AWS re:INVENT

DAT202: Getting started with Amazon Aurora

Debanjan Saha, General Manager, Amazon Web Services

Gurmit Singh Ghatore, Principal Database Engineer, Expedia

Brandon O'Brien, Principal Software Engineer, Expedia

AWS re:Invent

What is Amazon Aurora ……

Database reimagined for the cloud

- ☑ **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

Delivered as a **managed** service

Re-imagining relational database

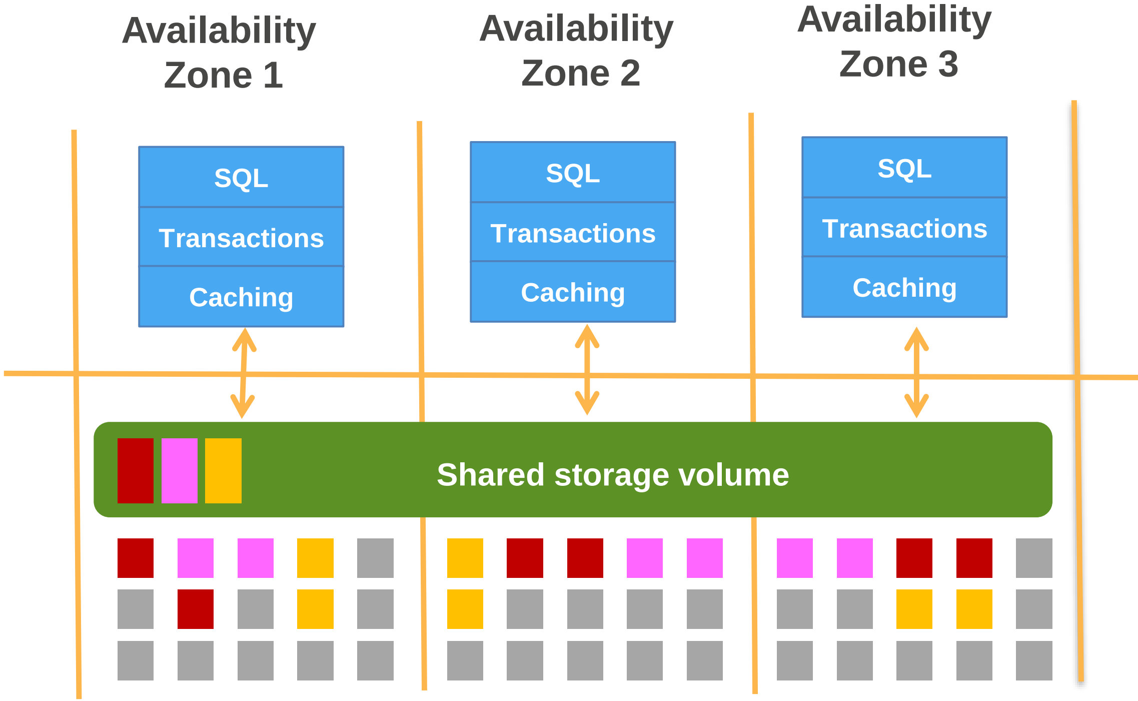
1. Scale-out and distributed design

2. Service-oriented architecture leveraging AWS services

3. Automate administrative tasks – fully managed service

Scale-out, distributed architecture

- Purpose-built log-structured distributed storage system designed for databases
- Storage volume is striped across hundreds of storage nodes distributed over 3 different availability zones
- Six copies of data, two copies in each availability zone to protect against AZ+1 failures
- Plan to apply same principles to other layers of the stack



Storage nodes with SSDs

Leveraging cloud ecosystem

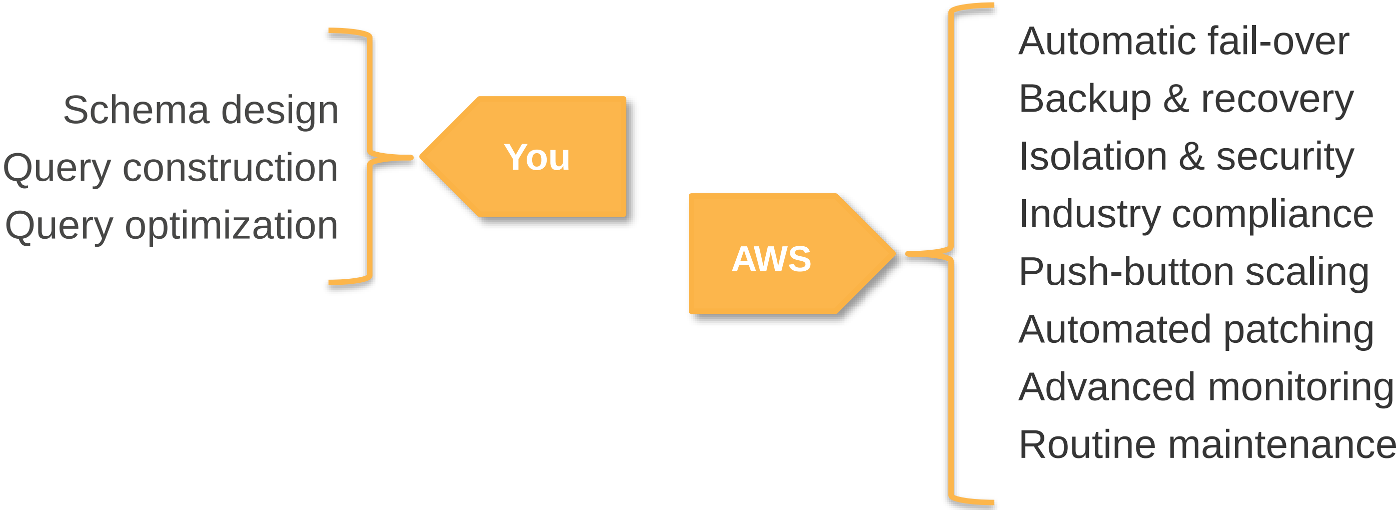
Lambda Invoke Lambda events from stored procedures/triggers.

S3 Load data from S3, store snapshots and backups in S3.

IAM Use IAM roles to manage database access control.

CloudWatch Upload systems metrics and audit logs to CloudWatch.

Automate administrative tasks



Schema design
Query construction
Query optimization

You

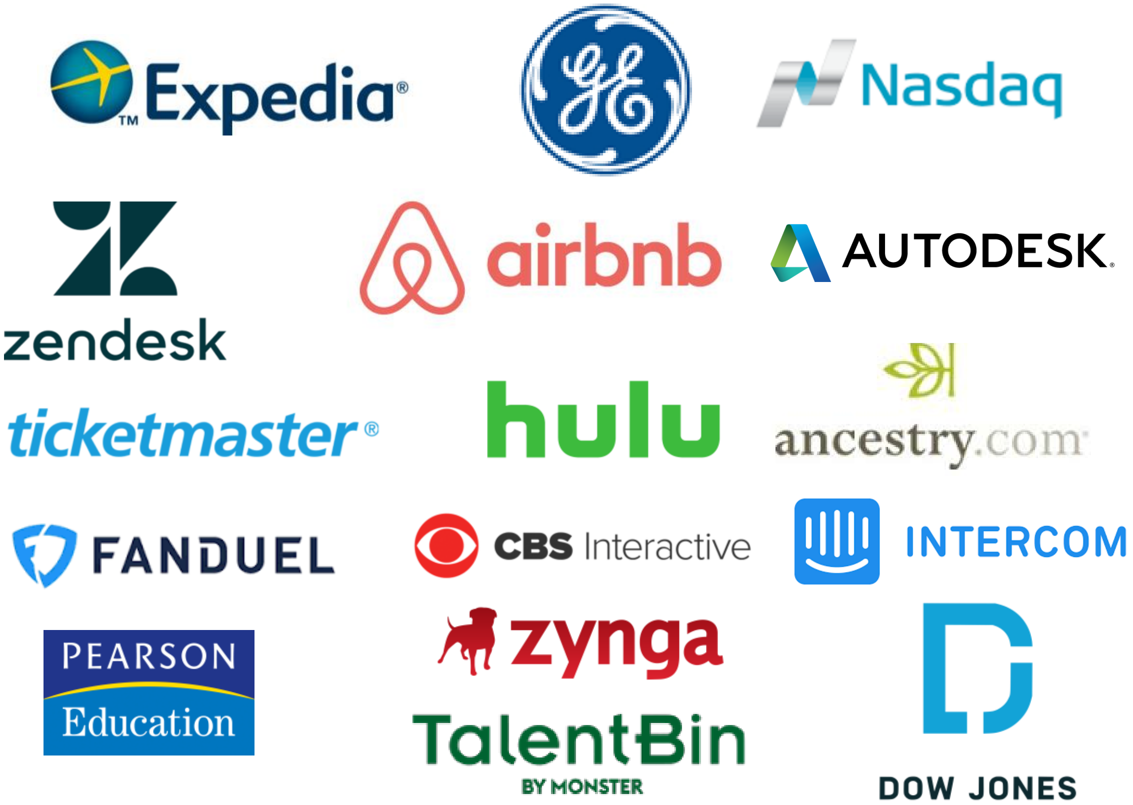
AWS

Automatic fail-over
Backup & recovery
Isolation & security
Industry compliance
Push-button scaling
Automated patching
Advanced monitoring
Routine maintenance

Takes care of your time-consuming database management tasks, freeing you to focus on your applications and business

Aurora customer adoption

Aurora is used by ¾ of the top 100 AWS customers



Fastest growing service in AWS history

Who are moving to Aurora and why?

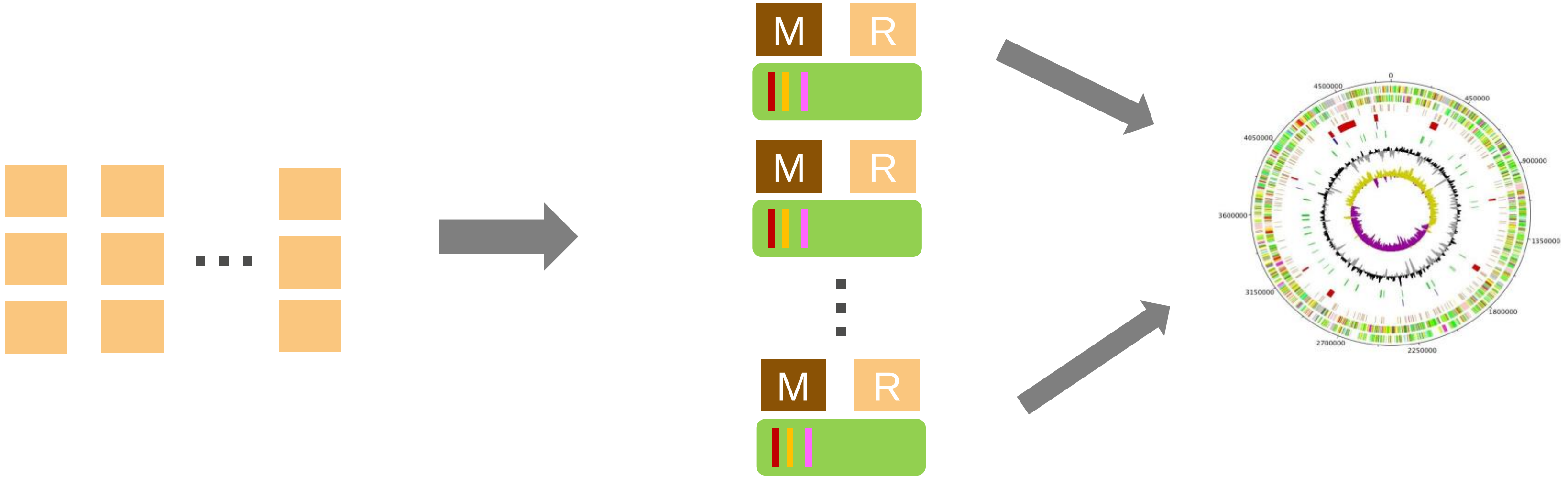
Customers using MySQL engines

- Higher performance – up to 5x
- Better availability and durability
- Reduces cost – up to 60%
- Easy migration; no application change

