



From Minutes to Seconds

Redefining MySQL Failover Reliability at Uber

Uber

About me

Siddharth Singh

Staff Software Engineer at Core Storage team Uber.
Focusing on the reliability, availability, quality, and resiliency
charters for MySQL/Vitess.



01 MySQL @ Uber

02 Challenges

03 Primary Failover Process

04 MySQL Group Replication Deployment

05 Routing optimizations

06 Performance Results

MySQL @Uber

2600+

Active Clusters

22000+

Database Nodes

99.99%

Availability

250+

Critical Services in Uber
use MySQL as Primary
DB

An aerial, top-down view of a city street grid. The streets are filled with cars, and there are several large buildings with flat roofs. The lighting is soft, suggesting late afternoon or early morning. The word "Challenges" is overlaid in the center in a large, white, sans-serif font.

Challenges

Historical Issues

Dependency Failures

Due to the need for external coordination for primary failover

Longer Primary Failover time

Due to lack of consensus between nodes

Accidental split brain scenarios

Due to lack of quorum

Disconnected Routing Plane

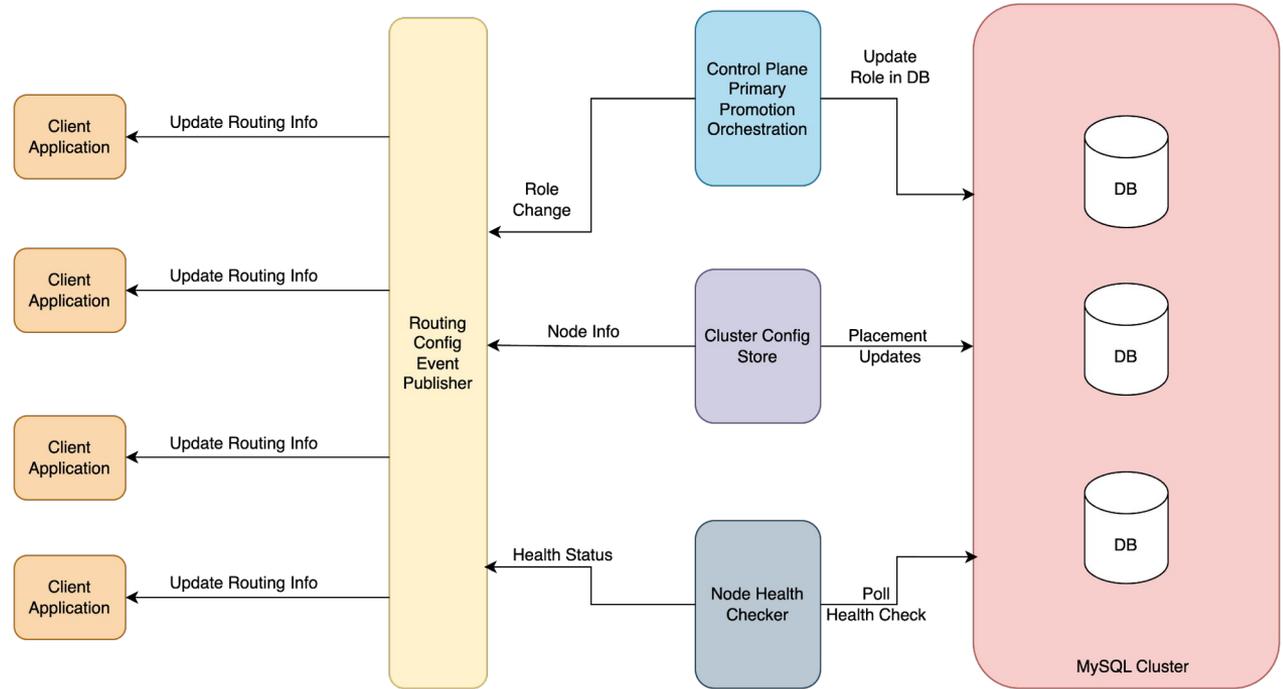
Due to signals from different sources to take routing decisions

Unbounded Replication Delay

Due to lack of replication aware throttler

Disconnected Routing Plane

- ❖ Multiple Source of truth for Routing Signals
- ❖ Not reflecting actual state of the cluster



An aerial, top-down view of a city street grid. The streets are filled with cars, and there are several large buildings with flat roofs. The overall color palette is muted, with greys, blues, and browns. The text 'Primary Failover Process' is overlaid in the center in a large, white, sans-serif font.

