AWS re:Invent

DAT404-R

Amazon Aurora Multi-Master: Scaling out database write performance

Eric Boutin
Software Development Manager
Amazon Aurora
Amazon Web Services

Steve Abraham
Principal Lead Data Architect
Amazon Aurora
Amazon Web Services

AWS re:Invent

aws

Agenda

Introducing Amazon Aurora Multi-Master

Architecture deep dive

Best practices

Demo

Amazon Aurora

Enterprise database at open source price

Delivered as a **managed** service

Speed and **availability** of high-end commercial databases

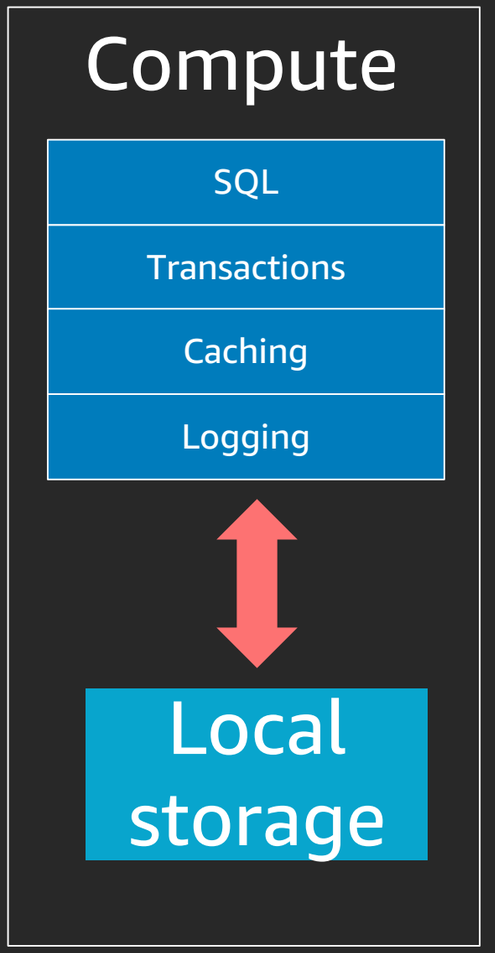
Simplicity and **cost-effectiveness** of open-source databases

