AWS SUMMIT

Deep Dive on Amazon Aurora

Ian Robinson
Specialist Solutions Architect, Data & Analytics, EMEA

18 May, 2017

What is Amazon Aurora?

Open source compatible relational database

Performance and **availability** of
commercial databases

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

Aurora Customer Adoption

Fastest growing service in AWS history

Business applications

Web and mobile

Content management

E-commerce, retail

Internet of Things

Search, advertising

BI, analytics

Games, media

ISCS

GE Oil & Gas

Alfresco™

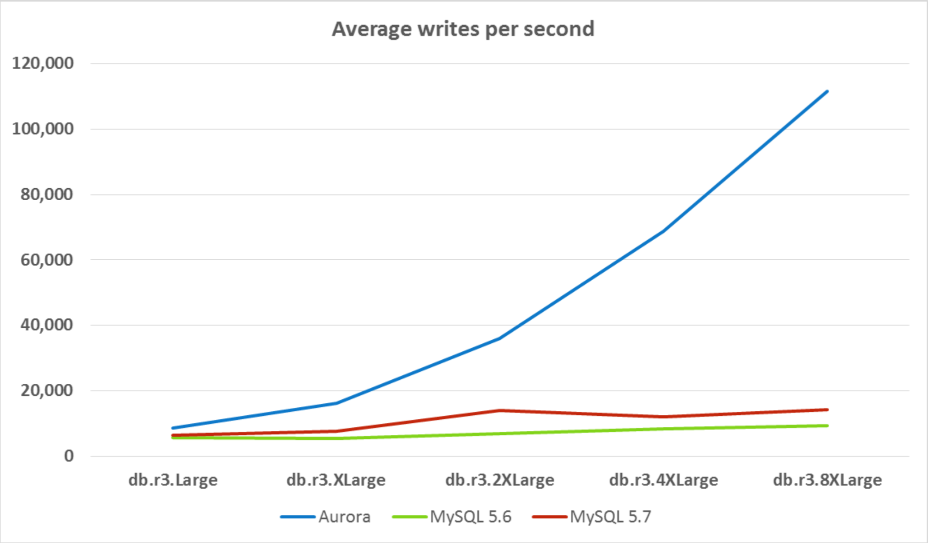
Earth Networks
Taking the Pulse of the Planet