Customers using commercial engines

- One tenth of the cost; no licenses
- Integration with cloud ecosystem
- Comparable performance and availability
- Migration tooling and services

Data store for high-performance applications



Cassandra (>100 nodes)

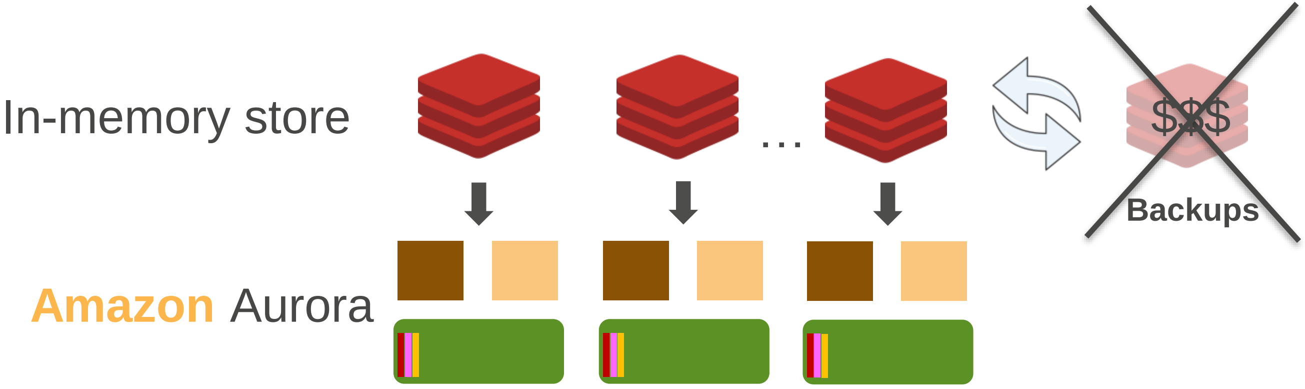
Aurora (~10 clusters)

DNA analysis and matching
(millions of reads and writes)

Large genealogy company achieved <10ms read latency and an order of magnitude reduction in projected costs by migrating to Aurora

- Data sharded across ~10 Aurora R3.XLarge clusters - sufficient room for vertical scaling
- OLAP + OLTP: DNA matching algorithms require millions of reads and batch updates

Highly-available persistent data store



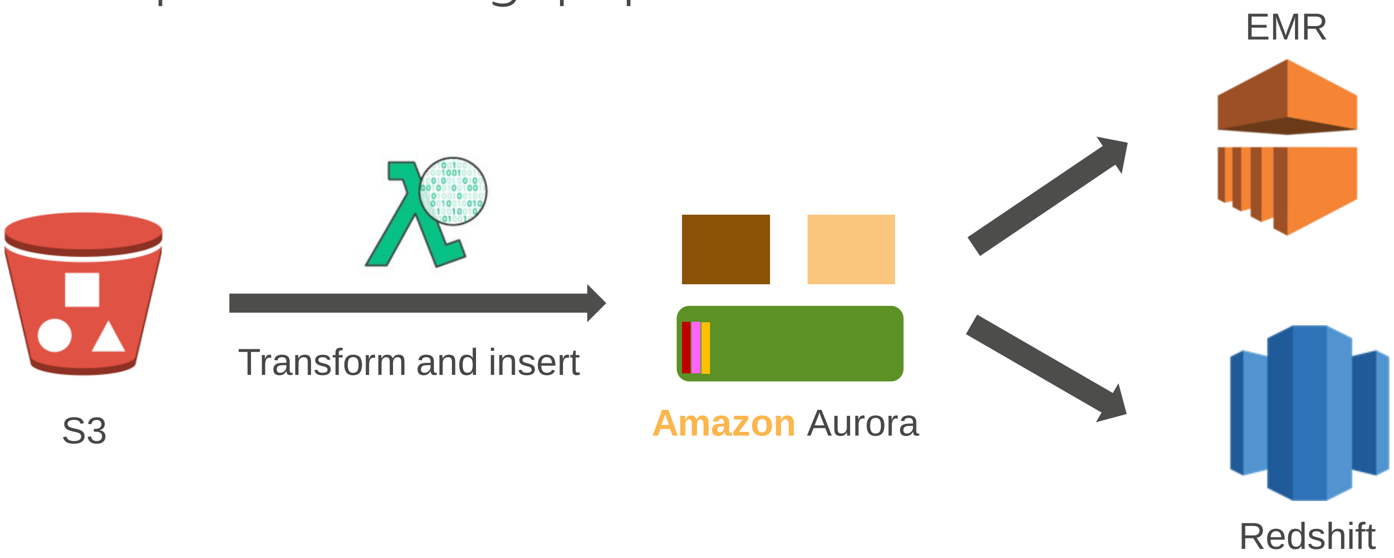
An on demand video streaming service stores metadata for millions of videos in in-memory data structures.

- Regular backups required; high and ever-increasing cost due to years worth of video metadata.
- Aurora used as the persistent data-store with Redis front-end cache to provide HA and optimal performance at lower costs, and eliminating the need for costly backups.

AWS re:Invent

aws

Data processing pipeline



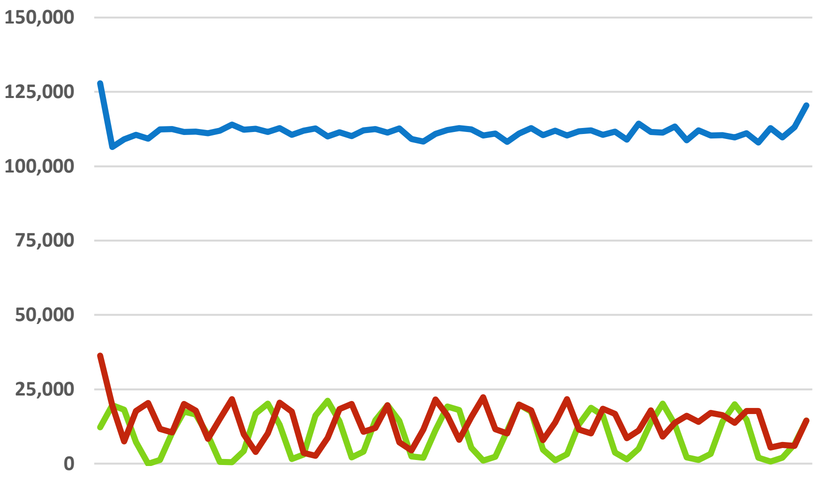
- Transaction streams from various sources comes to different S3 buckets.
- S3 event invoke a Lambda function, which reads the file and upload to Aurora
- Aurora is used for data processing, validation and operational reporting
- Processed data is loaded into Redshift and EMR for deep-analytics

Amazon Aurora is fast …

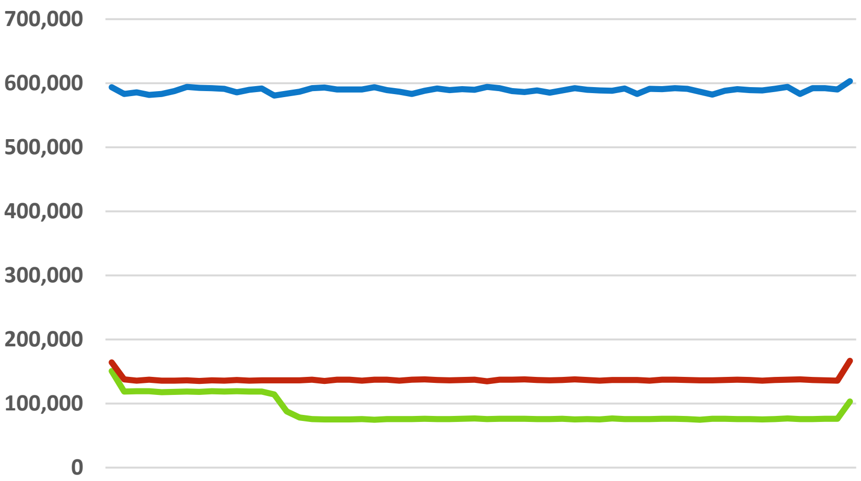
5x faster than MySQL

5X faster than RDS MySQL 5.6 & 5.7

WRITE PERFORMANCE



READ PERFORMANCE



MySQL SysBench results

R3.8XL: 32 cores / 244 GB RAM

Aurora 🟦 MySQL 5.6 🟩 MySQL 5.7 🟥

Five times higher throughput than stock MySQL based on industry standard benchmarks.

Aurora Scaling

With user connection

Connections	Amazon Aurora	RDS MySQL w/ 30K IOPS
50	40,000	10,000
500	71,000	21,000
5,000	110,000	13,000

UP TO

8x

FASTER

With number of tables

Tables	Amazon Aurora	MySQL I2.8XL local SSD	RDS MySQL w/ 30K IOPS (single AZ)
10	60,000	18,000	25,000
100	66,000	19,000	23,000
1,000	64,000	7,000	8,000
10,000	54,000	4,000	5,000

UP TO

11x

FASTER

With database size - SYSBENCH

DB Size	Amazon Aurora	RDS MySQL w/ 30K IOPS
1GB	107,000	8,400
10GB	107,000	2,400
100GB	101,000	1,500
1TB	26,000	1,200

UP TO

21

FASTER

With database size - TPCC

DB Size	Amazon Aurora	RDS MySQL w/ 30K IOPS
80GB	12,582	585
800GB	9,406	69

UP TO

136x

FASTER

How did we achieve this?

DO LESS WORK

Do fewer I/Os

Minimize network packets

Cache prior results

Offload the database engine

BE MORE EFFICIENT

Process asynchronously

Reduce latency path

Use lock-free data structures

Batch operations together

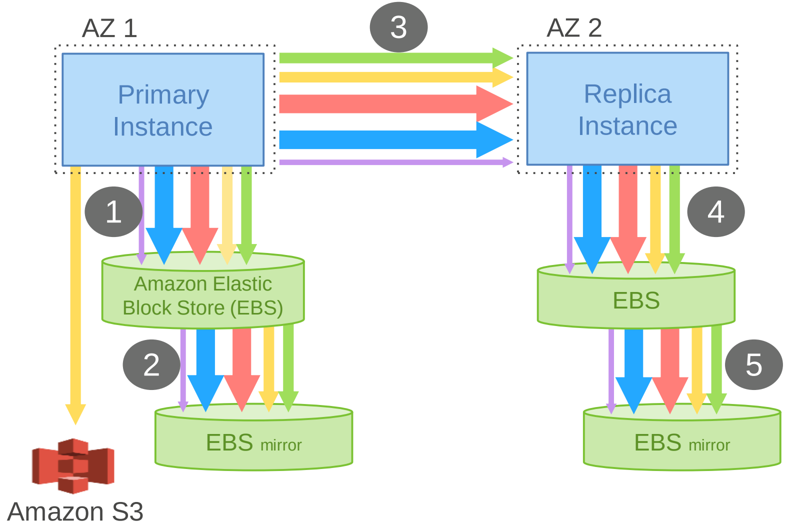
DATABASES ARE ALL ABOUT I/O

NETWORK-ATTACHED STORAGE IS ALL ABOUT PACKETS/SECOND

HIGH-THROUGHPUT PROCESSING IS ALL ABOUT CONTEXT SWITCHES

Aurora I/O profile

MYSQL WITH REPLICA



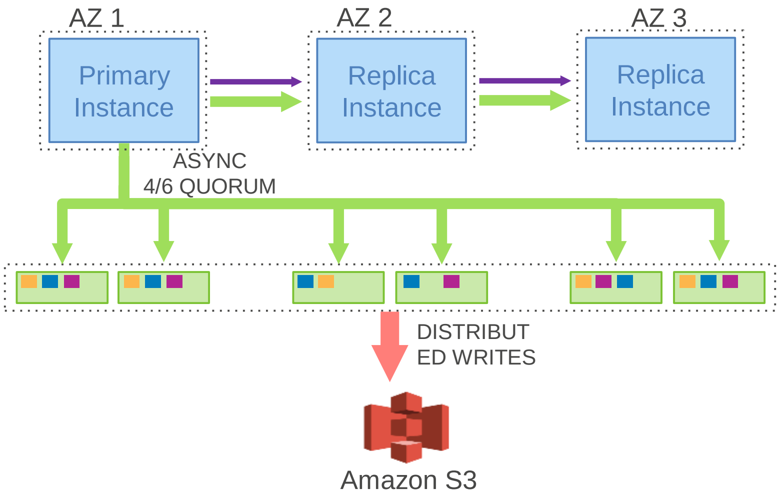
MySQL I/O profile for 30 min Sysbench run

