

Replication and Eventual Consistency

Dominic Duggan
Stevens Institute of Technolog

1

1

REPLICATION

2

2

Types of reliability

- Recoverability
 - Restart in a sensible state
- High availability
 - Operational during failure
 - **Replicate critical data**

3

3

Uses of replication

- High availability
- Share loads for scalability
- Lower latency, improve responsiveness

Save Discard Manual Failover Automatic Failover

Click on a location to add or remove regions from your Azure Cosmos DB account.

* Each region is billable based on the throughput and storage for the account. [Learn more](#)



4

4

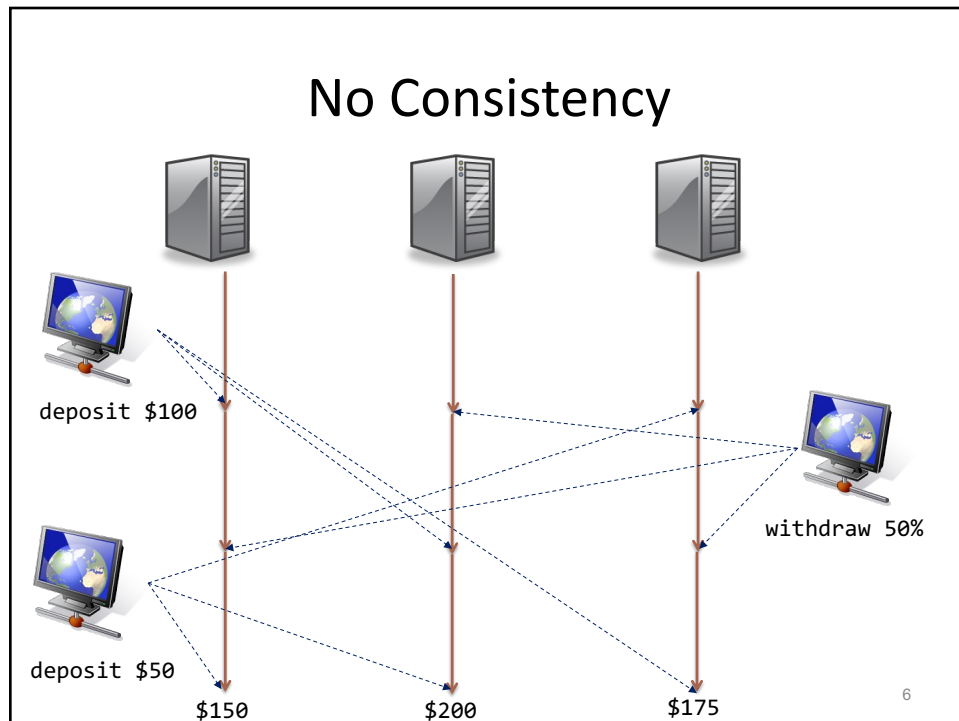
Replication for High Availability

- Active replication (State machine)
 - Peer-to-peer replicas
 - Each replica is **deterministic** state machine
 - **Operations** executed in same order on all replicas
 - All updates are totally ordered

