HQ Replication: A Hybrid Quorum Protocol for Byzantine Fault Tolerance

James Cowling, Daniel Myers, Barbara Liskov, Rodrigo Rodrigues, and Liuba Shrira

ECS265 Paper Review Presentation

Presenters: Shuyan Dai, Yuang Ma, Kaixin Xiao, Jiayu Jiang

Overview

- Introduction
- Quick Recap
- HQ Replication
  - System Architecture
  - Phases
  - Non-contention Example
  - Contention Example
  - Other Details
  - Optimizations
- Analysis
- Experimental Evaluation
- Conclusions and Recommendations

Introduction

- What are BFT and Q/U (quorum-based) protocols?
- What are their strengths and limitations?
- How can we do better?

Quick recap on BFT SMR

Byzantine-fault-tolerant state machine replication

BFT

- $3f + 1$ replicas for f failures but high latency
- Quadratic cost of inter-replica communication unnecessary when no contention
- All-to-all communications among replicas cause scalability issues

Q/U

- Fault-scalable: no steep performance/latency degradation with more server faults tolerated
- Requires a large number of replicas: $5f + 1$ for f failures
- Exponential back-off: poor performance during write contention

Hybrid Quorum (HQ)

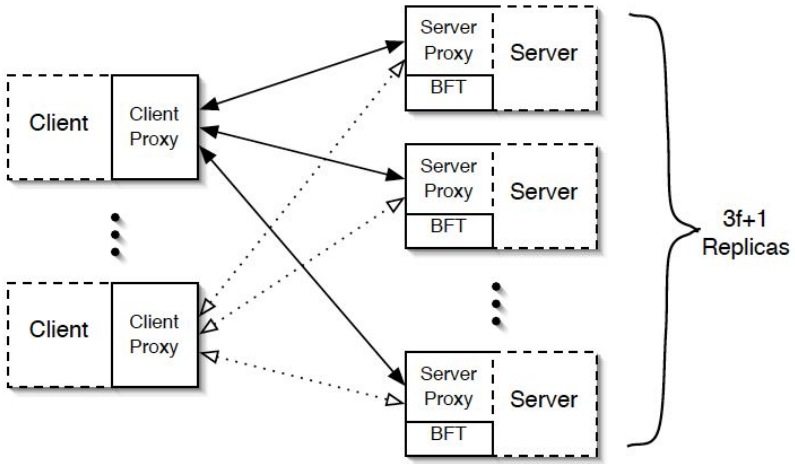
- Best of both worlds
- Contention ? (BFT SMR) : (quorum-based)
- $3f + 1$ replicas for f failures and scalable performance

HQ Replication

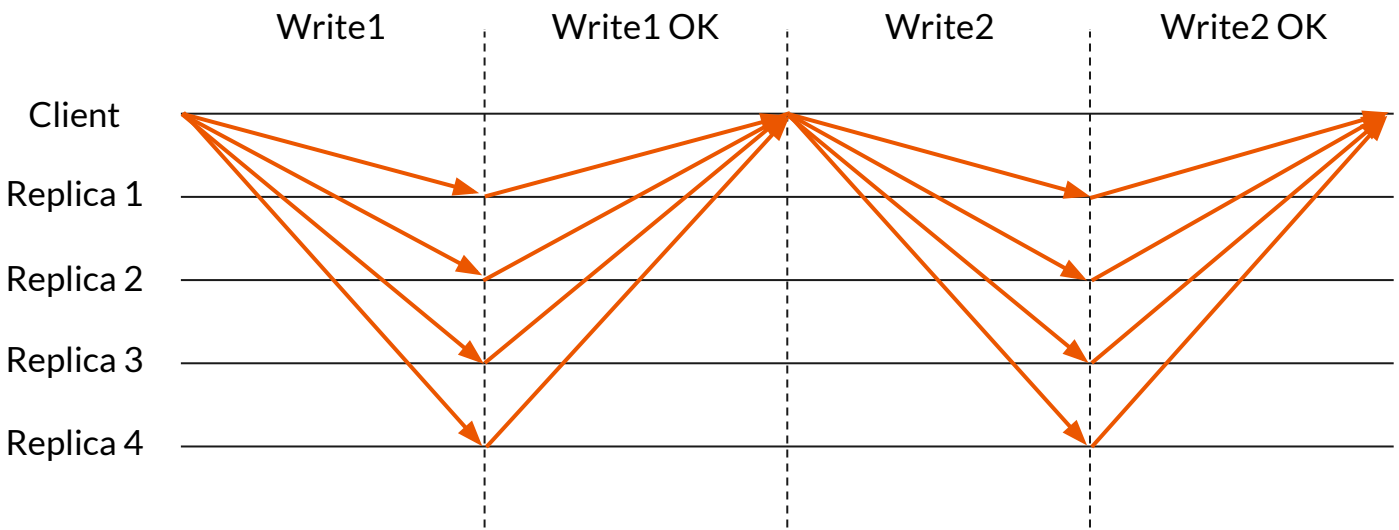
- System Architecture
- Normal Case (no contention)
- Contention Resolution (BFT gets involved)

System Architecture

- Client: application code calls an operation through client proxy.
- Server: server code gets invoked by server proxy in response to client (servers also run BFT state machine replication protocol).
- One-phase read.
- Two -phase write.



HQ Replication



HQ Replication Write Protocol

- Phase1:

  (1) client obtains timestamp grant from each replica.

- Phase2:

  (1) client forms certificate from 2f + 1 matching grants.

  (2) Sends to replicas to complete write.

- Operations are executed at given sequence number (Timestamp).

HQ Replication: Grant, Certificate and Replica State

Grant:

- Client ID
- Object ID
- Hash over requested operation
- Sequence Number (Timestamp)
- Replica signature

Certificate:

- Quorum (2f + 1) matching grants.
- Proves quorum of replicas agree to ordering of operation.

Replica State:

- Multiple independent objects
- State per object:

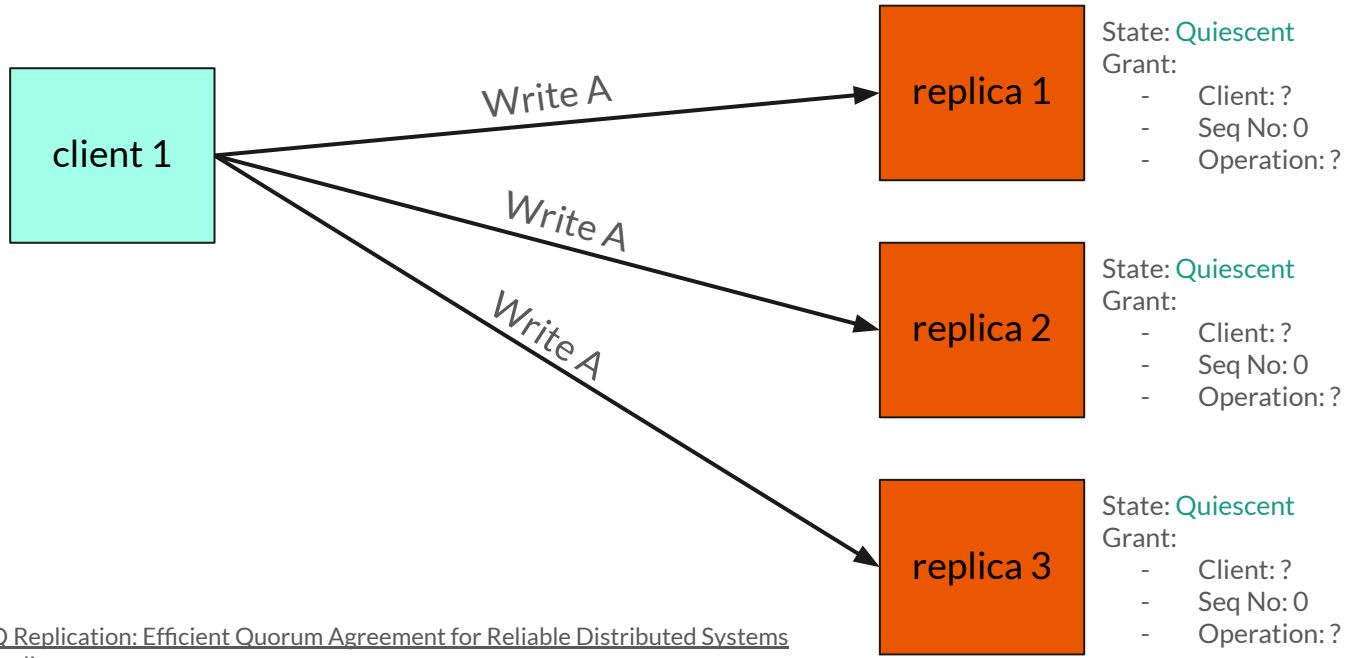
  Active: Write in progress

  Quiescent: No current write operation

Normal Case

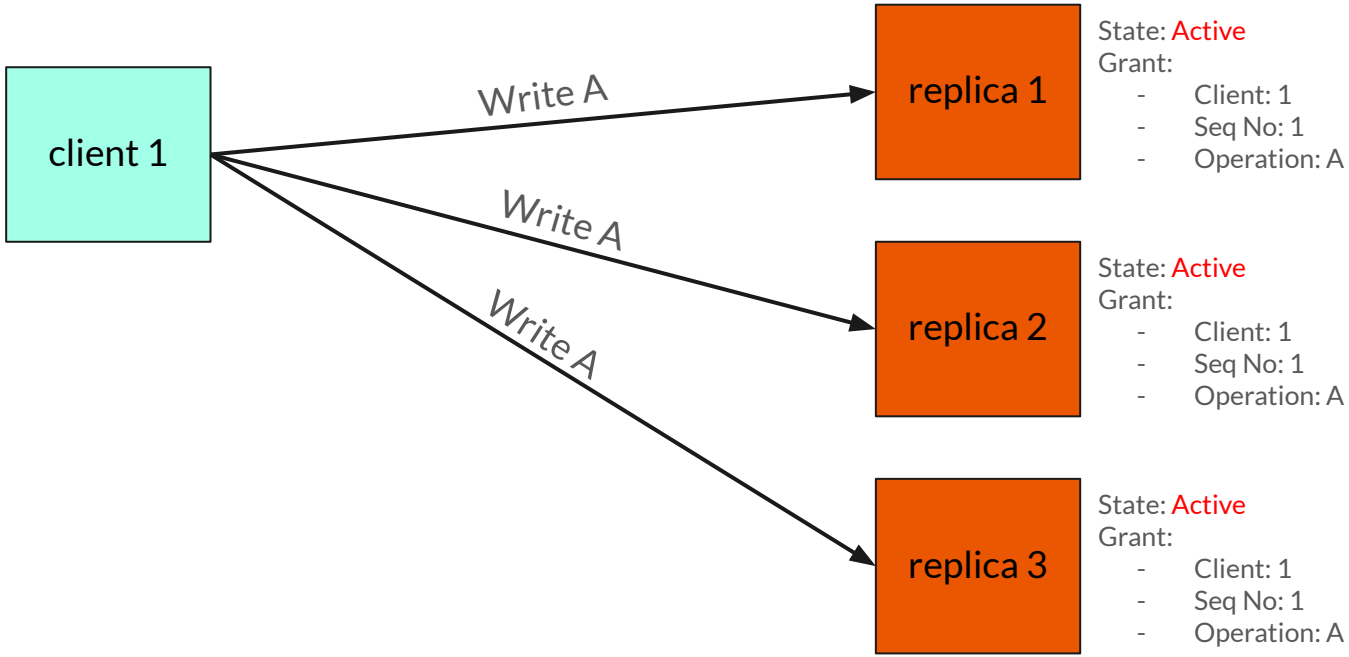
Isolated Write

Normal Case-Isolated Write

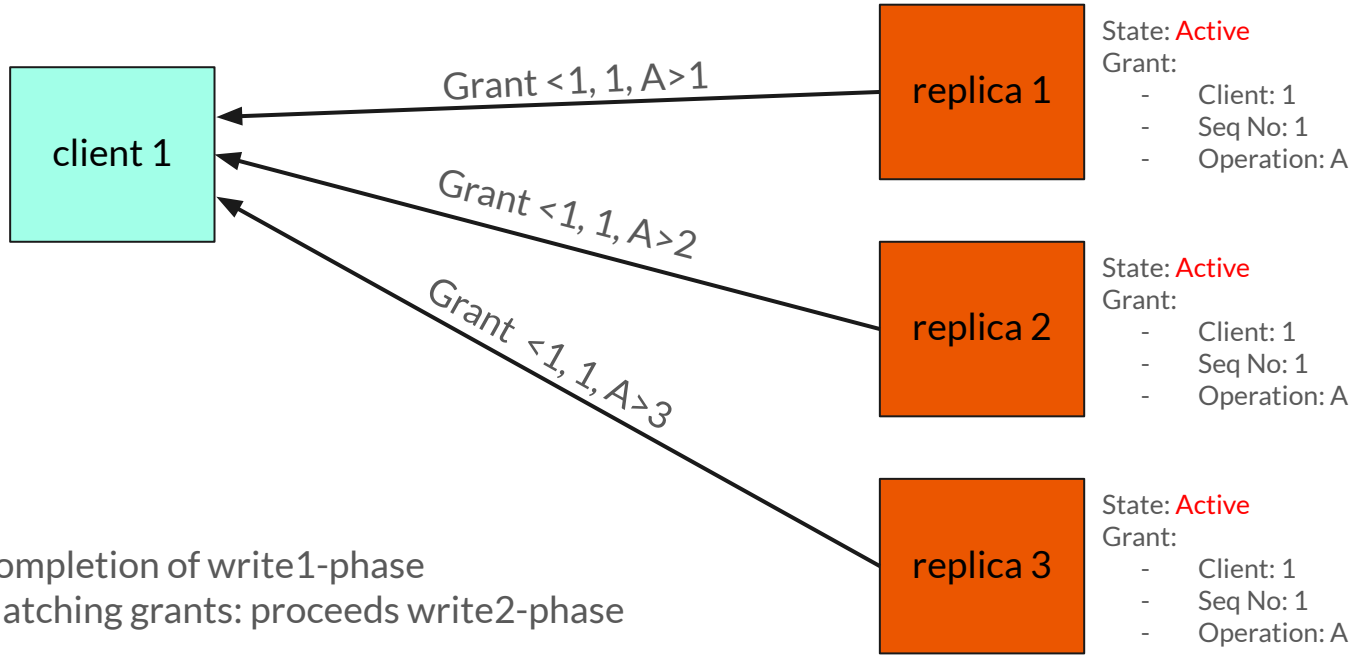


Citation: HQ Replication: Efficient Quorum Agreement for Reliable Distributed Systems
by James Cowling

