# Fault Tolerance

## State Replication with Primary Backup

Pedro F. Souto (pfs@fe.up.pt)

May 12, 2021

# Replication: What and Why?

What? **Replication** is the use of the multiple instances/copies of processes/data, that we call **replicas**

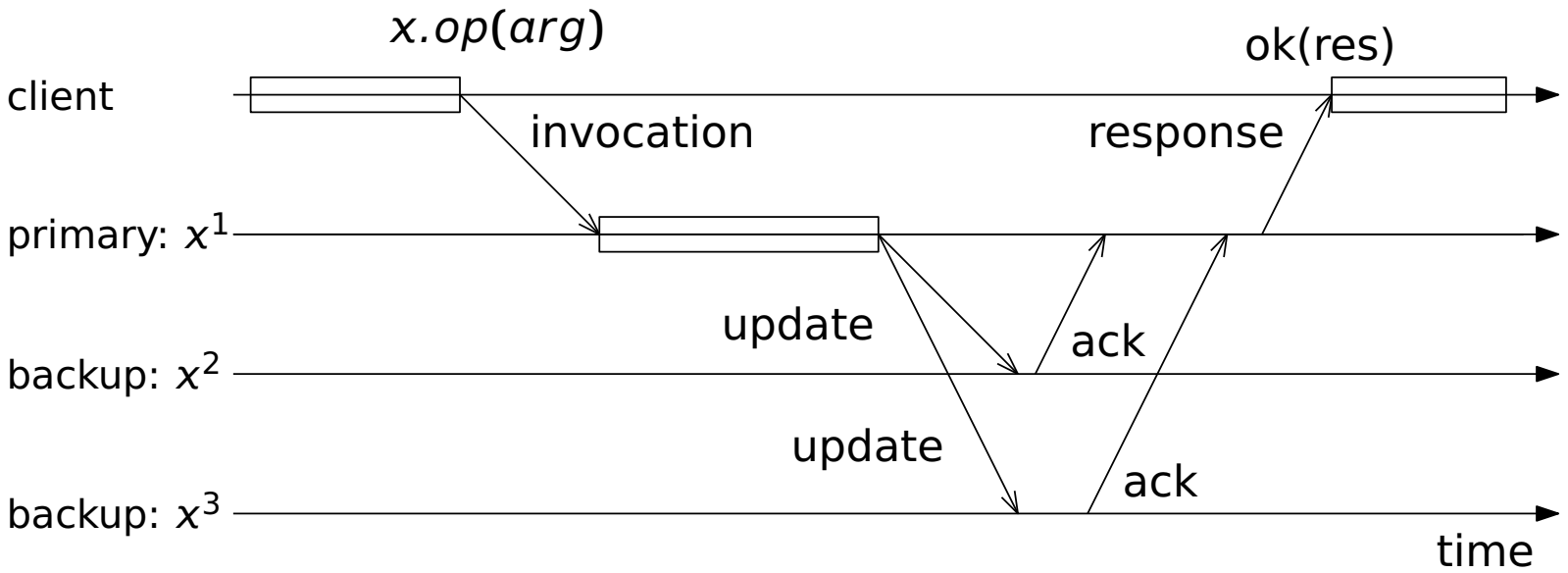
Why?

- Availability If one replica is down or unreachable, we can access the other replicas
- Scalability We can share the load among the replicas, and therefore handle higher loads by adding new replicas
- Performance By accessing a replica that is closer, a client will experience a lower latency

It is not easy to achieve all of these simultaneously

# Primary Backup Replication: Basic Algorithm

- One server is the **primary** and the remaining are **backups**
- The clients send requests to the primary only
- The primary executes the requests, updates the **state** of the backups and responds to the clients
  - After receiving **enough** acknowledgements from the backups
- If the primary fails, a **failover** occurs and one of the backups becomes the new primary.
  - May use leader election




Source: Guerraoui96

# Primary-Backup: Failure Detection and Failover

## Failure Detection

How? Usually sending:

either, I'M ALIVE messages periodically
or, acknowledgment messages

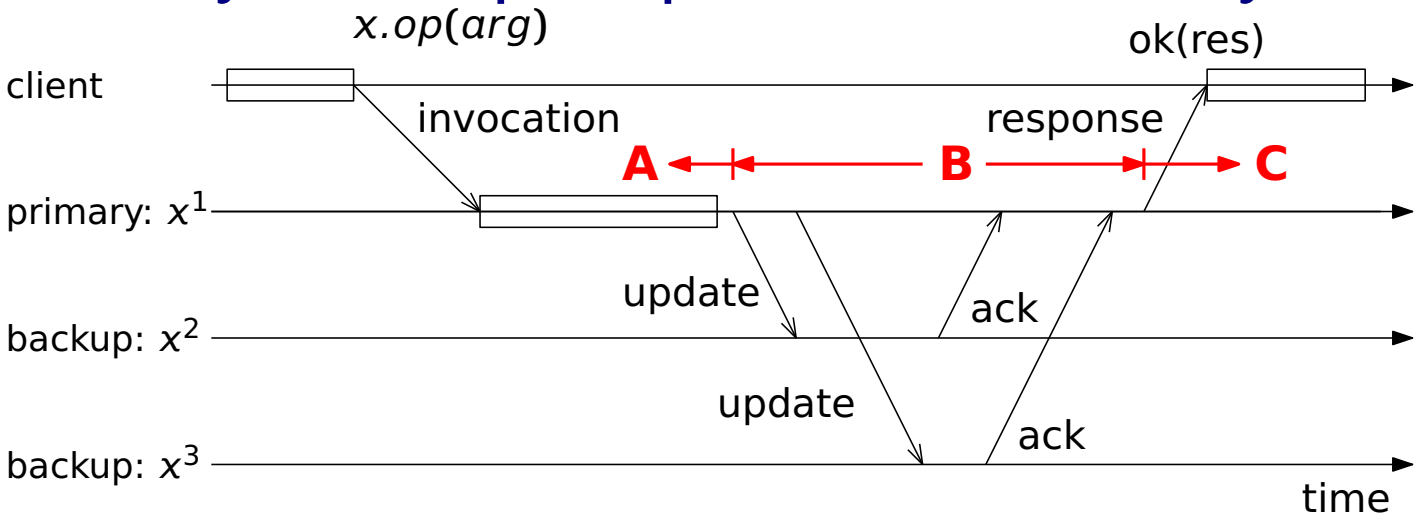
How reliable is it?

- It isn't, unless the system is synchronous ...

## Failover

- At least, "select" new primary

# Primary Backup Replication: Primary Failure (1/2)




Source: Guerraoui96

What if the primary fails?

Depends when the failure occurs

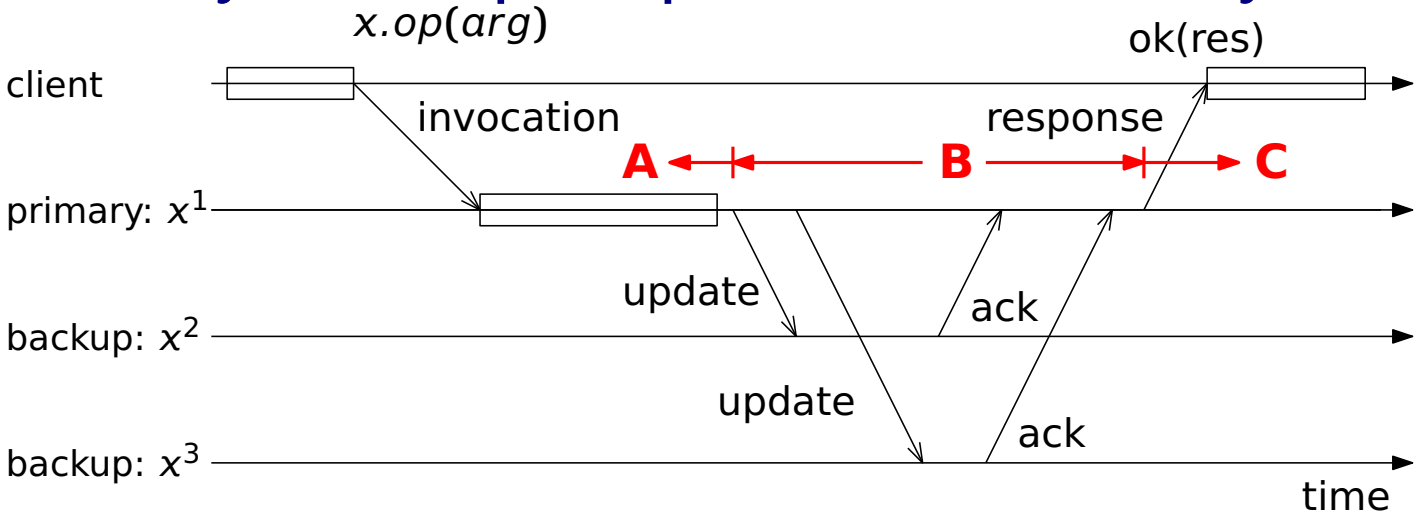
Primary crashes after sending response to client (C)

- Transparent to client
  - Unless response message is lost, and primary crashes before retransmitting it (case B)

Primary crashes before sending update to backups (A)

- No backup receives the update
- If client retransmits request, it will be handled as a new request by the new primary