Primary Failover Process

Goals

Less than 10s Write Unavailability

Zero Data Loss

No data loss during primary failover ensuring full durability

Always single primary

Completely avoid split brain scenarios

Deterministic failover decision logic

No external dependencies

Leader Election via consensus

Bounded Replication Lag

No unbounded delays in promotion time due to replication lag

MySQL Group Replication

To achieve consensus based replication

Routing Plane optimization

Unify routing signals by moving the source of truth to the nodes

MGR Deployment Architecture @ Uber

Group size 3

- Group size == 3
- 3 nodes from 3 AZs
- Quorum size 2 ⇒
Only 1 ack required ⇒
Latencies on par with semisync!

Continuous Rebalance

- Control plane identifies failed group members
- Replaces failed nodes with spare replicas from the read-only pool
- Ensures group size 3 eventually

Single Primary

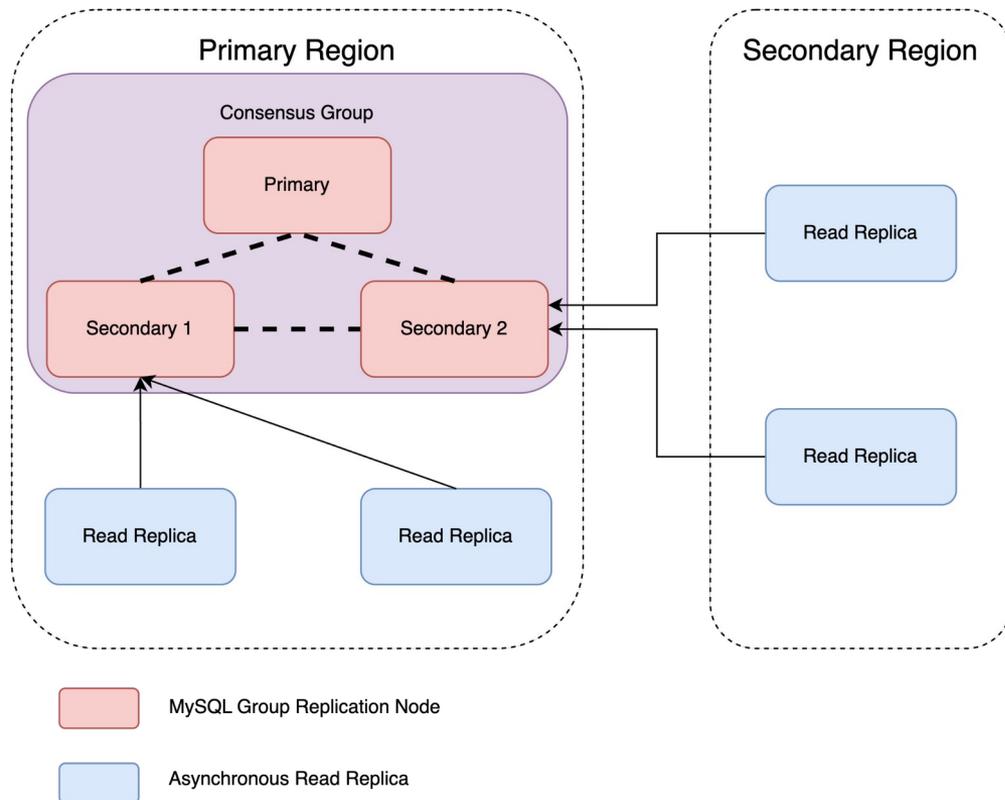
- No conflict potential
- Total order of transactions maintained.