Expedia®

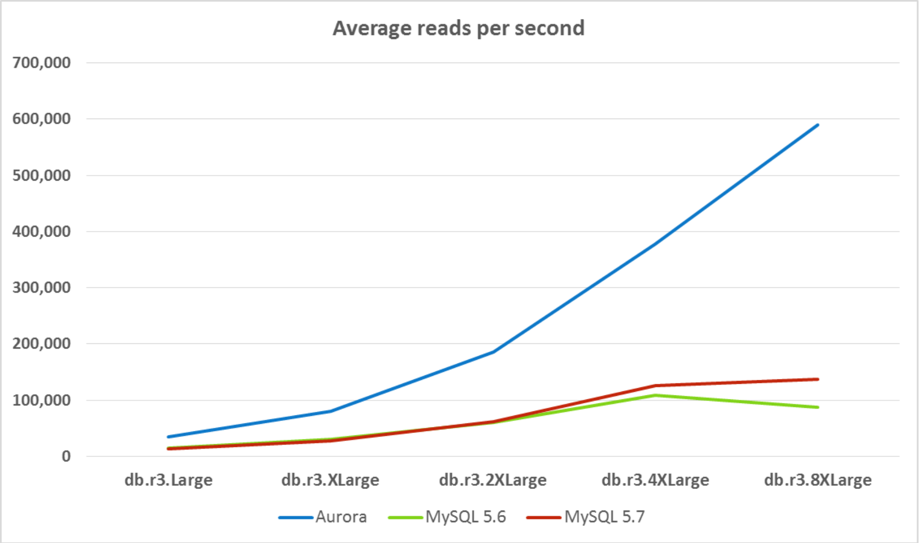
Performance

Scaling with Instance Sizes

WRITE PERFORMANCE



READ PERFORMANCE

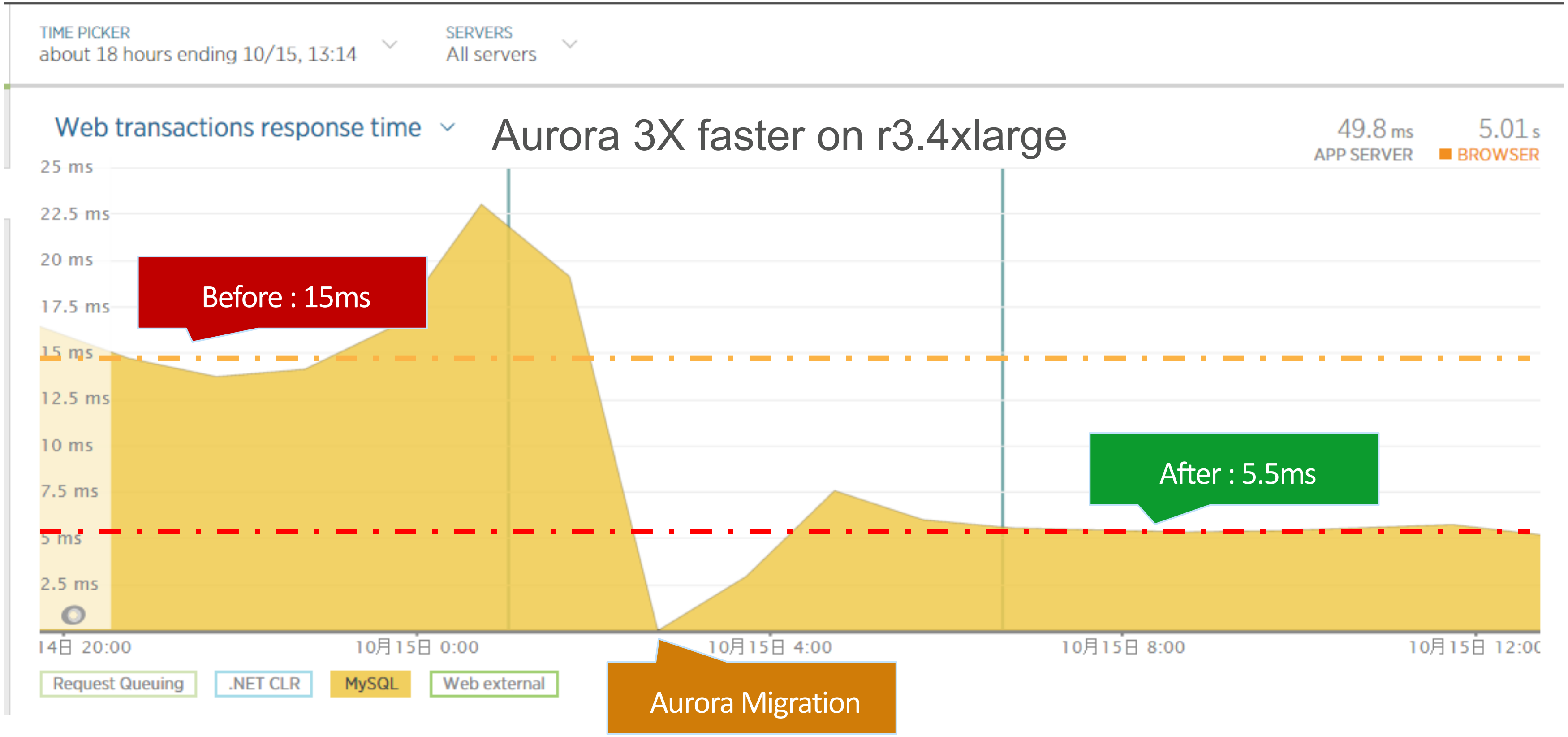


Aurora ━━ **MySQL 5.6** ━━ **MySQL 5.7** ━━

Aurora scales with instance size for both read and write.

Real-Life Data – Gaming Workload

Aurora vs. RDS MySQL – r3.4XL, MAZ



TIME PICKER
about 18 hours ending 10/15, 13:14

SERVERS
All servers

Aurora 3X faster on r3.4xlarge

Before : 15ms

After : 5.5ms

Aurora Migration

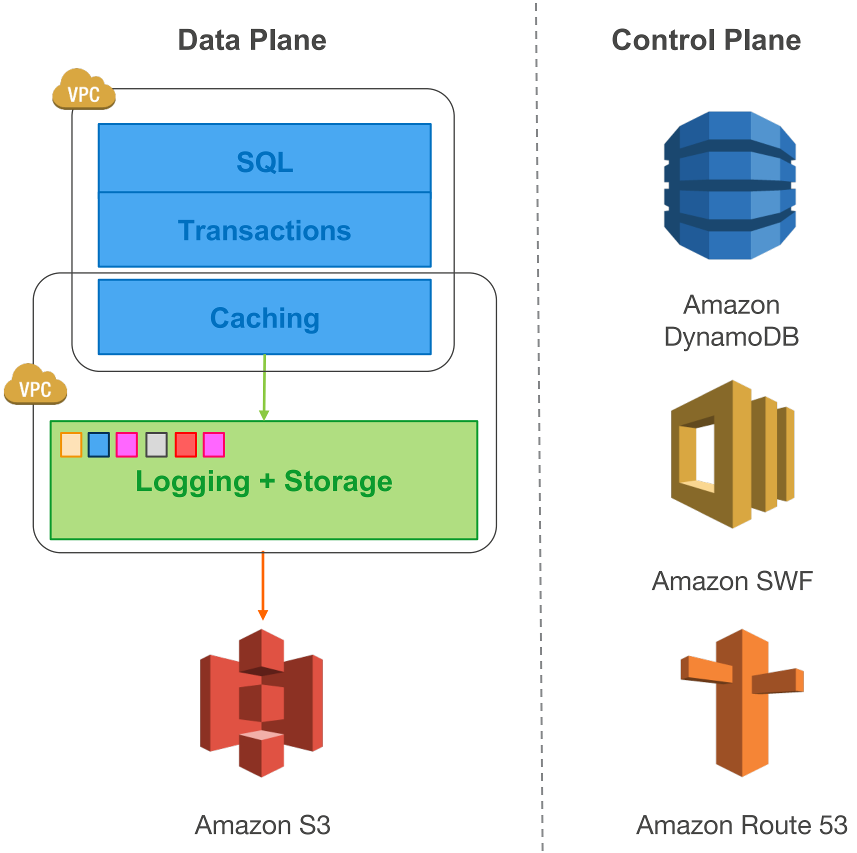
How Did We Achieve This?

I/O

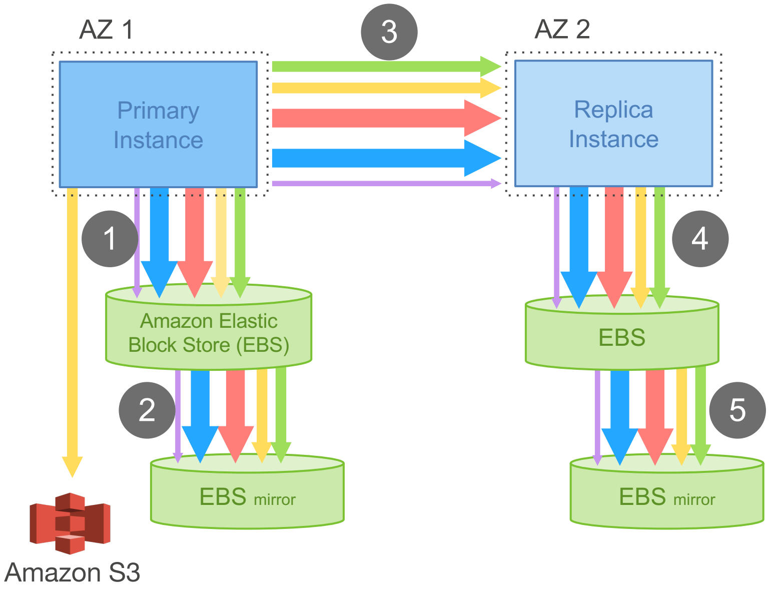
Packets/second

Context switching

A Service-Oriented Architecture Applied to Databases



I/O Traffic in MySQL



PERFORMANCE

780K transactions

7,388K I/Os per million txns (excludes mirroring, standby)