# Primary Backup Replication: Primary Failure (2/2)




Source: Guerraoui96

Primary crashes after sending update (and before sending a response) (B). Need to consider different cases:

No backup receives update as in case A

All backups receive update

- If client retransmits request, new primary will respond
  - Update message must include response, if operation is non-idempotent

Some backups, not all, receive update

- Backups will be in inconsistent state

Must ensure update delivery atomicity

# Primary Backup Replication: Recovery

**Problem** when a replica recovers, its state is stale

- It cannot apply the updates and send ACKS to the new primary

**Solution** Use a **state transfer** protocol to bring the state of the backup in synch with that of the primary

**State transfer protocol** Two main alternatives

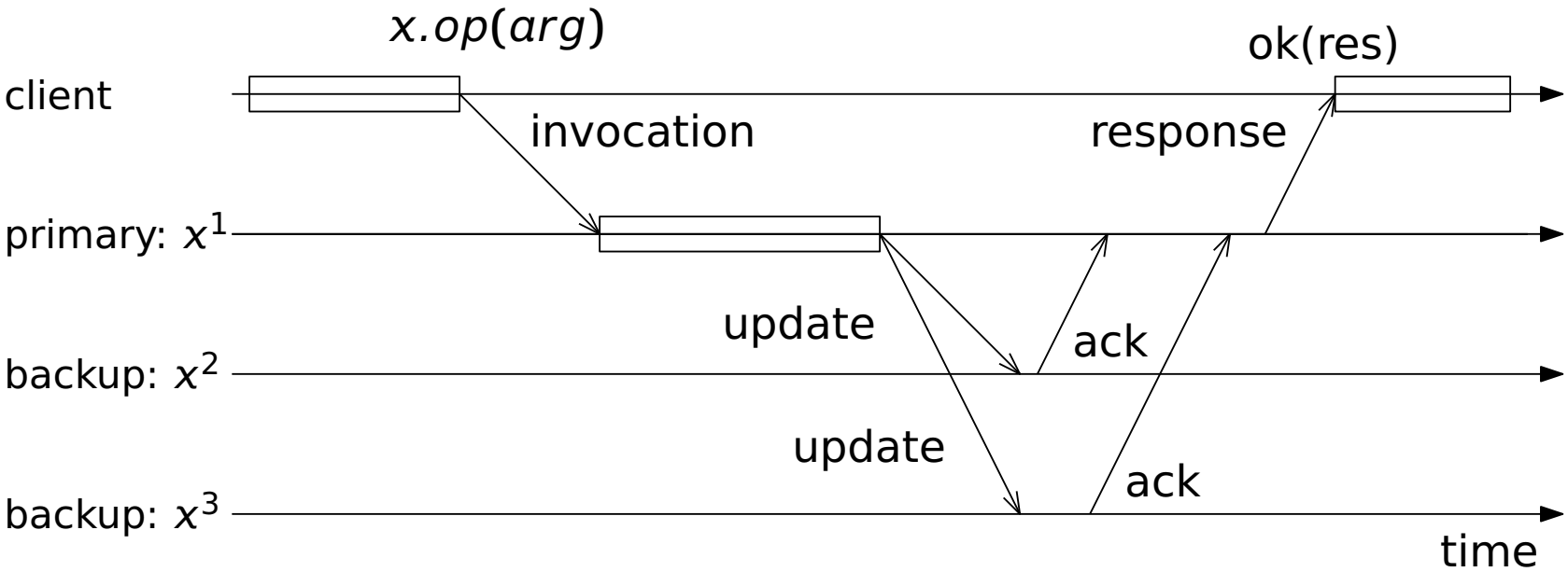
Resending missing UPDATES

Transferring the state itself

In both cases, the recovering replica can:

- Buffer the UPDATE messages received from the primary
- Process these UPDATES once its state is sufficiently up to date, i.e. reflects all previous UPDATES
  - Update the local replica
  - Send ACK to the primary

# Primary-backup fault-tolerance




**Question** What's the fault-tolerance?

**Answer** It depends on the failure model

**Crash-failure** Two faulty replicas

- In general, $n - 1$

**Omission** In this case, there is a need for a majority to prevent the existence of more than one primary at some time instant

