AWS 是如何改造 MySQL

ADD YOUR POWERPOINT TITLE HERE

演讲人：刘梦馨

跨界互联
数聚未来

第四届中国数据分析师行业峰会
CHINA DATA ANALYST SUMMIT
北京 中国大饭店 2017. 07

Agenda

- AWS Aurora background
- Log as database
- Durability and Availability
- Fast recovery
- High throughput
- Performance metric

Aurora Background

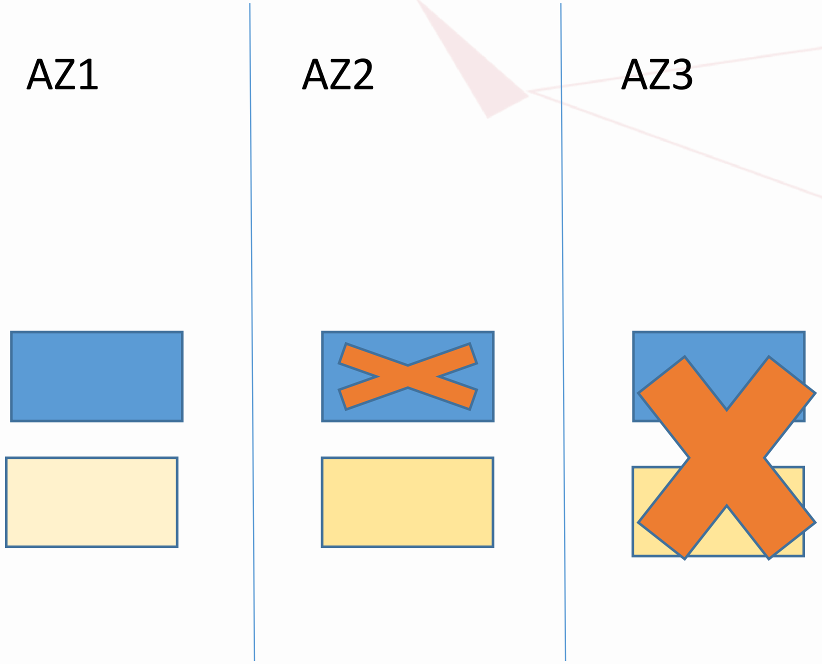
- First announce at Nov, 2014 and GA at Jul, 2015
- Compatible with MySQL 5.6
- 6 replica cross 3 AZ
- 99.99% available
- < 10s recovery
- 5X throughput compare MySQL 5.6

Durable and Available

- 6 replica cross 3 region
- Use 6 nodes quorum protocol
  - Write: 4/6 nodes
  - Read: 3/6 nodes
  - Gossip to fill the gap
- Async logs to S3 with 99.999999999% durable

Durable and Available

- Multi tenant environment
- Every time some nodes down
- If one AZ is un reachable …



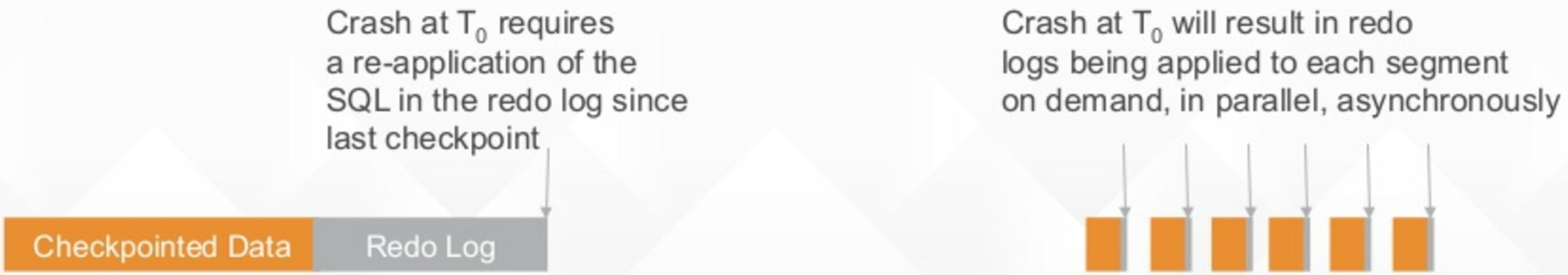
Durable and Available

- If one AZ and one instance is down
- Write is unavailable, read still work
- Use 3 read quorum to rebuild replica