Read-Only Replicas

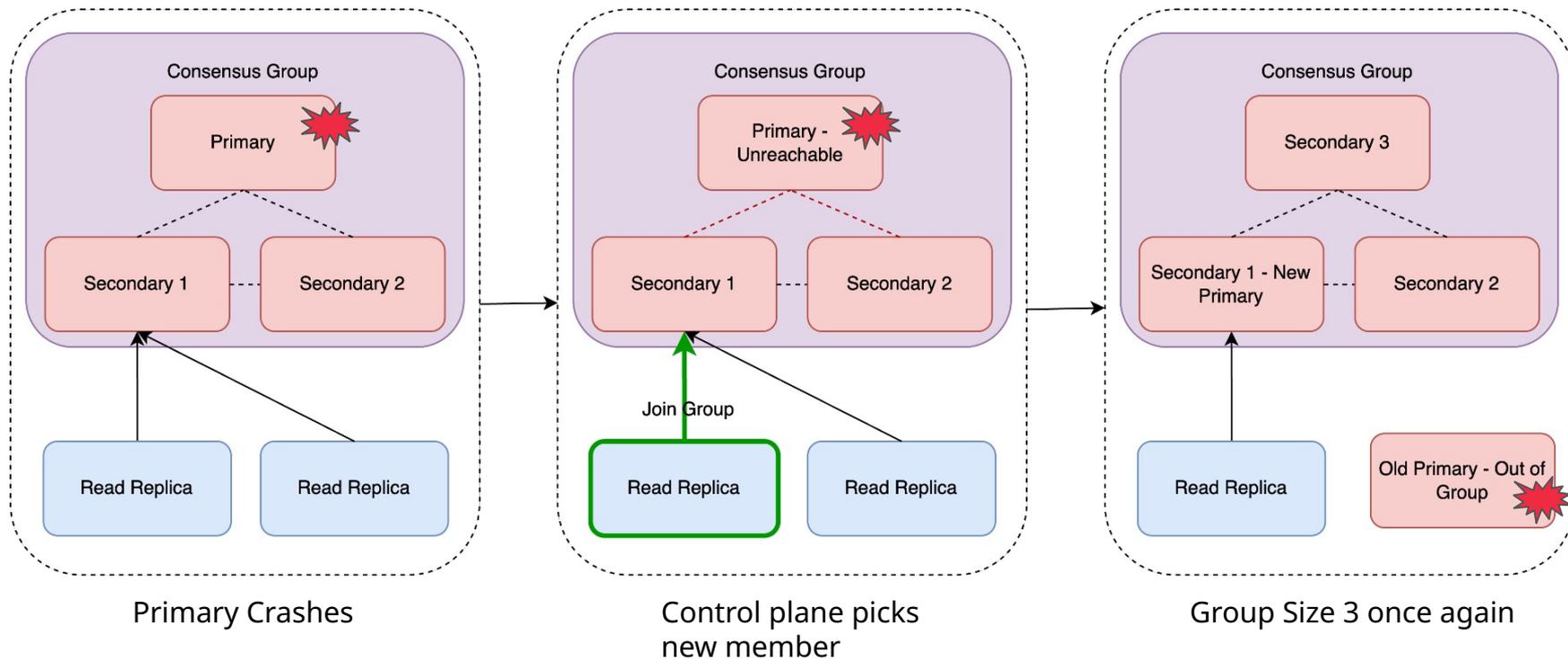
- Pool of read-only replicas
- Distributed across two regions
- Distributed across availability zones
- Not part of consensus group

Scalable Reads with MySQL Group Replication

- ❖ Group size 3
- ❖ Single Primary
- ❖ Read-Only Replicas



Continuous Rebalancing



Advantages of MGR @ Uber

Write Latencies

Comparable to semi-sync replication

Read Scaling

Secondaries along with read-only replicas can serve read requests that are eventually consistent

Fixed Group Size

Ability in control plane to rebalance and eventually reach the group size of 3

Zone Failure Tolerance

Keeping the nodes in consensus group zone distributed thereby ability to withstand loss of availability zone