Drop-in **compatibility** with MySQL and PostgreSQL

Simple **pay-as-you-go** pricing

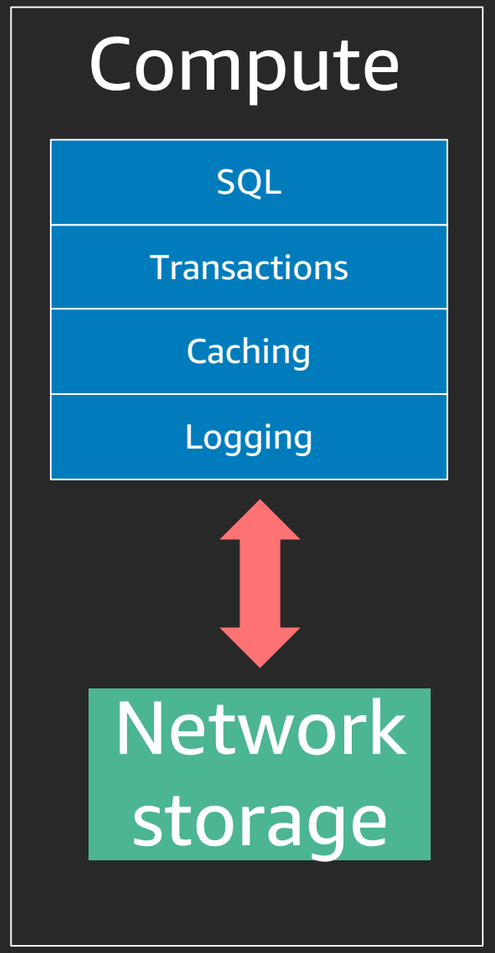
Traditional database architecture

Monolithic stack in a single box



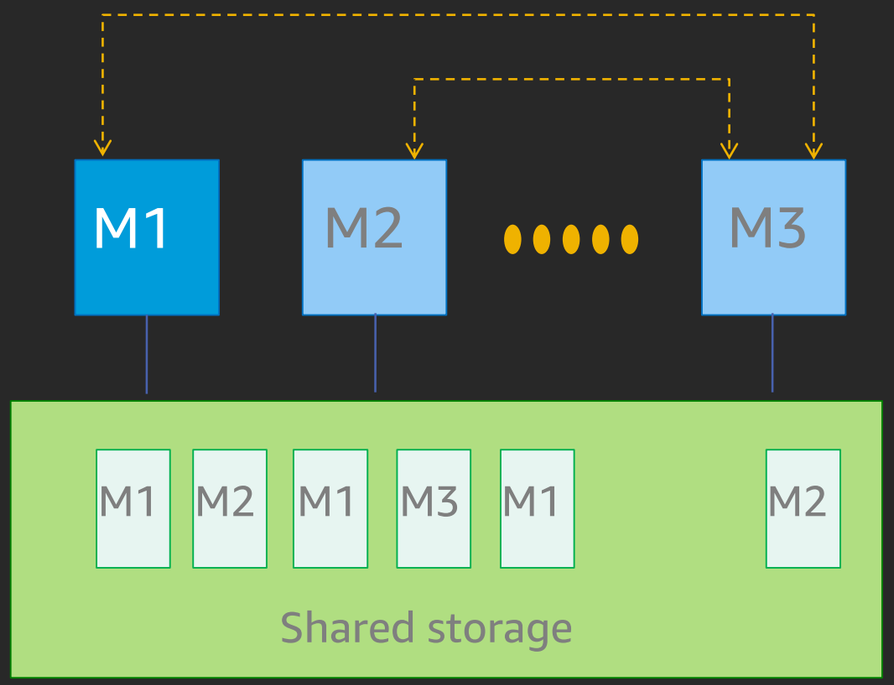
Traditional database architecture

Decoupled storage from compute



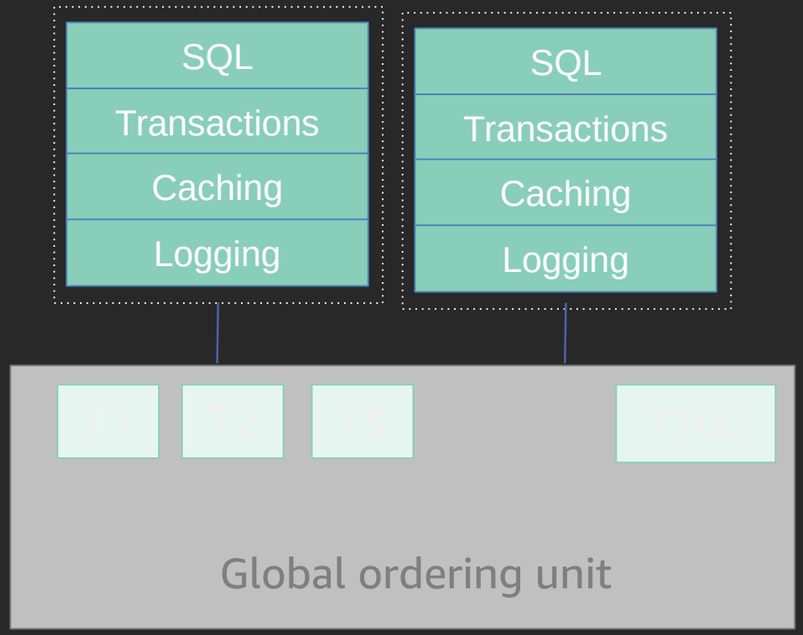
Existing Multi-Master solutions

Distributed lock manager



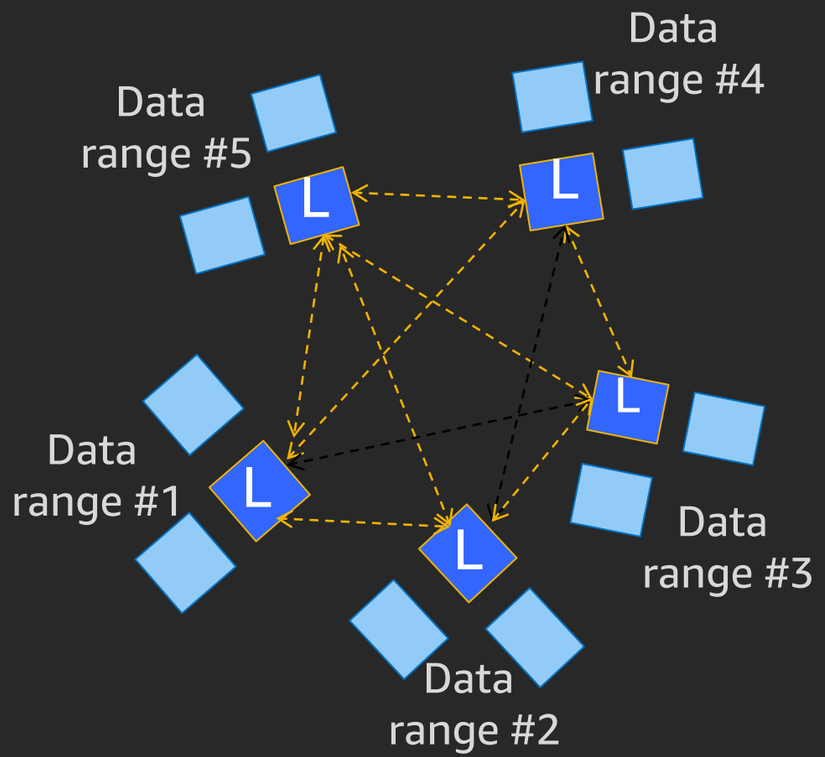
Heavyweight synchronization:
Pessimistic and negative scaling

Global ordering with read-write set



Global entity: Scaling bottleneck

Paxos leader with 2PC



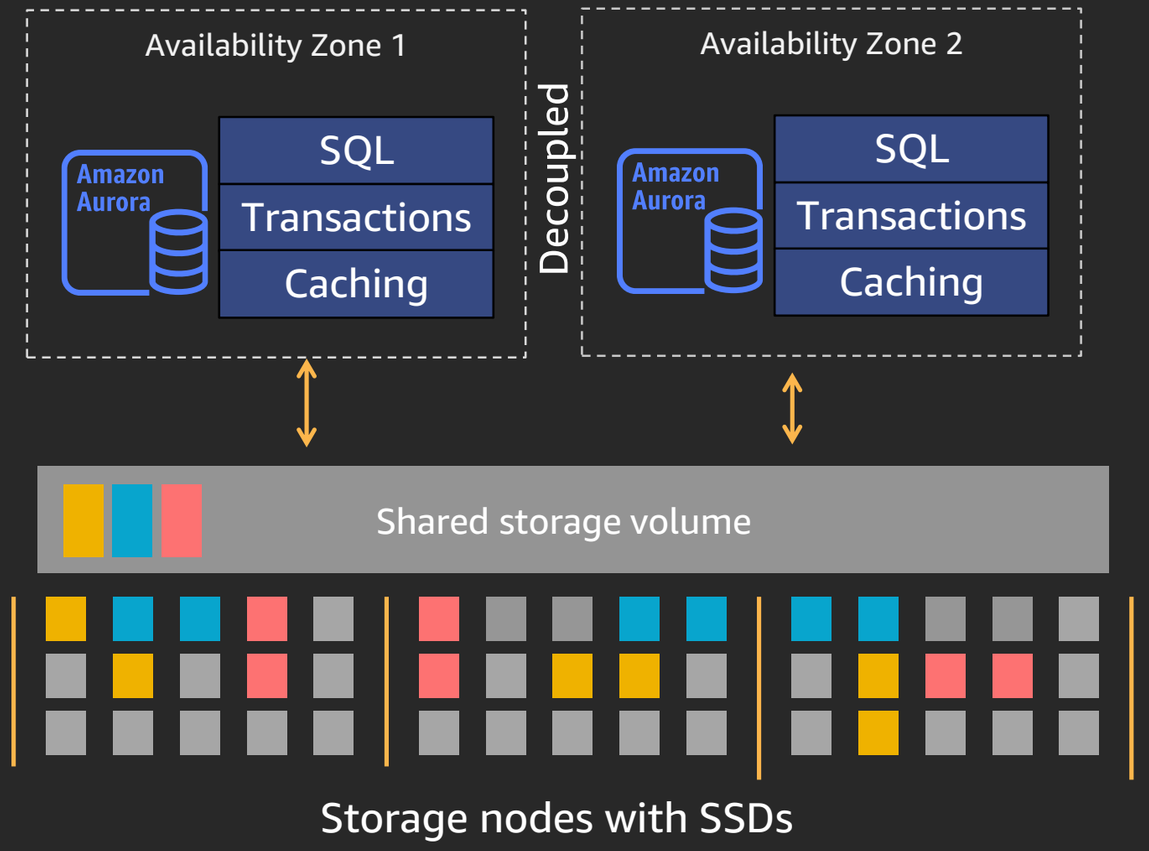
Heavy-weight consensus protocol:
Hot partitions and struggle with cross-partition queries

Scale-out, distributed architecture

Log applicator is pushed to storage to reduce write amplification

4/6 Write quorum provides fault tolerance and reduces performance jitter

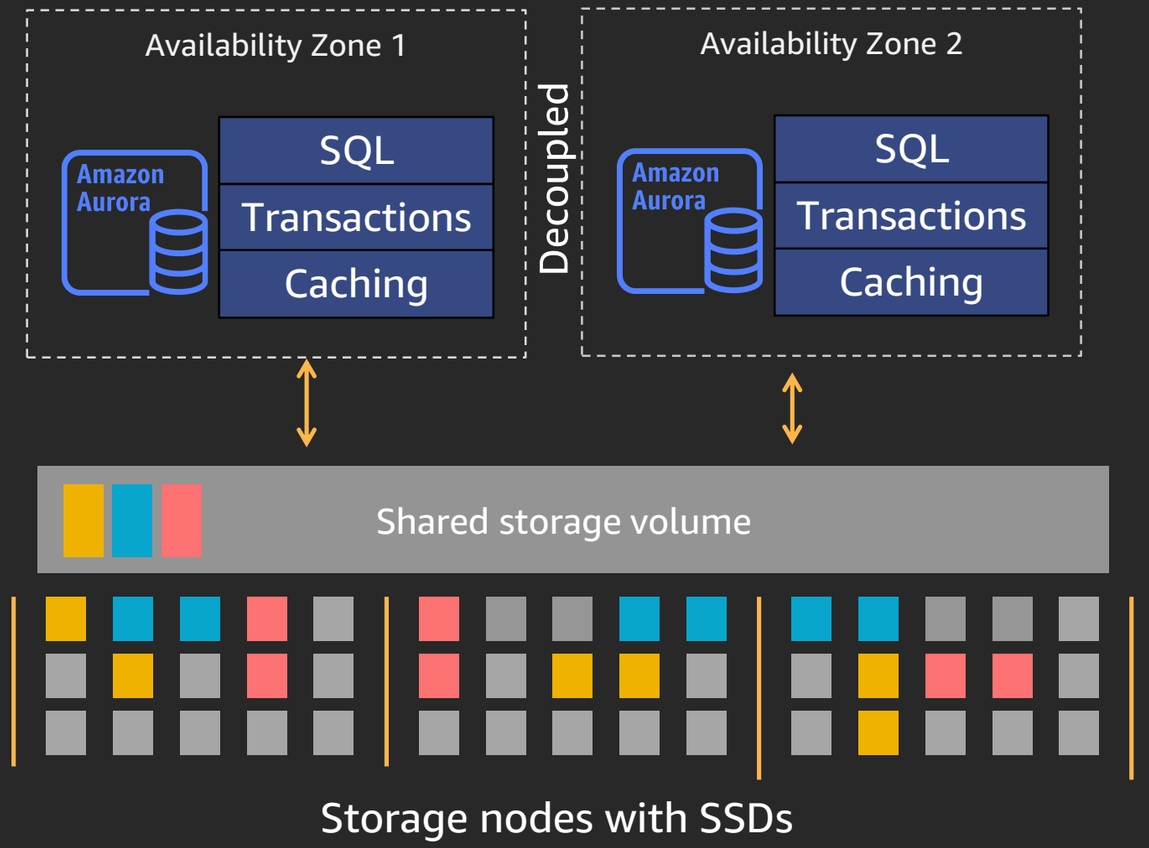
- ✓ Read scale out
- ✓ AZ + 1 fault tolerance
- ✓ Instant database recovery