780K transactions

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

Average 7.4 I/Os per transaction

AMAZON AURORA



Aurora IO profile for 30 min Sysbench run

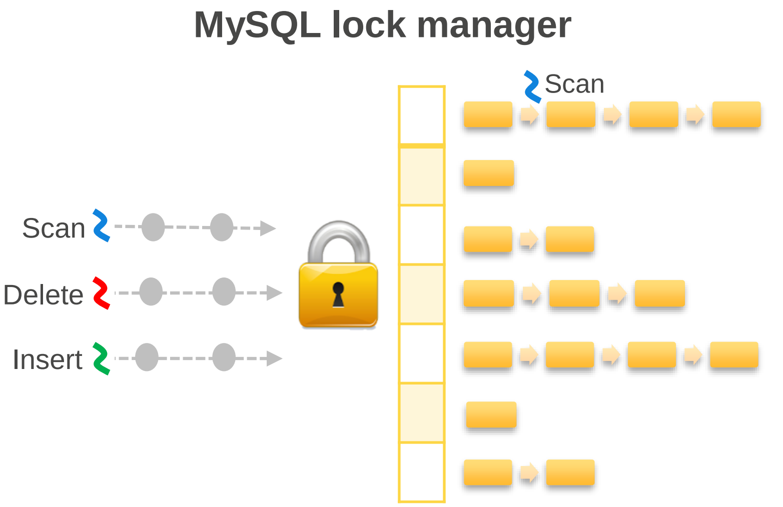
27,378K transactions **35X MORE**

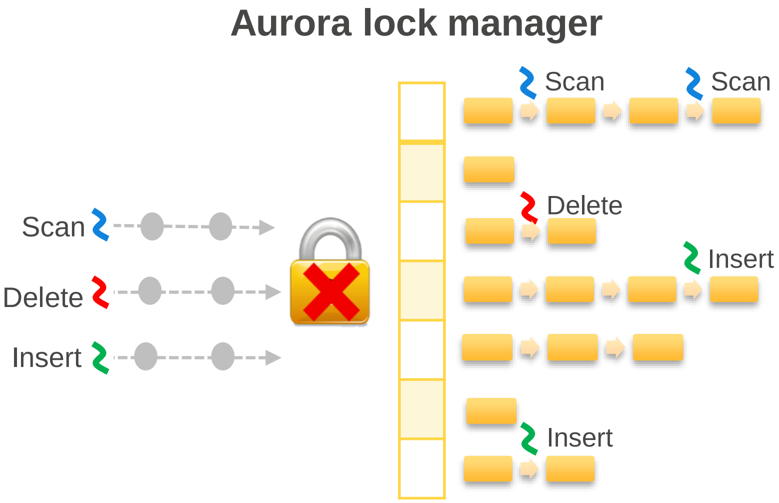
0.95 I/Os per transaction (6X amplification) **7.7X LESS**

TYPE OF WRITE

LOG   BINLOG   DATA   DOUBLE-WRITE   FRM FILES

Aurora lock management





- Same locking semantics as MySQL
- Concurrent access to lock chains

- Multiple scanners in individual lock chains
- Lock-free deadlock detection

Needed to support many concurrent sessions, high update throughput

New performance enhancements

Read performance

- Smart selector
- Logical read ahead
- Read views
- Hash joins *coming soon*
- Parallel query *coming soon*

Write performance

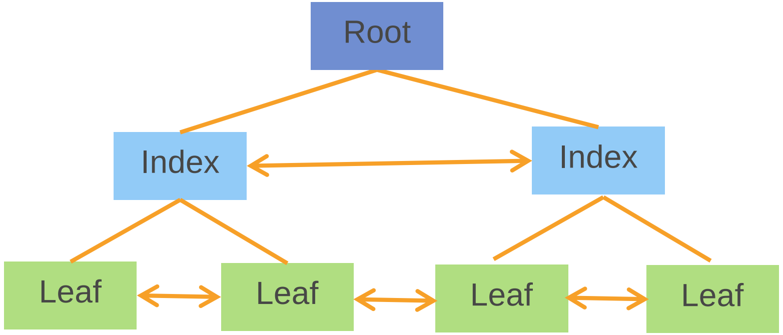
- NUMA aware scheduler
- Latch-free lock manager

Meta-data access

- Instant schema update
- B-Tree concurrency
- Catalog concurrency
- Faster index build

Online DDL: Aurora vs. MySQL

MySQL



- Full Table copy; rebuilds all indexes
- Needs temporary space for DML operations
- DDL operation impacts DML throughput
- Table lock applied to apply DML changes

Amazon Aurora

table name	**operation**	**column-name**	**time-stamp**
Table 1	add-col	column-abc	t1
Table 2	add-col	column-qpr	t2
Table 3	add-col	column-xyz	t3

- Use schema versioning to decode the block.
- Modify-on-write primitive to upgrade to latest schema
- Currently support add NULLable column at end of table
- Add column anywhere and with default coming soon.

Online DDL performance

On r3.large

	Aurora	MySQL 5.6	MySQL 5.7
10GB table	0.27 sec	3,960 sec	1,600 sec
50GB table	0.25 sec	23,400 sec	5,040 sec
100GB table	0.26 sec	53,460 sec	9,720 sec

On r3.8xlarge

	Aurora	MySQL 5.6	MySQL 5.7
10GB table	0.06 sec	900 sec	1,080 sec
50GB table	0.08 sec	4,680 sec	5,040 sec
100GB table	0.15 sec	14,400 sec	9,720 sec

What about availability

“Performance only matters if your database is up”

6-way replicated storage

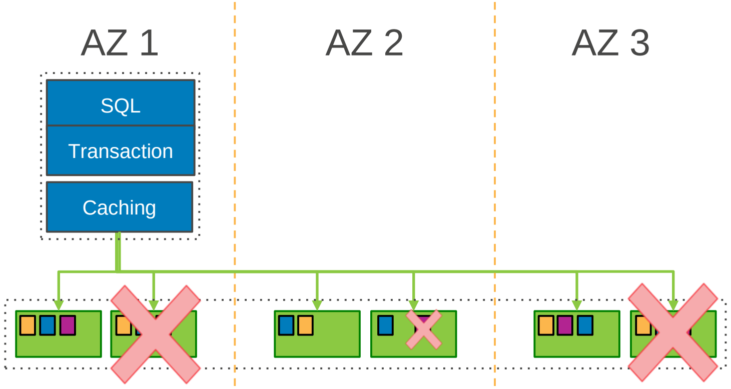
Survives catastrophic failures

Six copies across three availability zones

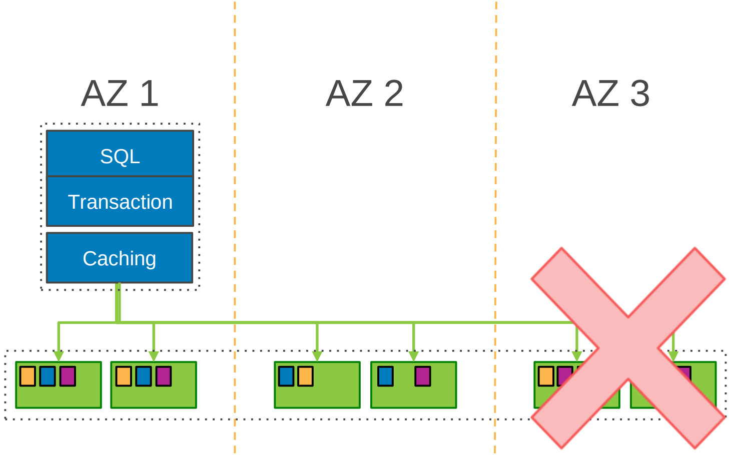
4 out 6 write quorum; 3 out of 6 read quorum