Average 7.4 I/Os per transaction

30 minute SysBench writeonly workload, 100GB dataset, **RDS MultiAZ**, 30K PIOPS

TYPE OF WRITE

→ LOG

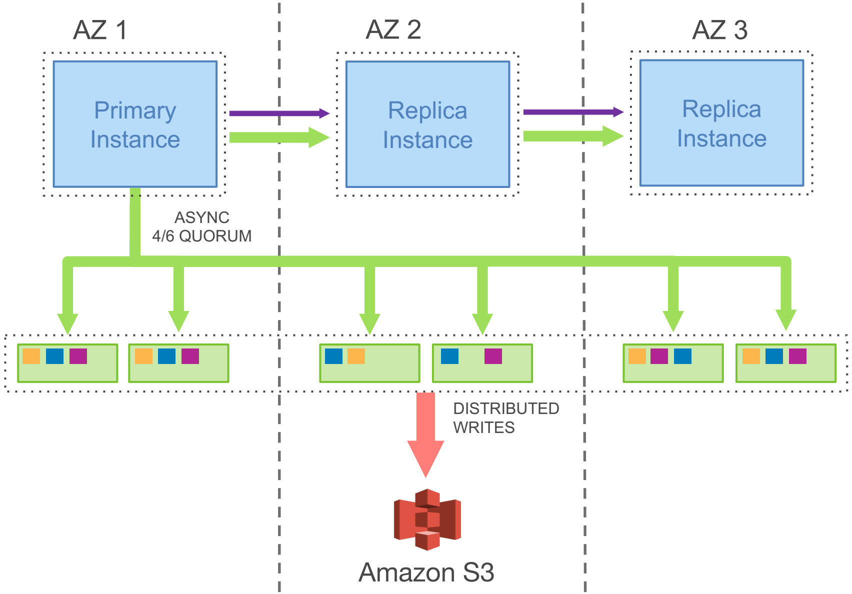
→ BINLOG

→ DATA

→ DOUBLE-WRITE

→ FRM FILES

I/O Traffic in Aurora



PERFORMANCE

27,378K transactions	35X MORE
950K I/Os per 1M txns (6X amplification)	7.7X LESS

TYPE OF WRITE

→ LOG

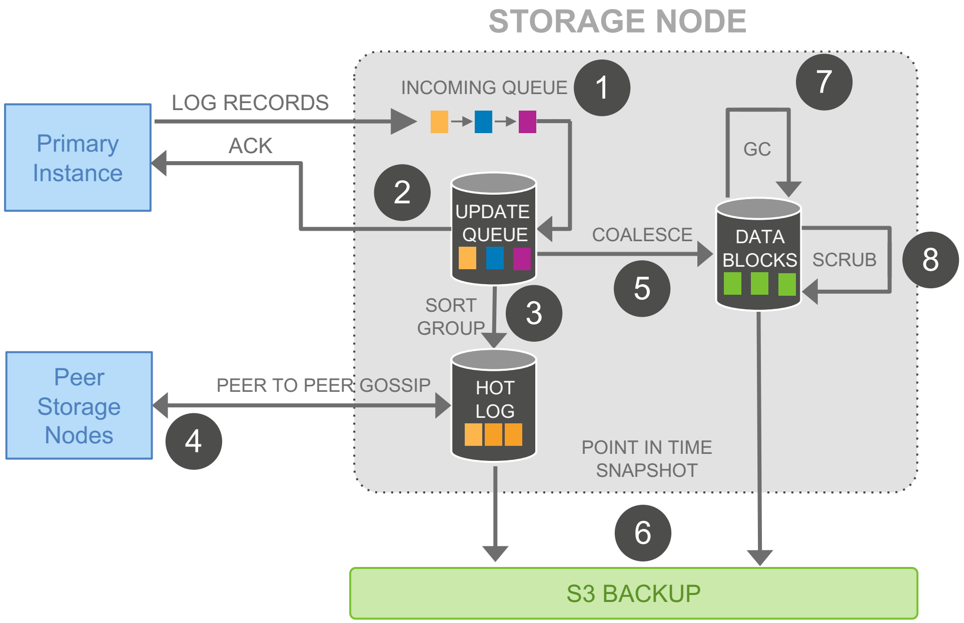
→ BINLOG

→ DATA

→ DOUBLE-WRITE

→ FRM FILES

I/O Traffic in Aurora (Storage Node)



OBSERVATIONS

All steps are asynchronous

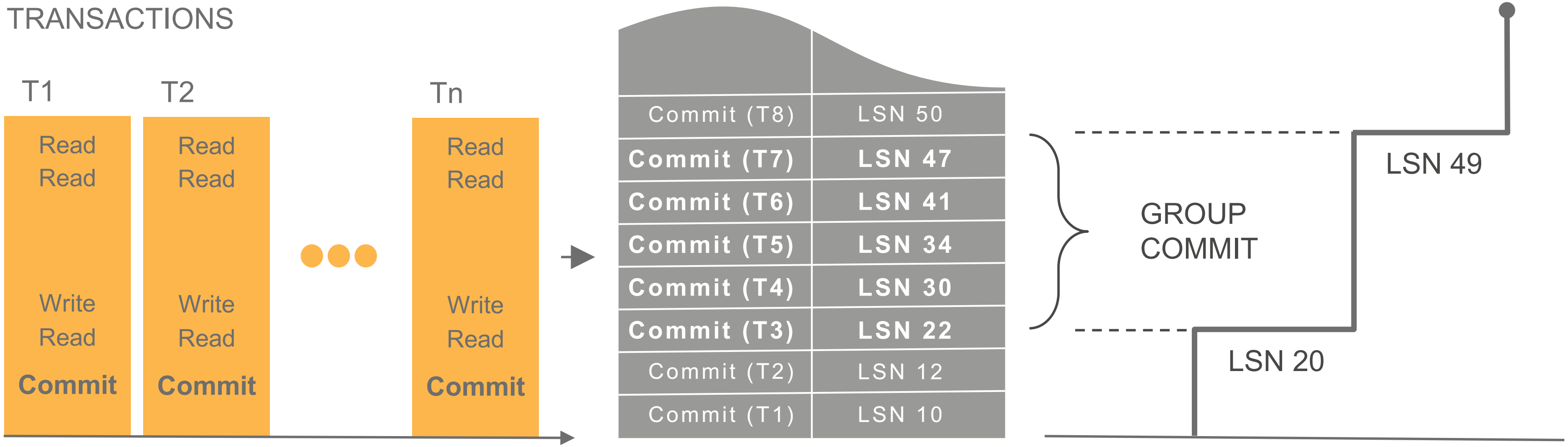
Only steps 1 and 2 are in foreground latency path