Scale-out, distributed architecture

New with Multi-Master

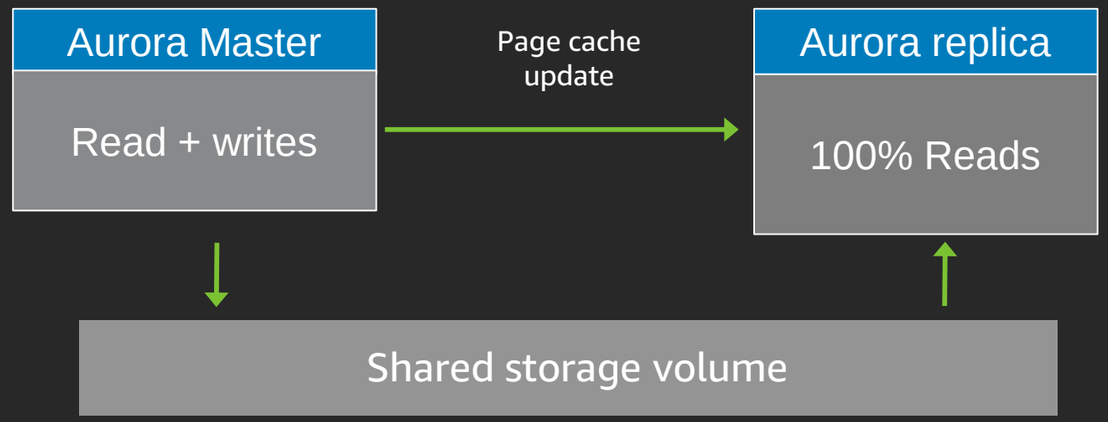
- ✓ Write scale out
- ✓ Continuous availability



Architecture deep dive

Read and write scale out

Aurora read scaling

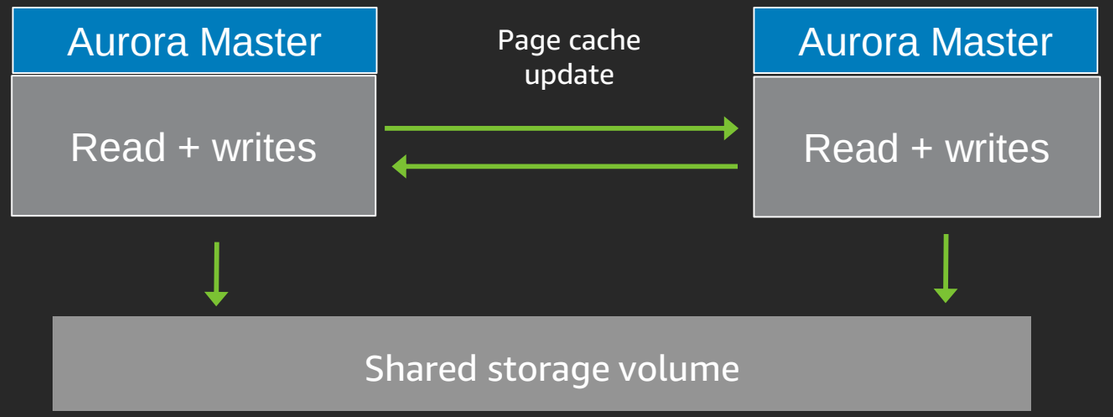


Page cache updated using physical delta change

No writes on replica

Shared storage

Aurora Multi-Master

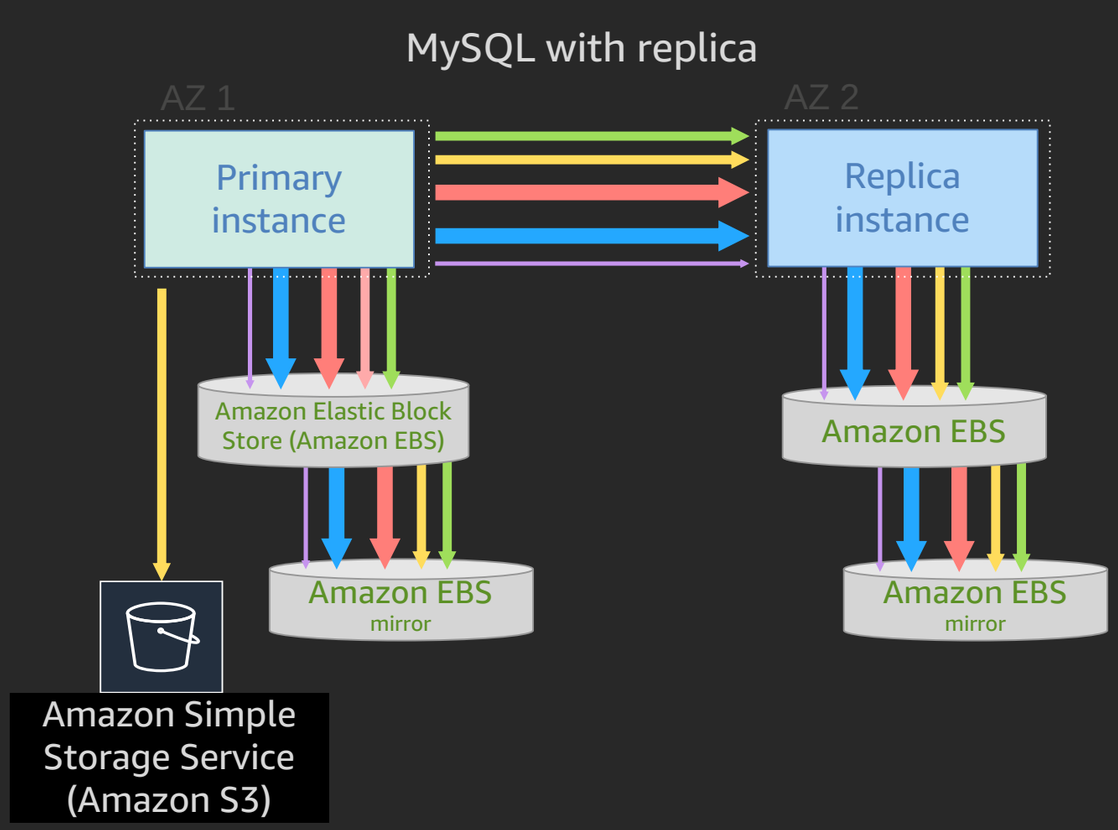


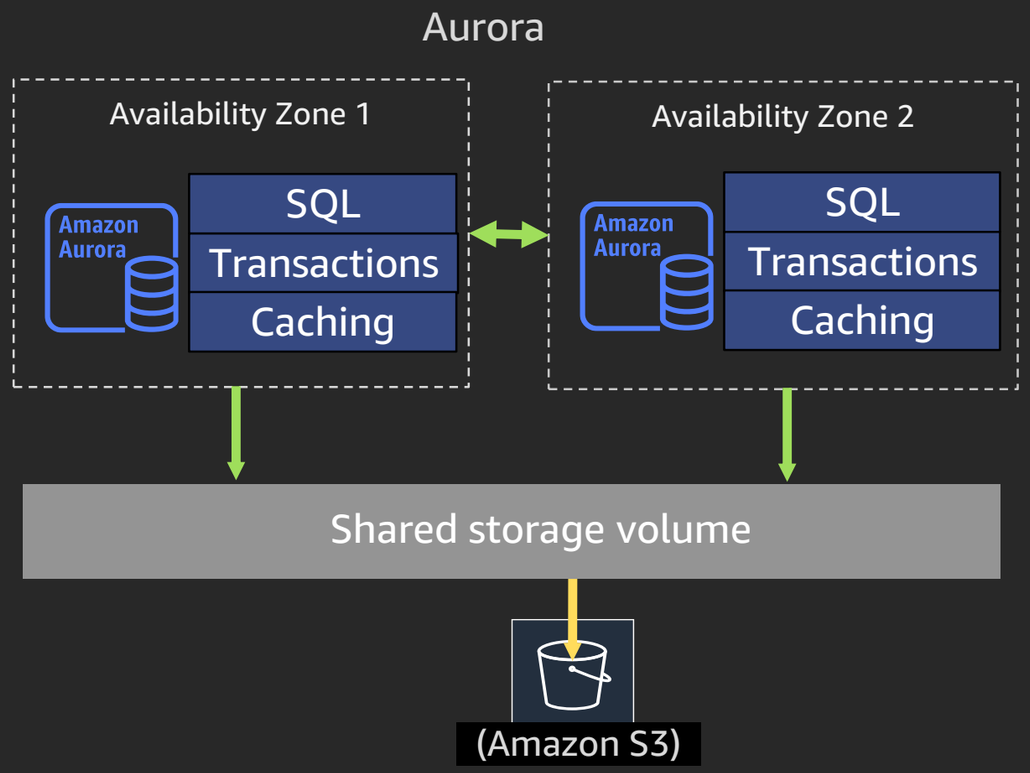
Page cache updated using physical delta change