Peer-to-peer replication for repairs

Volume striped across hundreds of storage nodes

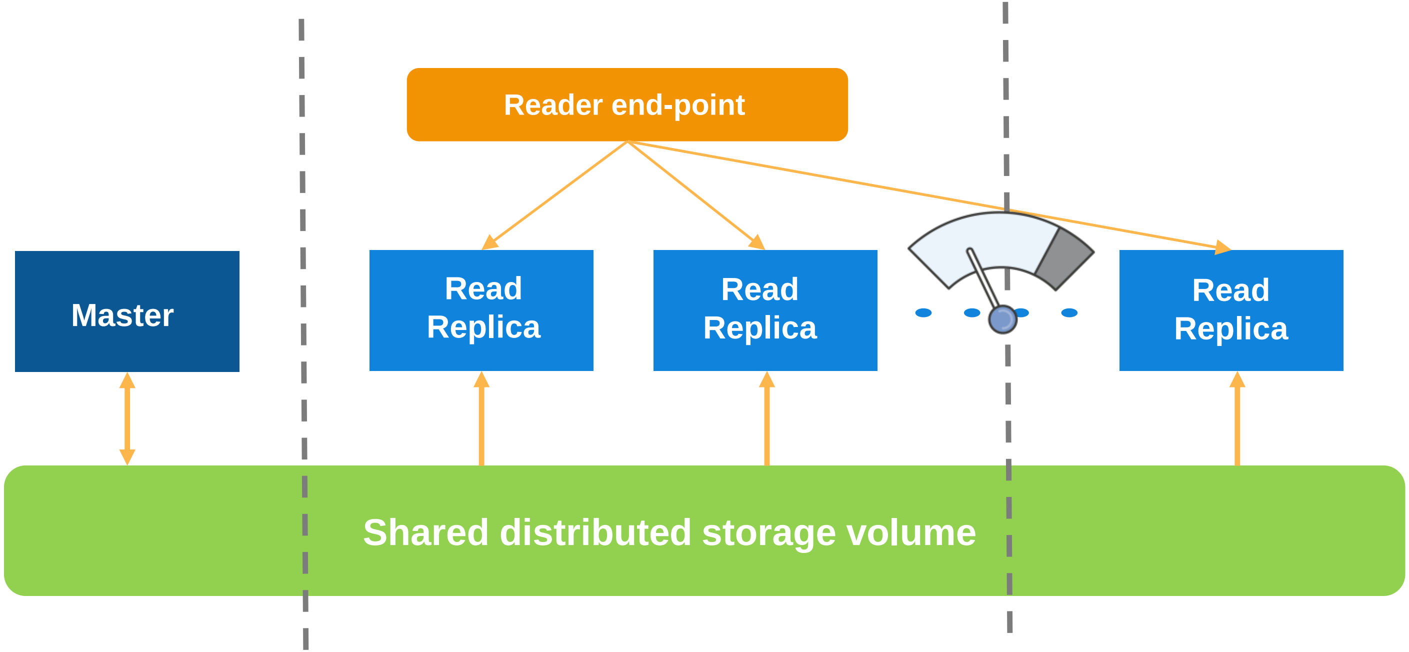


Read availability



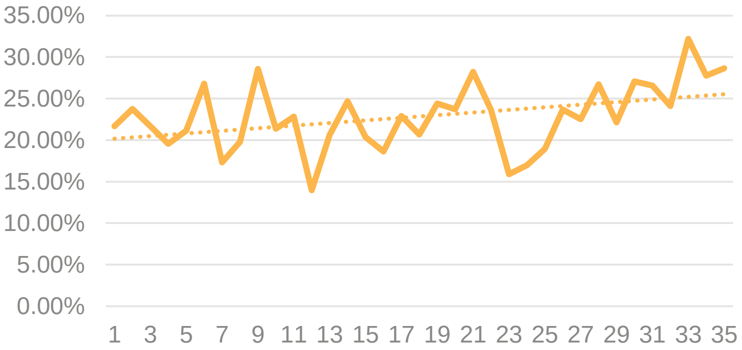
Read and write availability

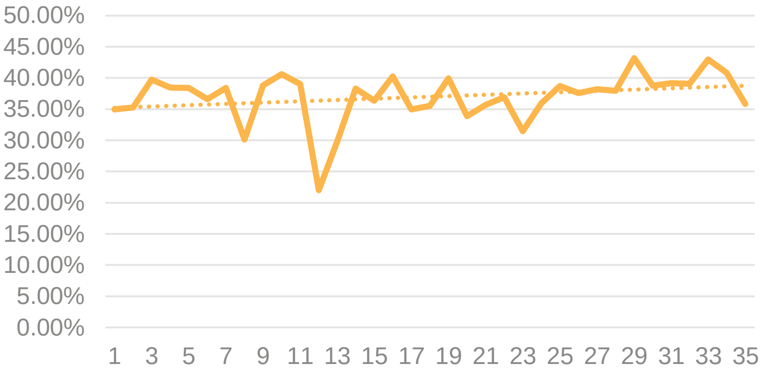
Up to 15 promotable read replicas

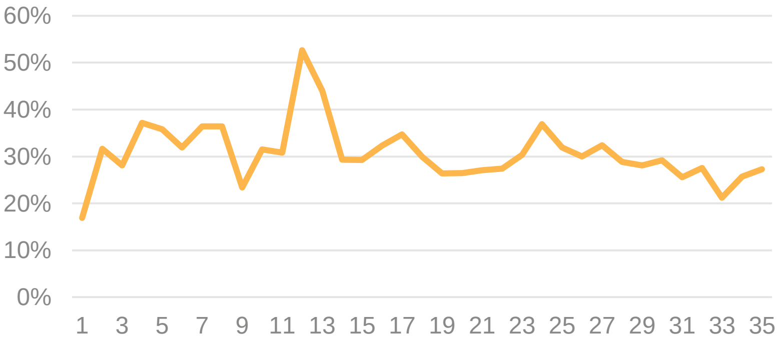


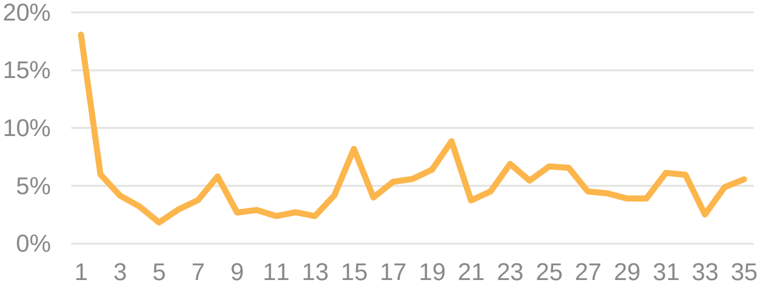
- Up to 15 promotable read replicas across multiple availability zones
- Re-do log based replication leads to low replica lag – typically < 10ms
- Reader end-point with load balancing and **auto-scaling** * NEW *

Database fail-over 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

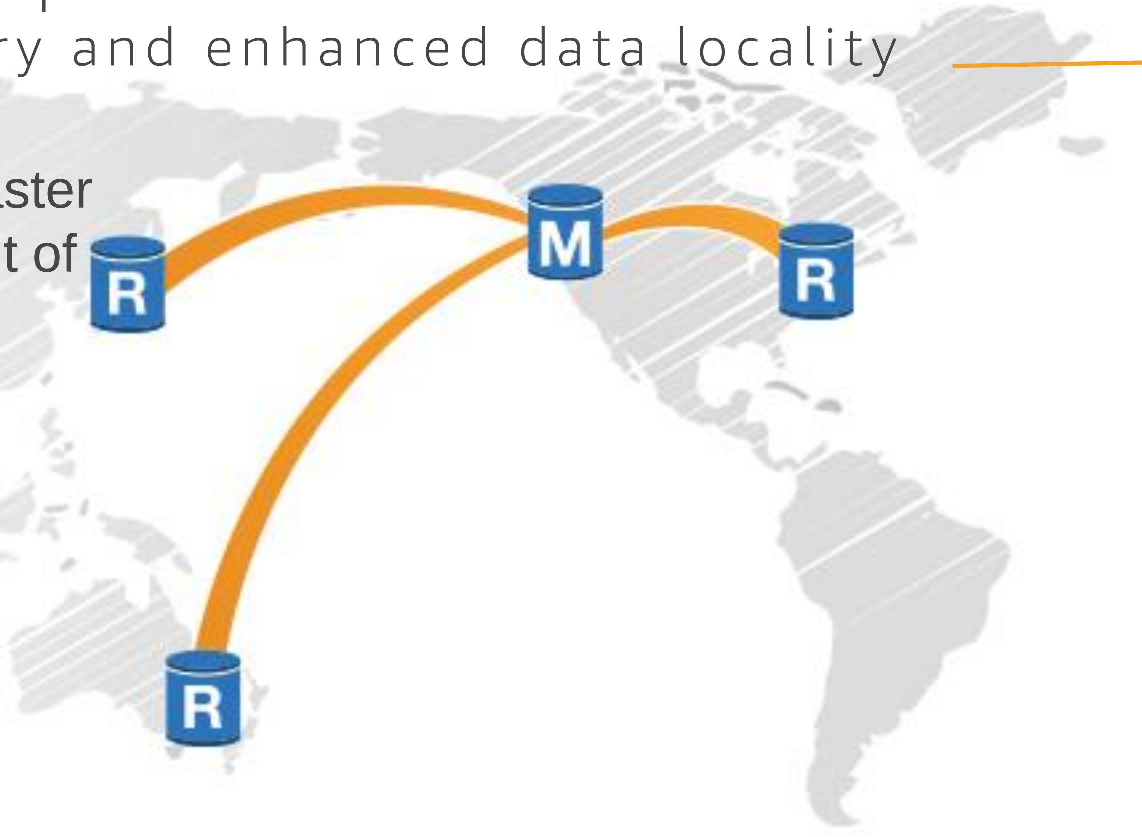
Cross-region read replicas

Faster disaster recovery and enhanced data locality

Promote read-replica to a master for faster recovery in the event of disaster

Bring data close to your customer's applications in different regions

Promote to a master for easy migration

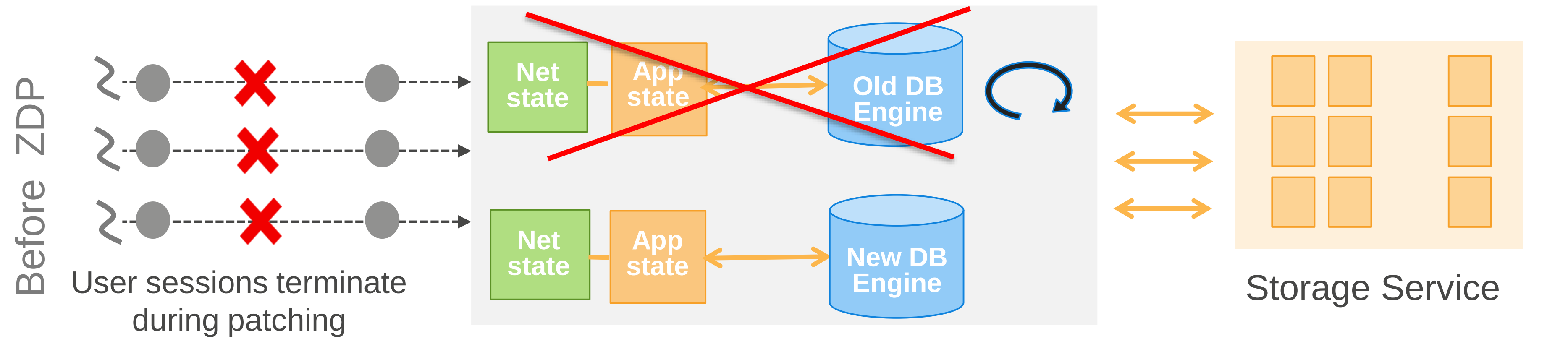


Availability is about more than HW failures

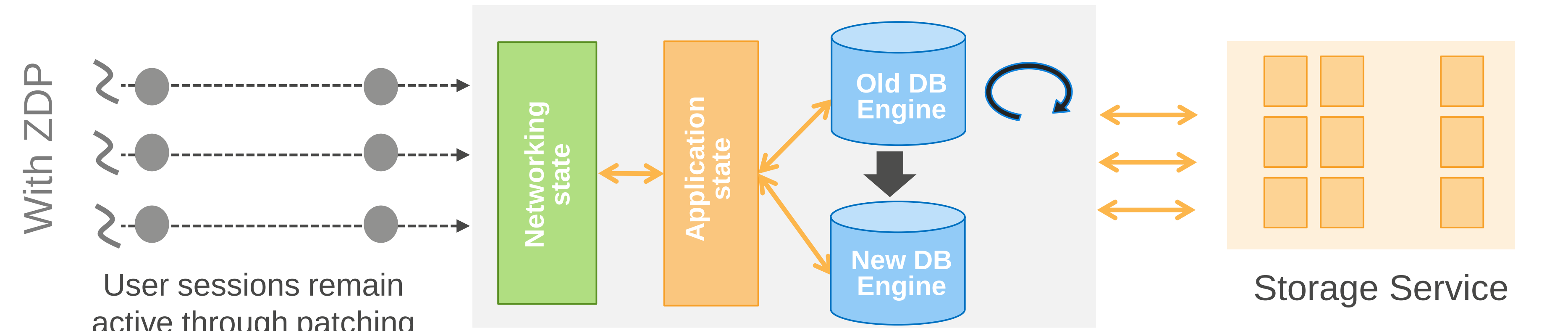
You also incur availability disruptions when you

1. Patch your database software – Zero Down Time Patch
2. Perform large scale database reorganizations – Fast Cloning
3. DBA errors requiring database restores – Online Point-in-time Restore

Zero downtime patching

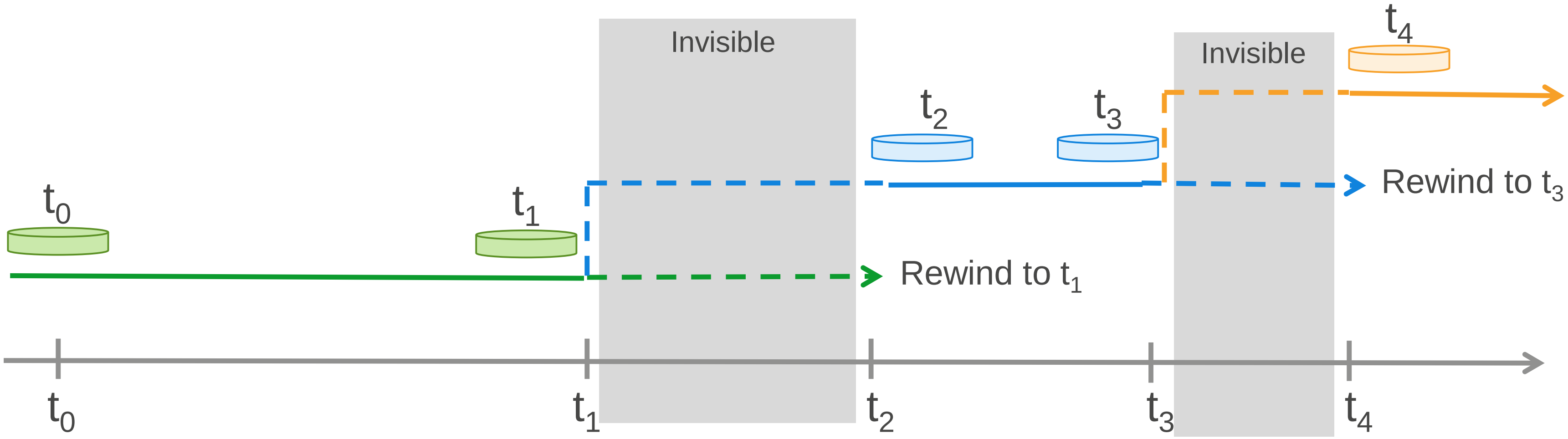


User sessions terminate during patching



User sessions remain active through patching

Database backtrack



Backtrack brings the database to a point in time without requiring restore from backups