Primary Failover Process | Routing Optimizations

Single Source of Truth

From the actual node - propagation via etcd

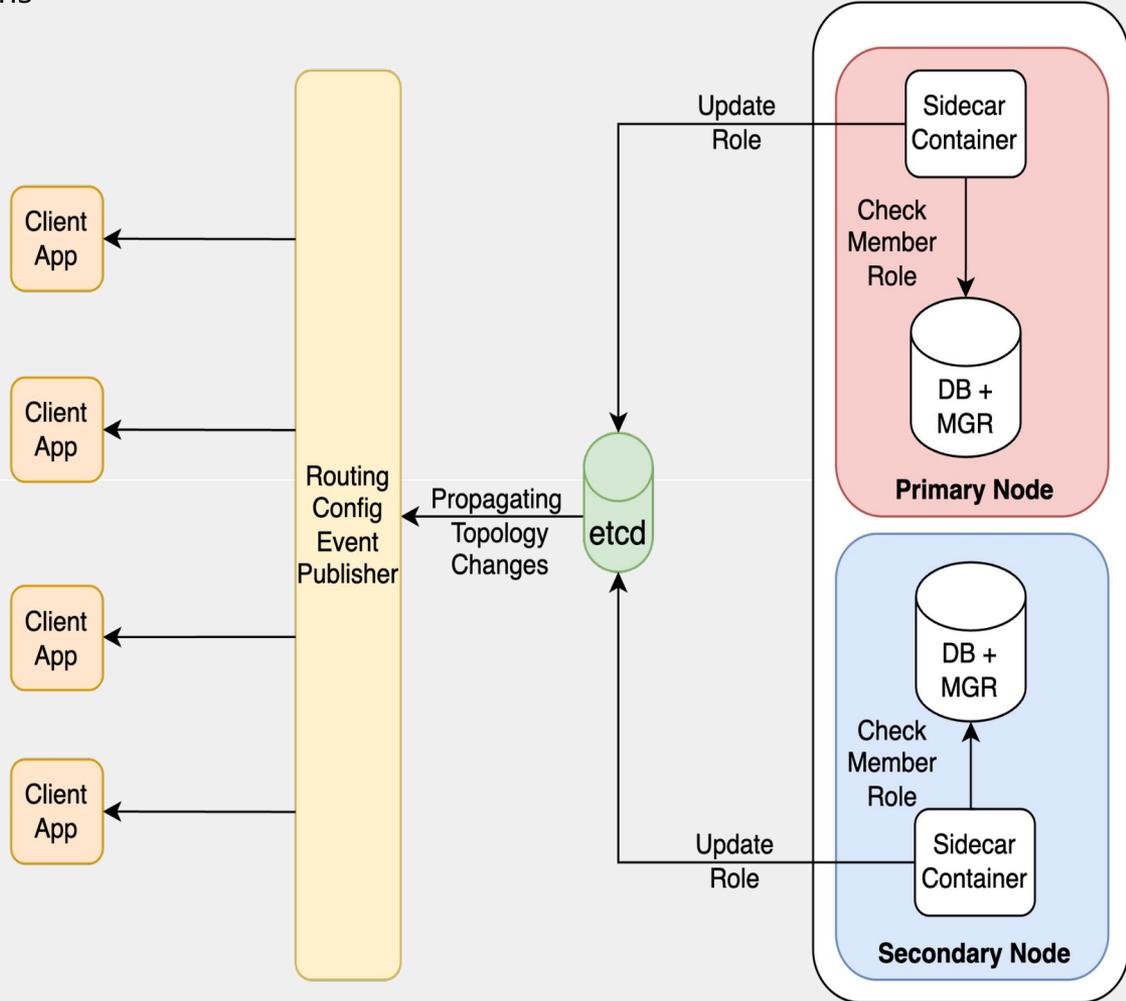
Single Primary

Write and strongly consistent read requests

Secondaries + Read Replicas

Eventually consistent reads

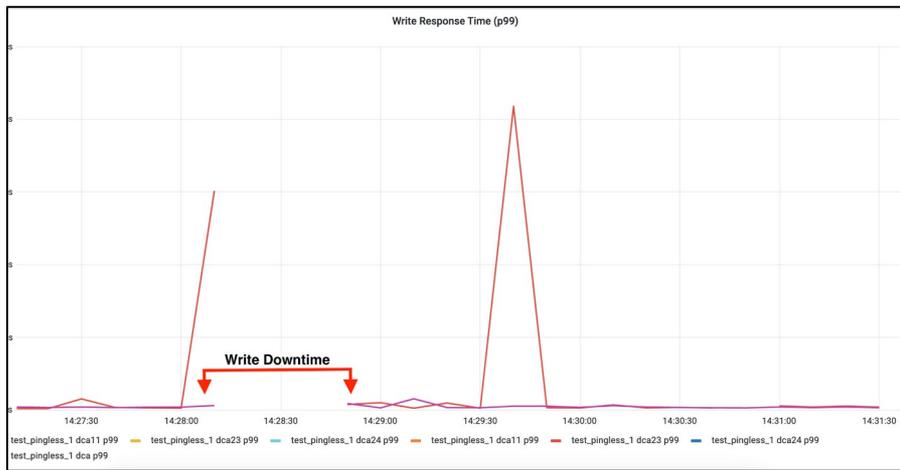
Enhancing MySQL Cluster From Minutes to Uber | Seconds Redefining MySQL Failover Reliability at Uber



An aerial, top-down view of a city street grid. The streets are filled with cars, and there are several large buildings with flat roofs. The lighting is somewhat dim, suggesting an overcast day or late afternoon. The overall color palette is muted, with greys, blues, and browns.