All instances can execute write transactions

Shared storage with optimistic conflict detection

I/O profile comparison





Type of write

Log

Binlog

Data

Double-write

FRM files

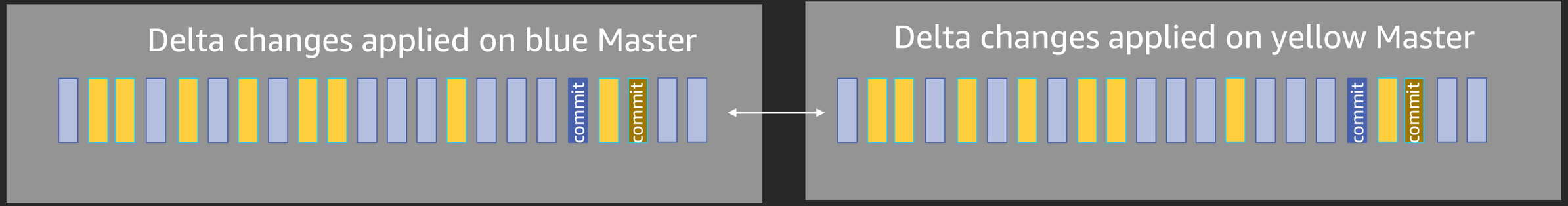
Decoupled transaction execution

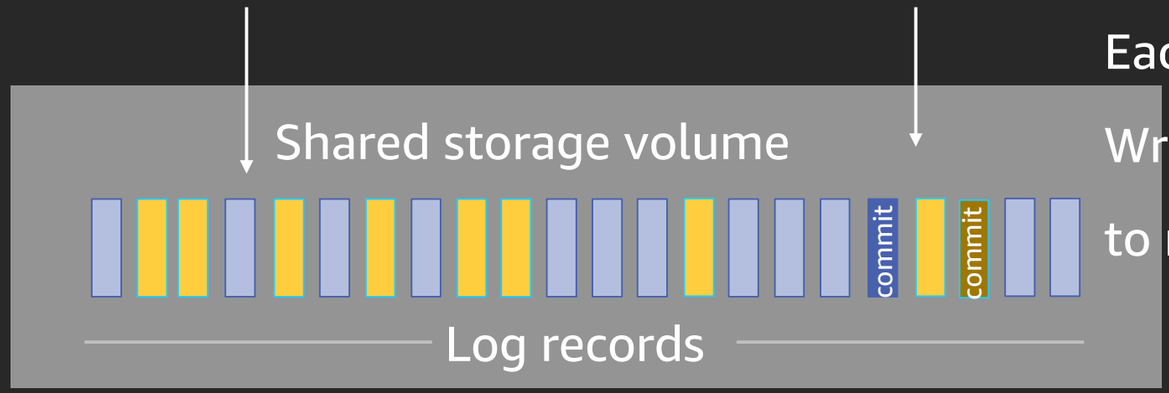
Commit

Transaction is committed after

all writes are confirmed

Commit



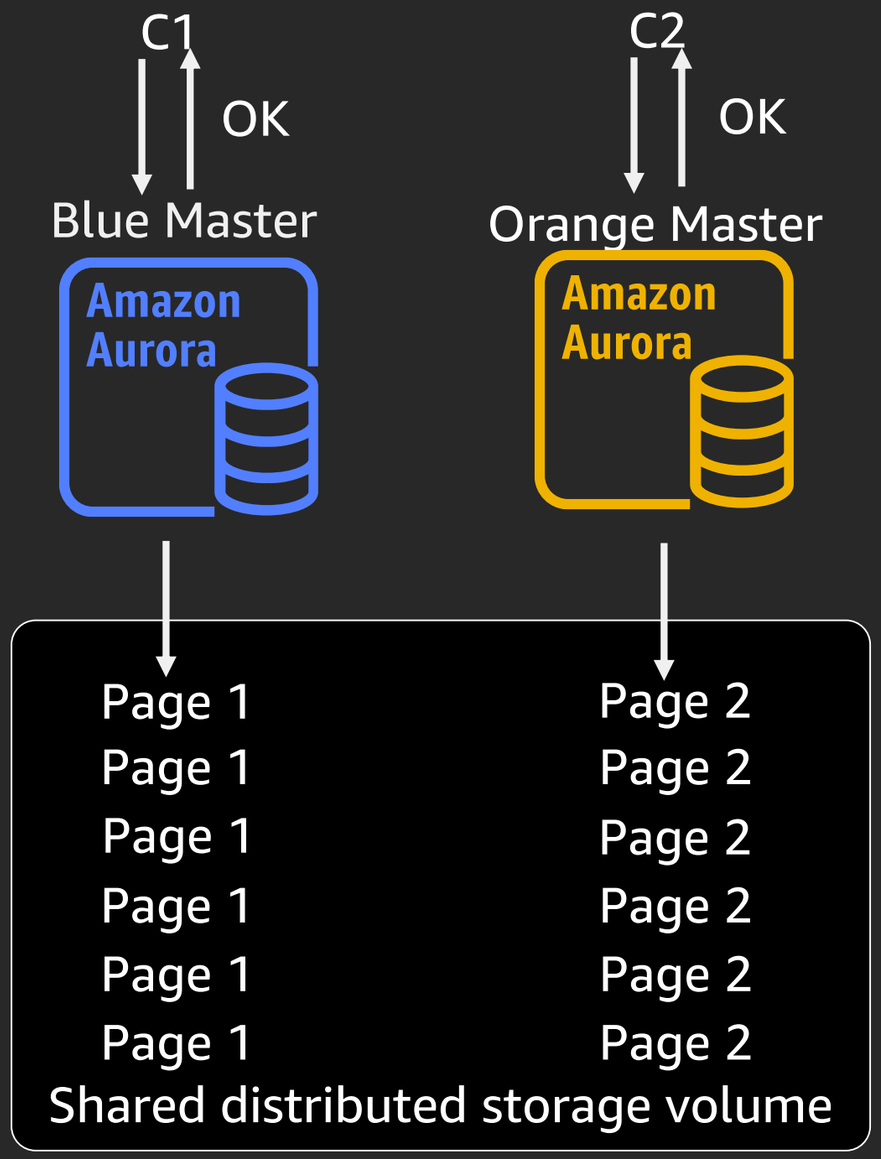


Each node sees its own write

Writes from other nodes are subject

to replication delay

Nonconflicting write

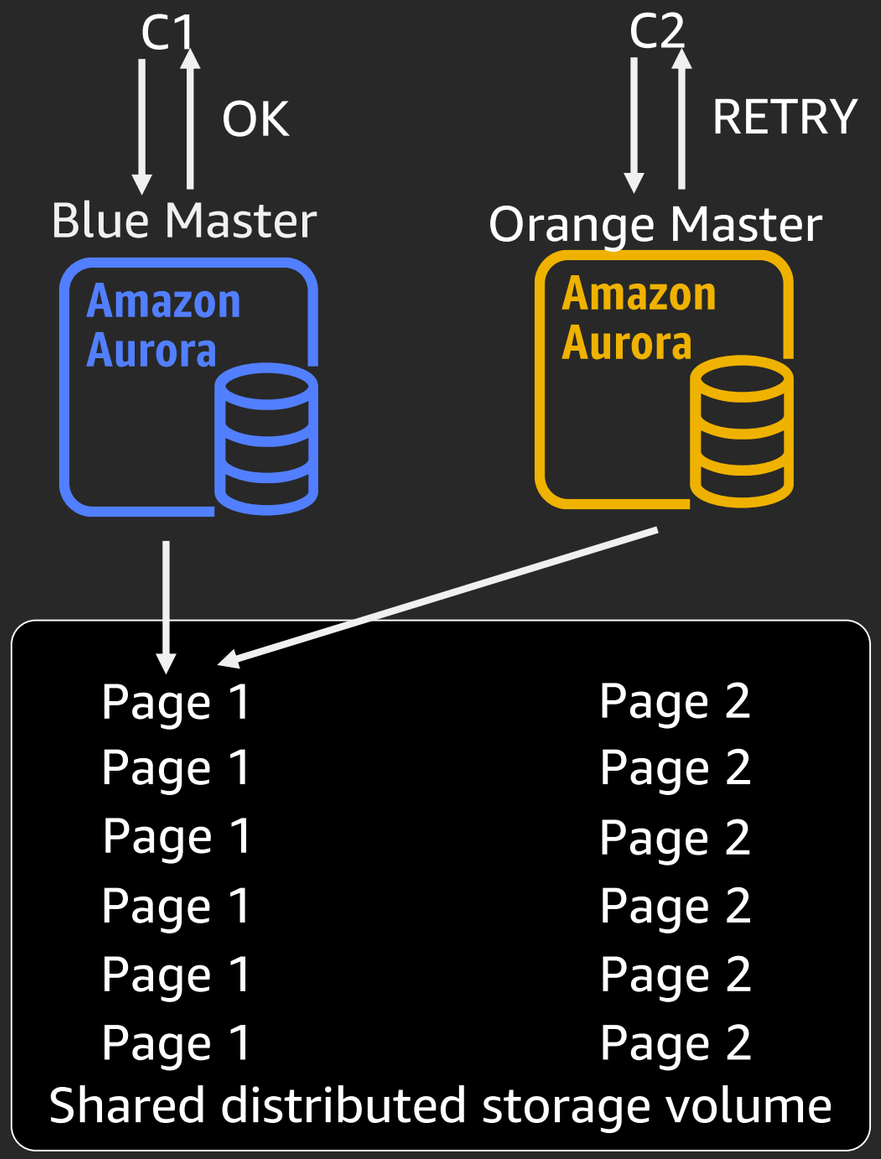


Nonconflicting writes originating on different masters on different tables

Time	Blue Master	Orange Master
1	Begin Trx (BT1)	Begin Trx (OT1)
2	Update (table1)	Update (table2)
3	Commit (BT1)	Commit (OT1)

No synchronization

Physical Conflict