Normal Case-Isolated Write

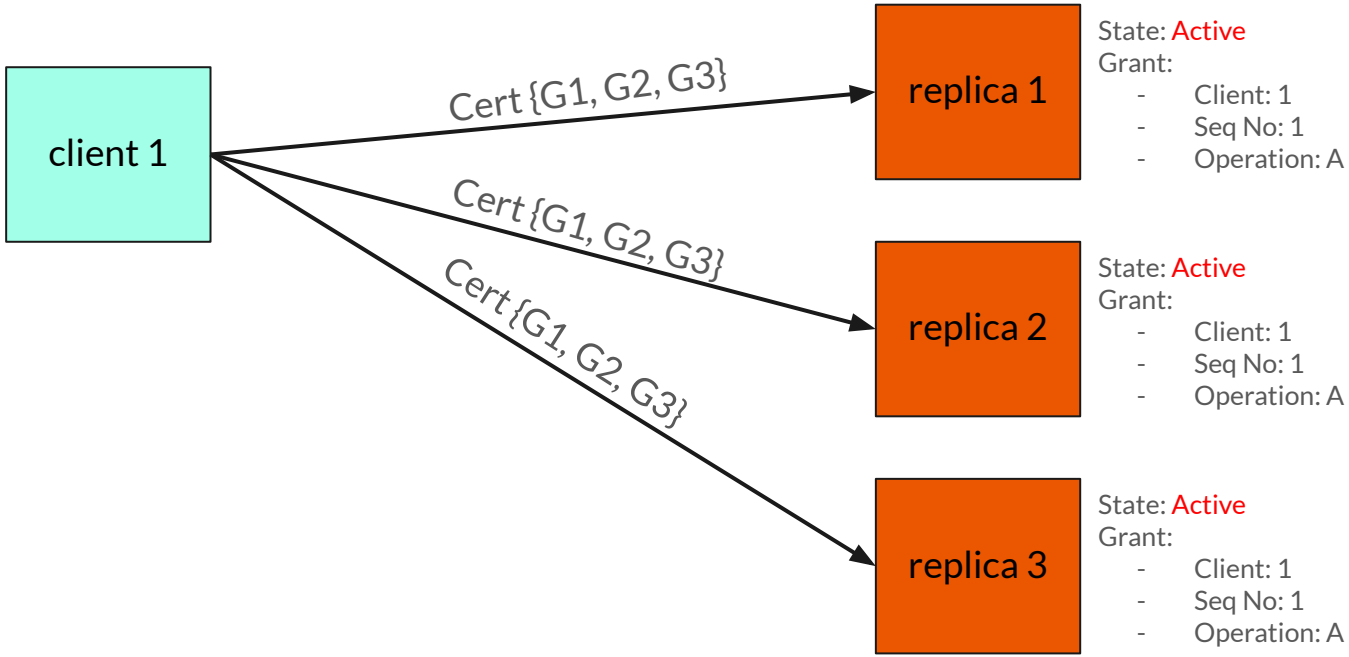


Normal Case-Isolated Write

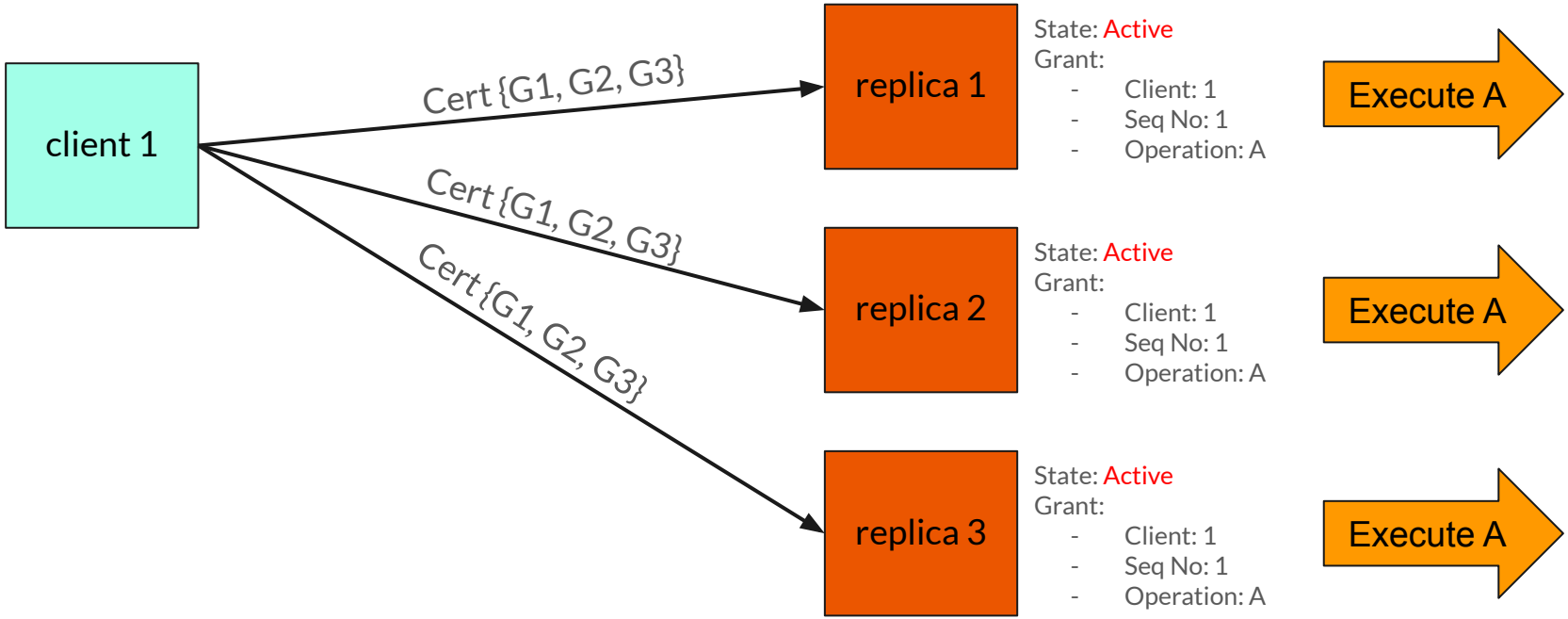


Completion of write1-phase
Matching grants: proceeds write2-phase

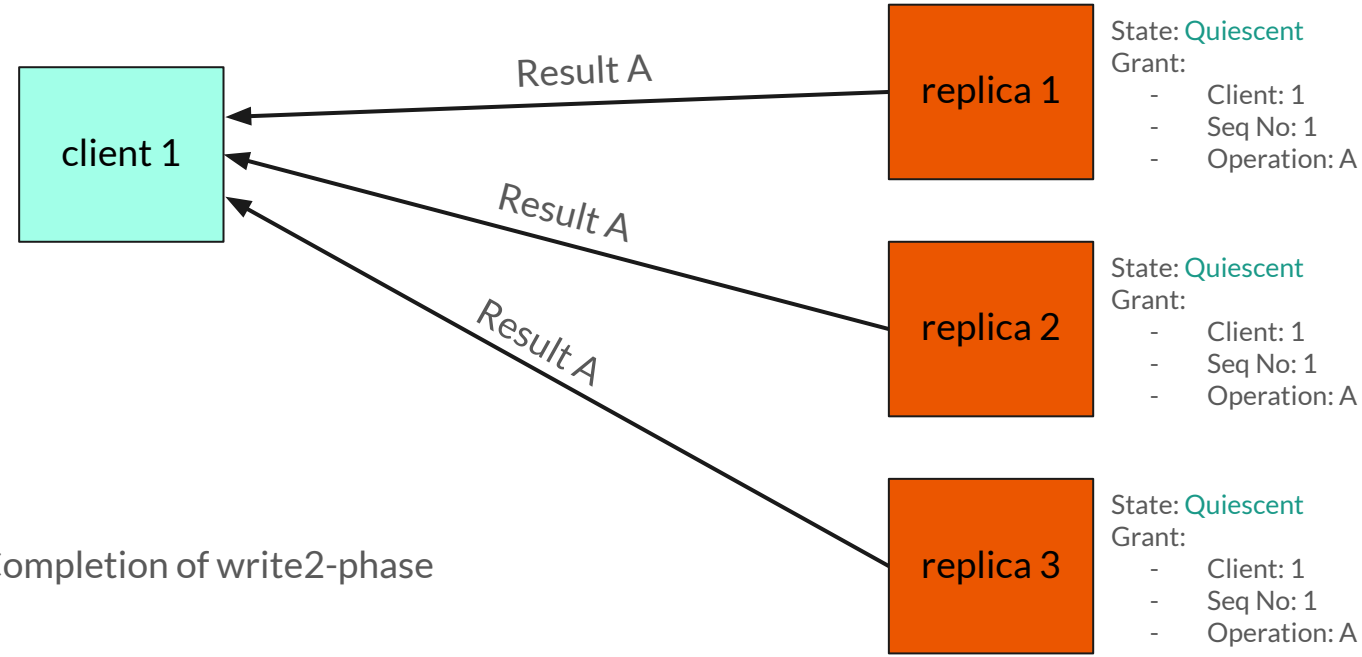
Normal Case-Isolated Write



Normal Case-Isolated Write



Normal Case-Isolated Write

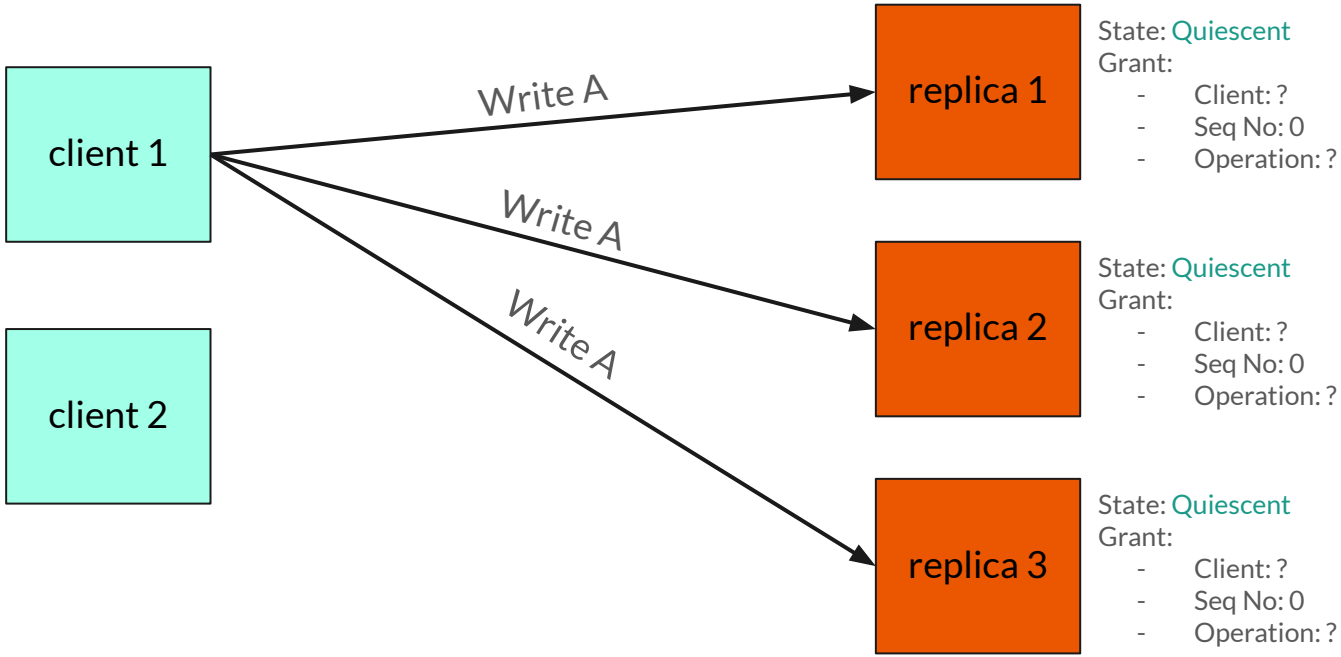


Completion of write2-phase

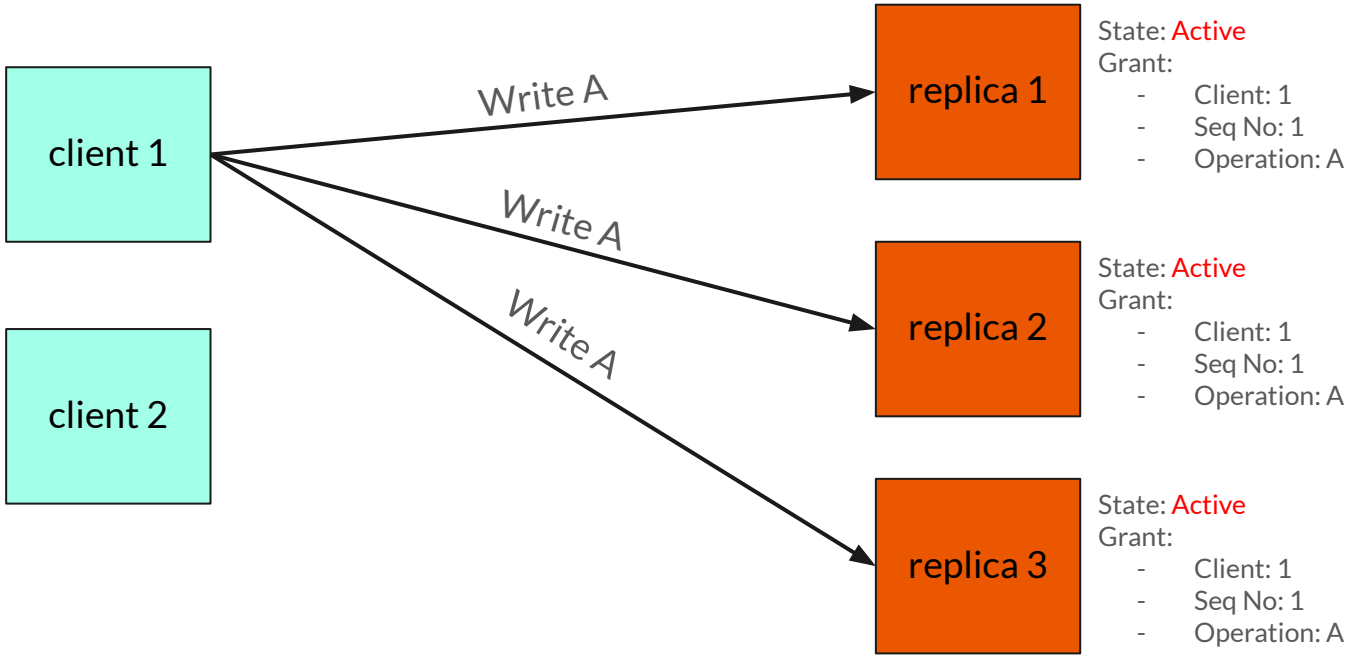
Normal Case

Incomplete Write

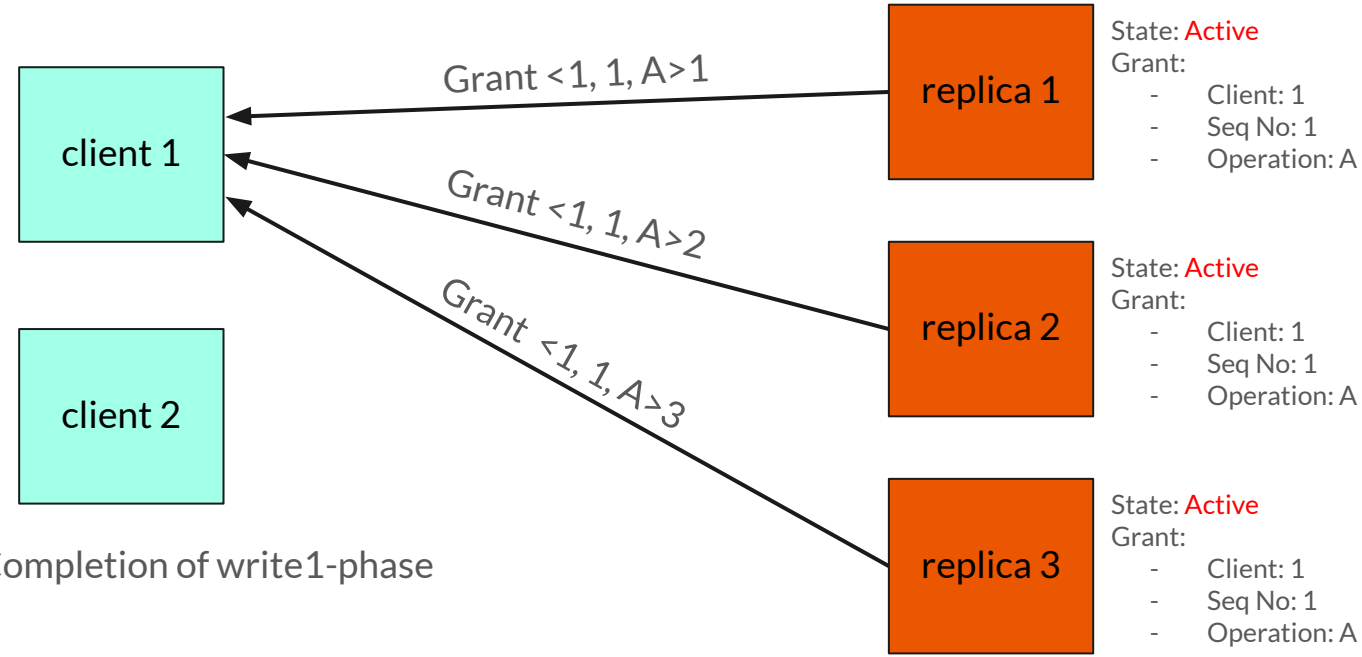