- If node get down during rebuilding replica …

Durable and Available

- Split replica to 10GB segment
- Only rebuild segment that not meet write quorom
- With 10GB network recover in 10 seconds



Durable and Available

- Operation benefit
  - Can rolling update system and software by just offline a node
  - Migrate user data by delete data
  - OverSale and keep performance by migrate data

Durable and Available

- Performance impact
  - Read/Write need network IO
  - IO times * 6
  - Slowest node will hang the request

- How to achieve 5X throughput?

Log is database

- MySQL write multiply times when commit a record
- Undo/Redo log, binlog, data pages, double write frm …
- Master and slave are in sync mode

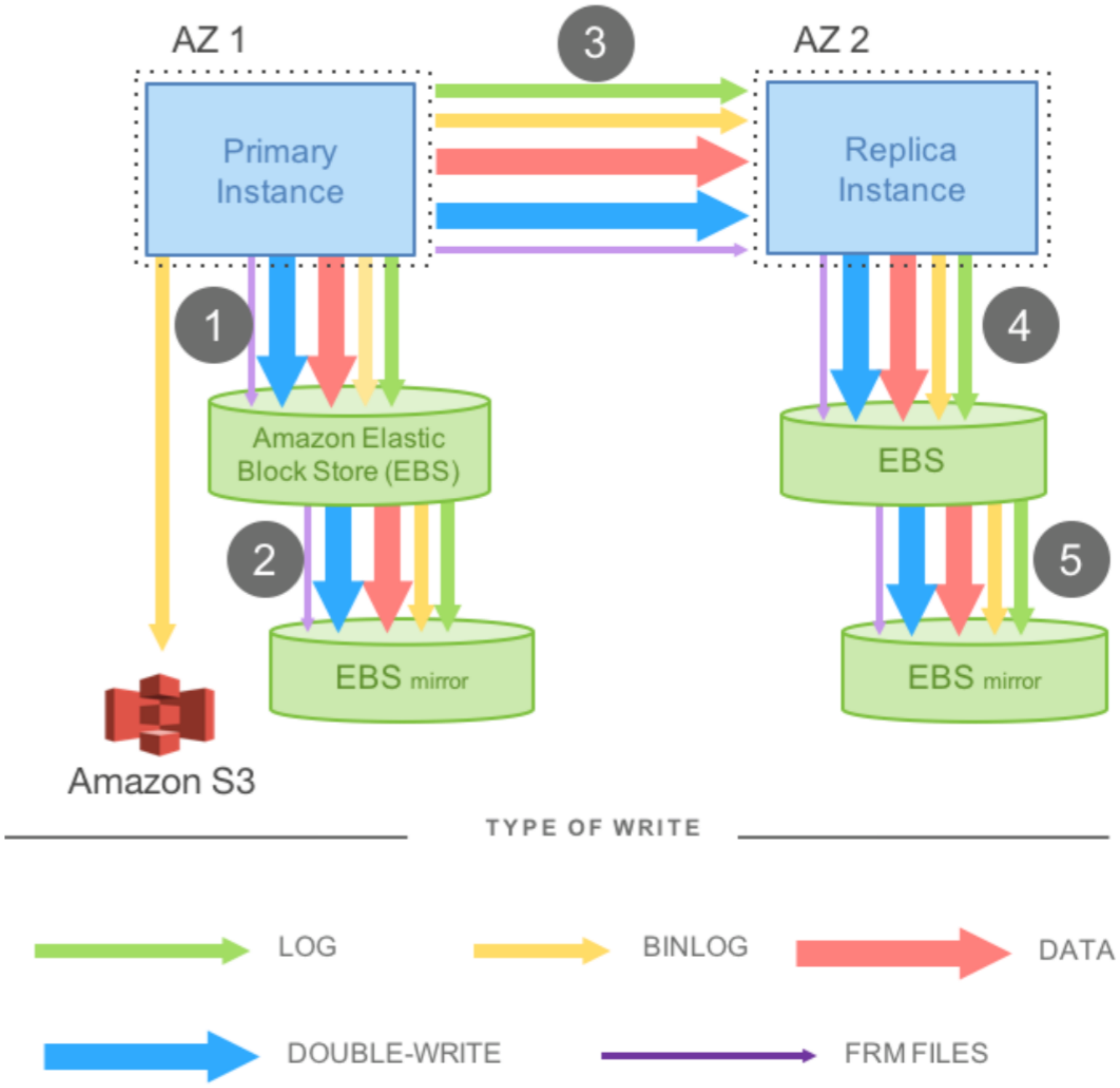
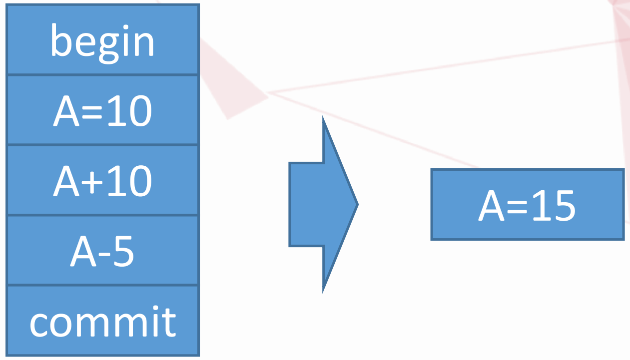