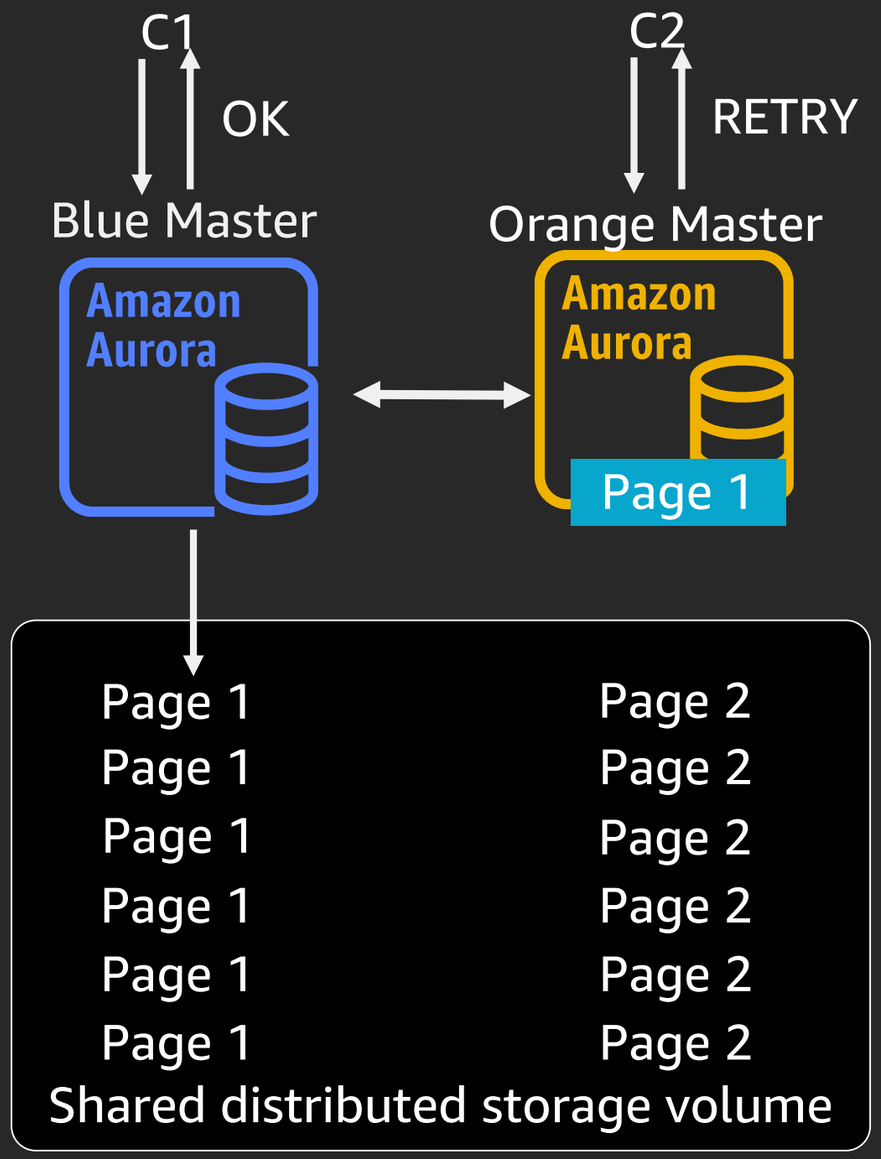


Conflicting writes originating on different masters on the same table

Time	Blue Master	Orange Master
1	Begin Trx (BT1)	Begin Trx (OT1)
2	Update (row1, table1)	Update (row1, table1)
3	Commit (BT1)	Rollback (OT1)

Optimistic conflict resolution

Logical conflict



Conflicting writes originating on different masters on the same table

Time	Blue Master	Orange Master
1	Begin Trx (BT1)	Begin Trx (OT1)
2	Update (row1, table1)	
3		Update (row1, table1) and Rollback (OT1)
4	Commit (BT1)	

No distributed locking

Conflict detection summary

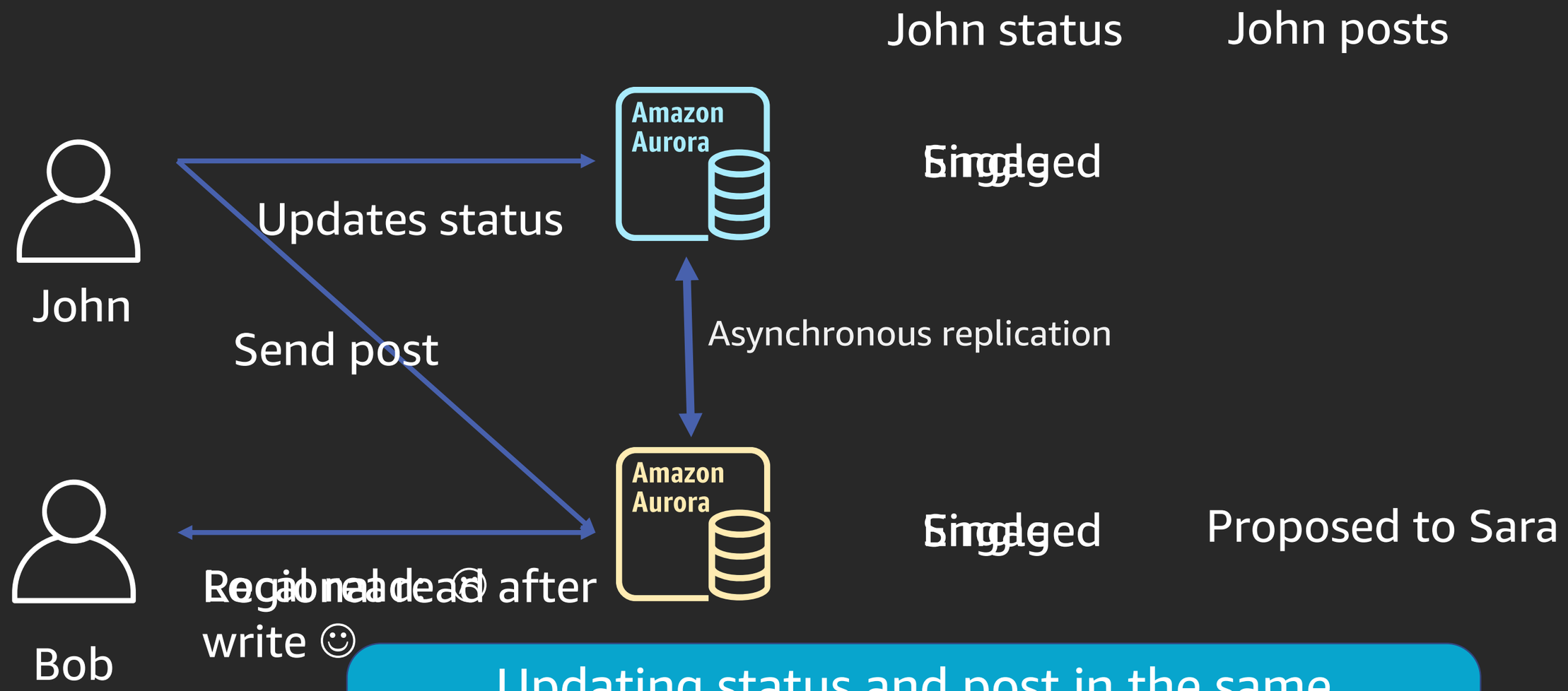
- Aurora Multi-Master uses optimistic conflict detection
- Storage nodes detect conflicts
- A transaction is only committed and acknowledged after the transaction is durable and conflicts have been resolved