- Backtracking from an unintentional DML or DDL operation
- Backtrack is not destructive. You can backtrack multiple times to find the right point in time

Security and Monitoring

Security and compliance

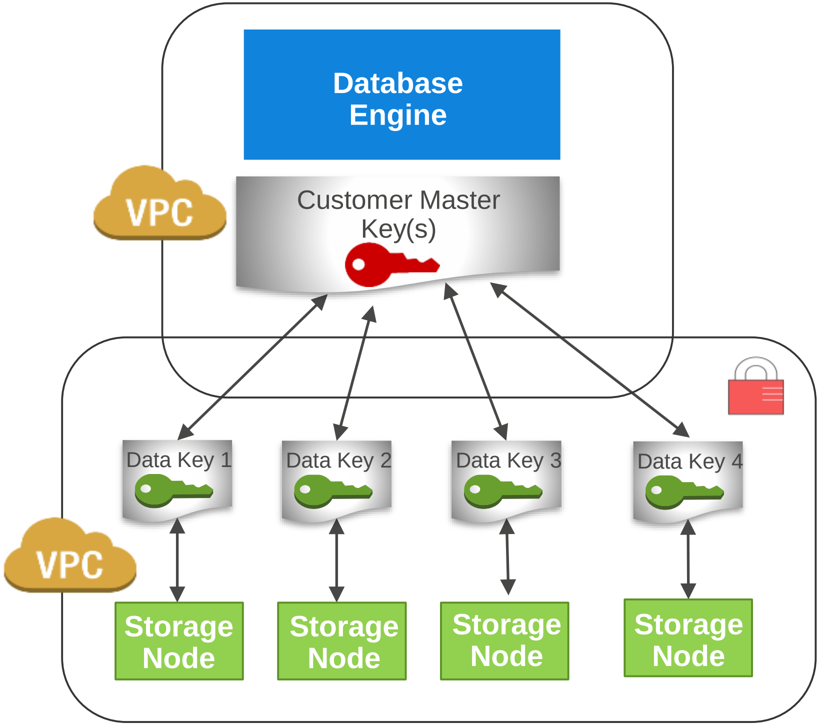
☑ Encryption to secure data at rest using customer managed keys

- AES-256; hardware accelerated
- All blocks on disk and in Amazon S3 are encrypted
- Key management via AWS KMS

☑ Encrypted cross-region replication, snapshot copy - SSL to secure data in transit

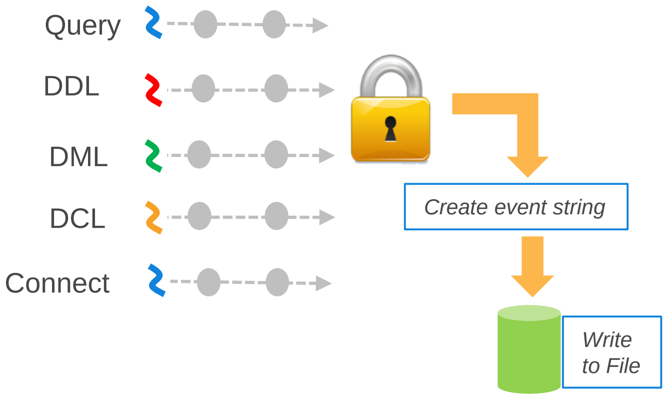
☑ Advanced auditing and logging without any performance impact

☑ Database activity monitoring *NEW*

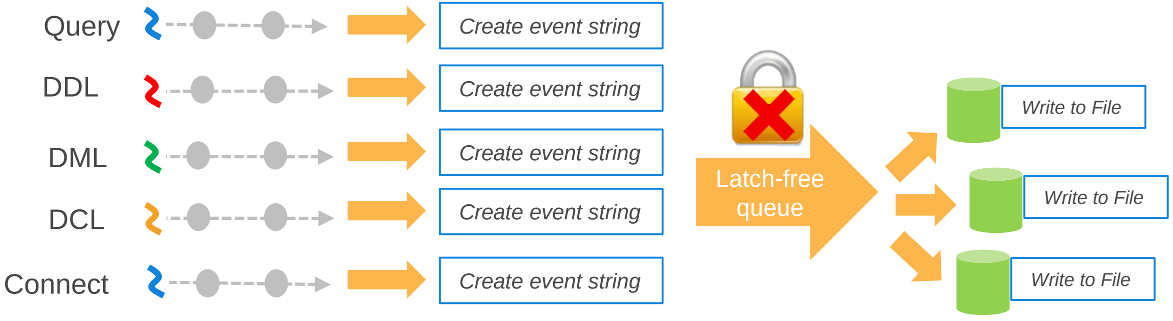


Aurora Auditing

MariaDB server_audit plugin



Aurora native audit support

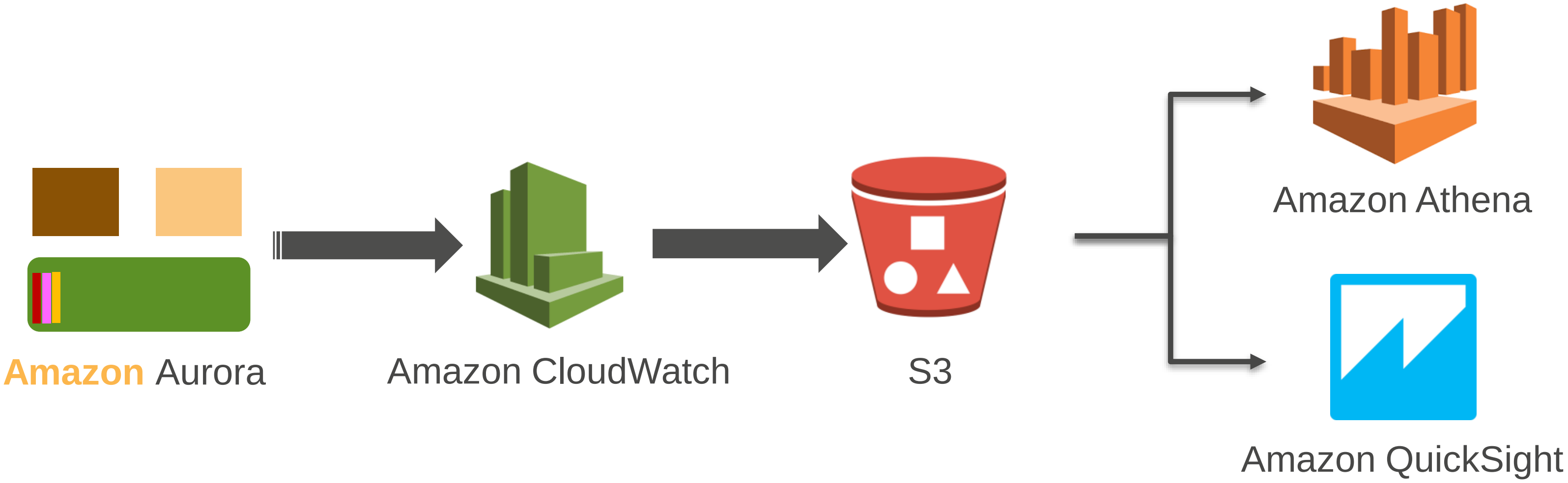


We can sustain over 500K events/sec

	MySQL 5.7	Aurora	
Audit Off	95K	615K	6.47x
Audit On	33K	525K	15.9x

Sysbench Select-only Workload on 8xlarge Instance

Database activity monitoring



- Continuously monitor activity in your DB clusters by sending these audit logs to CloudWatch Logs.
- Export to S3 for long term archival; analyze logs using Athena; visualize logs with QuickSight.

Search: Look for specific events across log files.

Metrics: Measure activity in your Aurora DB cluster.

Visualizations: Create activity dashboards

Alarms: Get notified or take actions

Industry certifications

- Amazon Aurora gives each database instance IP firewall protection
- Aurora offers transparent encryption at rest and SSL protection for data in transit
- Amazon VPC lets you isolate and control network configuration and connect securely to your IT infrastructure
- AWS Identity and Access Management provides resource-level permission controls

AICPA Service Organization Control Reports
AICPA
SERVICE ORGANIZATIONS
SOC
aicpa.org/soc
Formerly SAS 70 Reports

SOC3
SysTrust for Service Organizations

International Organization for Standardization
ISO
9001

Information Security Management System
ISO
27001
Certified

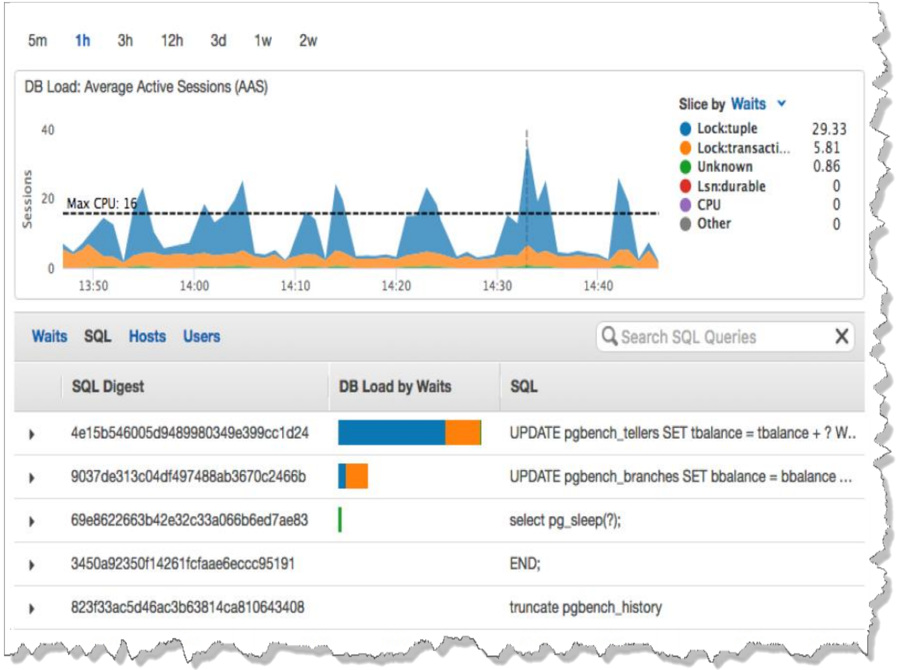
PAYMENT CARD INDUSTRY
PCI DSS
LEVEL 1 SERVICE PROVIDER
DATA SECURITY STANDARDS

New

New

New

Performance Insights



Slice by Waits	
● Lock:tuple	29.33
● Lock:transacti...	5.81
● Unknown	0.86
● Lsn:durable	0
● CPU	0
● Other	0

SQL Digest	DB Load by Waits	SQL
4e15b546005d9489980349e399cc1d24		UPDATE pgbench_tellers SET tbalance = tbalance + ? W..
9037de313c04df497488ab3670c2466b		UPDATE pgbench_branches SET bbalance = bbalance ...
69e8622663b42e32c33a066b6ed7ae83		select pg_sleep(?);
3450a92350f14261fcfaae6eccc95191		END;
823f33ac5d46ac3b63814ca810643408		truncate pgbench_history

Dashboard showing Load on Database

- Easy
- Powerful

Identifies source of bottlenecks

- Top SQL

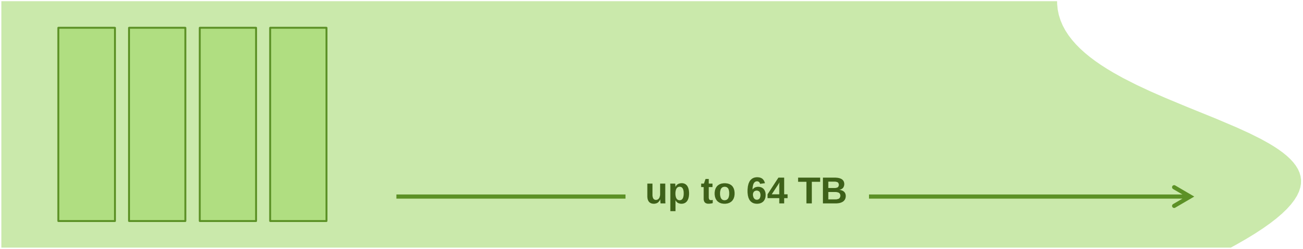
Adjustable time frame

- Hour, day, week , month
- Up to 35 days of data

Amazon Aurora is easy to use

Automated storage management, security and compliance, advanced monitoring, database migration.