5

5

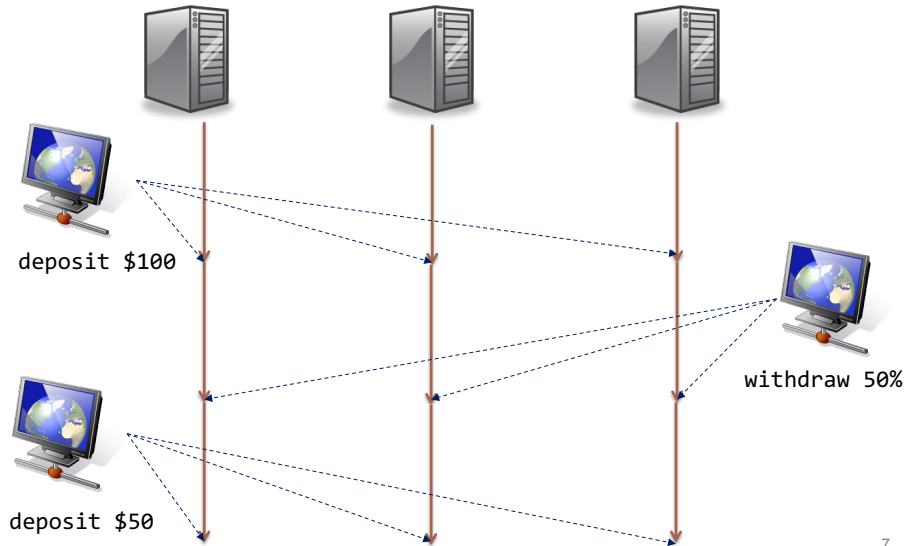
No Consistency



6

6

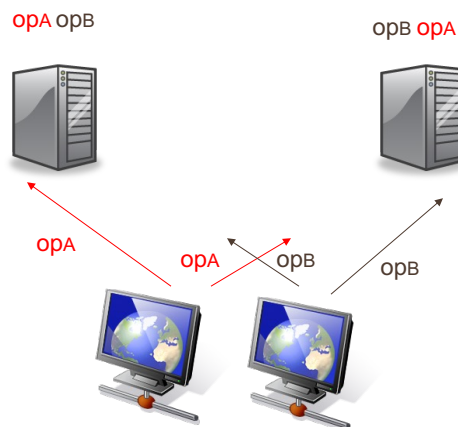
Strong Consistency



7

7

State-Machine Replication



- How to maintain a single order in the face of concurrent client requests?

8

8

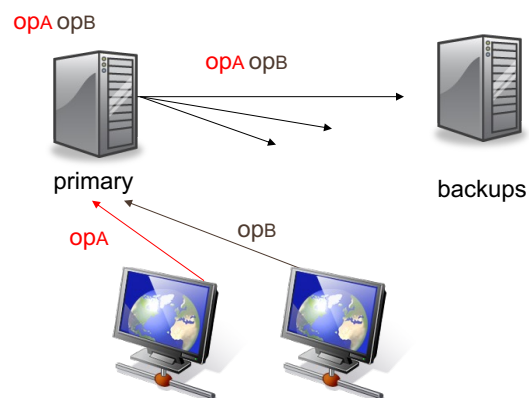
Replication for High Availability

- Passive replication (Primary-backup)
 - Primary replica with pool of backups
 - **Operation** executed on the primary
 - **Updates** performed in same order on all replicas
- Hybrid
 - Ex: Primary-backup where operation executed on all replicas

9

9

Primary-Backup Replication



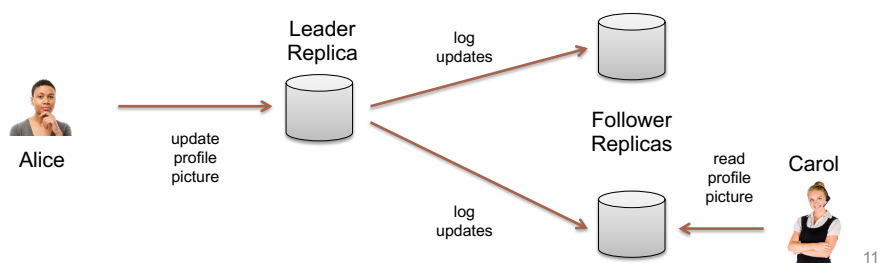
- Primary/backup: ensure a single order of ops:
 - Primary orders operations
 - Backups execute operations in order

10

10

Master-Slave Replication

- Leader sends log to backup
- Follower replays the log
 - applies committed transactions to its state
- If primary crashes, follower can take over



11

Challenges with Failover Recovery

- Detect leader has failed
 - Possible "split brain"
- Choose new leader
- Reconfigure for new leader