Normal Case-Incomplete Write



Normal Case-Incomplete Write

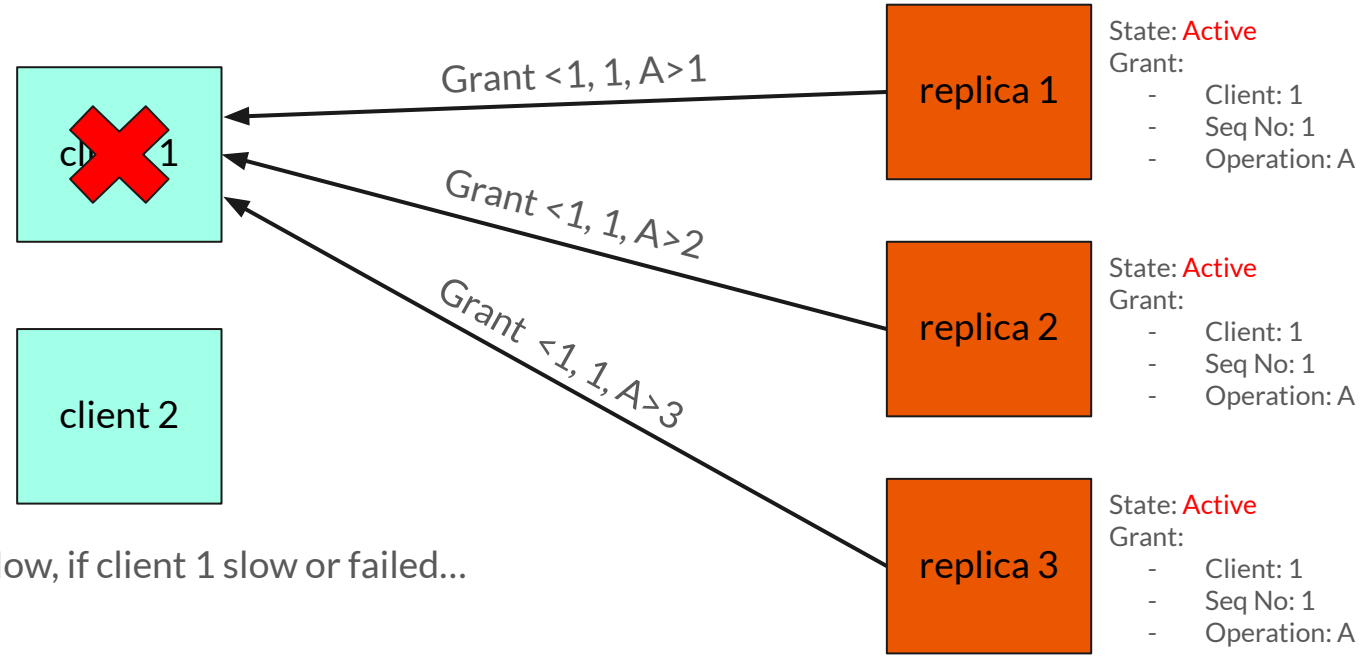


Normal Case-Incomplete Write



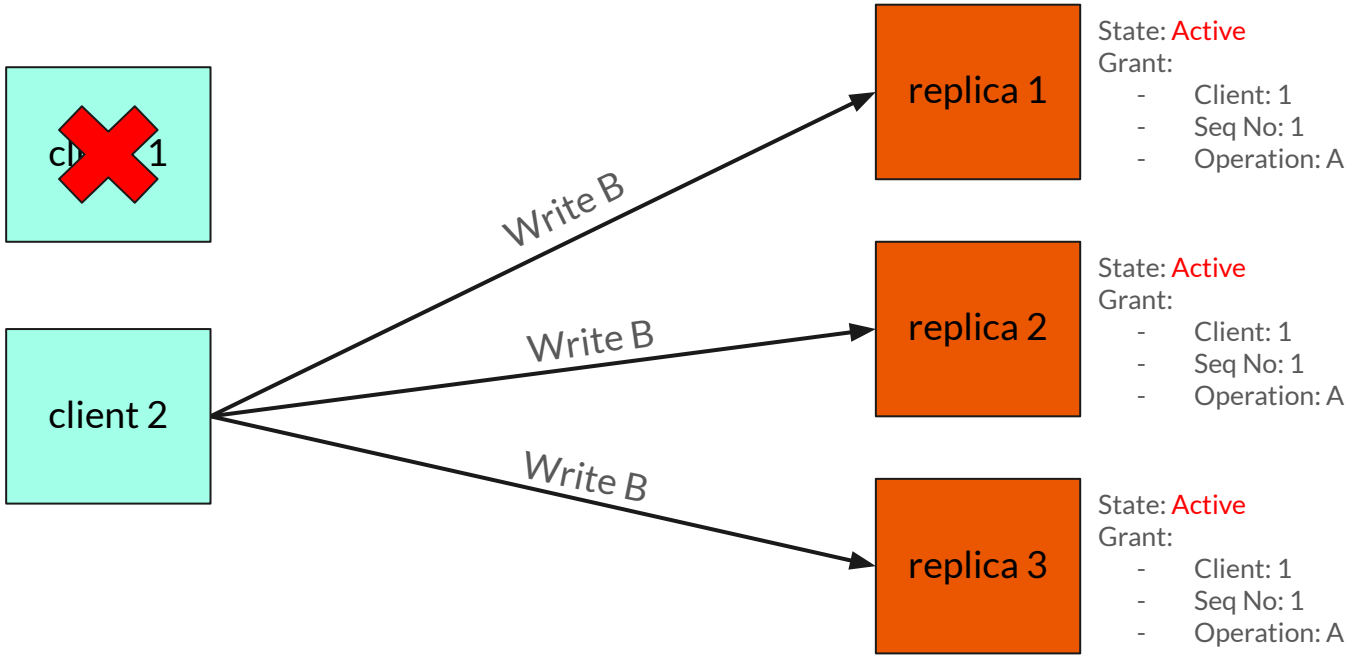
Completion of write1-phase

Normal Case-Incomplete Write

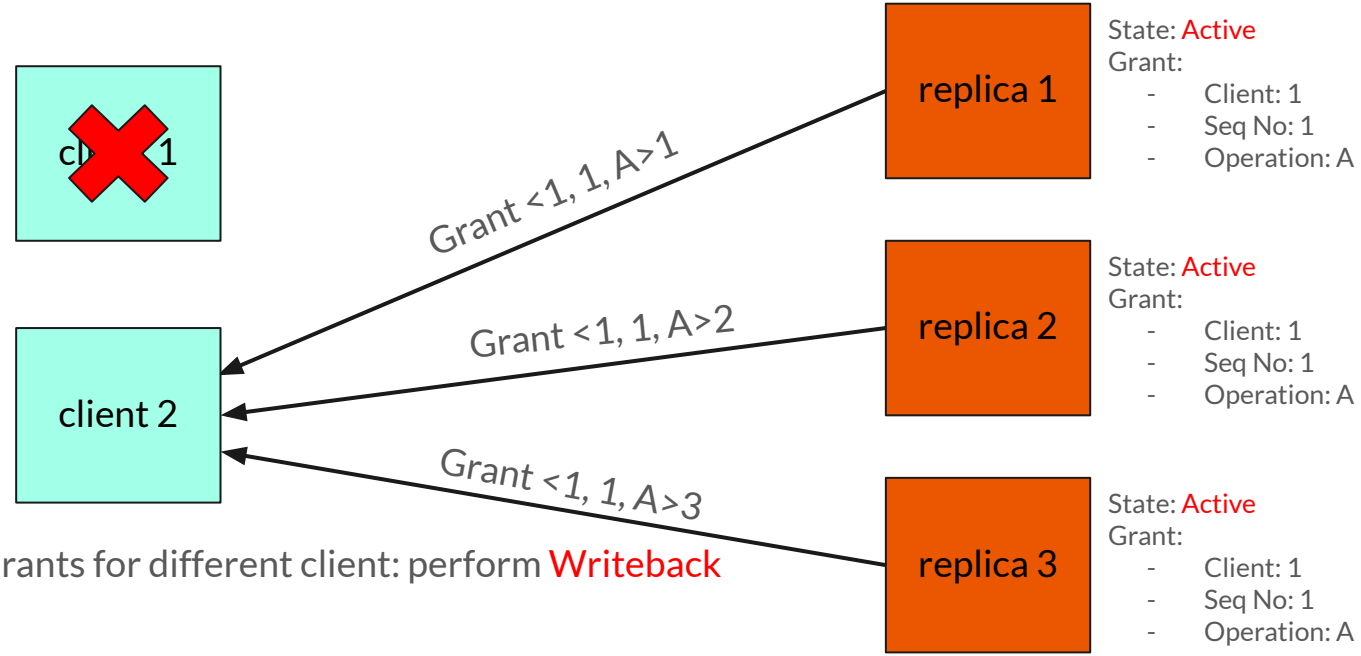


Now, if client 1 slow or failed…

Normal Case-Incomplete Write

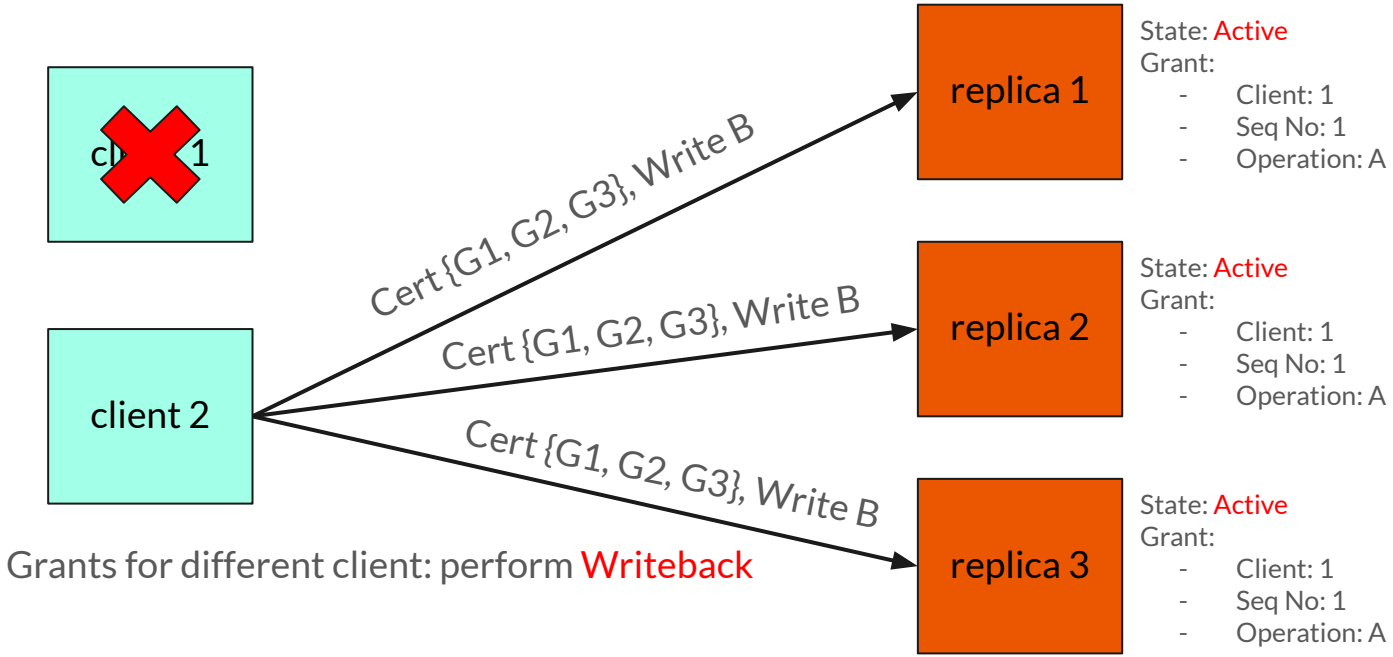


Normal Case-Incomplete Write



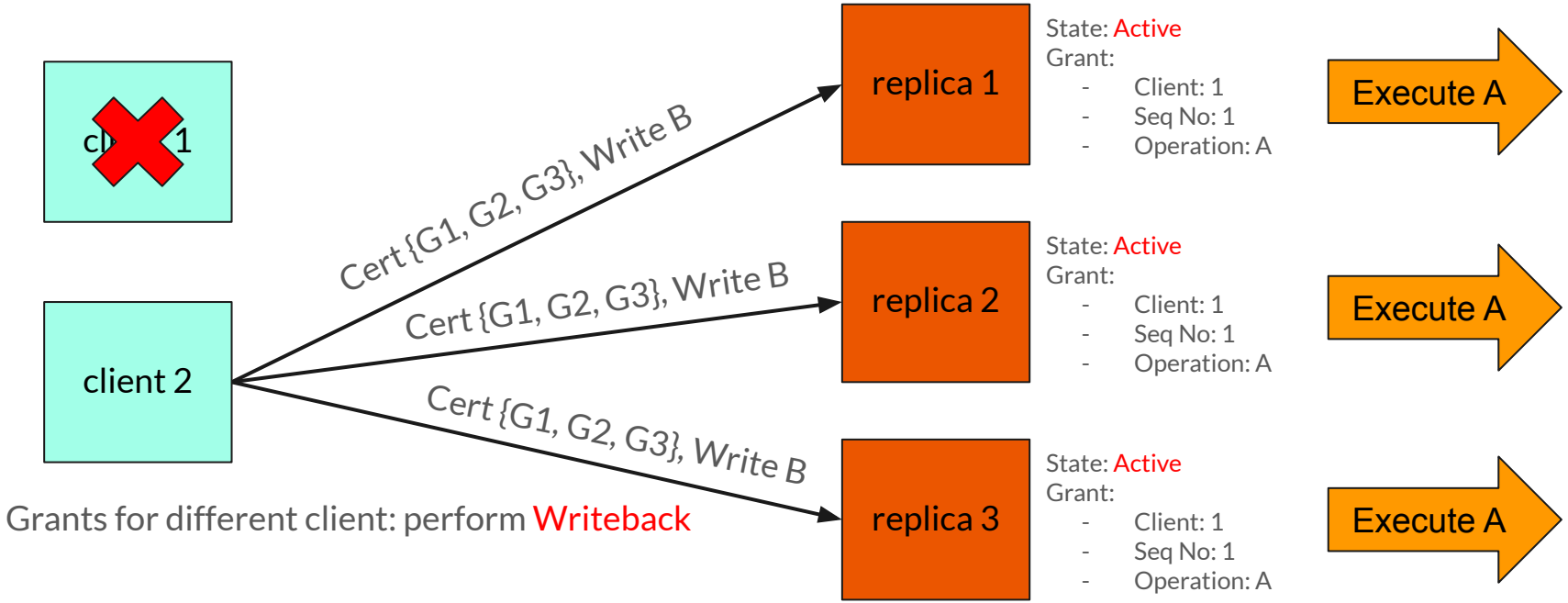
Grants for different client: perform Writeback