Simplify storage management



Up to 64TB of storage – auto-incremented in 10GB units

- Continuous, incremental backups to Amazon S3
- Instantly create user snapshots—no performance impact
- Automatic storage scaling up to 64 TB—no performance impact
- Automatic restriping, mirror repair, hot spot management, encryption

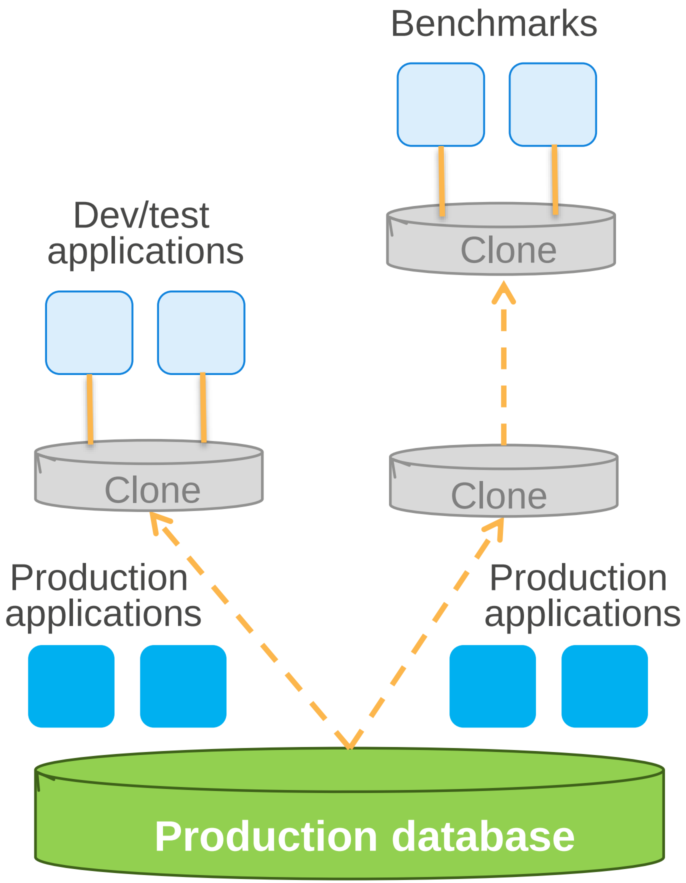
Fast database cloning

Create a copy of a database without duplicate storage costs

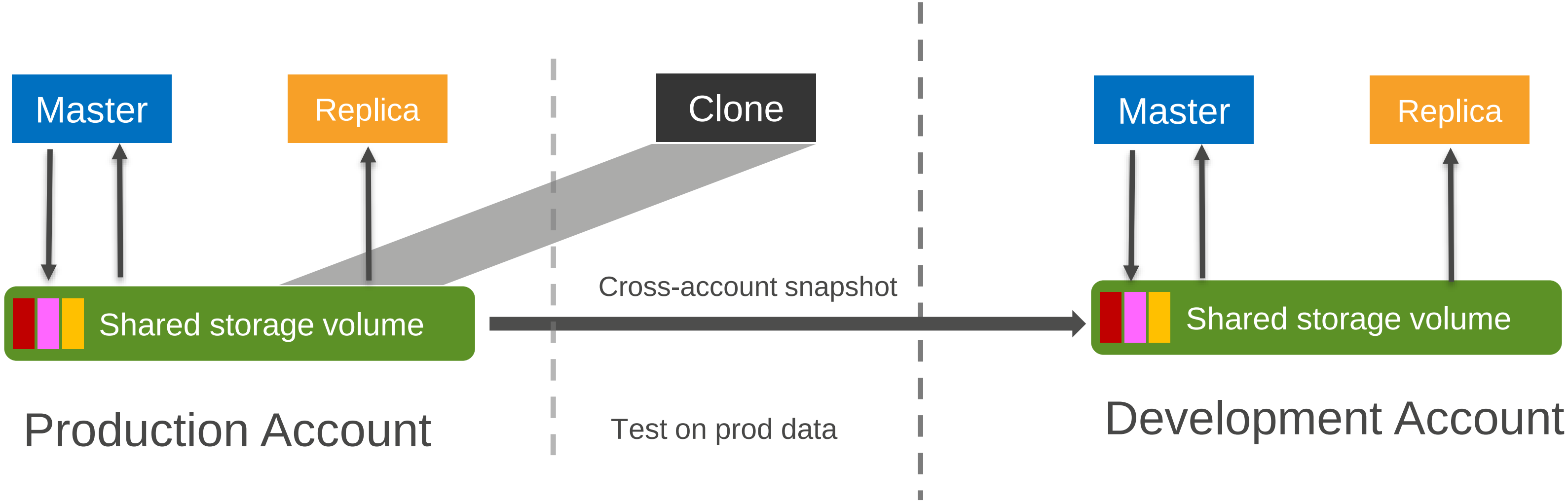
- Creation of a clone is nearly instantaneous – we don’t copy data
- Data copy happens only on write – when original and cloned volume data differ

Typical use cases:

- Clone a production DB to run tests
- Reorganize a database
- Save a point in time snapshot for analysis without impacting production system.



DevTest environments



- Create a clone of multi-terabyte DB clusters in minutes. Use for diagnosis and pre-prod testing
- Restore a production DB cluster in a development accounts for development, staging, or partnering with other teams

Leverage MySQL and AWS ecosystems

Business Intelligence	Data Integration	Query and Monitoring
tableau	talend	Webyog
ZOOMDATA	ATTUNITY	
looker	informatica	Navicat Premium for Aurora

Source: Amazon

AWS Ecosystem

Lambda

S3

IAM

KMS

CloudWatch

Route53

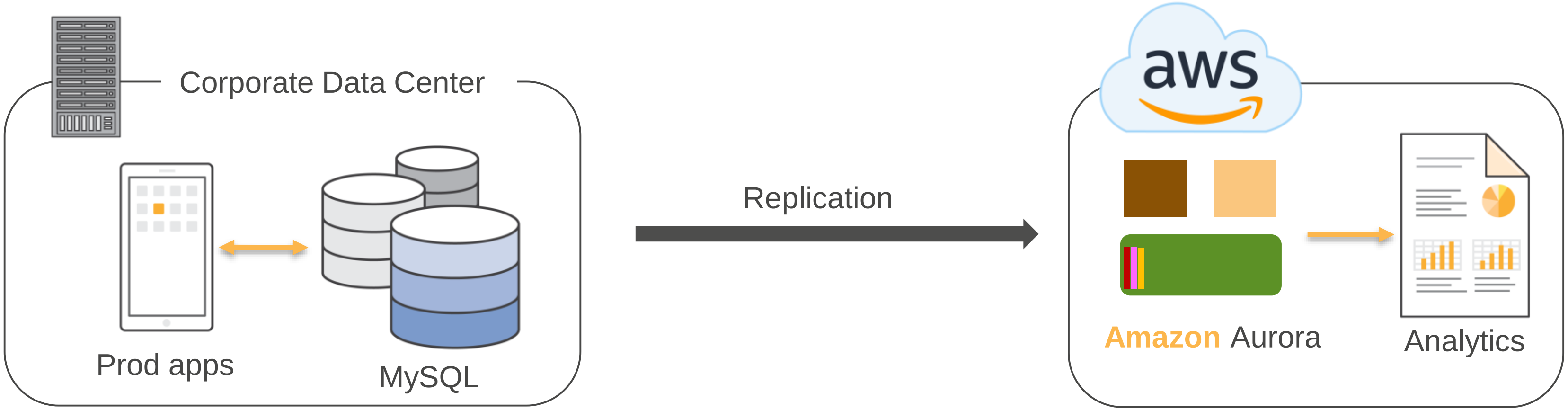
VPC

SWF

tableau

“We ran our compatibility test suites against Amazon Aurora and everything just worked." - Dan Jewett, Vice President of Product Management at Tableau

Hybrid operations and migration



Provider of an online marketplace for short-term rentals replicates on-premise production data into Aurora for read-only analytics workloads.

- Seamless replicate to Aurora: Full compatibility with MySQL and PostgreSQL
- Data replicated in minutes – run analytics on near real-time data
- Provides a clean waypoint towards full write/read migration

Amazon Aurora migration options

Source database	From where	Recommended option
MySQL	RDS	Console based automated snapshot ingestion and catch up via binlog replication.
MySQL MariaDB PERCONA	EC2, on premise	Binary snapshot ingestion through S3 and catch up via binlog replication.
ORACLE Microsoft SQL Server	EC2, on premise, RDS	Schema conversion using SCT and data migration via DMS.