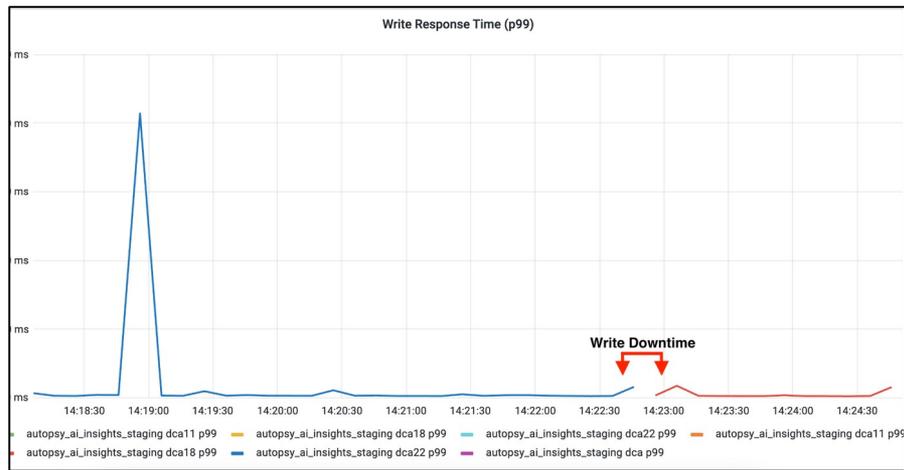
Performance Results

Improved Primary Promotion Time



Async replication

Cluster with Async replication: 40s write unavailability during primary failure



MGR Cluster

Cluster with MGR: 10s write unavailability during primary failure

Benchmark

Setup

CPU Cores: 8

Disk: 300 GB

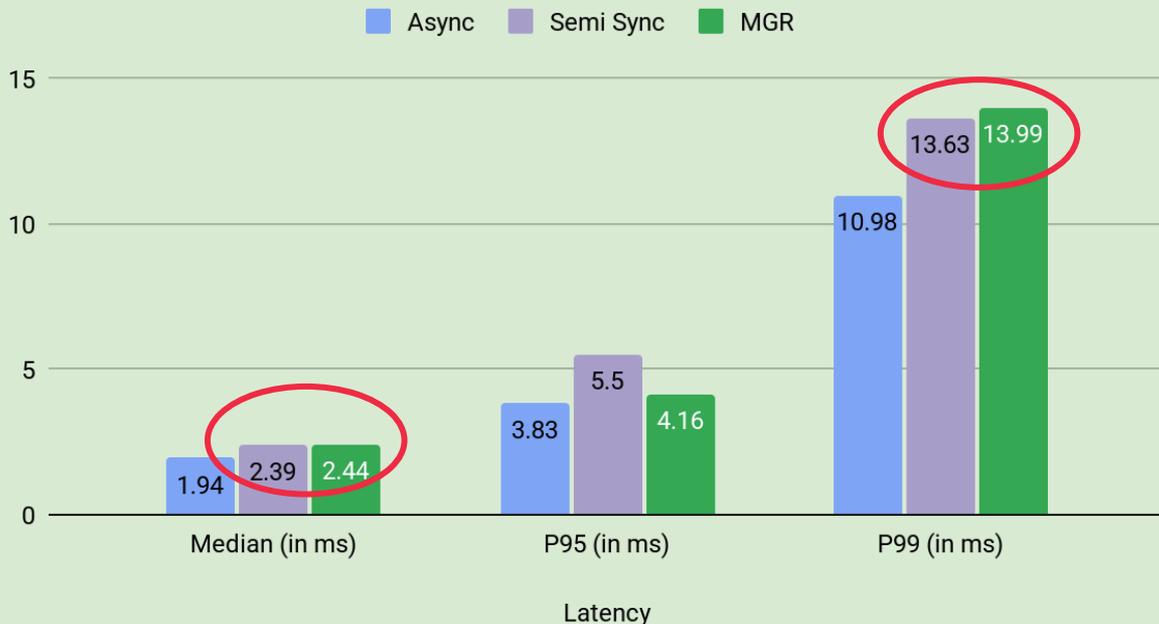
Memory: 42 GB

