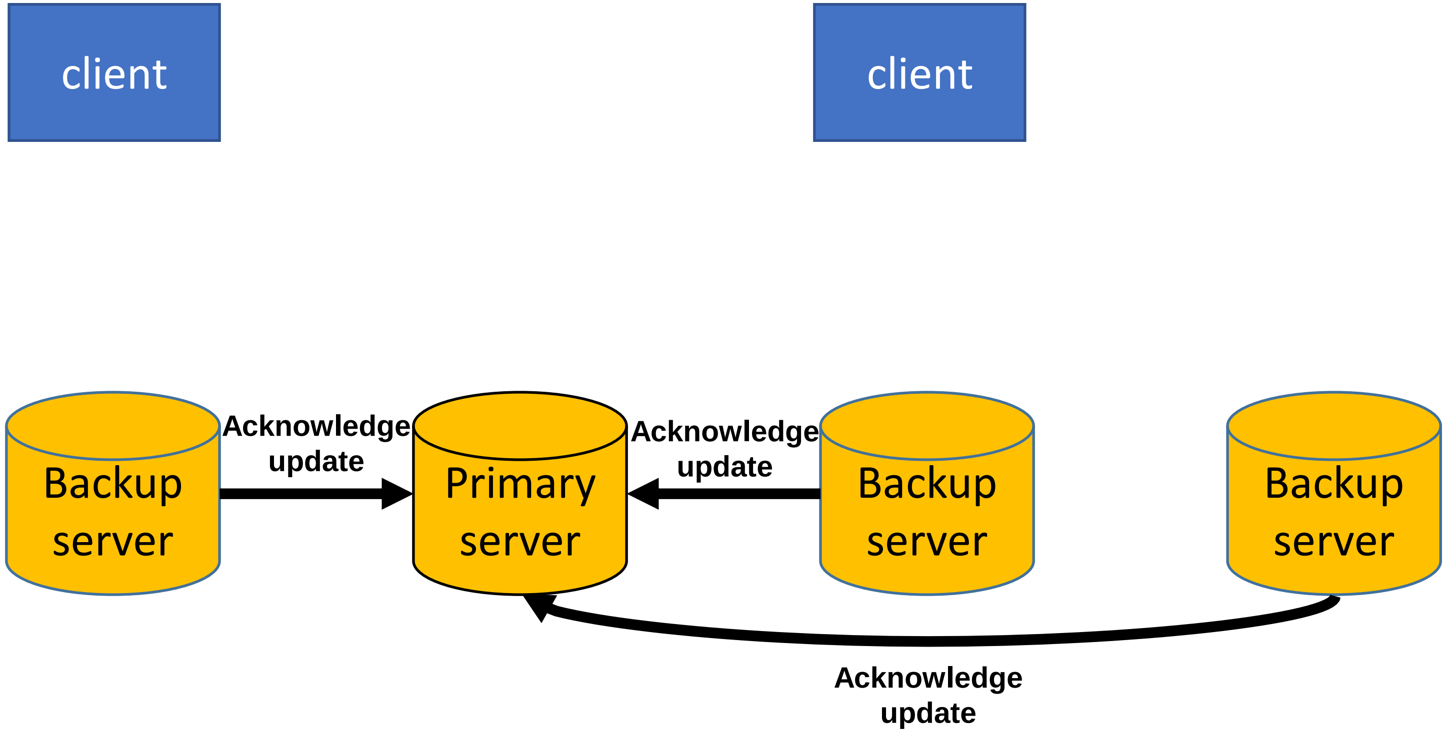
[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol



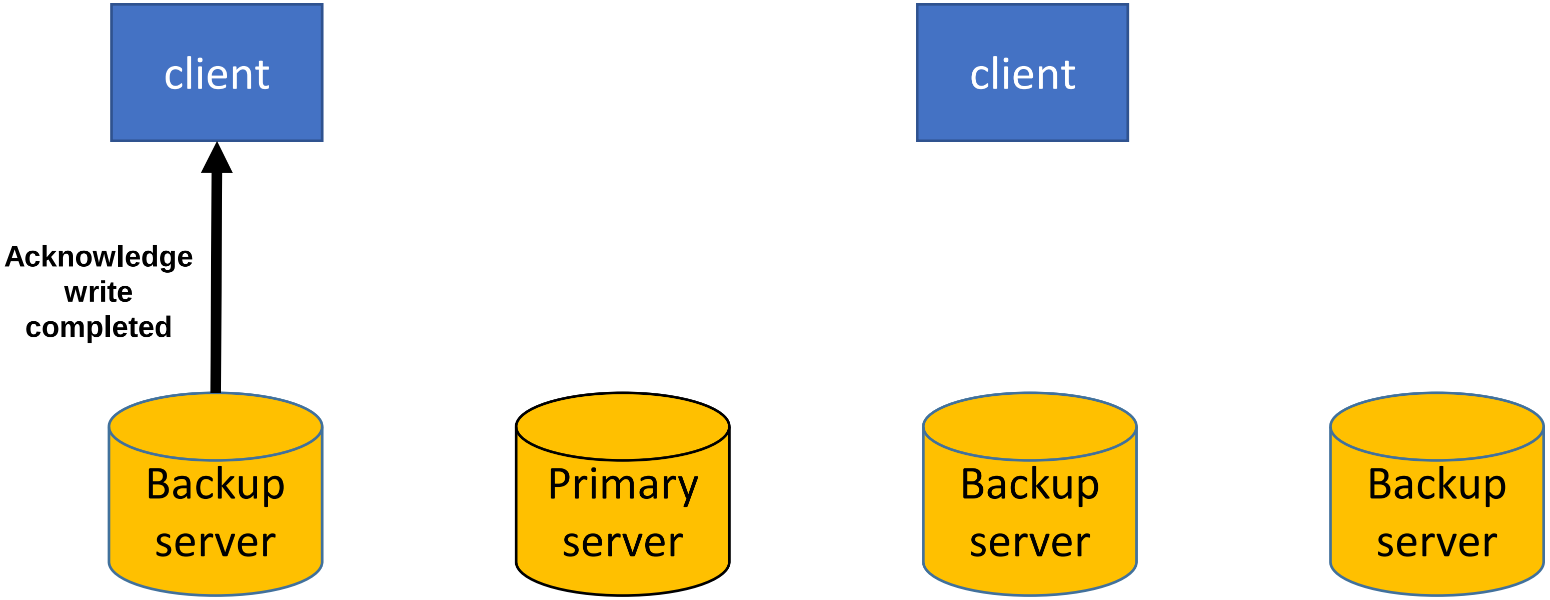
[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol



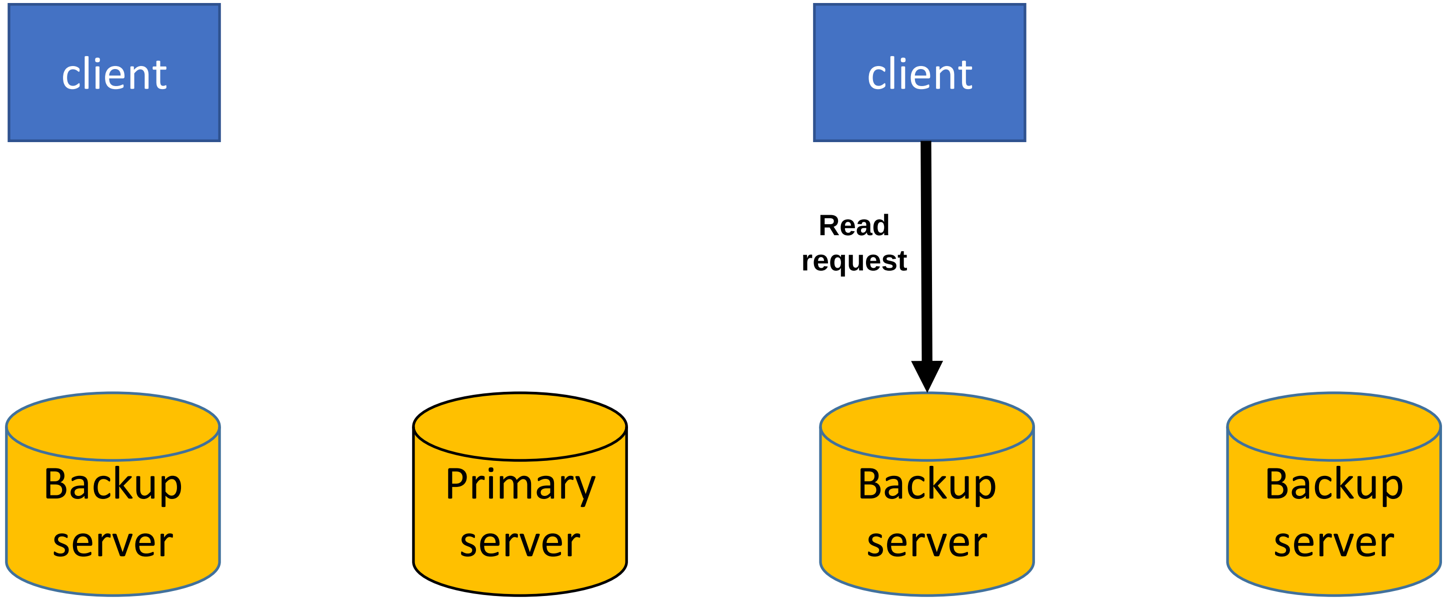
[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol



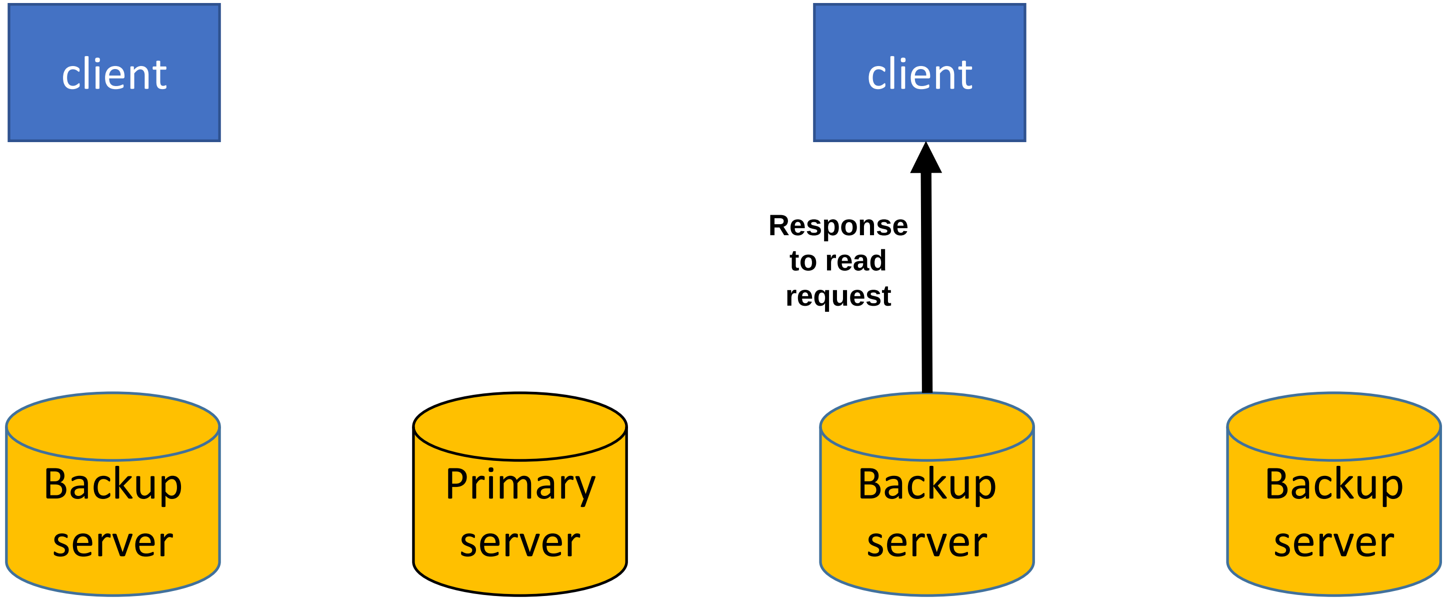
[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol



[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Remote-write protocol

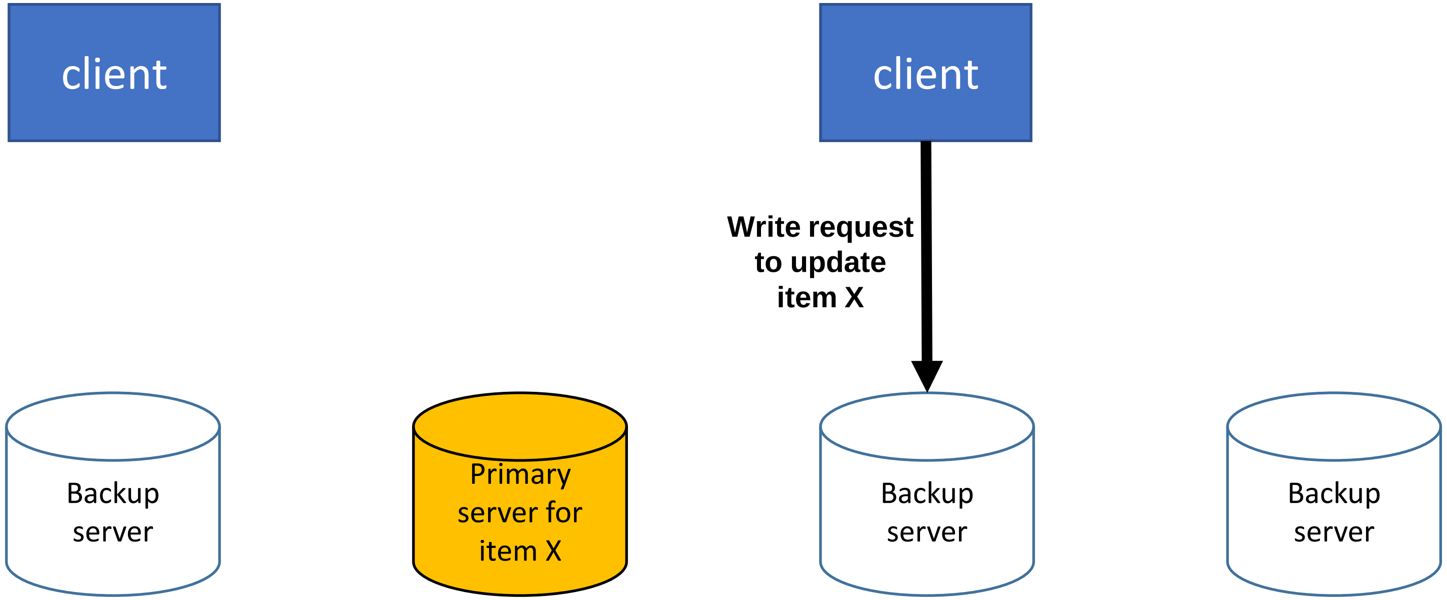


[1] p. 399, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Local-write protocol

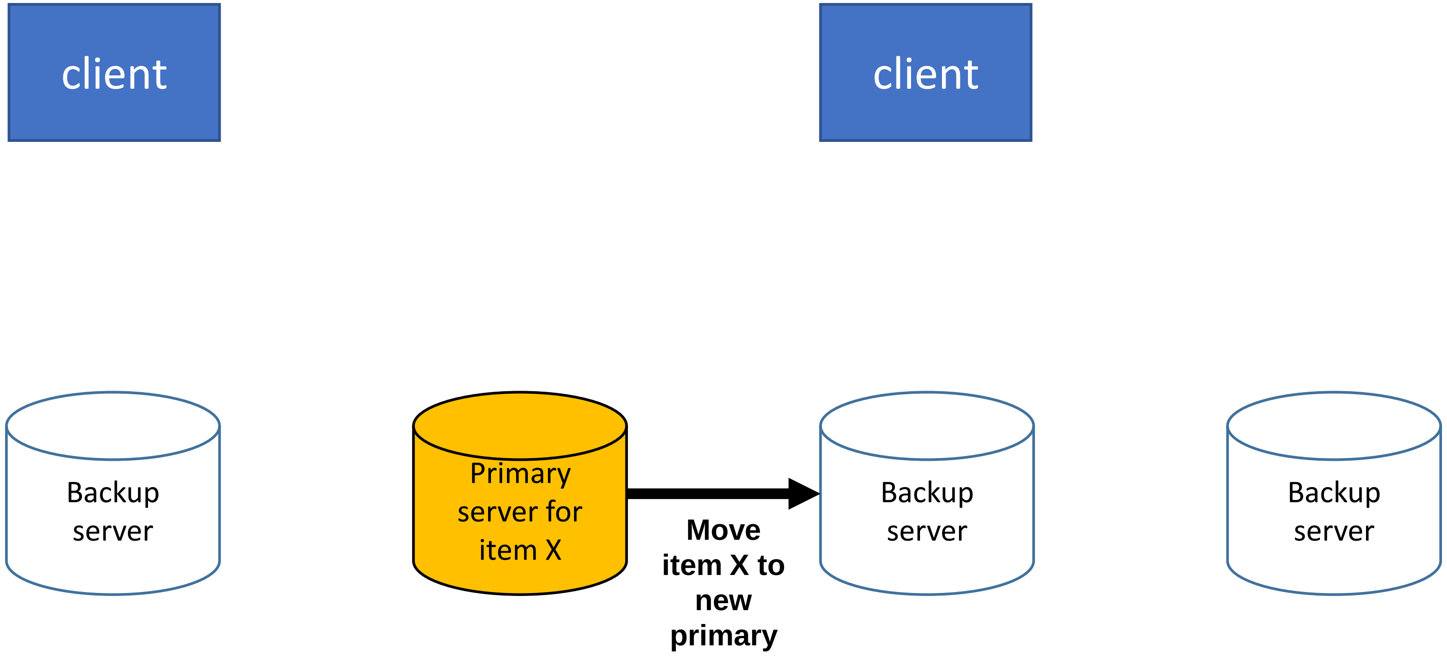
In the next example, notice how data item X migrates from one server to another server when performing an update on item X.