Input queue is **46X less** than MySQL (unamplified, per node)

Favor latency-sensitive operations

Use disk space to buffer against spikes in activity

Asynchronous Group Commits



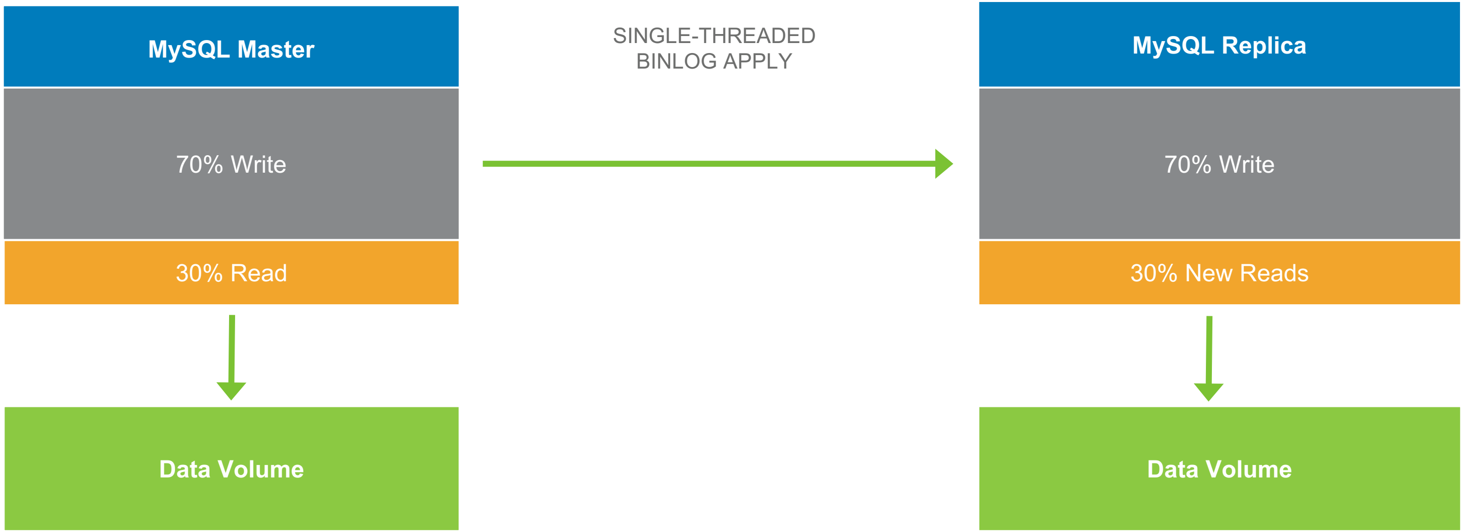
Commit (T8)	LSN 50
Commit (T7)	**LSN 47**
Commit (T6)	**LSN 41**
Commit (T5)	**LSN 34**
Commit (T4)	**LSN 30**
Commit (T3)	**LSN 22**
Commit (T2)	LSN 12
Commit (T1)	LSN 10