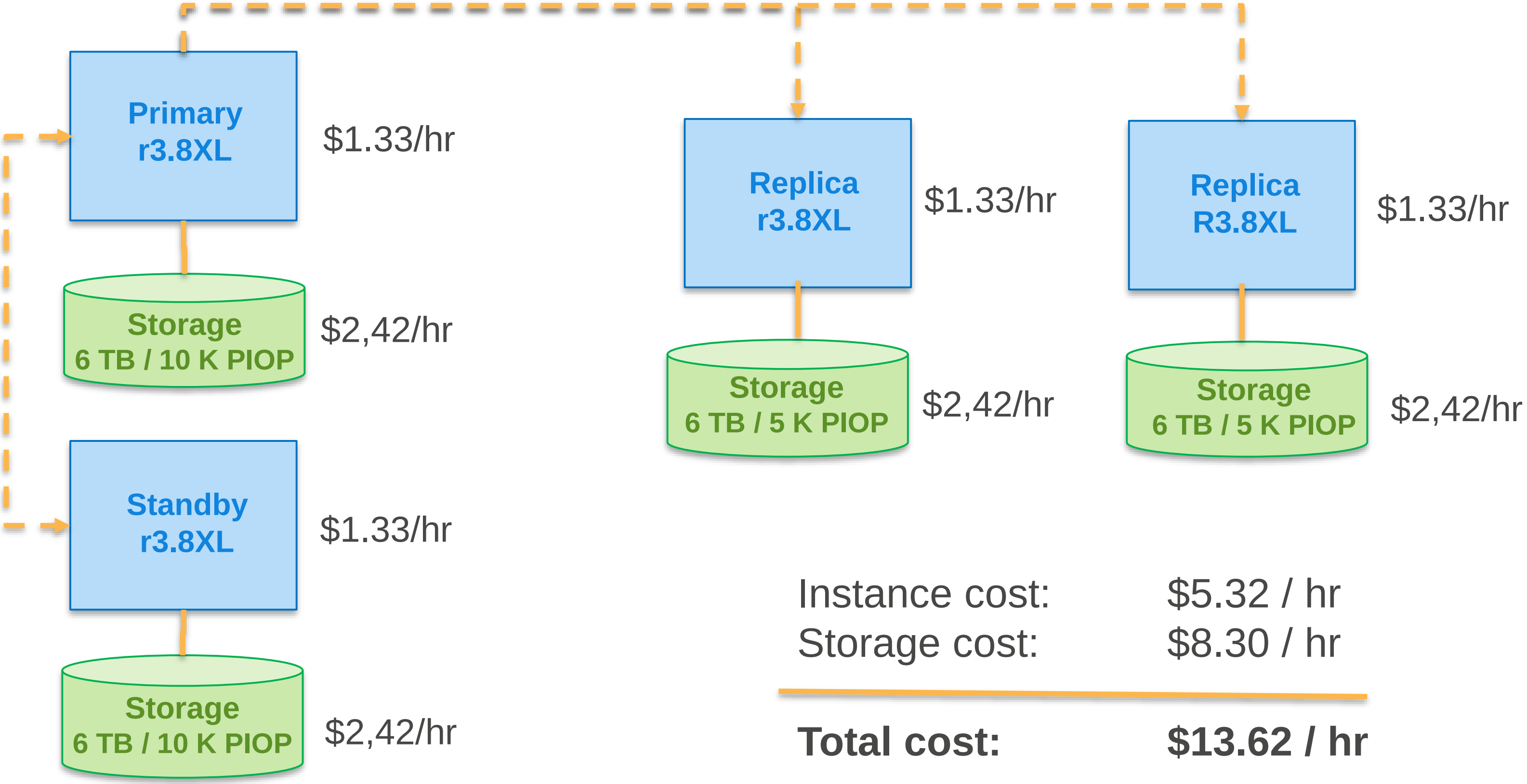
Amazon Aurora saves you money

1/10th of the cost of commercial databases

Cheaper than even MySQL

Cost of ownership: Aurora vs. MySQL

MySQL configuration hourly cost

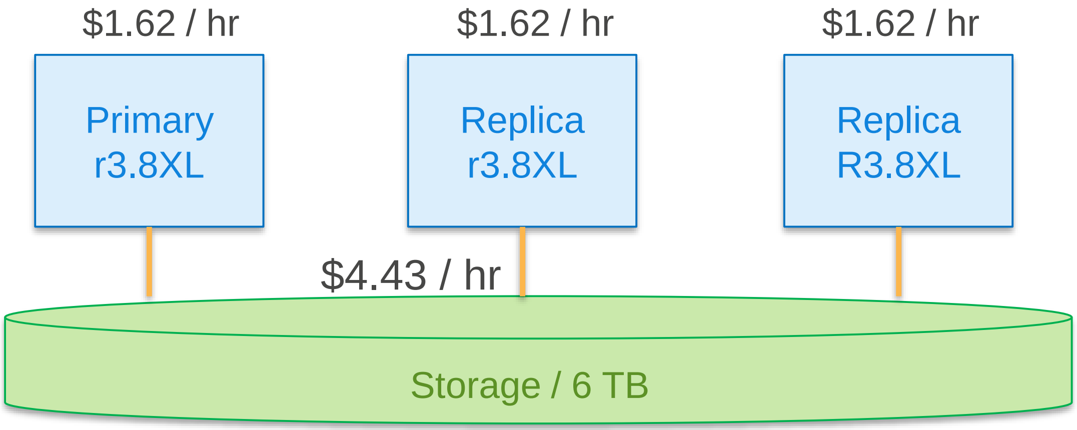


Instance cost:	$5.32 / hr
Storage cost:	$8.30 / hr
Total cost:	**$13.62 / hr**

Cost of ownership: Aurora vs. MySQL

Aurora configuration hourly cost

- No idle standby instance
- Single shared storage volume
- No PIOPs – pay for use I/O
- Reduction in overall IOP



$1.62 / hr

Primary r3.8XL

$1.62 / hr

Replica r3.8XL

$1.62 / hr

Replica R3.8XL

$4.43 / hr

Storage / 6 TB

Instance cost:	$4.86 / hr
Storage cost:	$4.43 / hr
Total cost:	**$9.29 / hr**

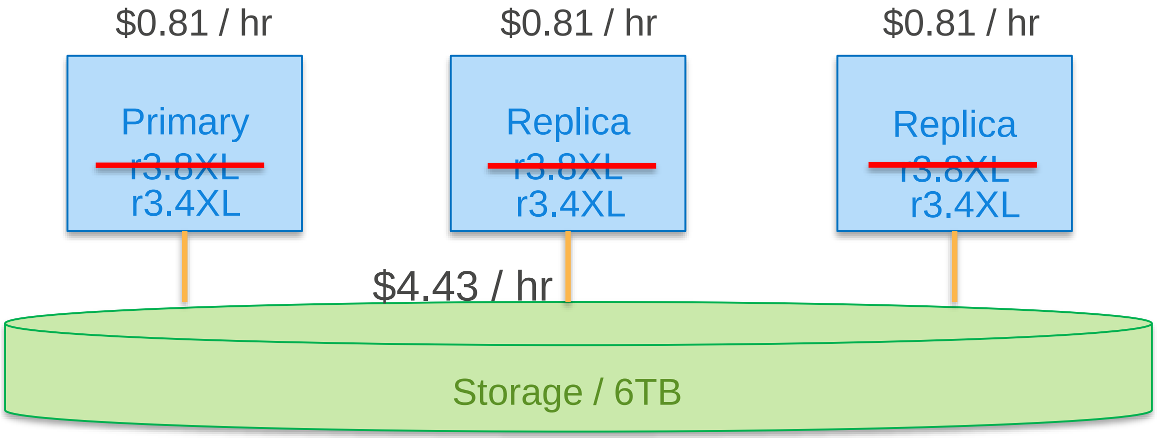
31.8% Savings

*At a macro level Aurora saves over 50% in storage cost compared to RDS MySQL.

Cost of ownership: Aurora vs. MySQL

Further opportunity for saving

- Use smaller instance size
- Pay-as-you-go storage



Instance cost:	$2.43 / hr
Storage cost:	$4.43 / hr
Total cost:	**$6.86 / hr**

49.6% Savings

Storage IOPs assumptions:

1. Average IOPs is 50% of Max IOPs
2. 50% savings from shipping logs vs. full pages

Higher performance, lower Cost

SAFE SOFTWARE

Safe.com lowered their bill by **40%** by switching from sharded MySQL to a single Aurora instance.

igt DOUBLE DOWN

Double Down Interactive (gaming) lowered their bill by **67%** while also achieving better latencies (most queries ran faster) and lower CPU utilization.

- Fewer instances needed
- Smaller instances can be used
- No need to pre-provision storage
- No additional storage for read replicas

Higher performance, lower Cost

AUTODESK®

⬆ 7x	⬇ 2x	⬇ 10x
Database connections	Response time	CPU utilization

> “Our application usage had grown exponentially over the last year. We were looking for horizontal scaling of our database to address the increased load. Amazon Aurora’s relatively low replication lag has helped us handle current load and positions us well for future growth.”

Expedia – Journey to Amazon Aurora

Gurmit Singh Ghatore, Principal Database Engineer

Brandon O’Brien, Principal Software Engineer

AWS re:Invent

PRODUCT

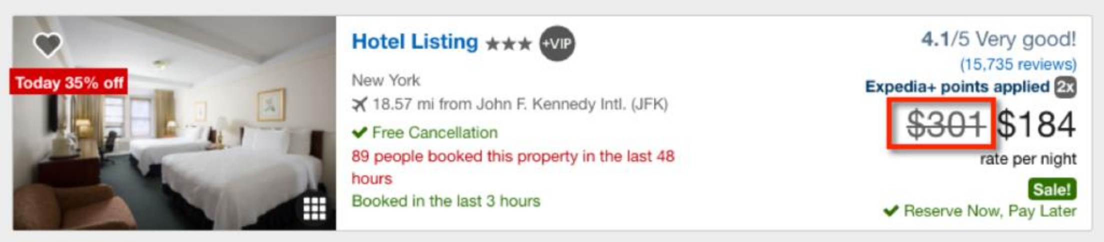
Real-time contextual pricing

- Enhance traveler’s shopping experience with contextual pricing e.g. strikethrough price

ENGINEERING

Low-latency pricing service

- Build a system that computes lodging price in real time
- Build a serving layer to surface prices to live Expedia traffic



Today 35% off

Hotel Listing ★★★ +VIP

New York

✈ 18.57 mi from John F. Kennedy Intl. (JFK)

✓ Free Cancellation

89 people booked this property in the last 48 hours

Booked in the last 3 hours

4.1/5 Very good!

(15,735 reviews)

Expedia+ points applied 2x

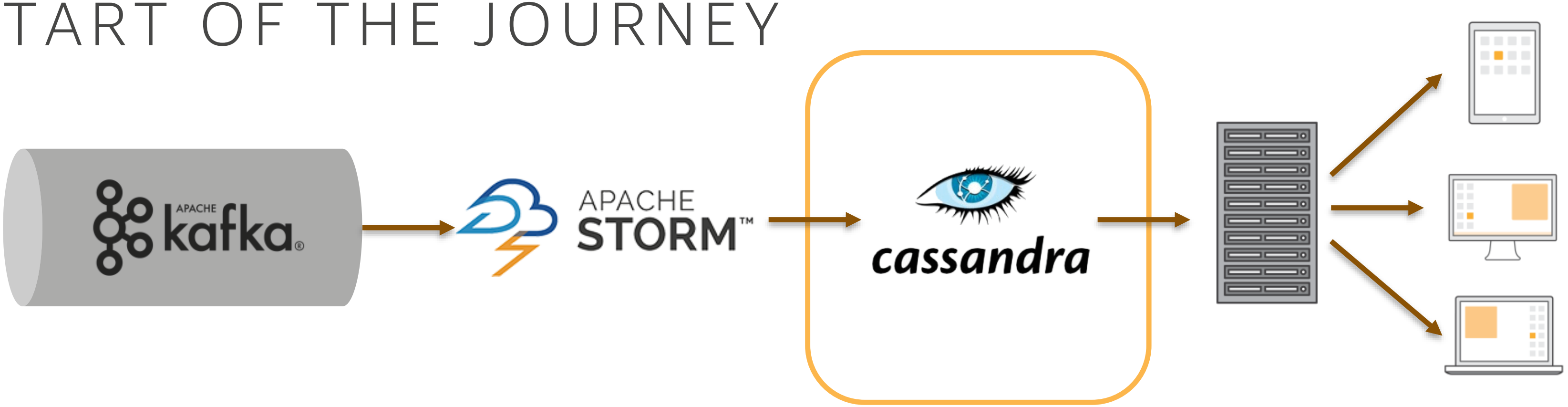
~~$301~~ $184

rate per night

Sale!