Consistency model

Instance Read-After-Write (INSTANCE_RAW): A transaction can observe all transactions previously committed on this instance, and transactions executed on other nodes, subject to replication lag

Regional Read-After-Write (REGIONAL_RAW): A transaction can observe all transactions previously committed on all instances in the cluster

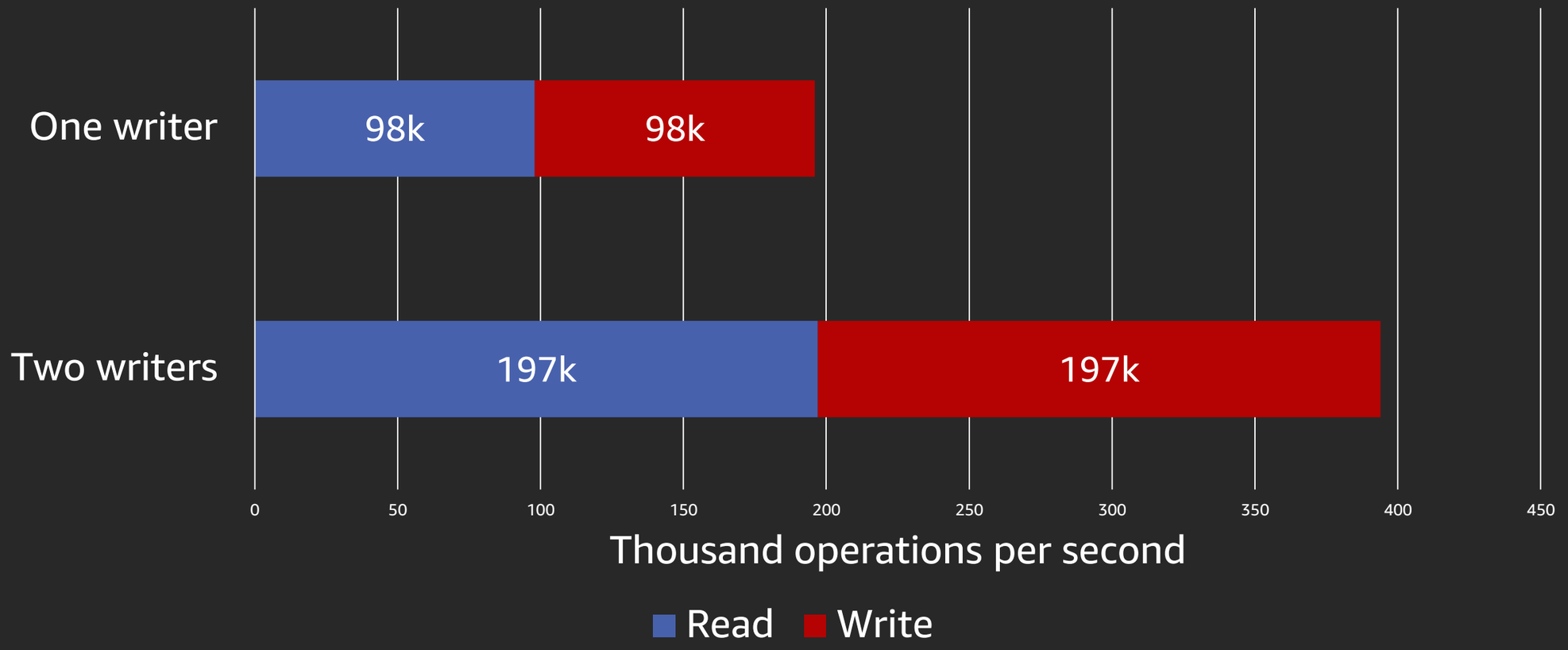
Consistency model



Changing the consistency model of a session

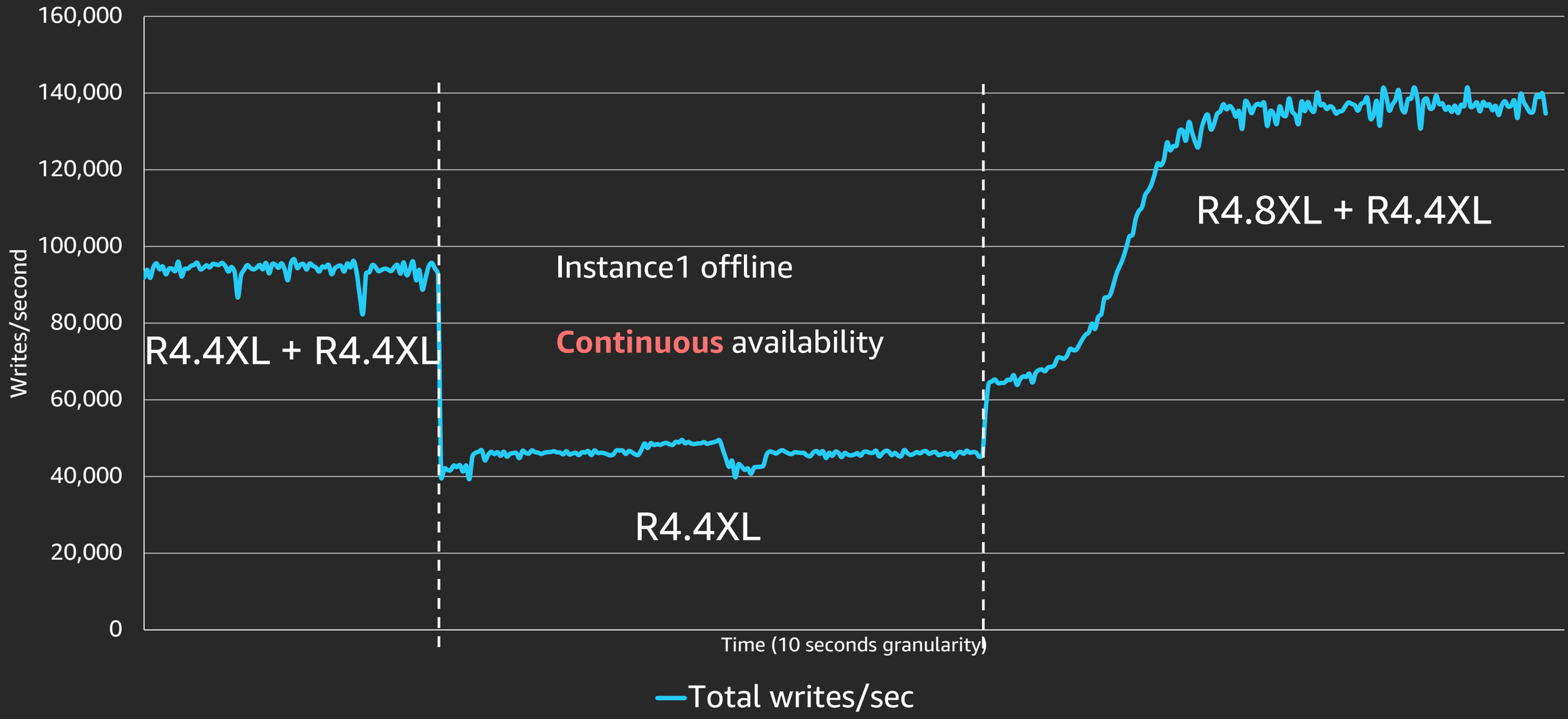
```
mysql> select @@aurora_mm_session_consistency_level;
+-------------------------------------+
| @@aurora_mm_session_consistency_level |
+-------------------------------------+
| INSTANCE_RAW                        |
 +-------------------------------------+
1 row in set (0.01 sec)

mysql> set session aurora_mm_session_consistency_level = 'REGIONAL_RAW’;
Query OK, 0 rows affected (0.00 sec)

mysql> select @@aurora_mm_session_consistency_level;
+-------------------------------------+
| @@aurora_mm_session_consistency_level |
+-------------------------------------+
| REGIONAL_RAW                        |
+-------------------------------------+
1 row in set (0.03 sec)
```