Figure 2: Network IO in mirrored MySQL

- Think about 6 replica

Log is database

- WAL can regenerate all data record
  - Like checkout from git by commit ids
  - Modern database use WAL to store data
- Only redo log need to be replicated
- Periodically checkpoint to reduce log length



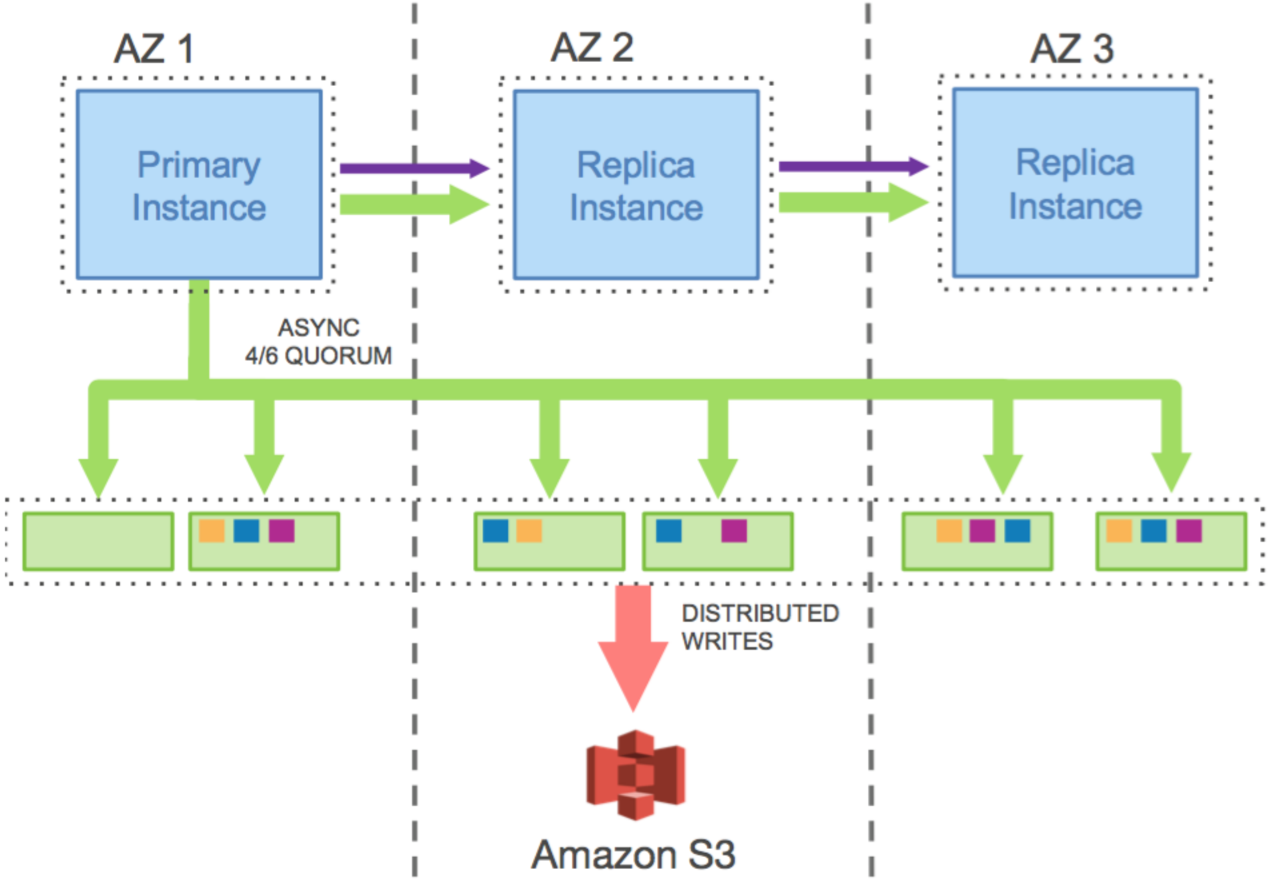


Figure 3: Network IO in Amazon Aurora

- Transfer redo log and meta changes to replica to update cache

Log is database

- Write Operation
  - Client: make SQL changes and commit
  - Worker thread: commit redo logs to log service and return
  - Notify thread: async wait log service complete quorum and notify client
- Write will not block thread