Local-write protocol



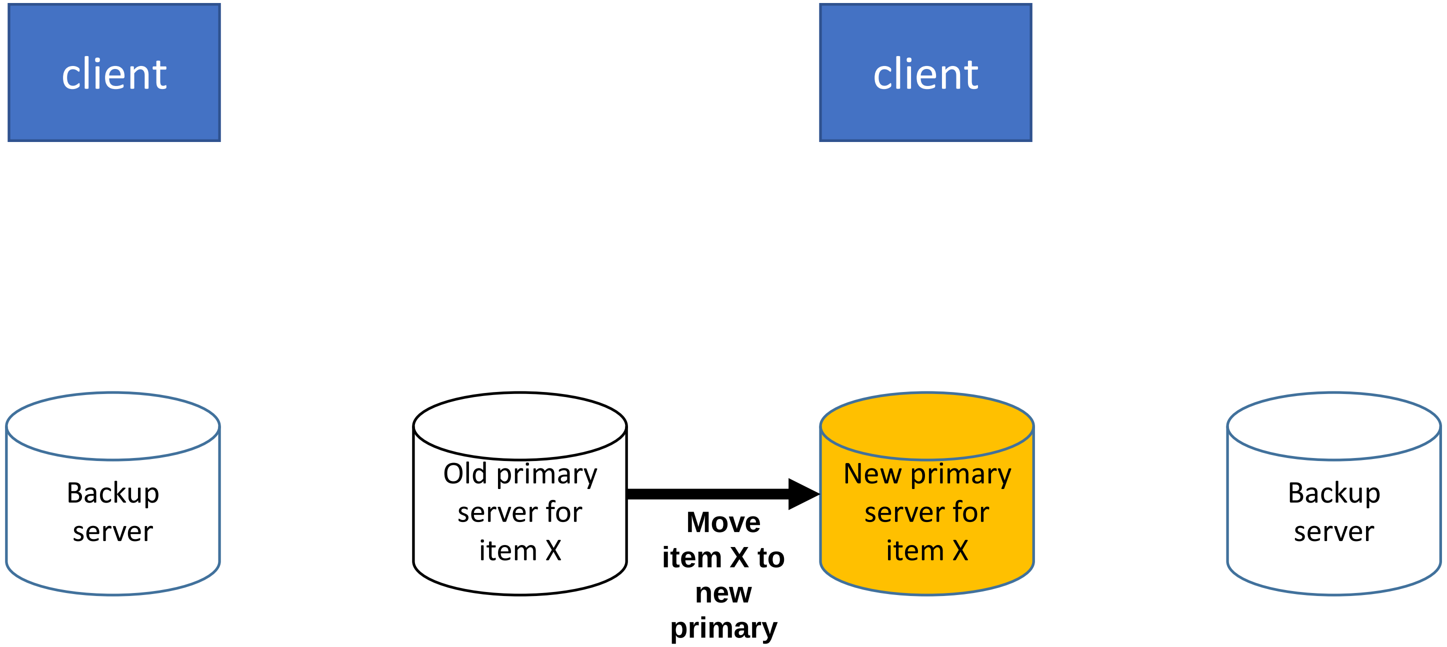
[1] p. 401, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Local-write protocol



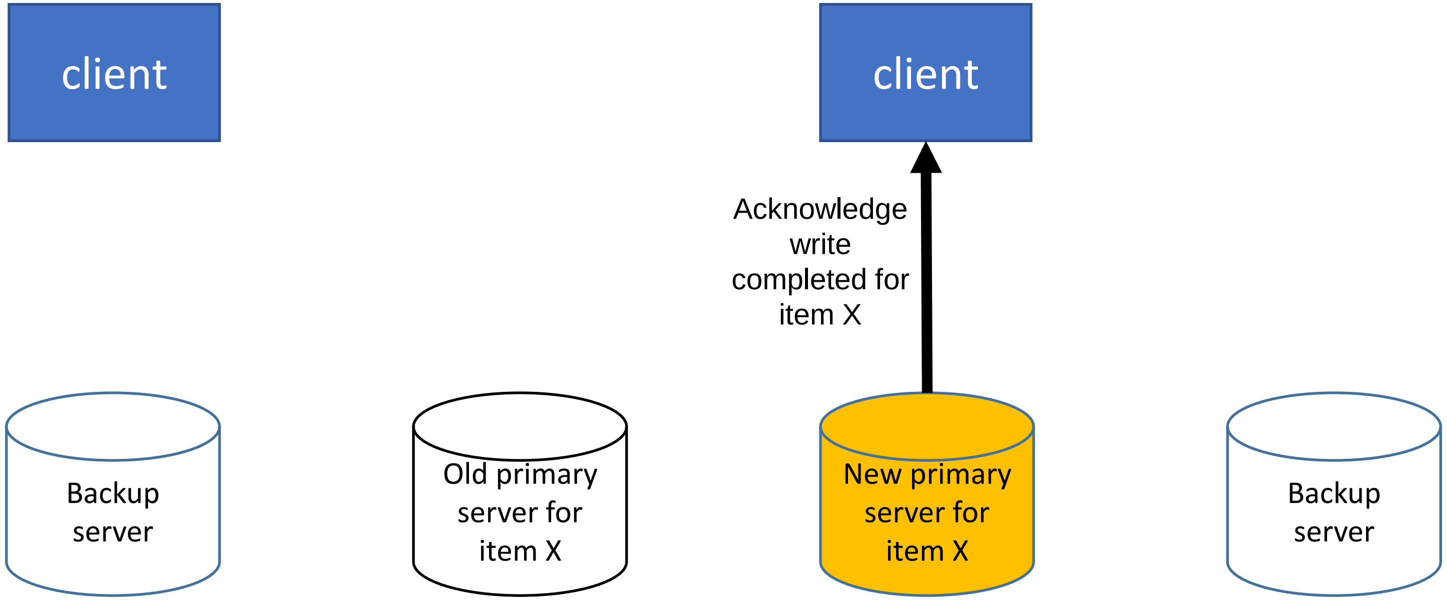
[1] p. 401, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Local-write protocol