Scalability



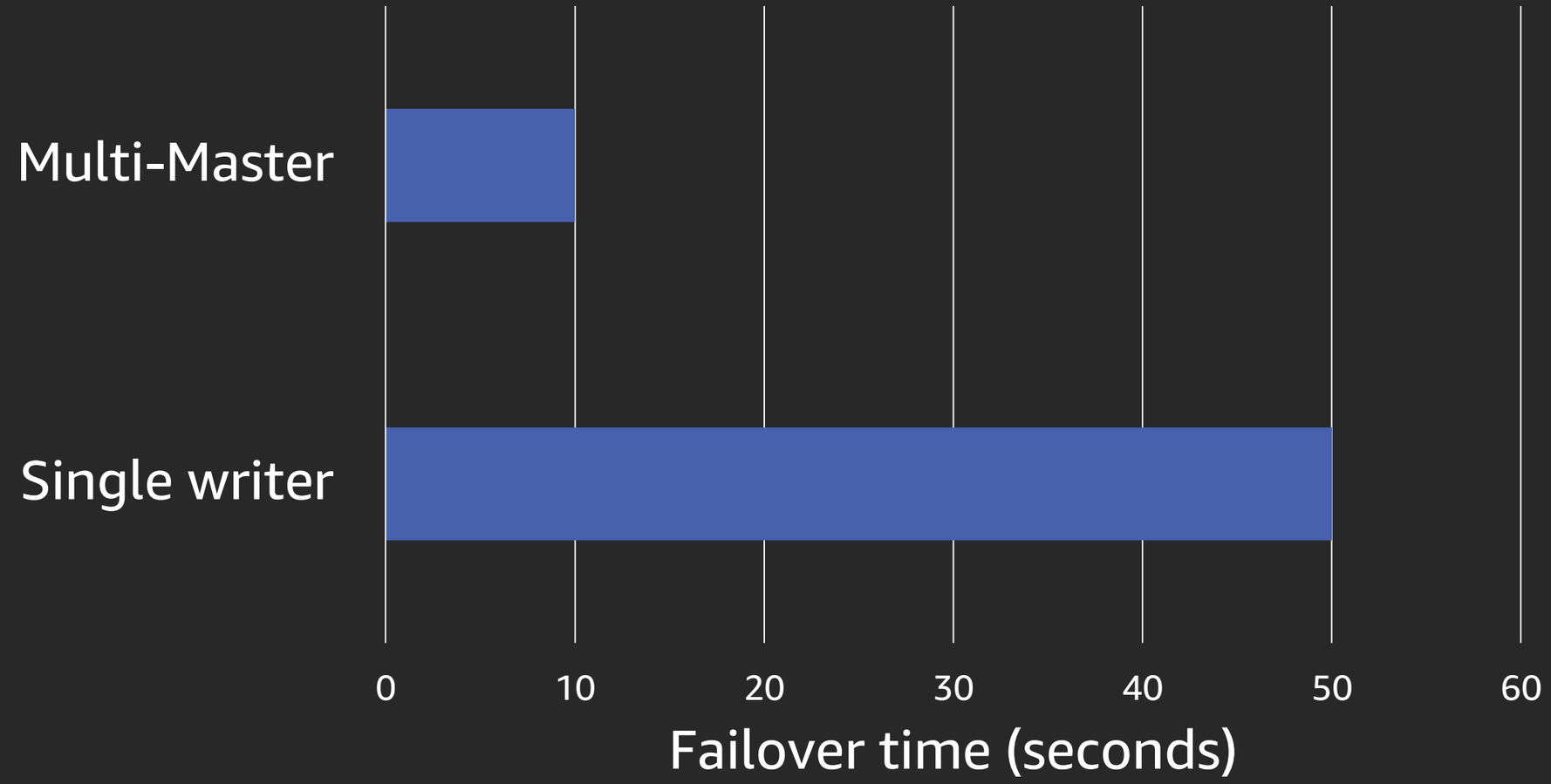
Sysbench, 50% read-write mix on 250 tables, containing 2.5M rows each on 8XL instance size

Scaling writes/seconds in Aurora Multi-Master



Sysbench write only, 500 tables, 2.5M rows per tables, no conflicts

Failover time for an unmodified application



Failover time when using the MariaDB driver connecting to the cluster endpoint

Best practices

Connecting to the cluster

The cluster endpoint follows an available instance. The cluster endpoint will be changed in case of a failover.

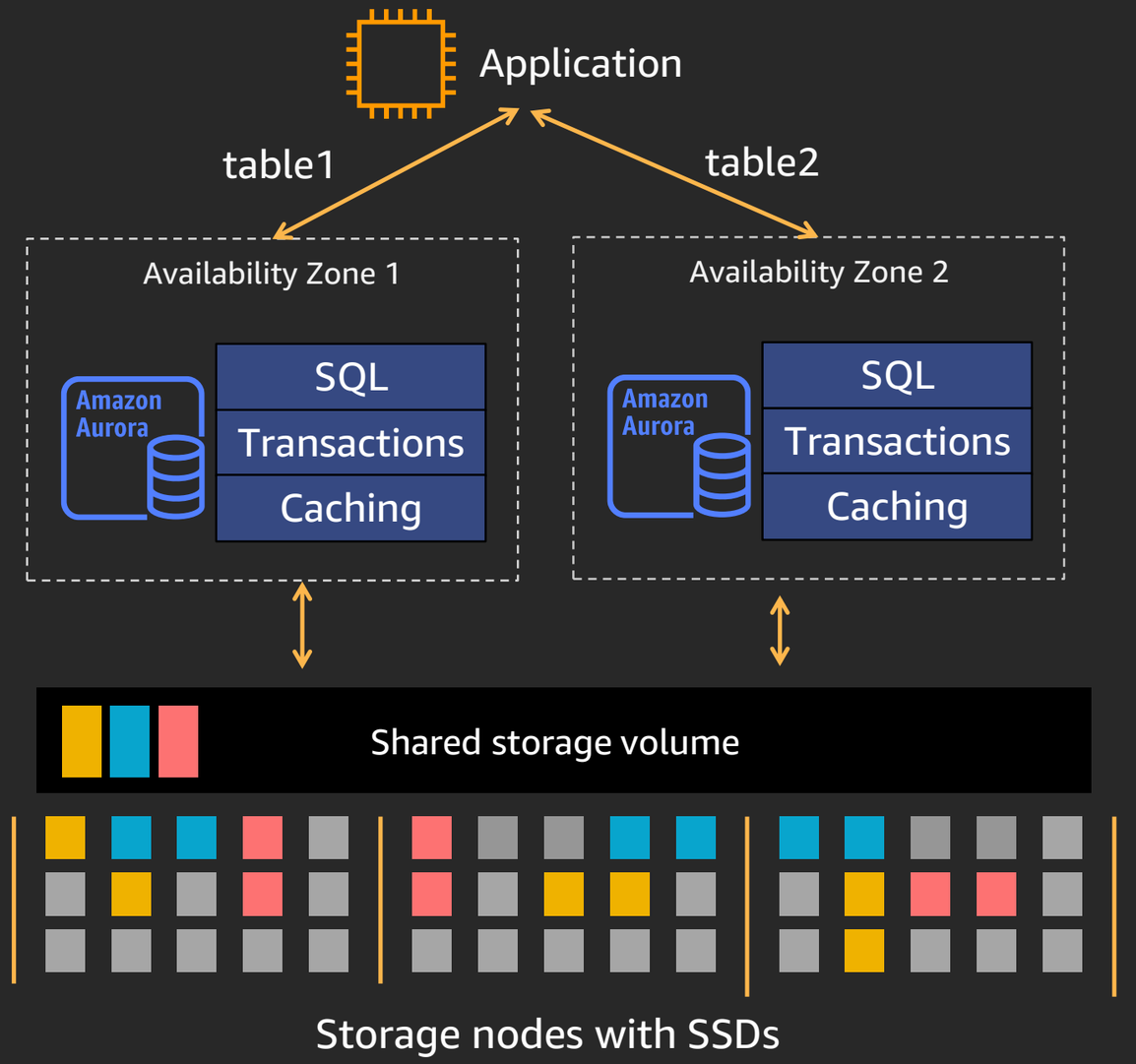
Each instance has an instance endpoint pointing to a specific instance

Custom endpoints allow more flexibility

Performance best practices

Structure the workload to limit conflicts between database instances. For example, *prefer* writing to a table from a single database instance.