Normal Case-Incomplete Write



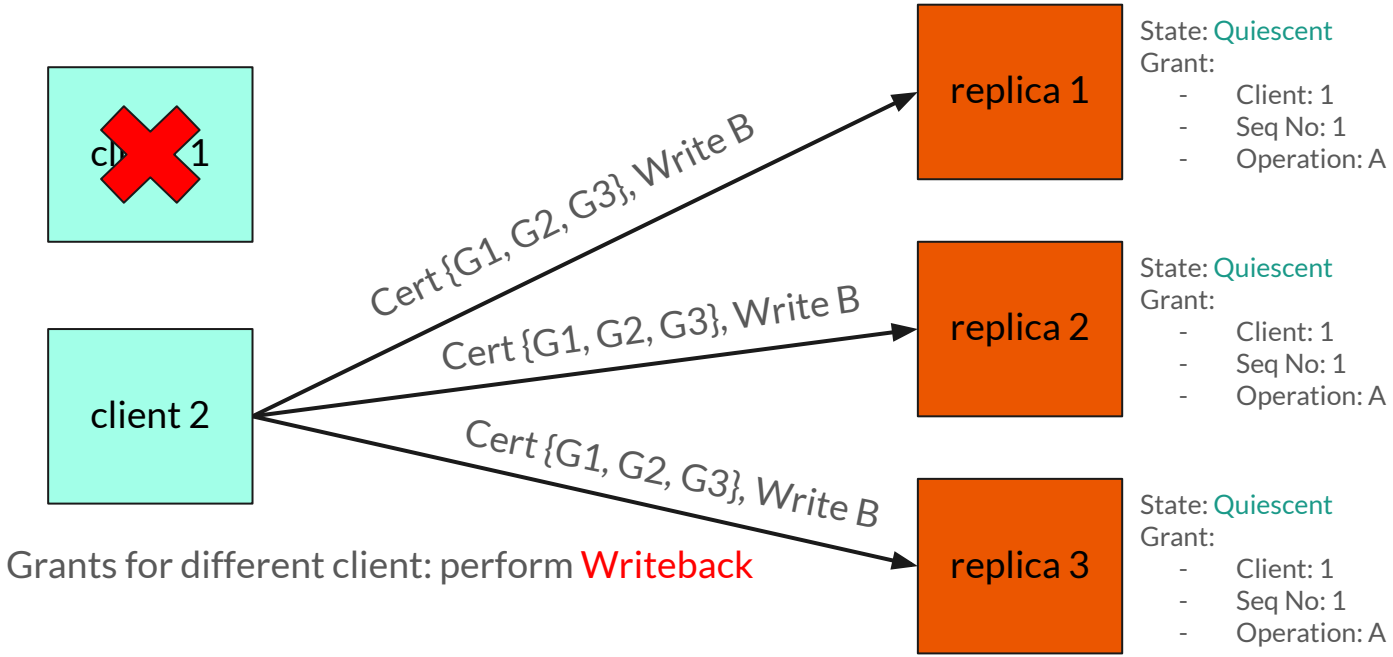
Grants for different client: perform Writeback

Normal Case-Incomplete Write



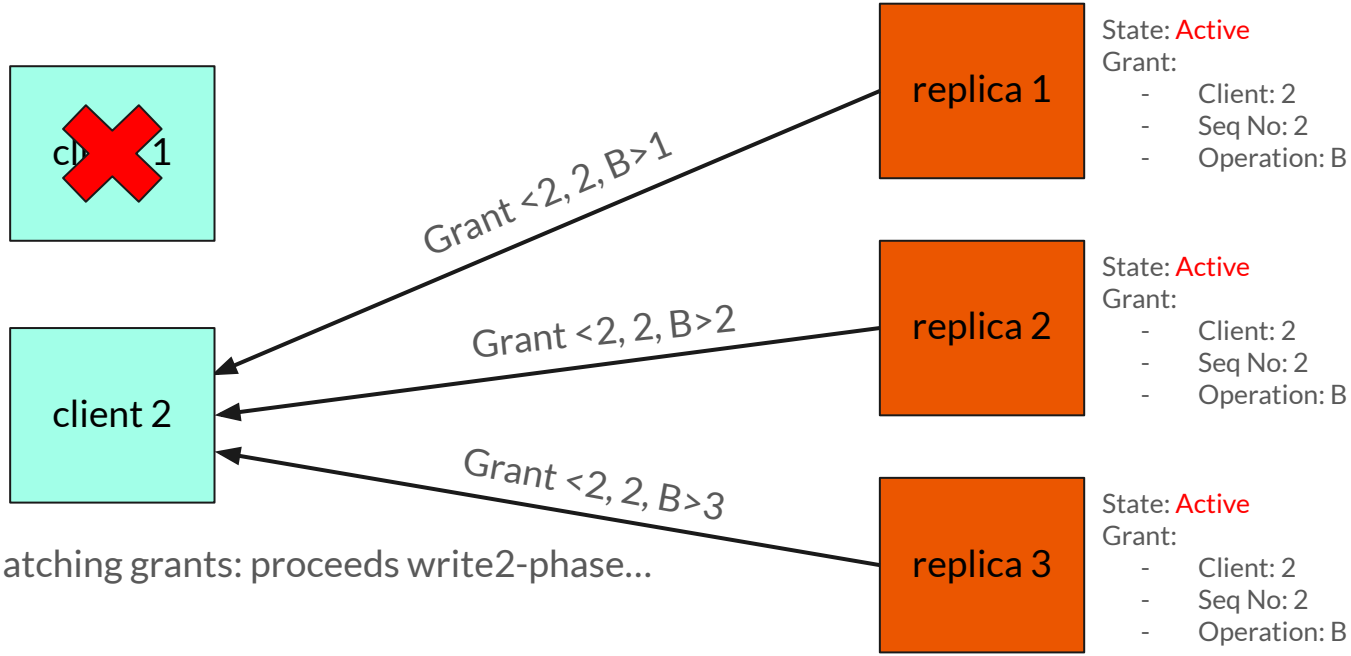
Grants for different client: perform Writeback

Normal Case-Incomplete Write



Grants for different client: perform Writeback

Normal Case-Incomplete Write



Matching grants: proceeds write2-phase…

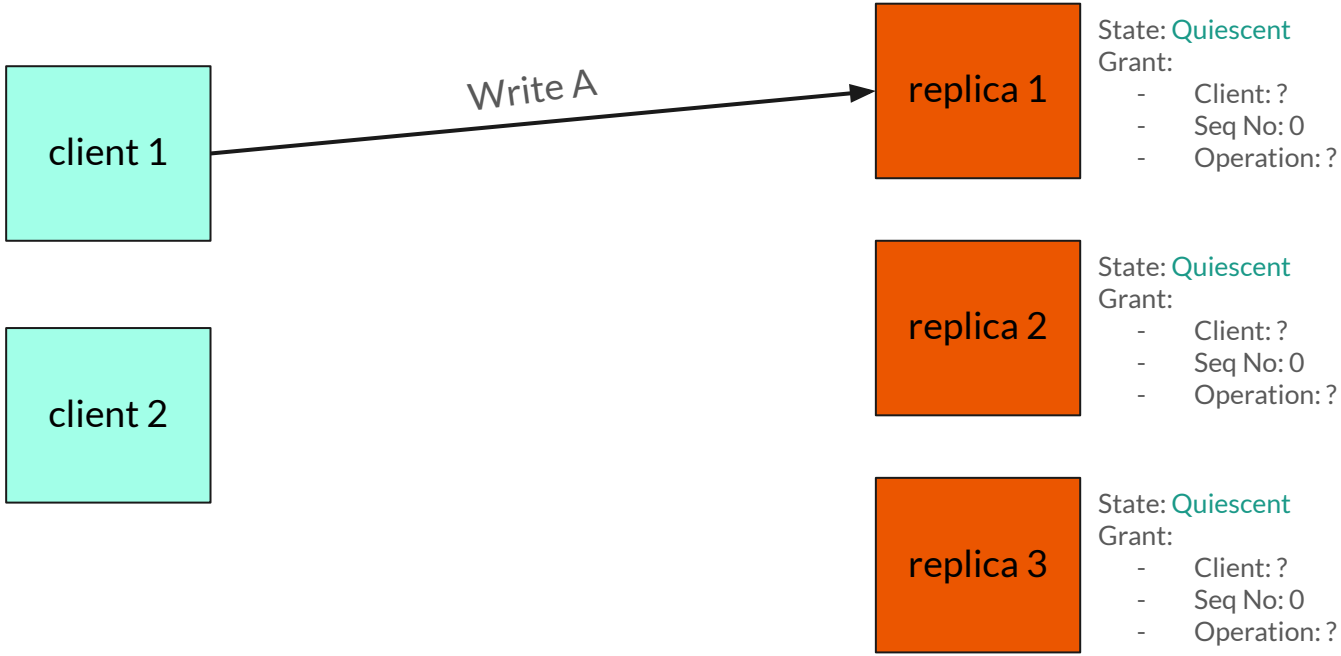
Contention Resolution

- Contention occurs because: several clients want to write at the same timestamp or faulty client.
- Use BFT module to resolve contention: establish sequential order on contending ops.
- When receive resolve request:
  (1) Freeze local object state.
  (2) Send state to primary.
- Primary replica runs BFT on combined state.
- Replicas execute contending operations.