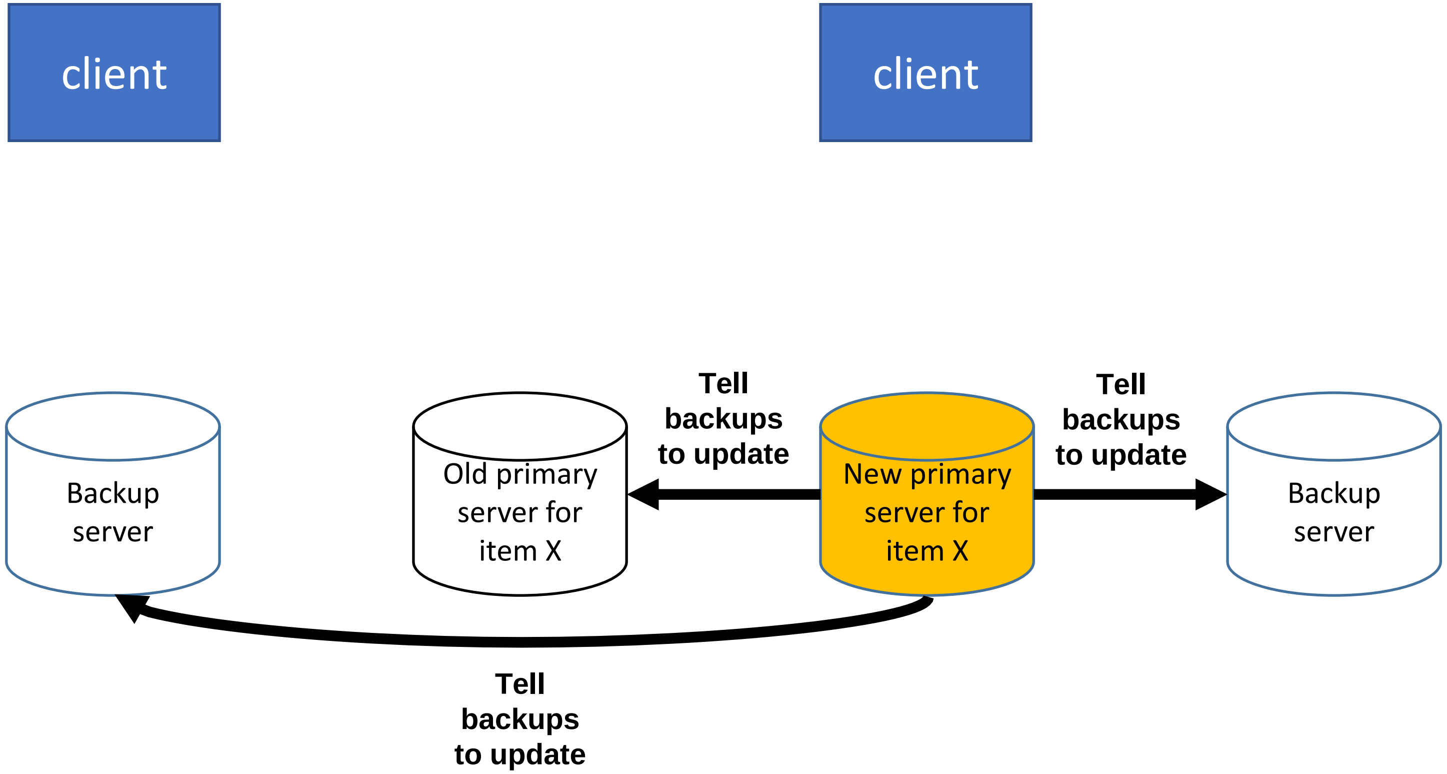
[1] p. 401, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Local-write protocol