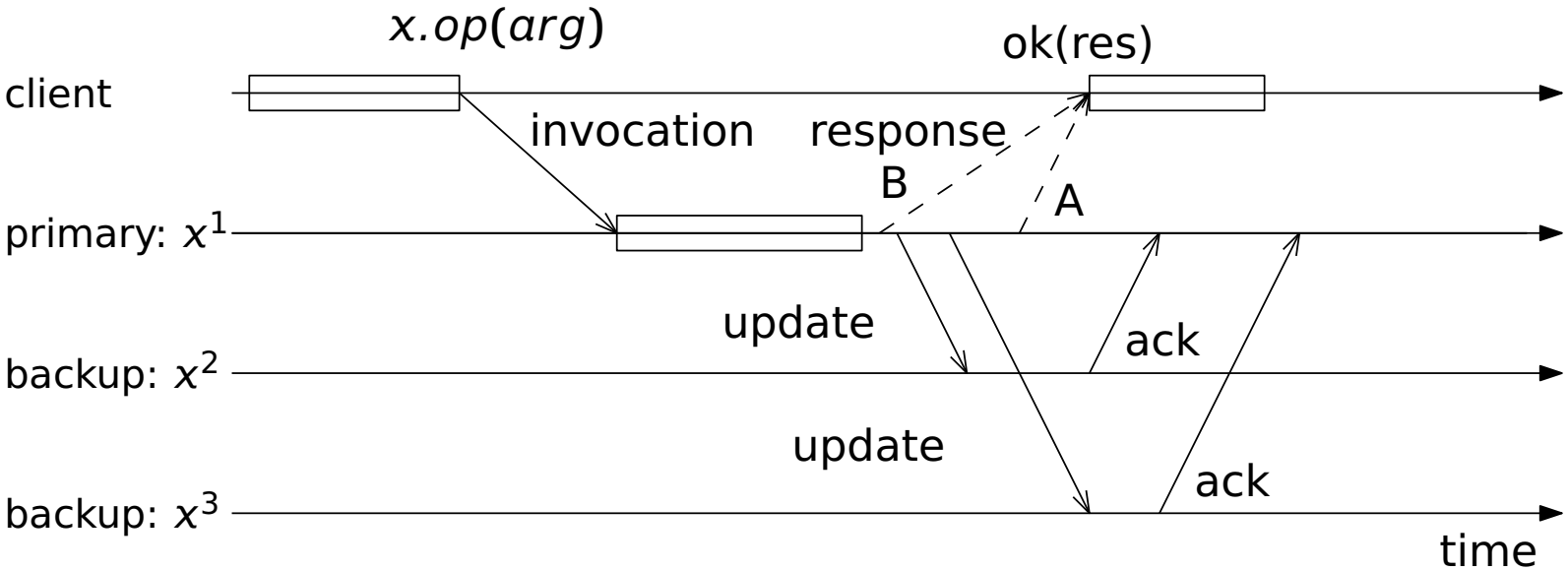
# Primary Backup Replication: Non-blocking Algorithm

**Observation** Waiting for backup acknowledgements increases latency

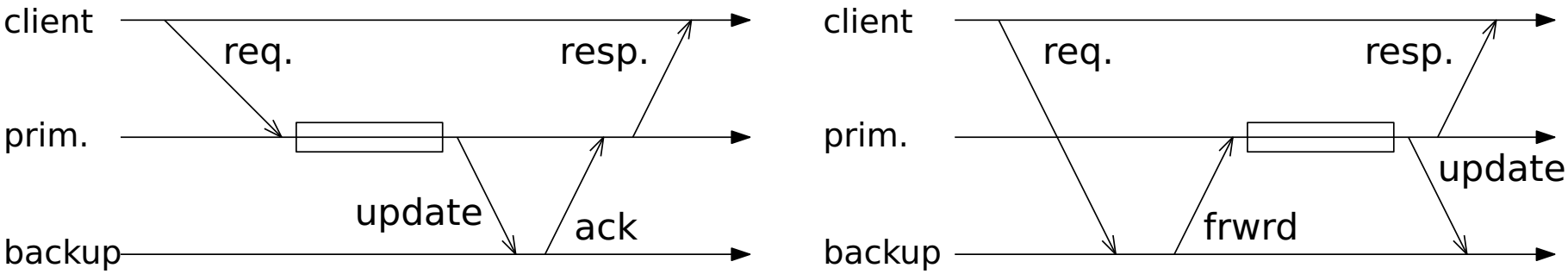
**Solution** Primary may send response to client before receiving `ack`'s from backups (A)

**Question 1** What is the trade-off?

**Question 2** What about sending response before the update to the backups (B)?



# Primary Backup, with one Backup (Alsberg and Day)




**Failure detection** need to prevent "split-brain"

- E.g. use redundant links between replicas
- Or else use the same network interface to communicate with clients and other replica
  - This way faults caused by the network interface should affect the communication with both the other replica and the clients

**Question** Can we change the order in which the update and the response are sent in the RHS image?

# Further Reading

- van Steen and Tanenbaum, *Distributed Systems*, 3rd Ed.
  - Section 7.5.2: Primary-Based Protocols
- R. Guerraoui and A. Schiper, *Software-based replication for fault-tolerance*, in IEEE Computer, (30)4:68-74 (April 1997)(in Moodle)