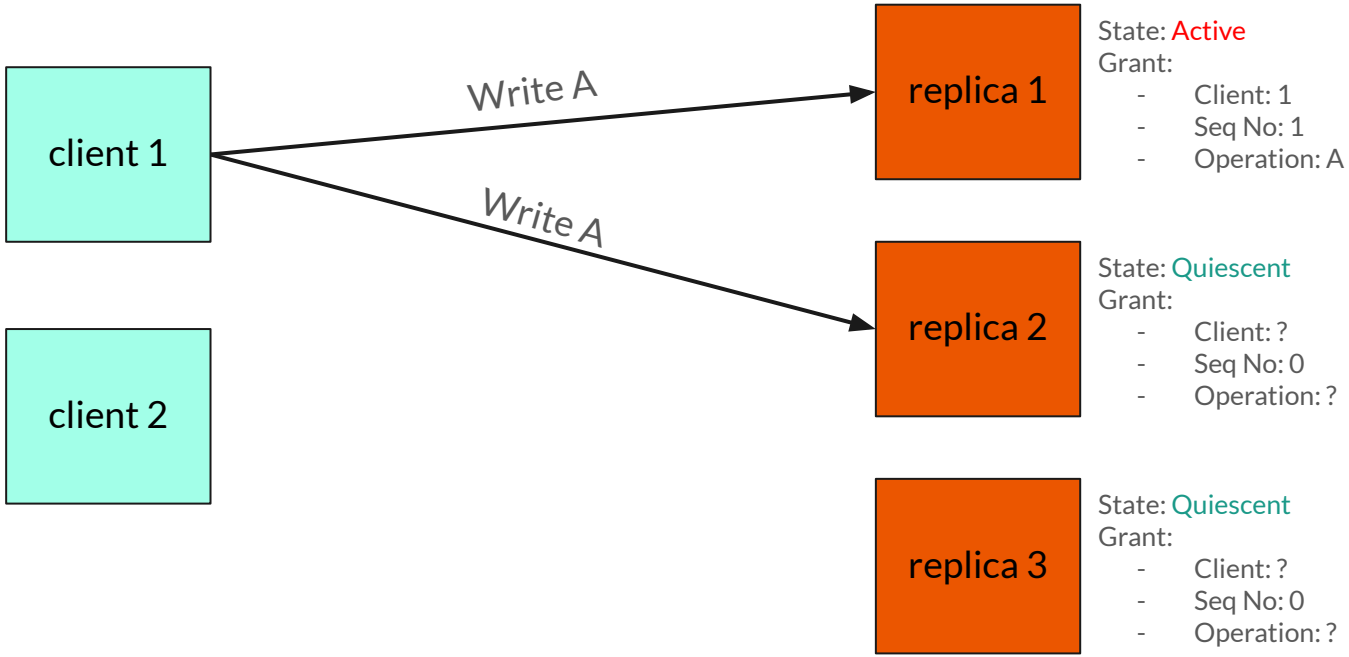
Contention Resolution

Write Contention

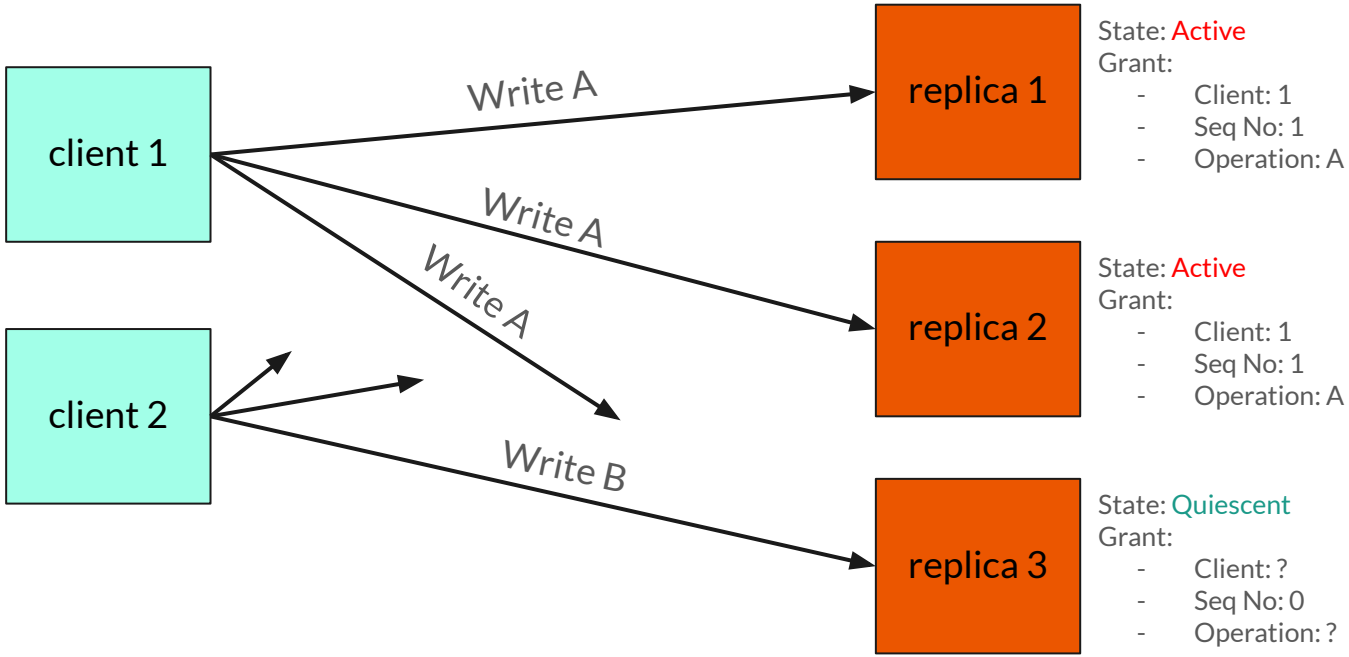
Write Contention



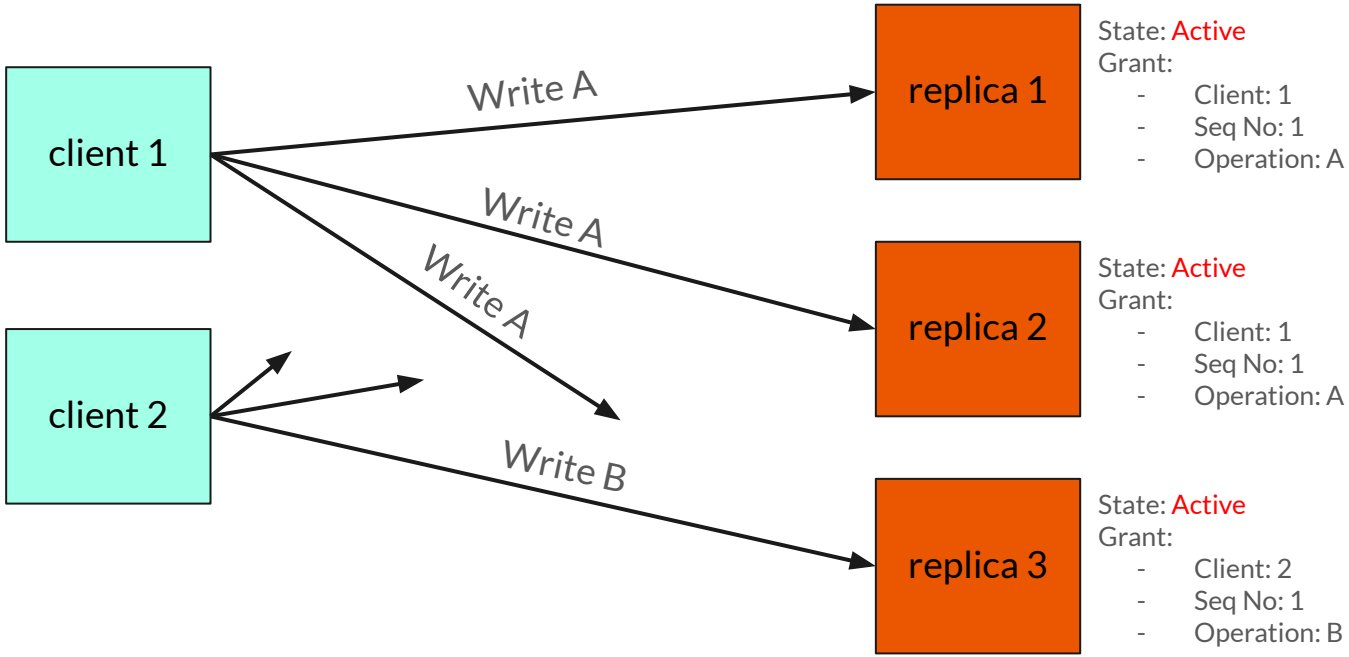
Write Contention



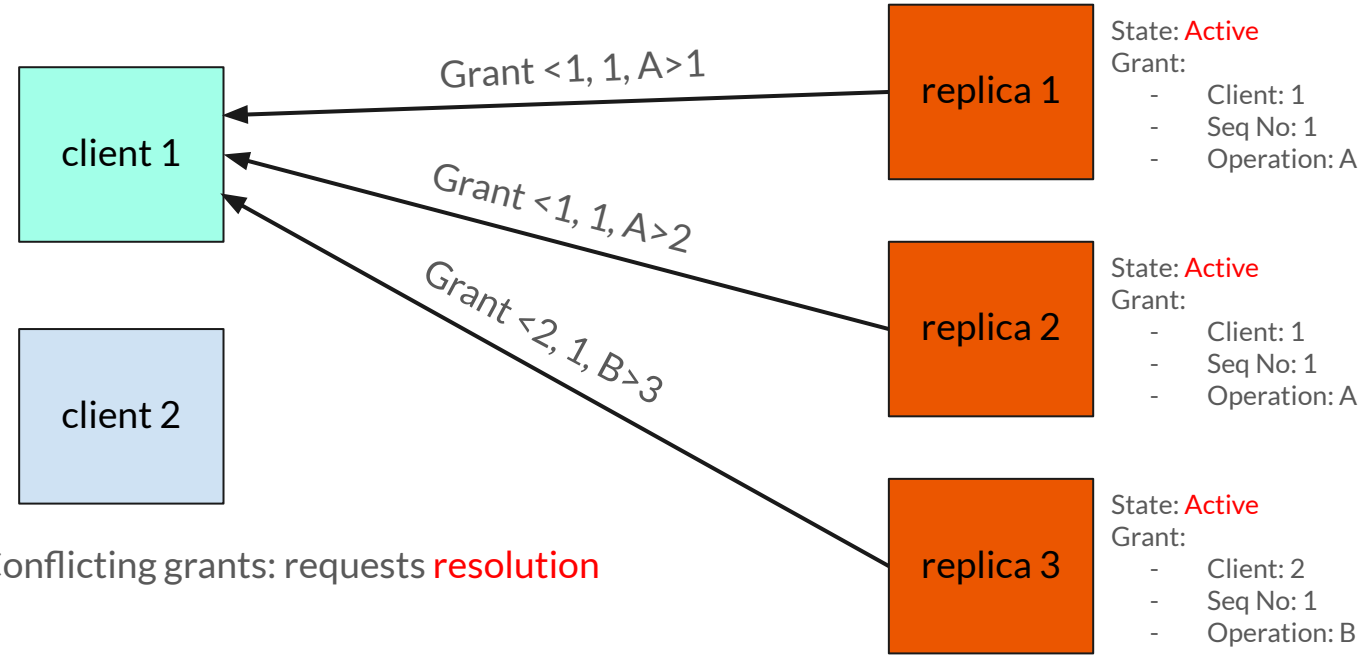
Write Contention



Write Contention

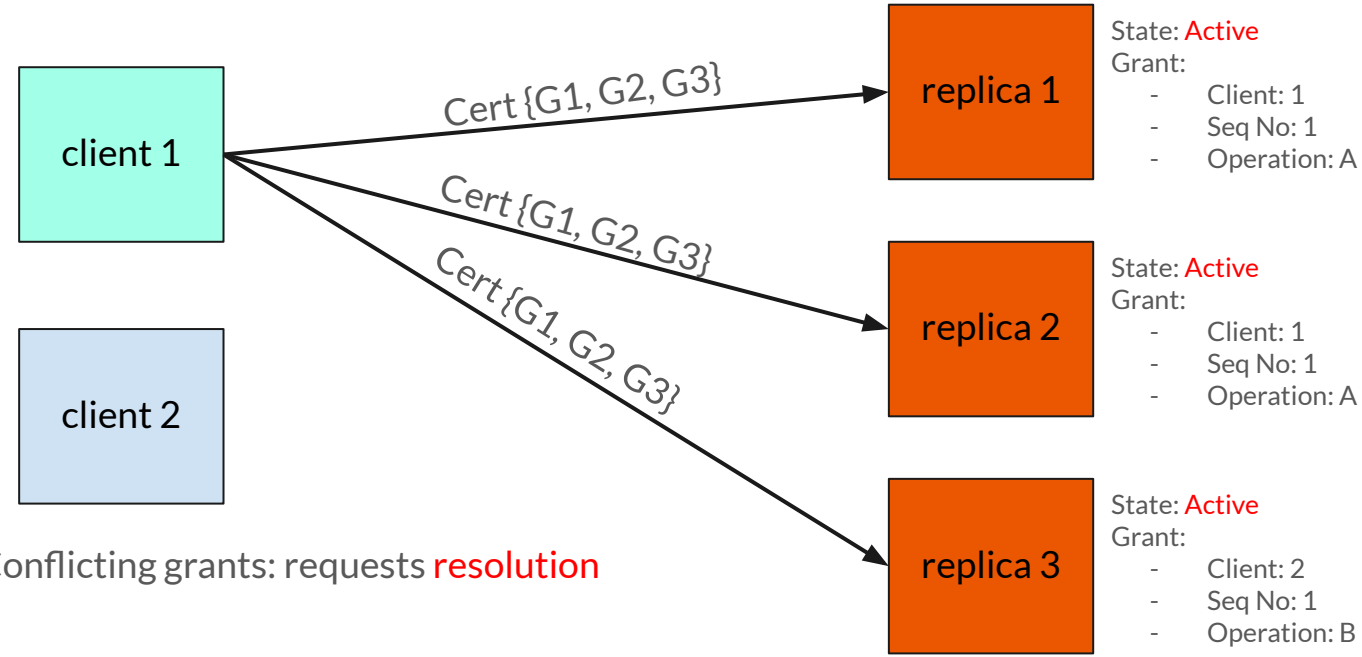


Write Contention



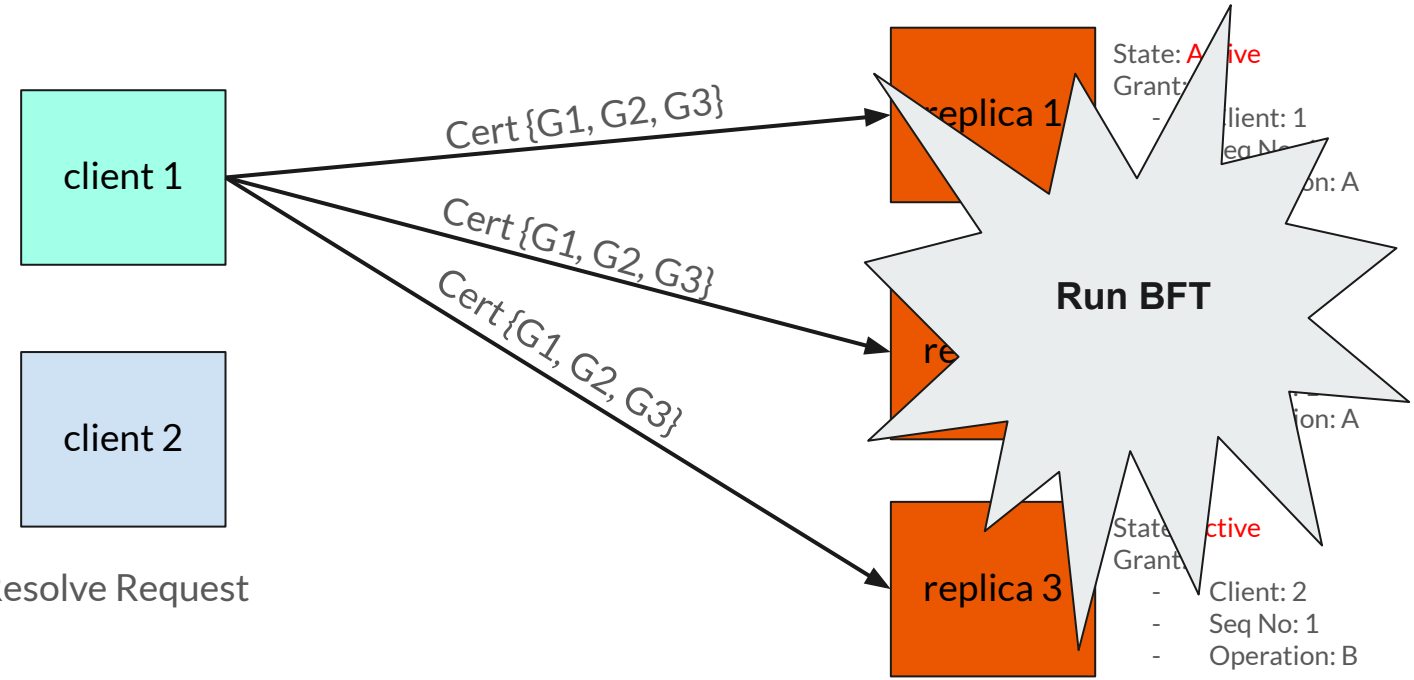
Conflicting grants: requests resolution

Write Contention



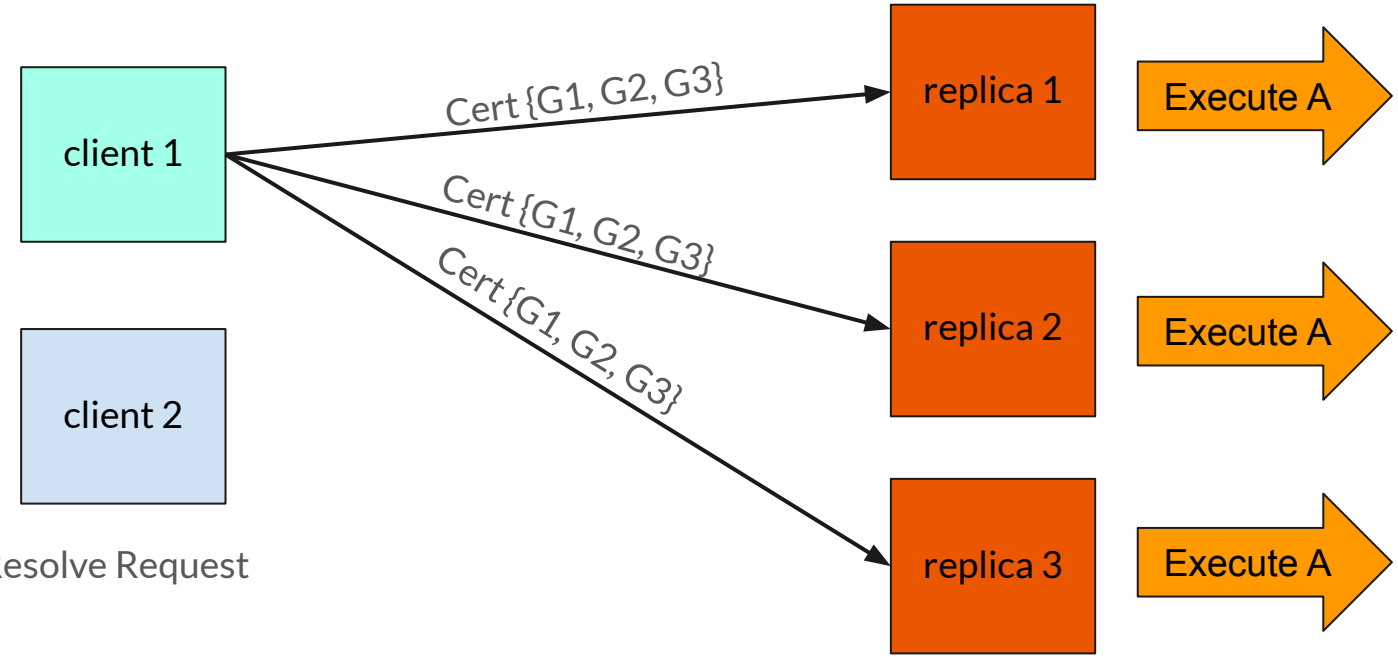
Conflicting grants: requests resolution

Write Contention



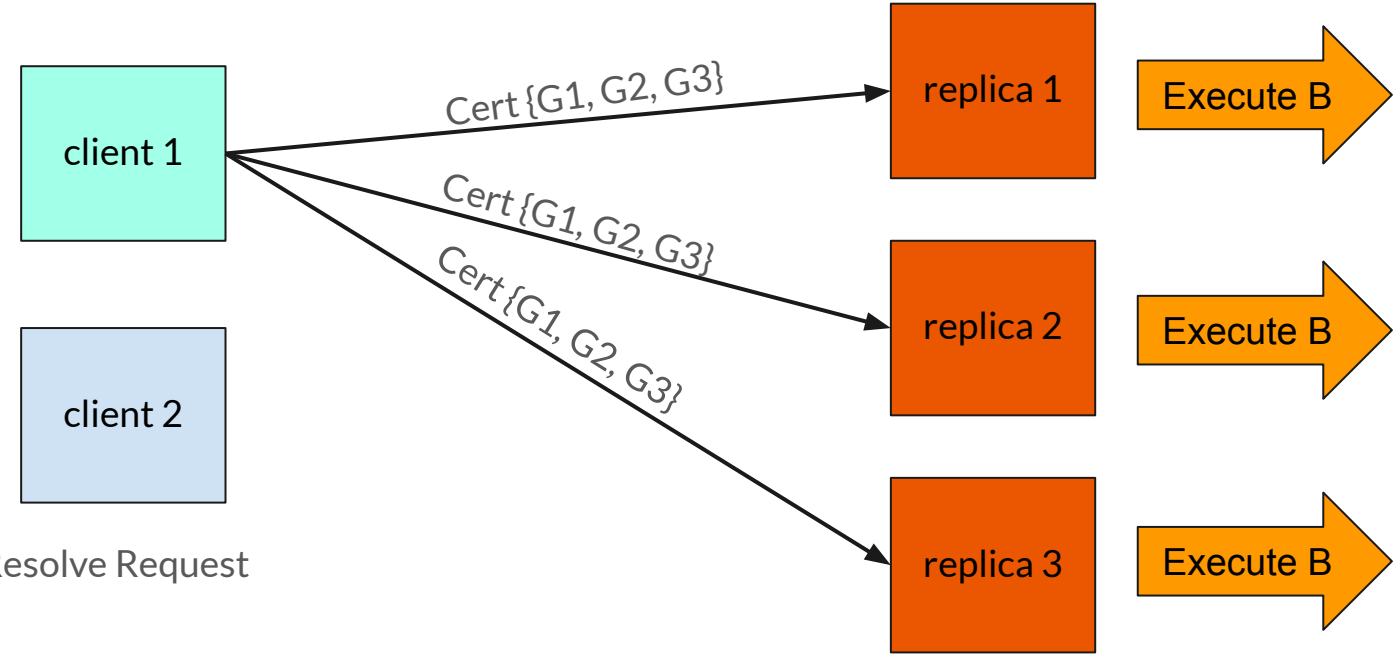
Resolve Request

Write Contention



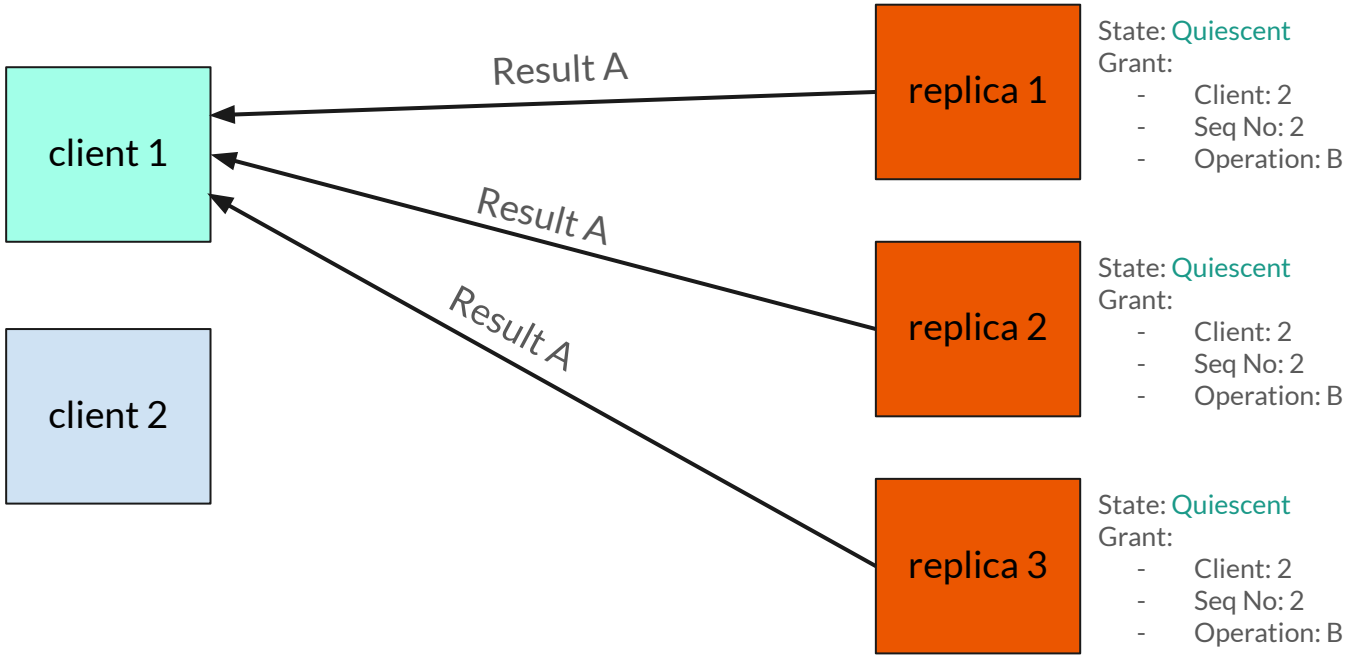
Resolve Request

Write Contention

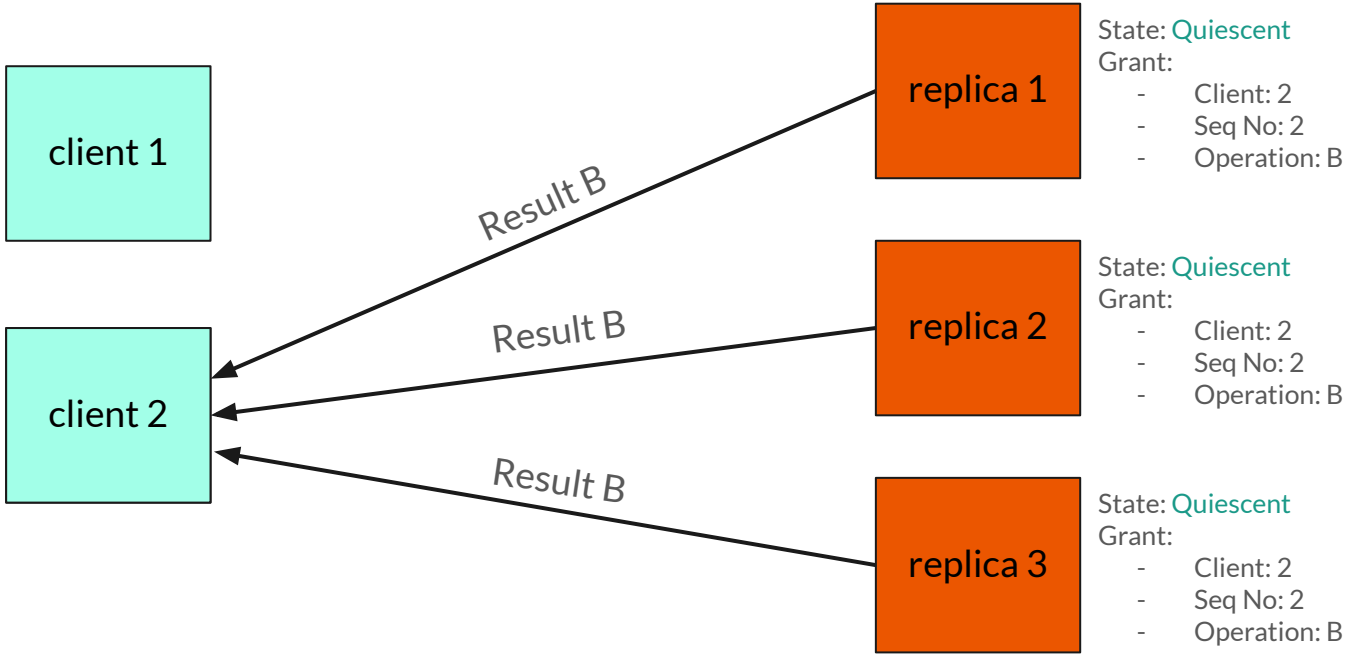


Resolve Request

Write Contention



Write Contention



HQ Replication-Details

- **State Transfer**
  - confirm hash
  - Log

HQ Replication-Details

- **State Transfer**
  - confirm hash
  - Log
- **Correctness**
  - Linearizability

HQ Replication-Details

- **State Transfer**
  - confirm hash
  - Log
- **Correctness**
  - Linearizability
- **Multi-object transactions**

Optimizations

- Early Grants

Optimizations

- Early Grants
- Avoiding Signing