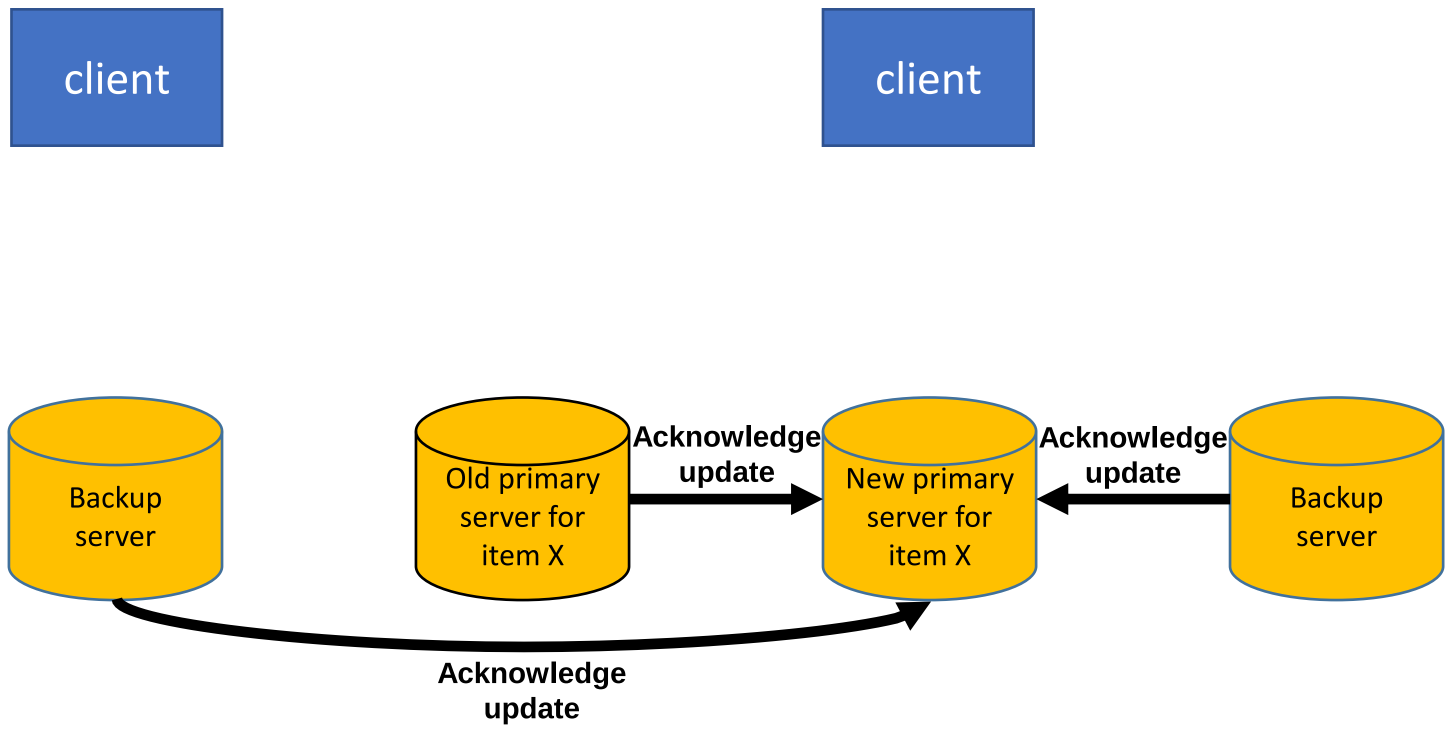
[1] p. 401, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Local-write protocol



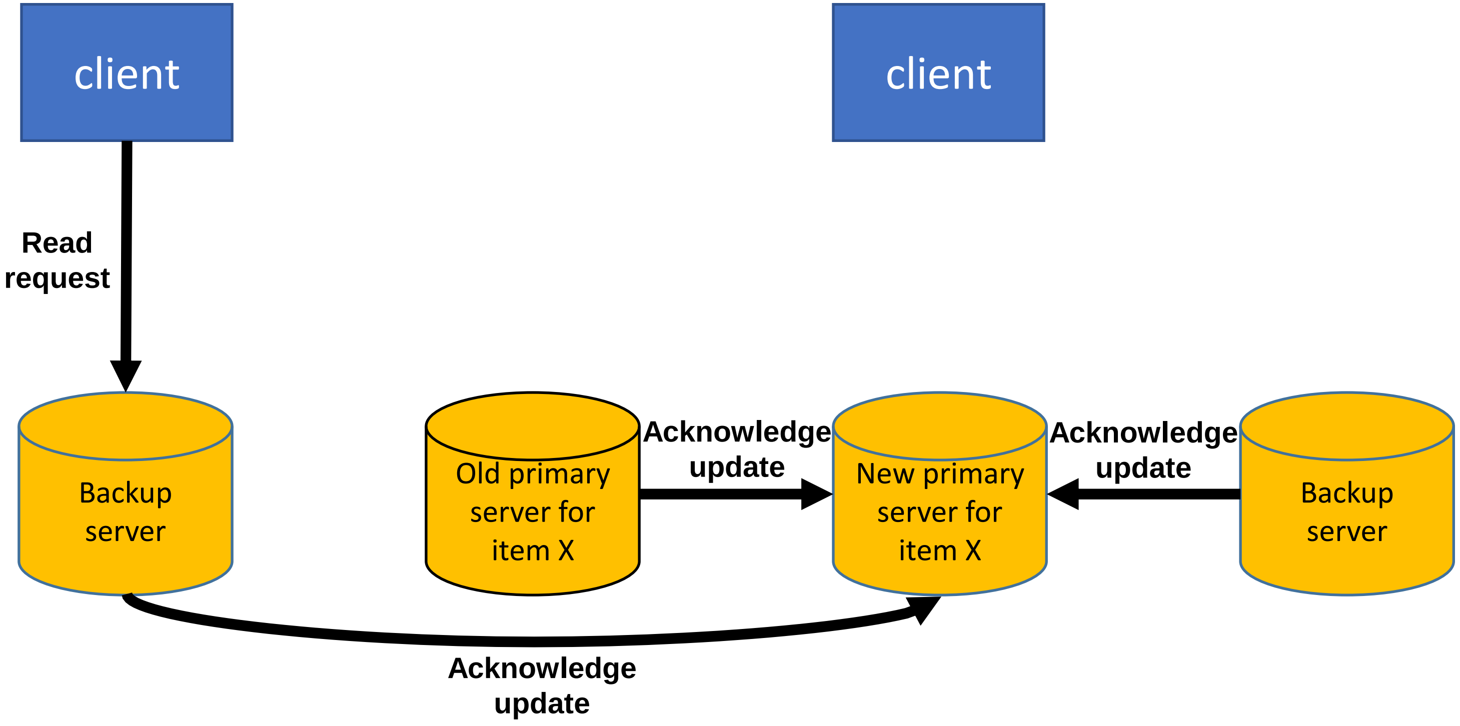
[1] p. 401, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Local-write protocol