12

12

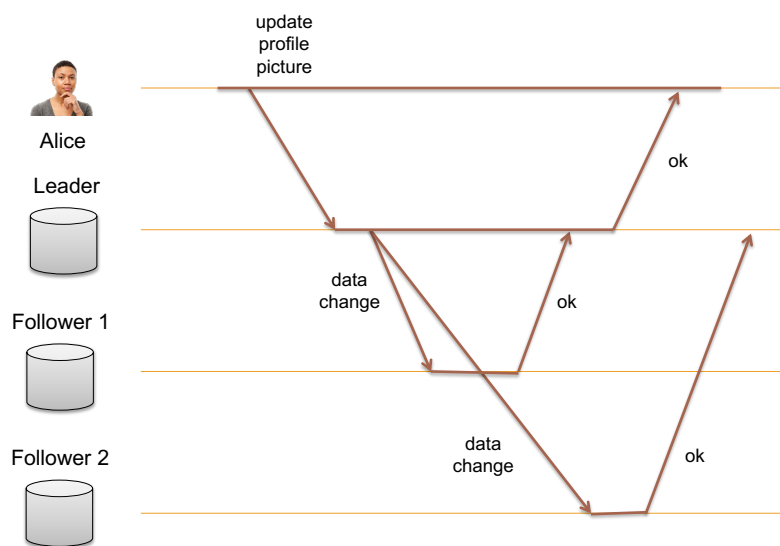
Real systems

- Primary-backup with logging to backup
- Synchronous: Wait for replicas to ACK
 - Problem: Latency
- Asynchronous: Don't wait for ACK
 - Backup may lag state of primary
- Semi-synchronous

13

13

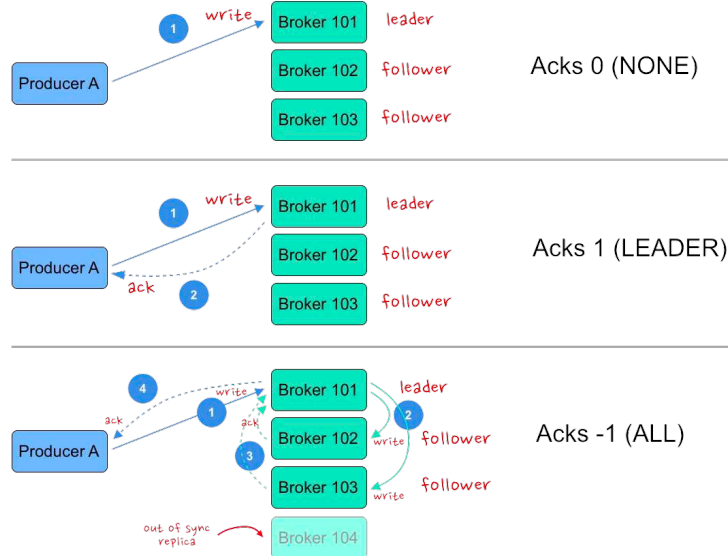
Semi-Synchronous



14

14

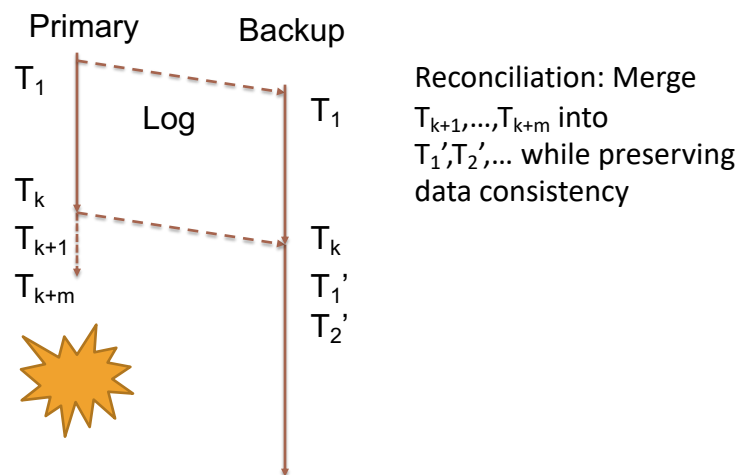
Kafka Producer Guarantees



15

15

Reconciliation



16

16

Reconciliation

- Fix transactions impacted by loss of tail of log
 - Apply the missing updates
 - Cascaded rollback
 - Worst case: human intervention
- Similar to compensations in long-lived transactions

17

17

Transactional Replication

- One-Copy Serializability (1SR)
 - Effect of transactions on *replicated* data items are same as if performed serially on *single* data items
 - Key: Failures and recoveries must be serialized with respect to transactions
 - Reason: Updates only performed on *available copies*

18

18