Optimizations

- Early Grants
- Avoiding Signing
- Preferred Quorums

Optimizations

- Early Grants
- Avoiding Signing
- Preferred Quorums
- BFT Improvements
    - TCP -> UDP
    - MACs -> authenticators

Analysis

The performance characteristics of HQ, BFT, and Q/U.

- HQ and Q/U: Using both MACs/authenticators and preferred quorums
- Original BFT: Using authenticators but not preferred quorums
- BFT-MACs: Using MACs but not preferred quorums
- BFT-opt: Using both MACs and preferred quorums

Analysis - Message handled (non-contention)

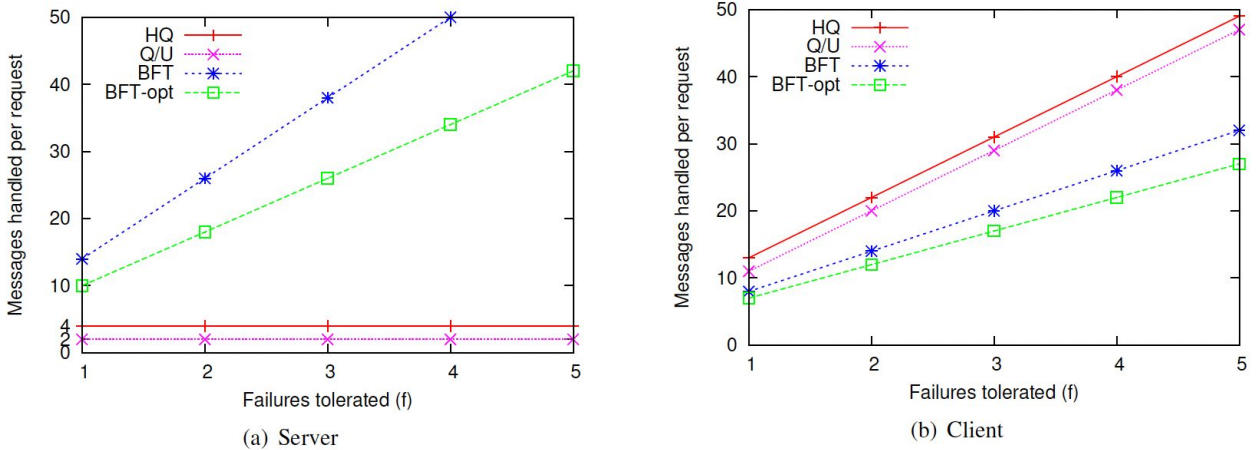


Figure: Total messages sent/received per write operation in BFT, Q/U, and HQ.

Analysis - Traffic (non-contention)

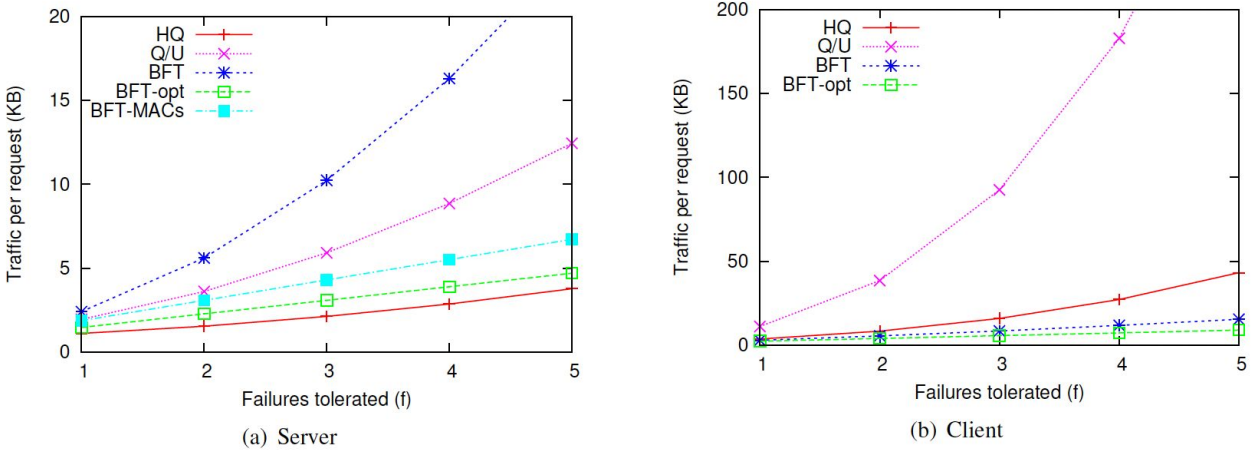


Figure: Total traffic sent/received per write operation in BFT, Q/U, and HQ.

Experimental Evaluation - Setup

- Deployed on Emulab
  - Up to 16 replicas (f = 5) and 100 clients (2 per machine)
  - Each experiment runs for 100,000 client operations and 5 repeat runs were recorded
- New codebase
  - Derived from a common C++ codebase
- Counter service
  - Multiple objects
    - Non-contention -> private objects
    - Contention -> shared objects