TIME

COMMIT QUEUE
Pending commits in LSN order

LSN GROWTH
Durable LSN at head-node

MySQL Read Scaling



MySQL Master

70% Write

30% Read

Data Volume

SINGLE-THREADED
BINLOG APPLY

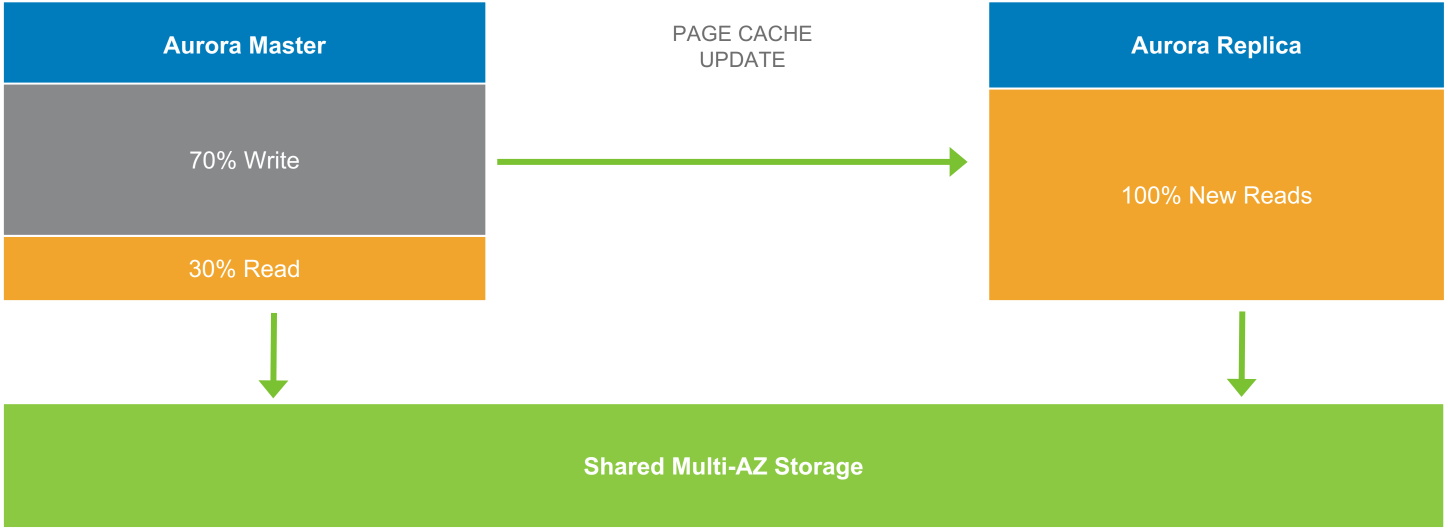
MySQL Replica

70% Write

30% New Reads

Data Volume

Aurora Read Scaling



Aurora Master

70% Write

30% Read

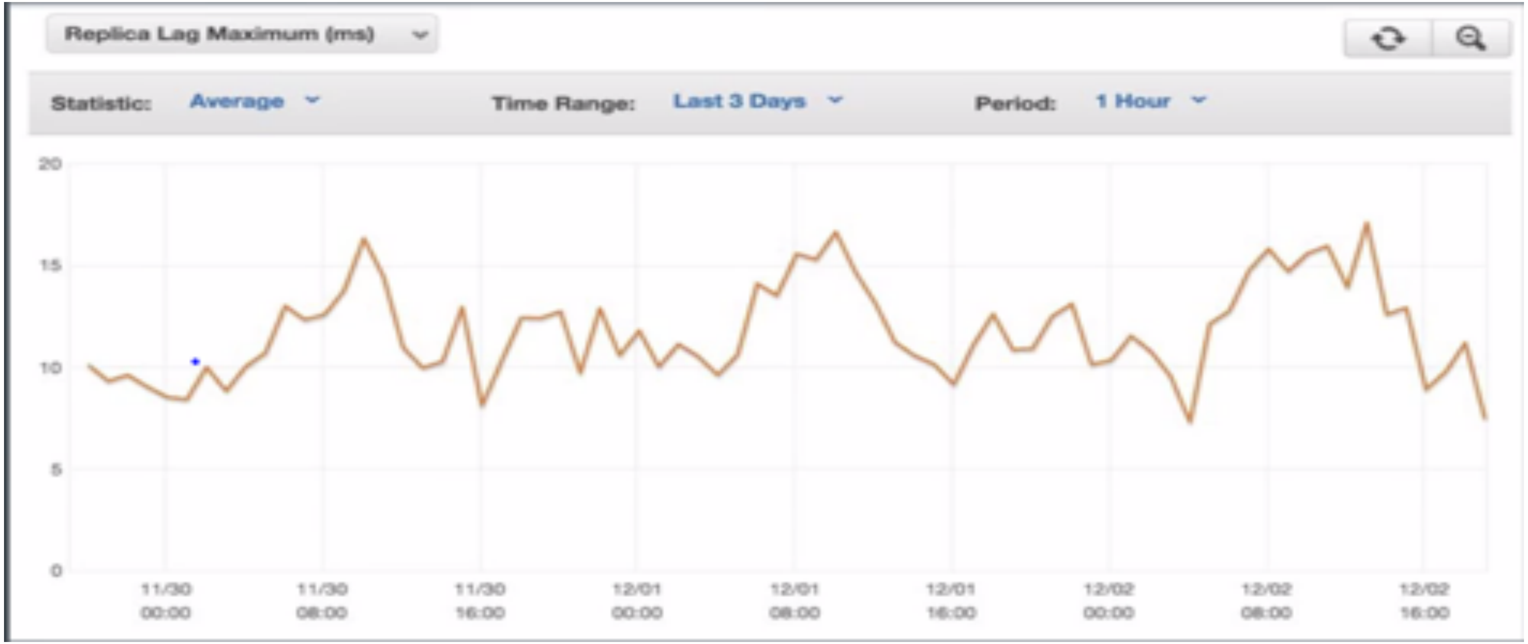
PAGE CACHE
UPDATE

Aurora Replica

100% New Reads

Shared Multi-AZ Storage

Real-Life Data - Read Replica Latency



Replica Lag Maximum (ms)

Statistic: Average

Time Range: Last 3 Days

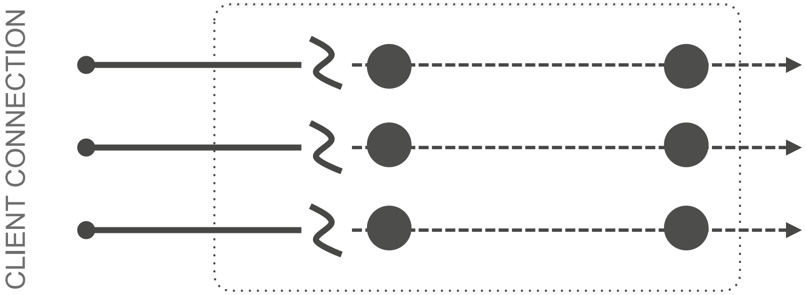
Period: 1 Hour

GoGuardian

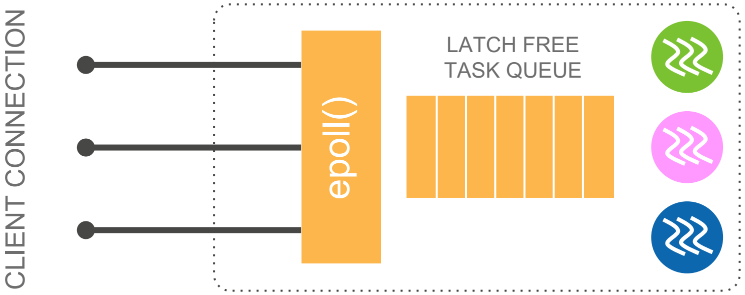
“In MySQL, we saw replica lag spike to almost 12 minutes which is almost absurd from an application’s perspective. With Aurora, the maximum read replica lag across 4 replicas never exceeded 20 ms.”

Adaptive Thread Pool

MYSQL THREAD MODEL



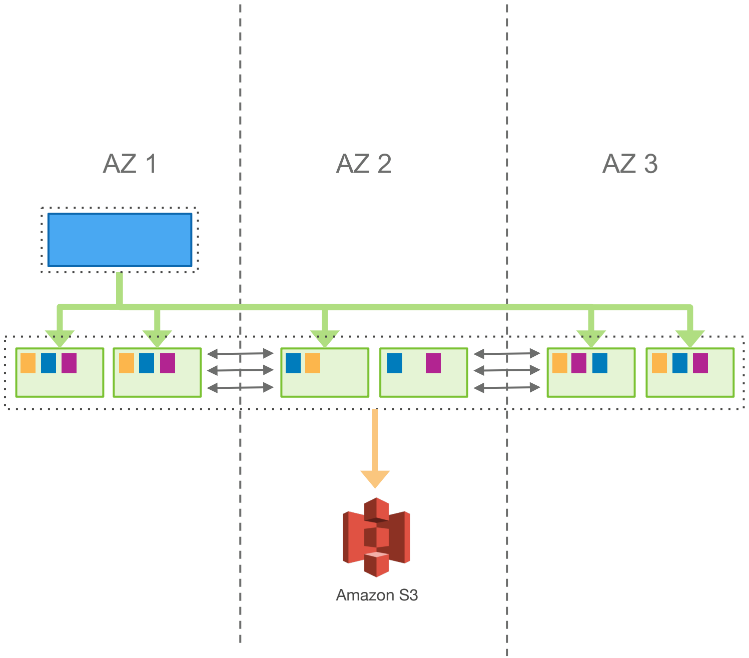
AURORA THREAD MODEL



Availability

“Performance only matters if your database is up”