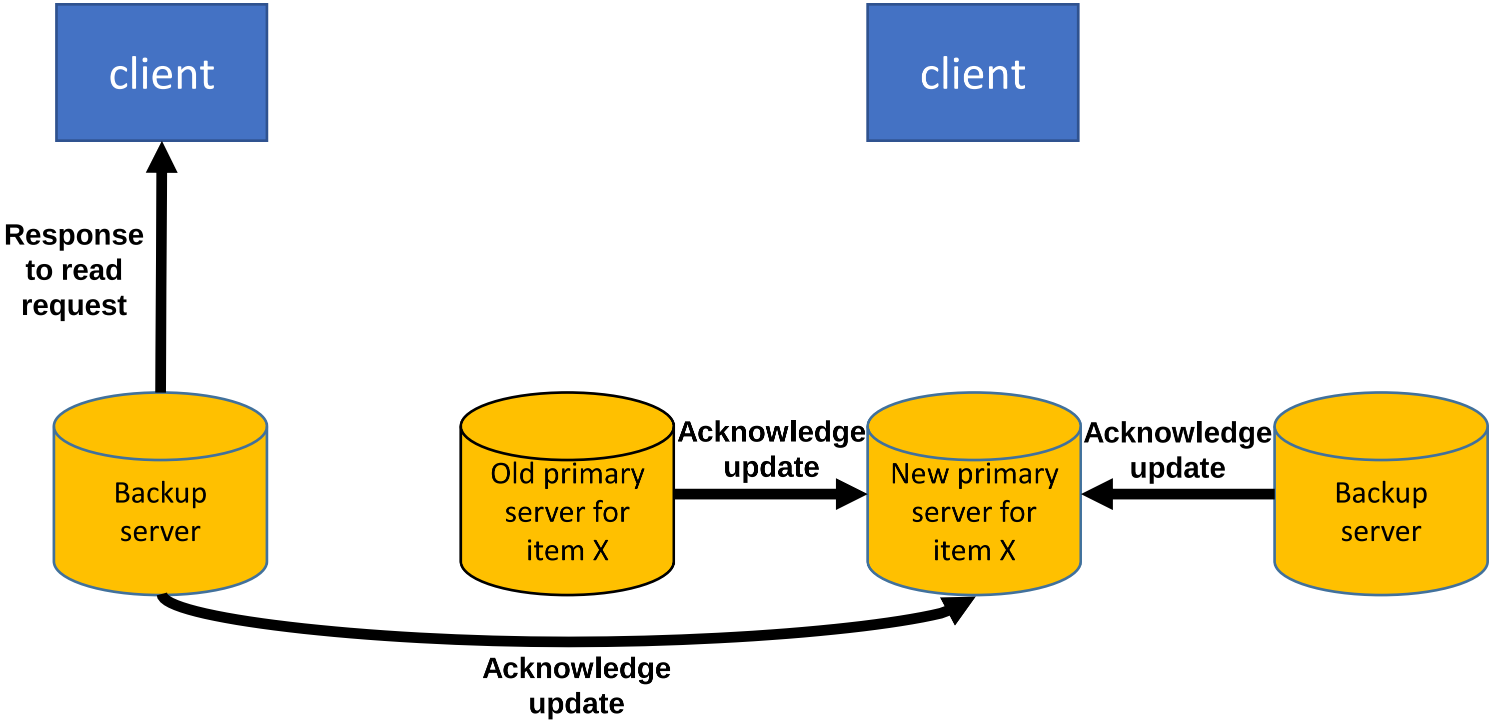
[1] p. 401, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Local-write protocol



[1] p. 401, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Local-write protocol



[1] p. 401, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Quorum-based protocol

In the following example, write requests outnumber read requests. Therefore, a write (an update) is performed.

Quorum-based protocol

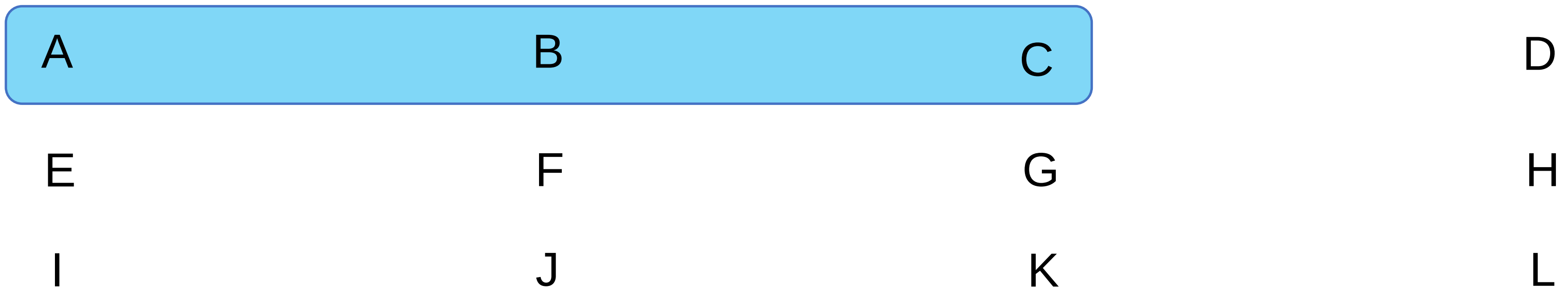
A	B	C	D
E	F	G	H
I	J	K	L

Objective: To prevent read-write conflicts.