✓ Reserve Now, Pay Later

START OF THE JOURNEY



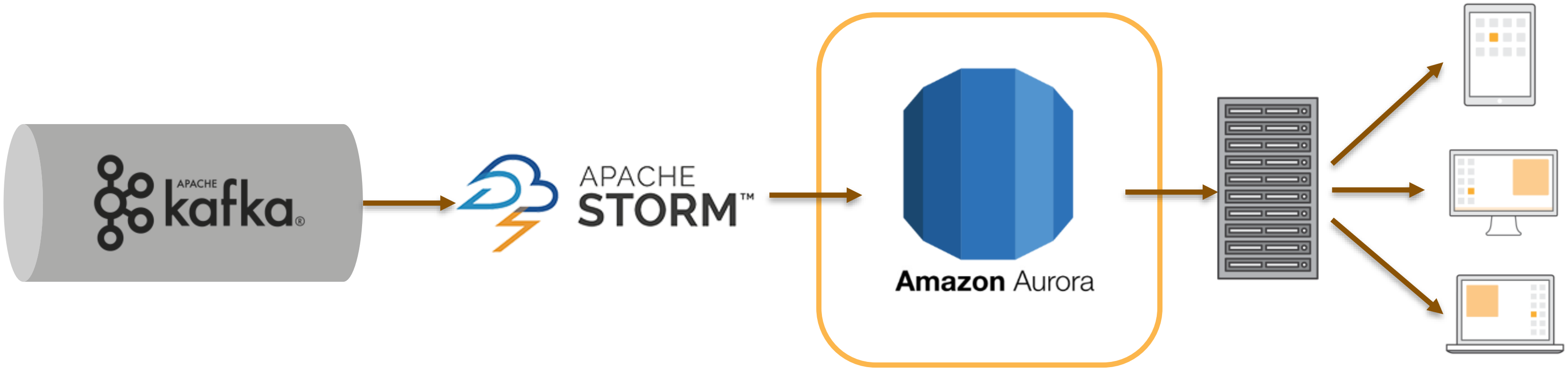
$$ High operational costs

✗ Ad hoc queries

✗ Flexibility

✗ Single use case

TRANSITION TO AMAZON AURORA

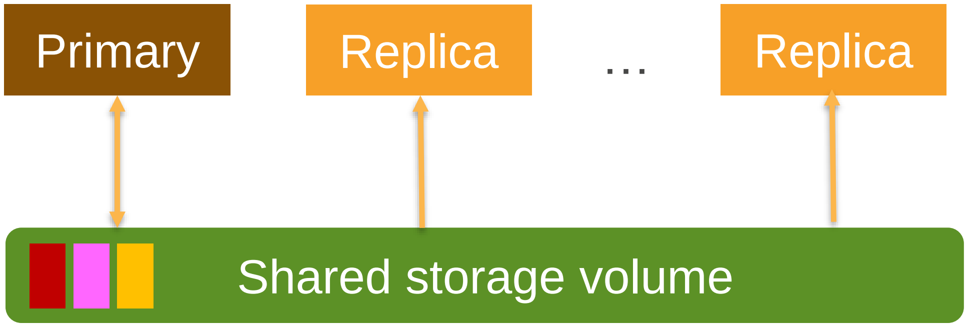


Scalable | Low maintenance | High performance

WORKLOAD HIGHLIGHTS

1B+	4,000	25,000
rows	writes/second	reads/second

ON THE PATH - INFRASTRUCTURE DETAILS



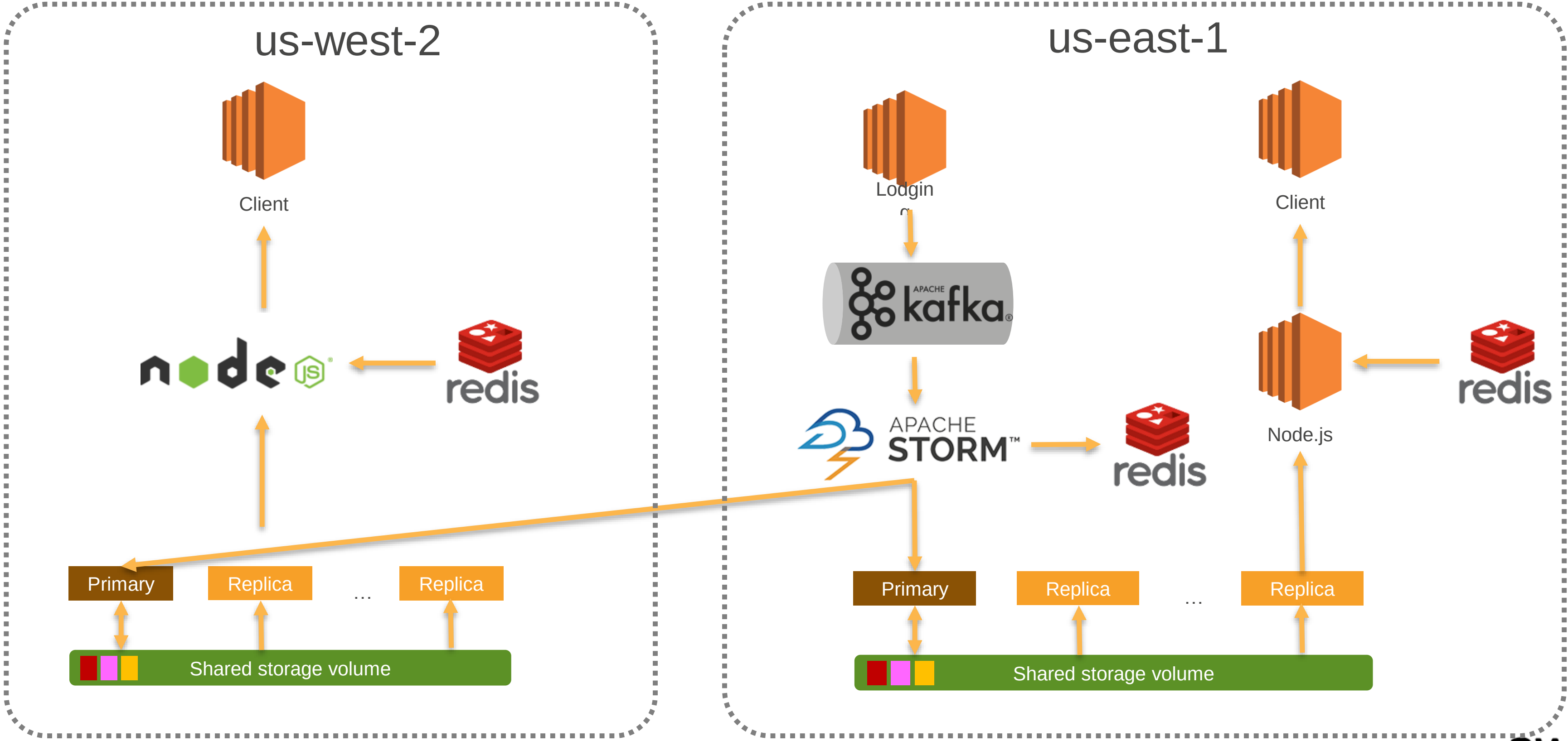
12 EC2 instances

c3.8xlarge

6 Aurora Replicas

db.r3.8xLarge

ARCHITECTURE

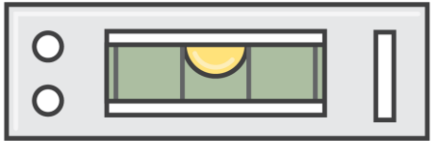


REQUIREMENTS AND SOLUTIONS

<100 ms

99% of read responses

- Node.js **splits the incoming read request** into smaller batches and fires off to all **read replicas in parallel**.
- Results returned by Aurora are combined in Node.js



Balanced resource utilization

- **Aurora provides connection load balancing** rather than individual queries.
- We built **application connection pools to load balance** across all read replicas evenly.

REQUIREMENTS AND SOLUTIONS (CONT.)

High-volume data ingestion

- Storm cluster **parallelizes workload** across storm nodes.
- Each node uses JDBC micro batch inserts.

Multi-region presence

- **Storm persists data** to independent Aurora clusters **in local and remote regions**

Why didn’t we use in-built cross-region read replica feature?

Aurora (as of now) supports only one cross-region replica. Also, we can’t create replicas from replica. Adopted solution allows creating multiple replicas from local primary for regional scale-out with cross region data-out cost only once.

REQUIREMENTS AND SOLUTIONS (CONT.)

Schema changes

Fast DDL

We leveraged Aurora Fast DDL to make in-place simple schema changes to live tables.

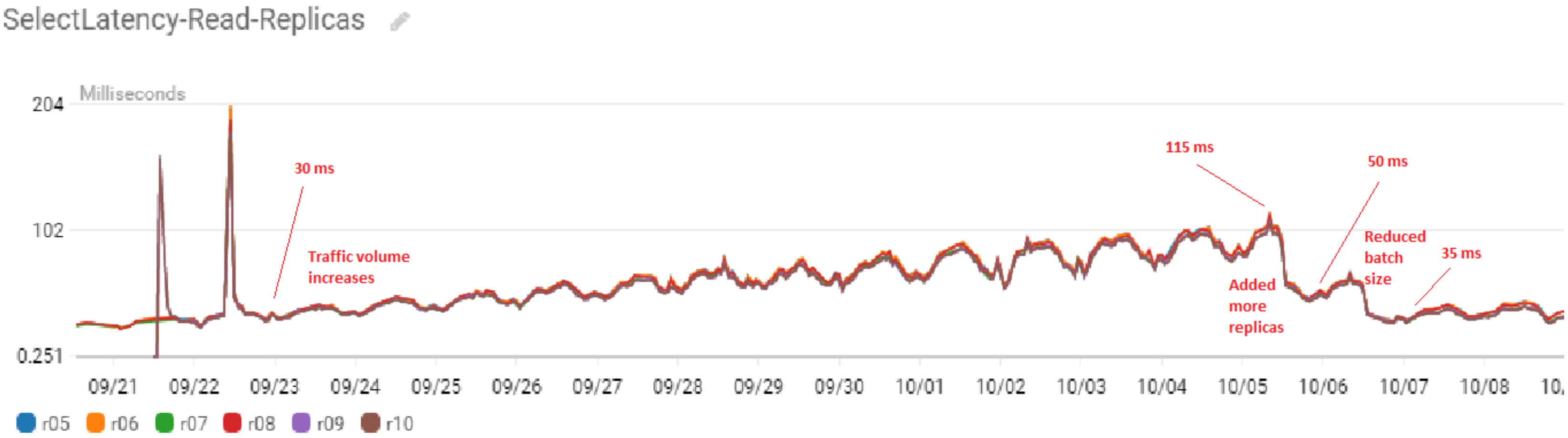
AWS Data Migration Service

For complex schema changes, create new schema table, transfer data and switch.

We utilized AWS DMS service for transferring data to new schema table.

PERFORMANCE SOLUTIONS

Performance degradation over time as client requests and volume increased.



Scale out database → **Parallelize queries**

PERFORMANCE NUMBERS

<20 ms

Replication lag

<50 ms

Select latency

<10 %

CPU utilization

<1 ms

Insert latency

40+ GB

Free-able memory

AWS re:Invent

aws

MONETARY AND MAINTENANCE BENEFITS

10-15%

Cost saving vs. Cassandra

- Real saving comes from multiple use of same data with different read pattern.
- Adhoc querying capability became a non-event.

<10 minutes

End-to-end downtime

- Applying patches and updates is easy with very low tolerable downtime
- Initially time was spent in resolving performance bottlenecks. Later on maintenance time was on upgrades and monitoring only.

AWS re:Invent

aws

KEY TAKEAWAYS

- Parallelize. Parallelize. Parallelize.
- Custom load balance reads
- **Cross-region replication** could be challenging (esp. for large volumes). Explore alternatives.
- **Complex schema/index changes** can be handled with ease **using Aurora features**. Leverage Aurora features as much as possible.
- Aurora’s **low maintenance** resulted in increased product velocity.

Some Other Aurora Sessions

DAT301	Deep Dive on Amazon Aurora MySQL compatible Edition	#1 Wednesday 4:45p #2 Friday 11:30am
DAT315	A Practitioner's Guide on Migrating to, and Running on Amazon Aurora	Thursday 4pm
DAT334	Amazon Aurora Performance Optimization	Wednesday 12pm
DAT331	Airbnb Runs on Amazon Aurora	Wednesday 1pm
DAT336	Amazon Aurora Storage Demystified: How It All Works	Wednesday 4pm
DAT338	Migrating from Oracle to Amazon Aurora	Thursday 5:30pm
DAT402	Deep Dive on the Amazon Aurora PostgreSQL-compatible Edition	Wednesday 4pm

Thank You

AWS re:Invent

aws