Storage Durability



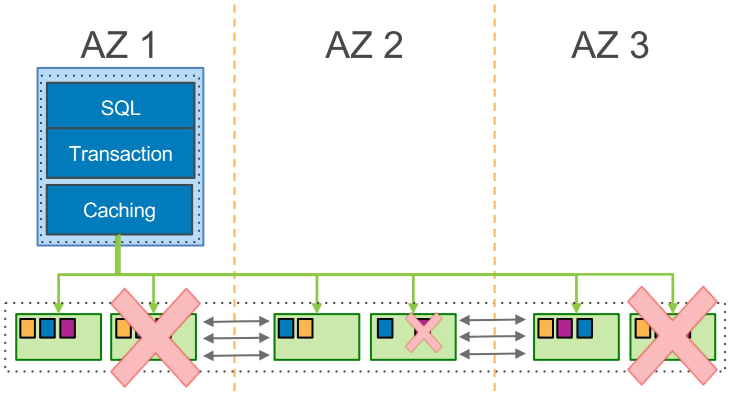
Amazon Aurora Storage Engine Fault-Tolerance

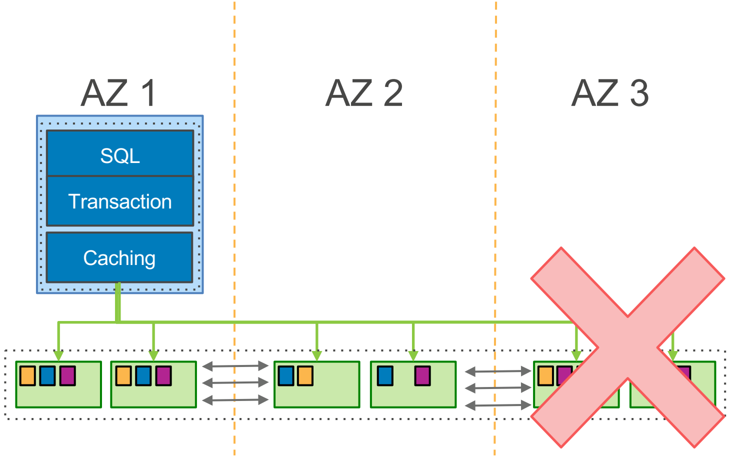
What can fail?

- Segment failures (disks)
- Node failures (machines)
- AZ failures (network or datacenter)

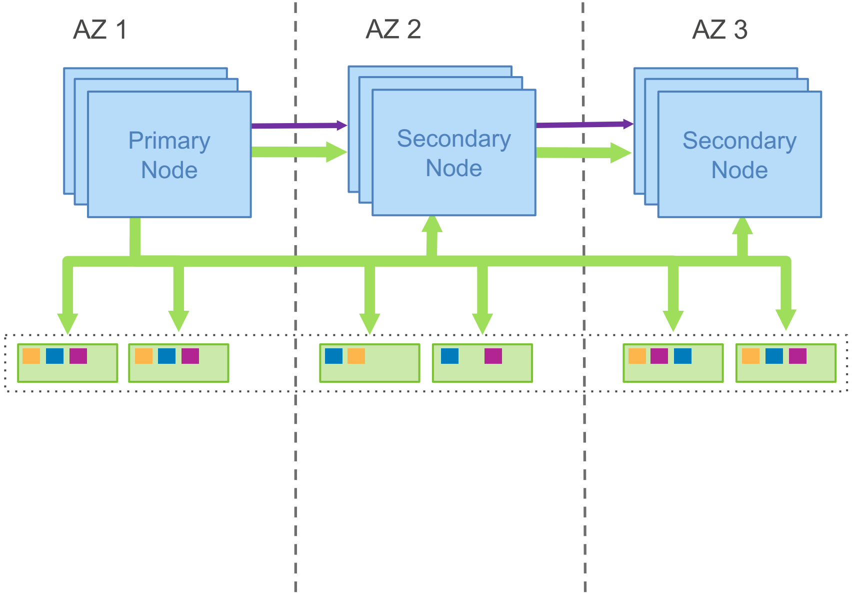
Optimizations

- 4 out of 6 write quorum
- 3 out of 6 read quorum
- Peer-to-peer replication for repairs