N = 12

Source: p. 403, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

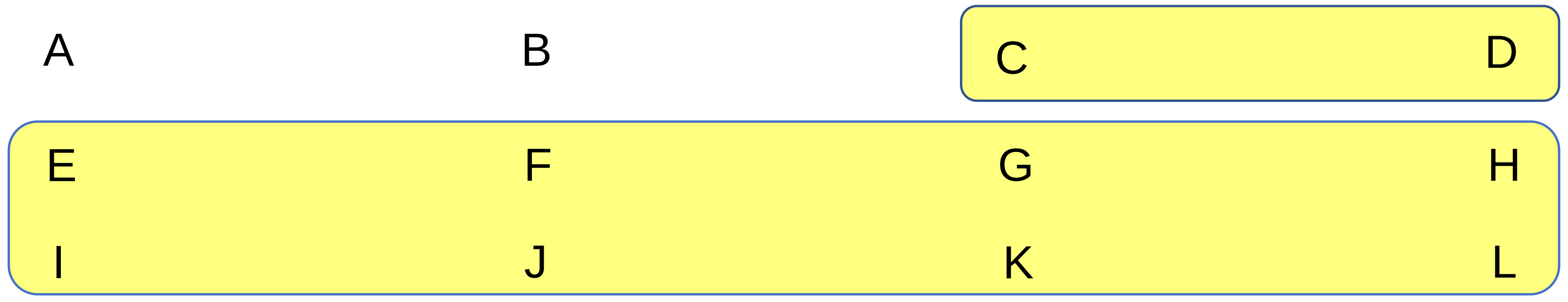
Quorum-based protocol



Blue denotes read quorum: N_R = 3

Source: p. 403, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

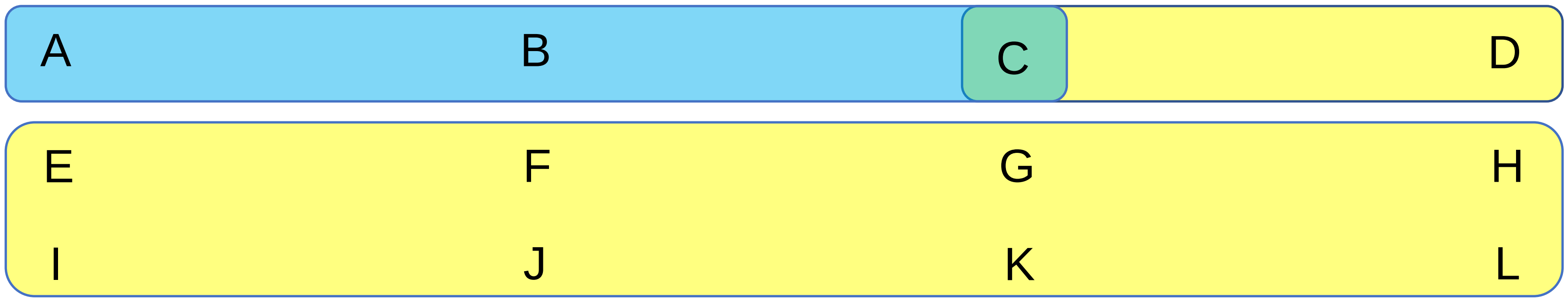
Quorum-based protocol



Yellow denotes write quorum: $N_W = 10$

Source: p. 403, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Quorum-based protocol



Green area (machine "C") denotes read-write conflict.

Constraints in the quorum-based protocol:

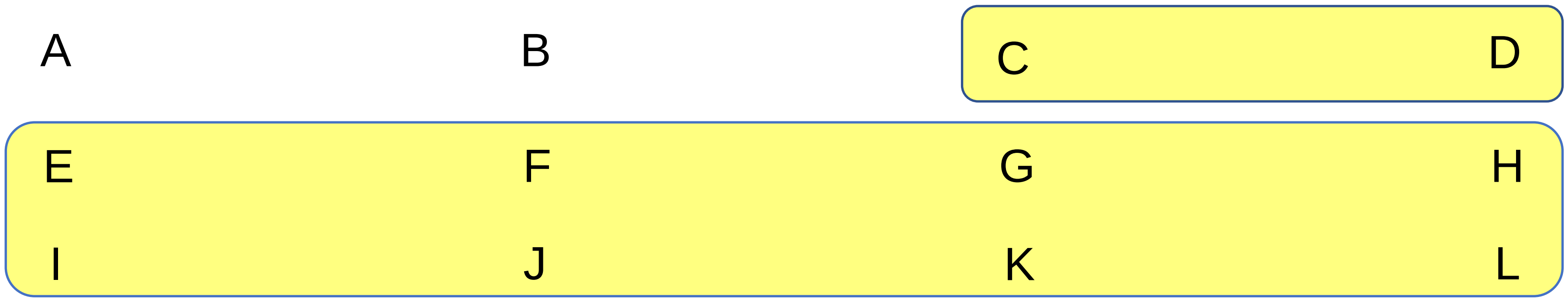
- $N_R + N_W > N$
- $N_W > N / 2$

Given $N_R = 3$, $N_W = 10$, and $N = 12$:

- The first constraint is satisfied $N_R + N_W > N$
- The second constraint is satisfied $N_W > N / 2$
- The write quorum wins since $N_W > N_R$

Source: p. 403, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Quorum-based protocol



The write quorum wins the vote.

Source: p. 403, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.

Quorum-based protocol

In the following example, there is an equal number of read and write requests. Since there was no majority vote, no operation is performed.

Quorum-based protocol

Green area (machine "H") denotes read-write conflict.

Given $N_R = 7$, $N_W = 6$, and $N = 12$:

- The first constraint is satisfied $N_R + N_W > N$
- The second constraint is **NOT** satisfied $N_W > N / 2$

Since the second constraint is not satisfied, nothing happens (i.e. neither read nor write wins the quorum). This avoids read-write conflict.

Source: p. 403, *Distributed Systems* (3rd edition) by Maarten van Steen and Andrew S. Tanenbaum.