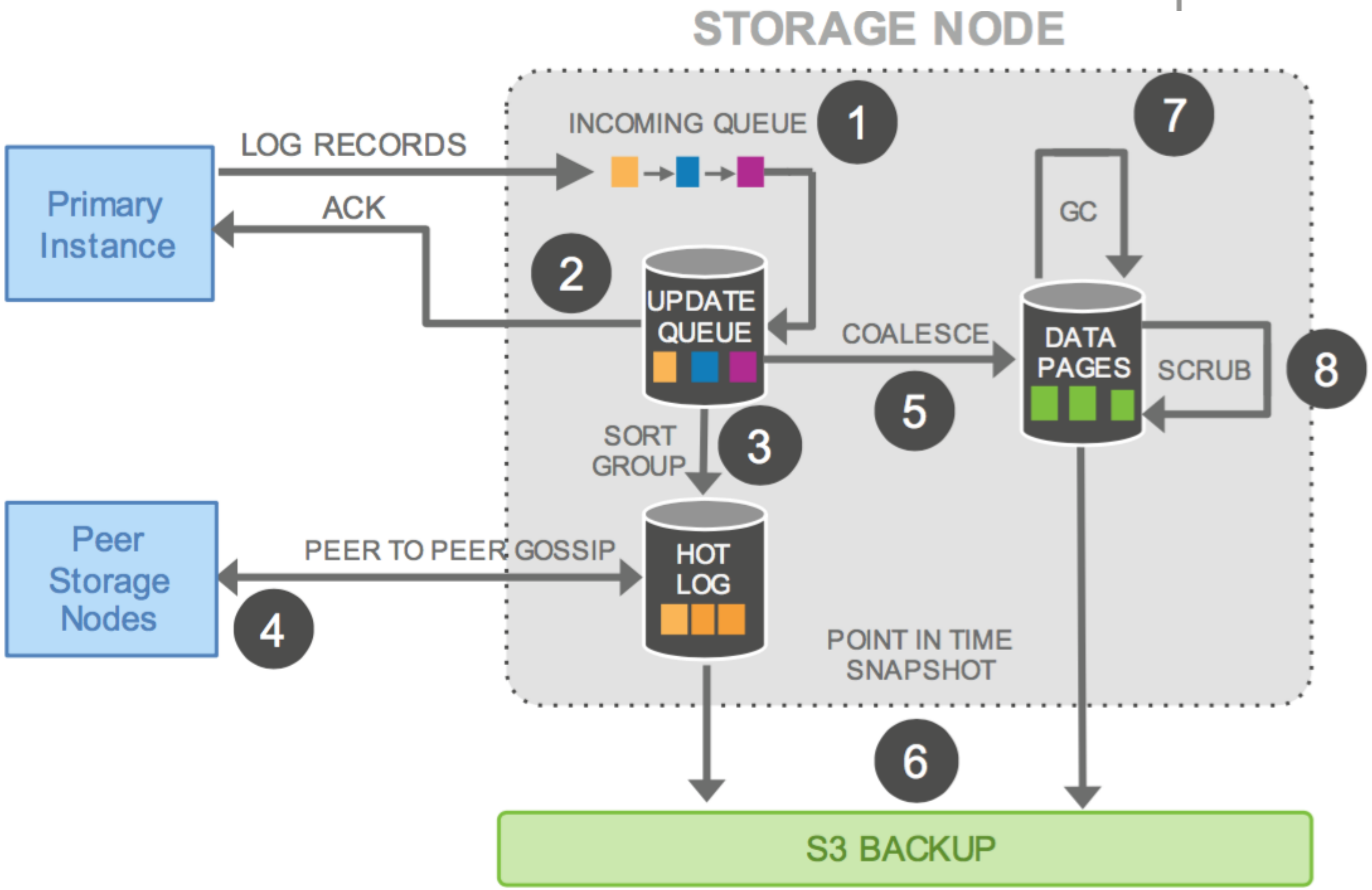
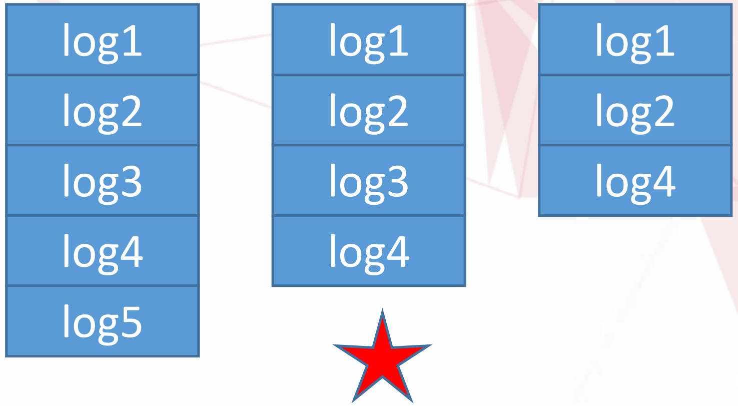


Figure 4: IO Traffic in Aurora Storage Nodes

Log is database

- Read Operation
  - Most time no quorum read
  - Read local cache
  - Engine knows the durable commit log id
  - Only quorum read during failover



Fast recovery

- Tradition MySQL recovery
  - Checkpoint every day/hour
  - Use WAL to replay data update
  - >10 minutes
- Semi Sync
  - No consistence guarantee between master and slave
  - Add overhead for each transaction

Fast recovery

- Aurora separate compute engine and storage
- Storage node recovery
  - Use read quorum
  - Periodical CRC check and recovery
- Compute node recovery
  - WAL redo offload to storage node
  - Compute and truncate uncommitted log
  - < 10s recover time