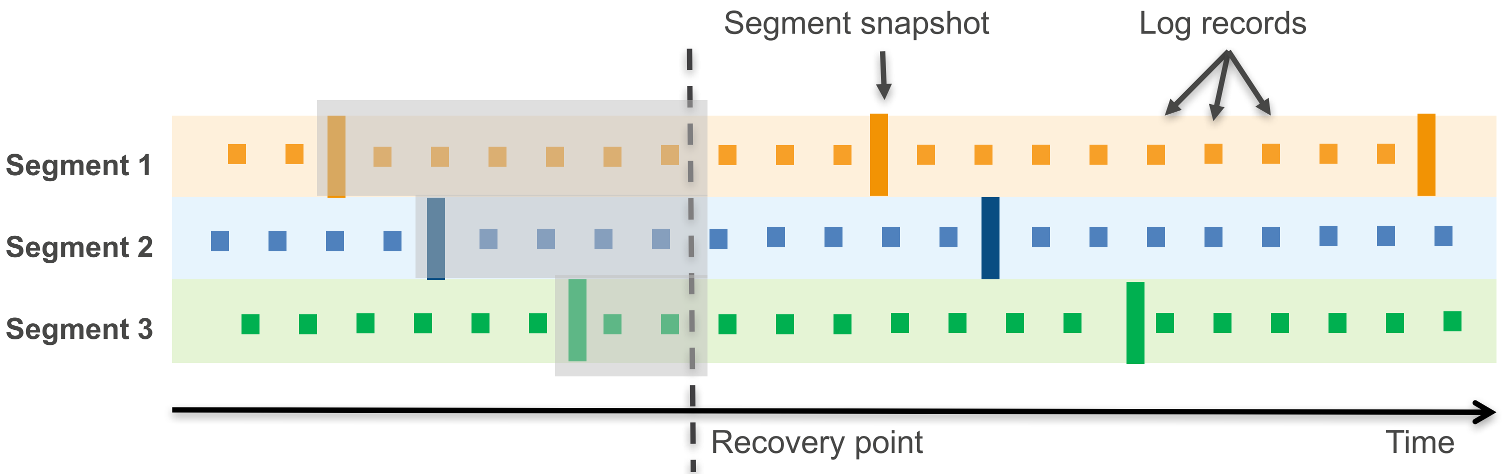




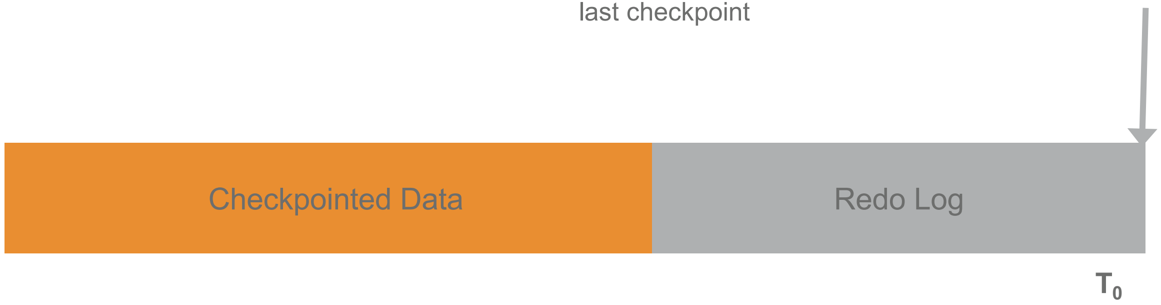
Aurora Replicas



Continuous Backup



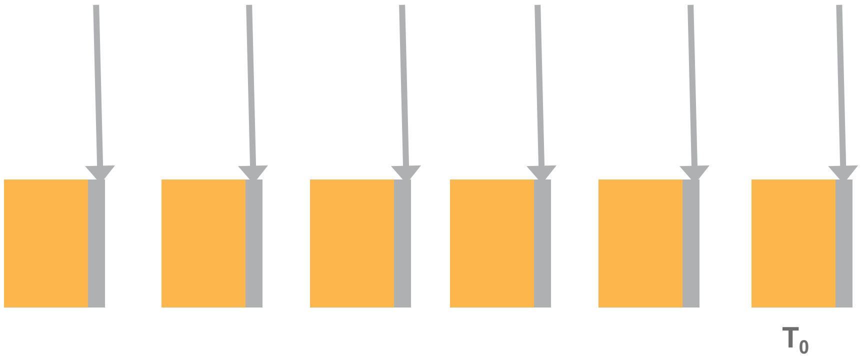
Traditional Crash Recovery



Crash at T_0 requires a re-application of the SQL in the redo log since last checkpoint

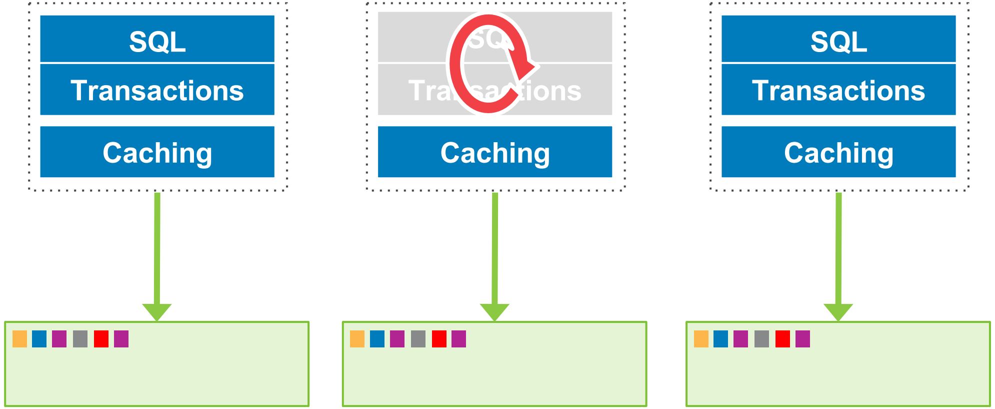
Amazon Aurora – Instant Crash Recovery

Crash at T_0 will result in redo logs being applied to each segment on demand, in parallel, asynchronously