  - Negligible (10 byte) operation payload

Experimental Evaluation - Non-batched write

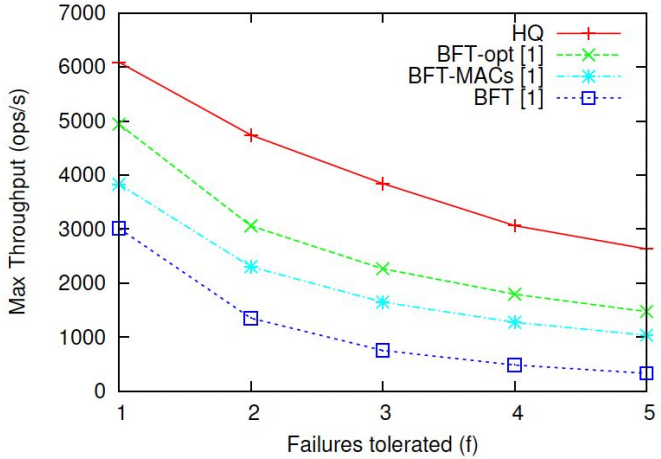


Figure: Maximum non-batched write throughput under varying f.

Experimental Evaluation - Batched write

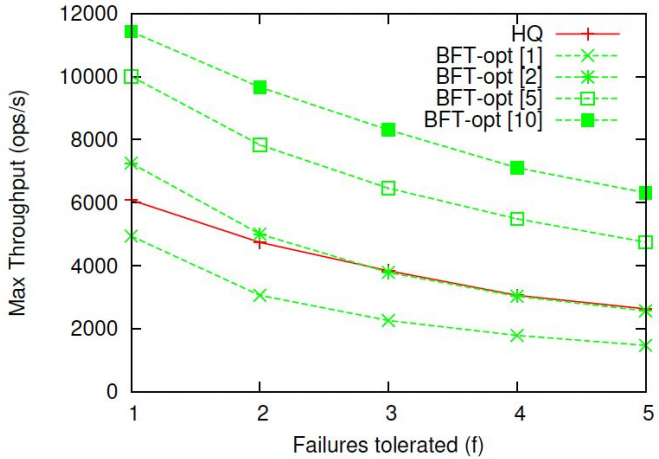


Figure: Effect of BFT batching on maximum write throughput.

Experimental Evaluation - Write Contention

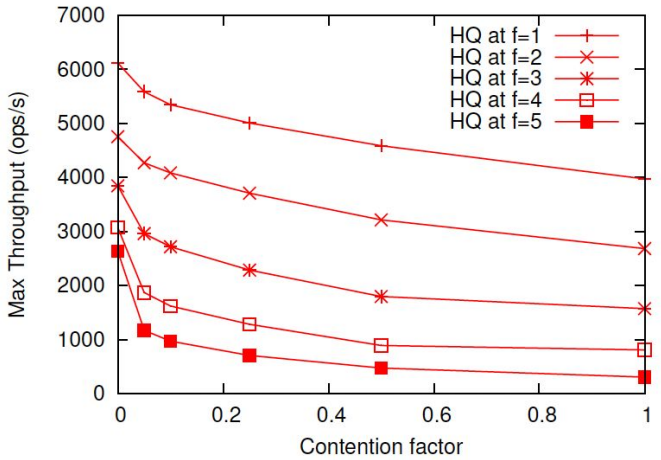


Figure: HQ throughput under increasing write contention.

Conclusions

- A new quorum based protocol with 3f+1 replicas
  - By using a two-phase instead of a one-phase write protocol
- A new way of handling contention in quorum-based protocols
  - By using BFT when there is contention
  - The hybrid approach can be used broadly to other systems
- A new implementation of BFT
  - Was developed to scale with f

Recommendations

- Choose Q/U if
  - Low contention
  - Low latency is the main issue
  - 5f+1 replicas is acceptable
- Choose BFT if
  - High contention
  - High throughput
- Choose HQ if
  - Moderate contention
  - Throughput degrades slowly as f grows

Thanks!

ECS265 Paper Review Presentation

Presenters: Shuyan Dai, Yuang Ma, Kaixin Xiao, Jiayu Jiang