Performance Metrics

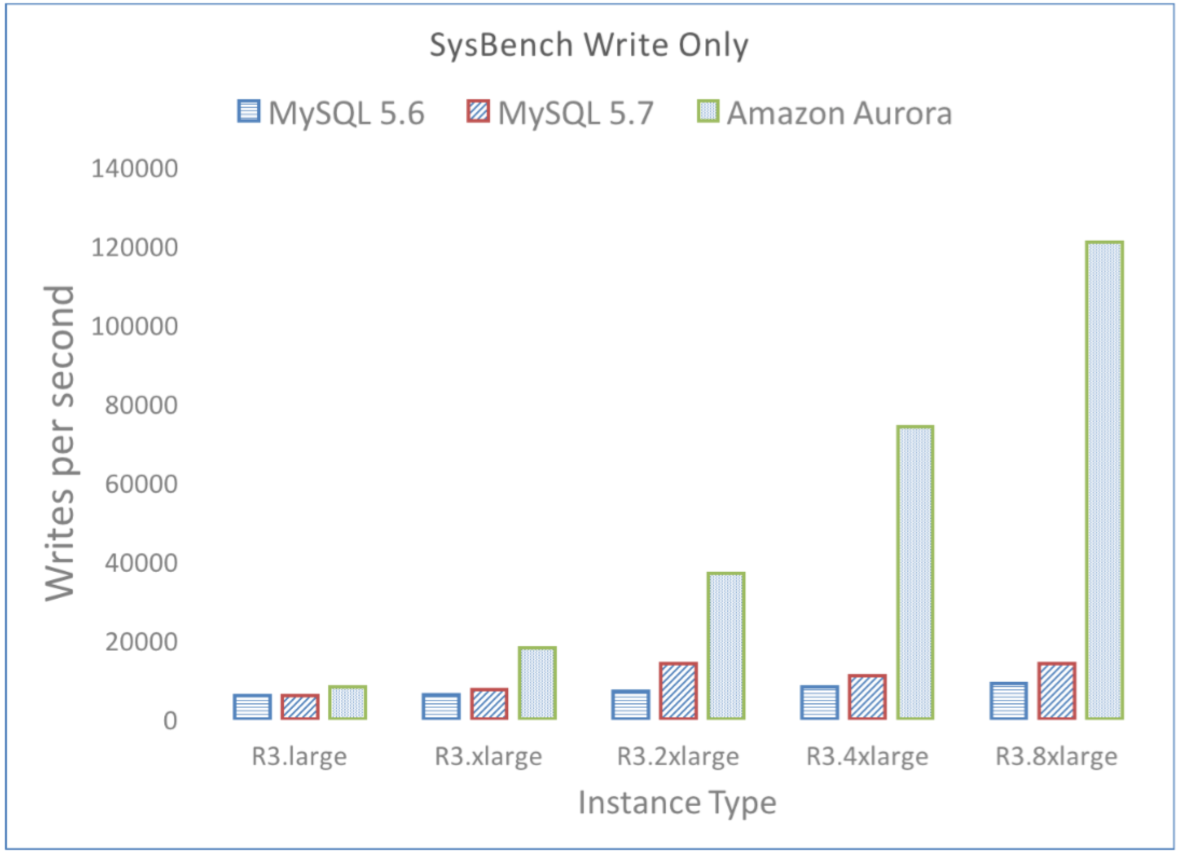


Figure 7: Aurora scales linearly for write-only workload

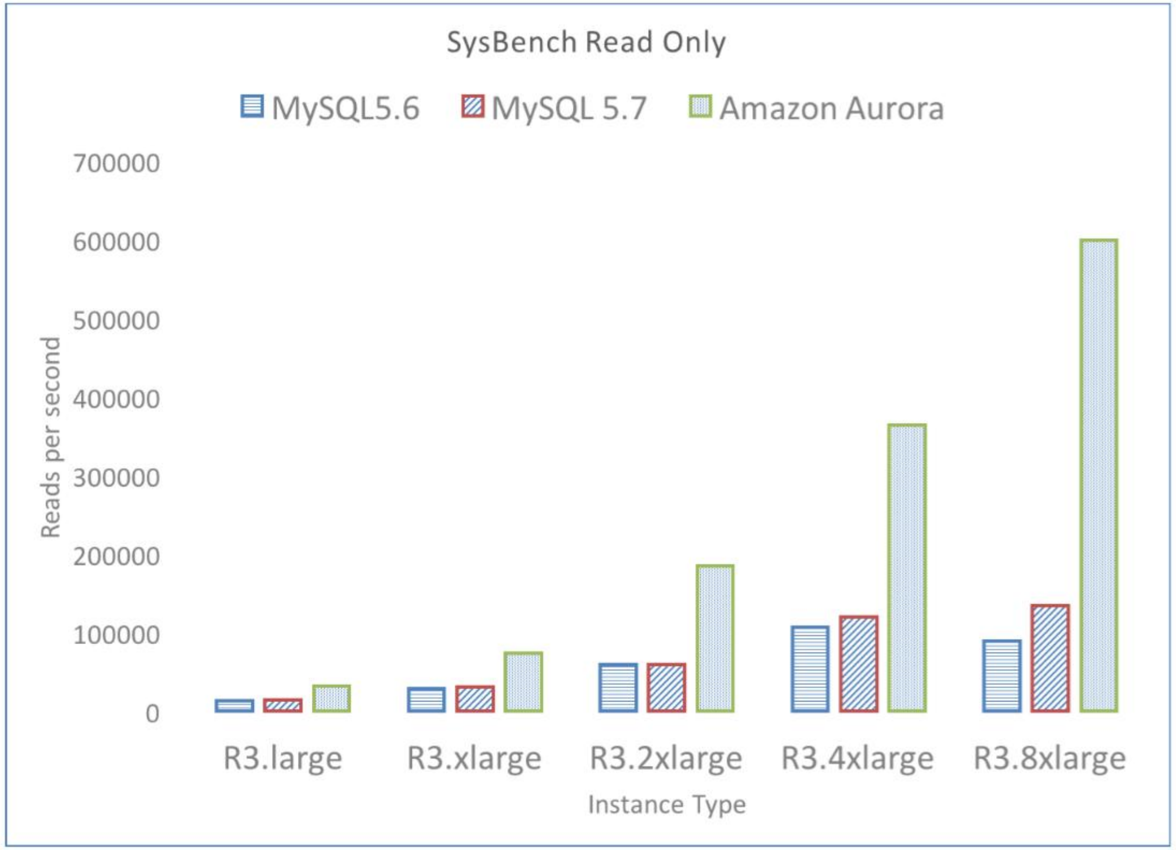


Figure 6: Aurora scales linearly for read-only workload

Performance Metrics

Table 2: SysBench Write-Only (writes/sec)

DB Size	Amazon Aurora	MySQL
1 GB	107,000	8,400
10 GB	107,000	2,400
100 GB	101,000	1,500
1 TB	41,000	1,200

Table 3: SysBench OLTP (writes/sec)

Connections	Amazon Aurora	MySQL
50	40,000	10,000
500	71,000	21,000
5,000	110,000	13,000

Table 4: Replica Lag for SysBench Write-Only (msec)

Writes/sec	Amazon Aurora	MySQL
1,000	2.62	< 1000
2,000	3.42	1000
5,000	3.94	60,000
10,000	5.38	300,000

Summary

- 6 replica with 10G segment to guarantee available
- Log as database to reduce log traffic
- Offload replication and recovery work to Logservice
- A robust and high performance infra can relieve application

- How other software: postgres, redis, es, hadoop …

CHINA DATA ANALYST SUMMIT

CDA 数据分析师 www.cda.cn

THANKS

跨界互联
数聚未来

第四届中国数据分析师行业峰会
CHINA DATA ANALYST SUMMIT