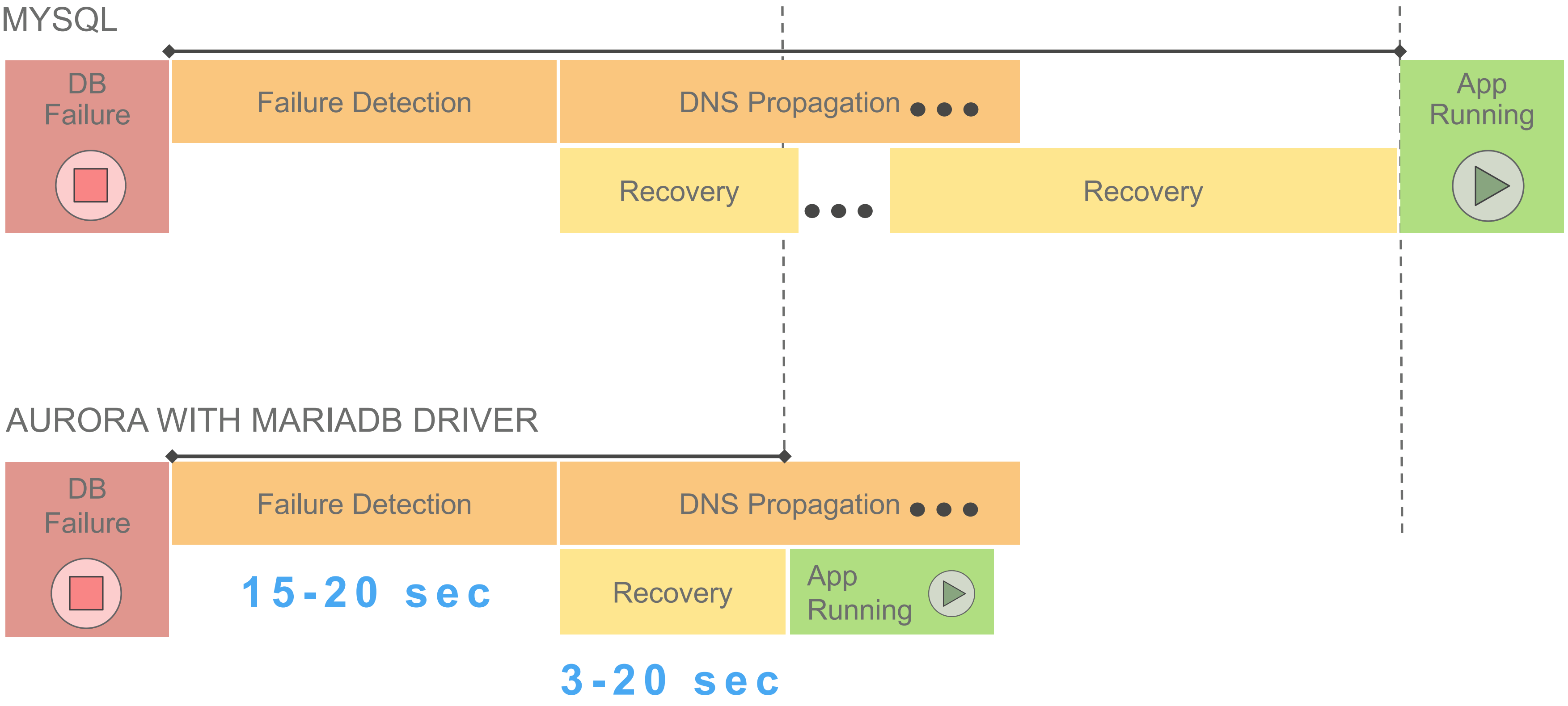


Survivable Caches

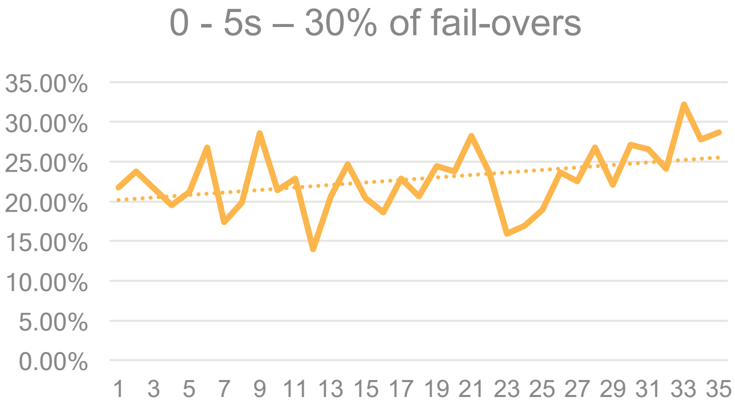
Caching process is outside the DB process
and remains warm across a database restart



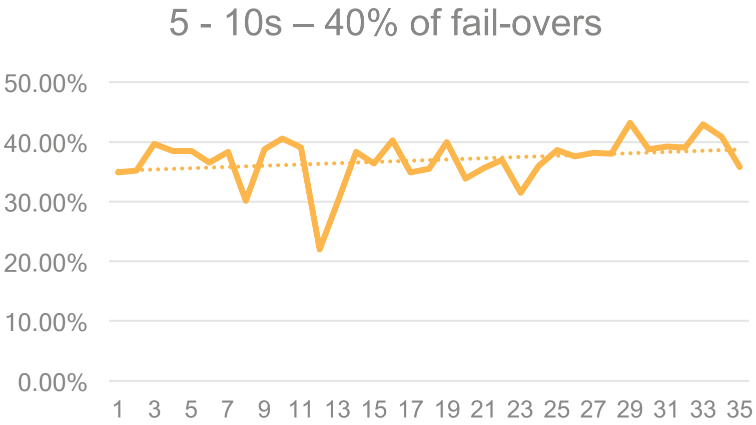
Faster Failover



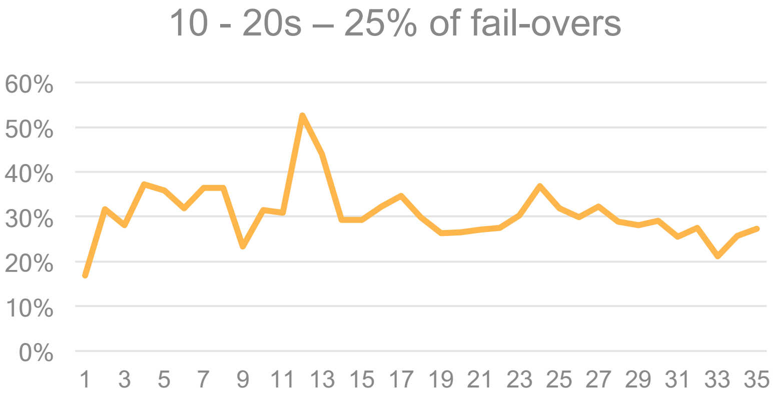
Database Failover Time



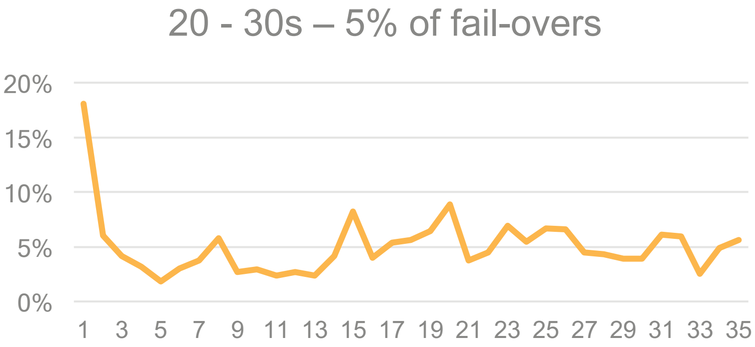
0 - 5s – 30% of fail-overs



5 - 10s – 40% of fail-overs



10 - 20s – 25% of fail-overs



20 - 30s – 5% of fail-overs

AWS SUMMIT

Thank You

amazon web services