IOPS: 40,000

QPS : 4000

Major reliability improvements with minimal latency regression!

Async, Semi Sync and MGR - Insert Latency Comparison



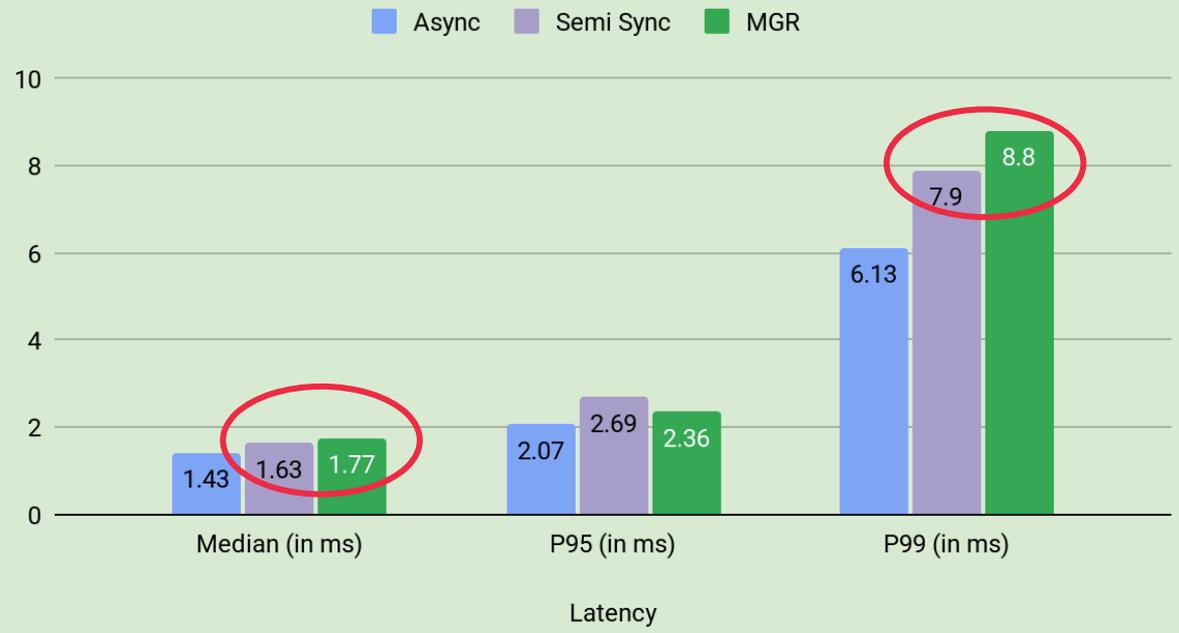
Benchmark

Setup

CPU Cores: 8
Disk: 300 GB
Memory: 42 GB
IOPS: 40,000
QPS : 4000

Major reliability improvements with minimal latency regression!

Async, Semi Sync and MGR - Update Latency Comparison



Q&A