Use the instance endpoint to distribute writes to many database instances. The cluster endpoint does not load balance.



Handling conflict in the application

Physical and logical conflicts cause a deadlock error to be returned

Conflicts can be monitored from `information_schema multiwriter_conflict_statistics` and from Amazon CloudWatch

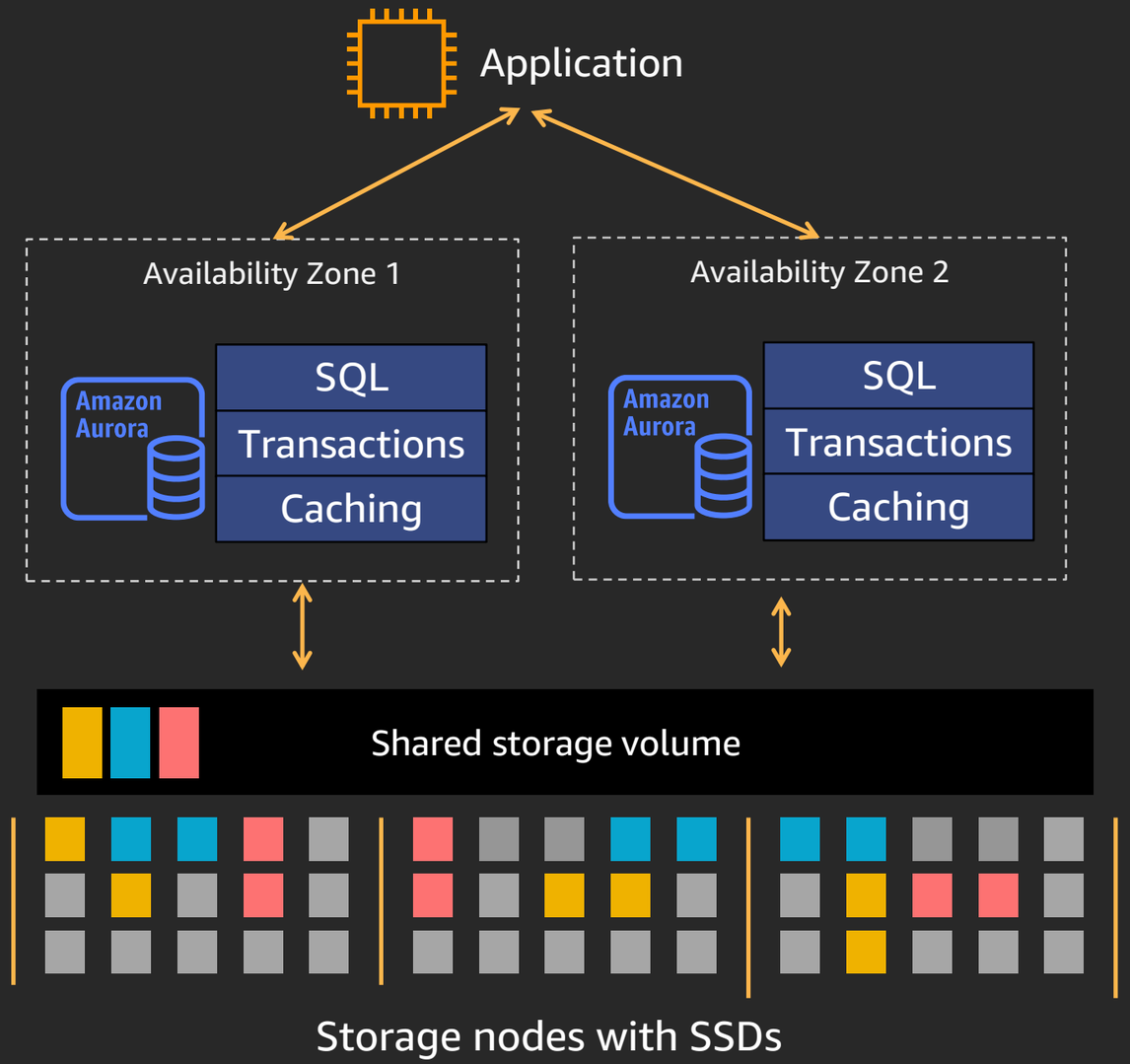
```
mysql> select `table`, `index`, physical_conflicts, logical_conflicts, split_conflicts from
information_schema.multiwriter_conflict_statistics limit 10;
+--------+---------+--------------------+-------------------+-----------------+
| table  | index   |physical_conflicts  | logical_conflicts | split_conflicts |
+--------+---------+--------------------+-------------------+-----------------+
| test90 | PRIMARY |                  0 |                 0 |               0 |
| test90 | k_90    |                  0 |                 0 |               0 |
| test94 | PRIMARY |                  0 |                 0 |               0 |
| test94 | k_94    |                  0 |                 0 |               0 |
| test97 | PRIMARY |                  0 |                 0 |               0 |
| test97 | k_97    |                  0 |                 0 |               0 |
| test2  | PRIMARY |                  0 |                 0 |               0 |
| test2  | k_2     |                  0 |                 0 |               0 |
| test98 | PRIMARY |                  0 |                 0 |               0 |
| test98 | k_98    |                  0 |                 0 |               0 |
+--------+---------+--------------------+-------------------+-----------------+
10 rows in set (0.16 sec)
```

Availability best practices

Application connect to both database instances at the application layer

Monitor the availability of database instances at the application layer and redirect the load to the available instance

The cluster endpoint follows an available instance



Monitoring cluster health

```
mysql> select server_id, replica_lag_in_milliseconds, last_reported_status from
information_schema.replica_host_status;

+----------------------+------------------------------+----------------------+
| server_id            | replica_lag_in_milliseconds  | last_reported_status |
+----------------------+------------------------------+----------------------+
| mm-reinvent-demo-1-1 |                       37.302 | Online               |
| mm-reinvent-demo-1-2 |                       39.907 | Online               |
+----------------------+------------------------------+----------------------+
```

Demo

AWS Management Console

AWS services

Find Services

You can enter names, keywords or acronyms.

Example: Relational Database Service, database, RDS

Recently visited services

- CloudFormation
- EC2
- CloudWatch
- RDS
- Lambda

All services

Compute
- EC2
- Lightsail
- ECR
- ECS
- EKS

Management & Governance
- AWS Organizations
- CloudWatch
- AWS Auto Scaling
- CloudFormation

AWS Cost Management
- AWS Cost Explorer
- AWS Budgets
- AWS Marketplace Subscriptions

Access resources on the go

Access the Management Console using the AWS Console Mobile App. Learn more

Explore AWS

Free Digital Training

Get access to 350+ self-paced online courses covering AWS products and services. Learn more

AWS IQ

Complete your AWS projects faster with help from AWS Certified third-party experts. Get started

AWS Security Hub

Summary

Aurora Multi-Master enables:

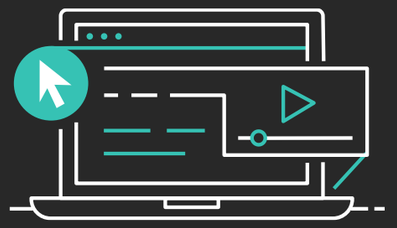
- Have multiple writers to scale write throughput

- Have writers in multiple Availability Zones for continuous availability

Available now for Aurora compatible with MySQL 5.6

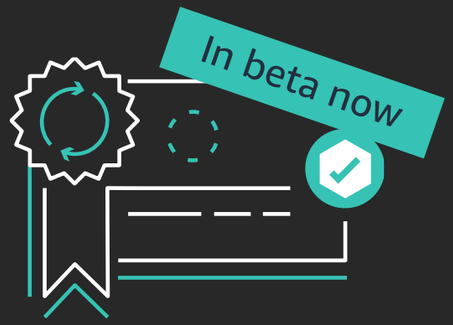
Learn databases with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate database skills



25+ free digital training courses cover topics and services related to databases, including:

- Amazon Aurora
- Amazon Neptune
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Redshift
- Amazon RDS



Validate expertise with the new **AWS Certified Database - Specialty** beta exam

Visit aws.training

aws training and certification

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.