AWS re:Invent

DAT328-R

Deep dive on Amazon Aurora with PostgreSQL compatibility

Grant McAlister
Senior Principal Engineer
Amazon Web Services

Kevin Jernigan
Principal Product Manager,
Amazon Aurora PostgreSQL
Amazon Web Services

AWS re:Invent

aws

Amazon Relational Database Service (Amazon RDS): A managed service

Amazon Aurora | MySQL | PostgreSQL | MariaDB | Microsoft SQL Server | Oracle

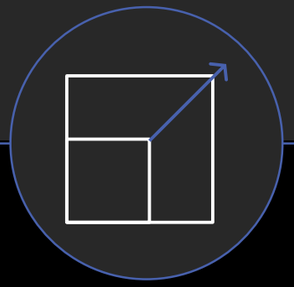
Easy to administer

Easily deploy and maintain hardware, OS and database software; built-in monitoring

Available & durable

Automatic Multi-AZ data replication; automated backup, snapshots, failover

Performant & scalable

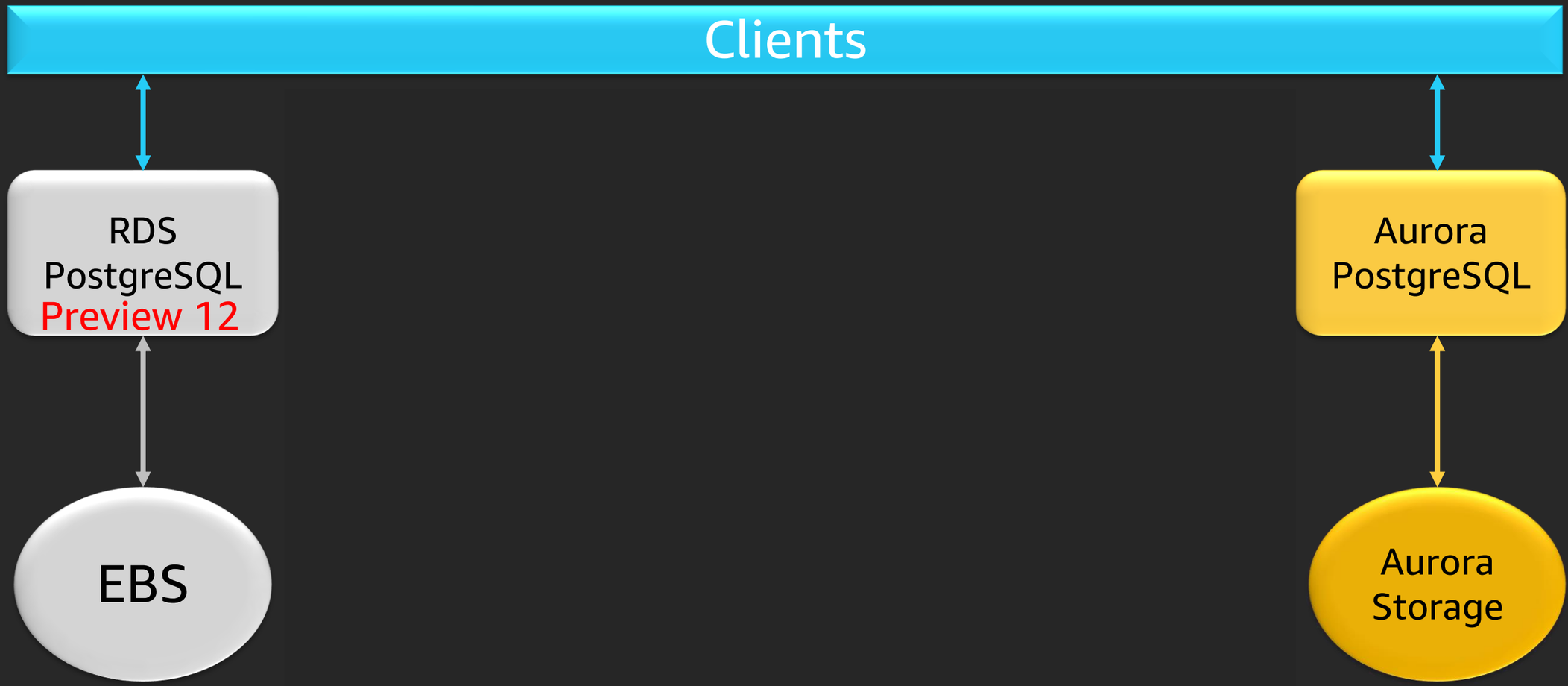


Scale compute and storage with a few clicks; minimal downtime for your application

Secure & compliant

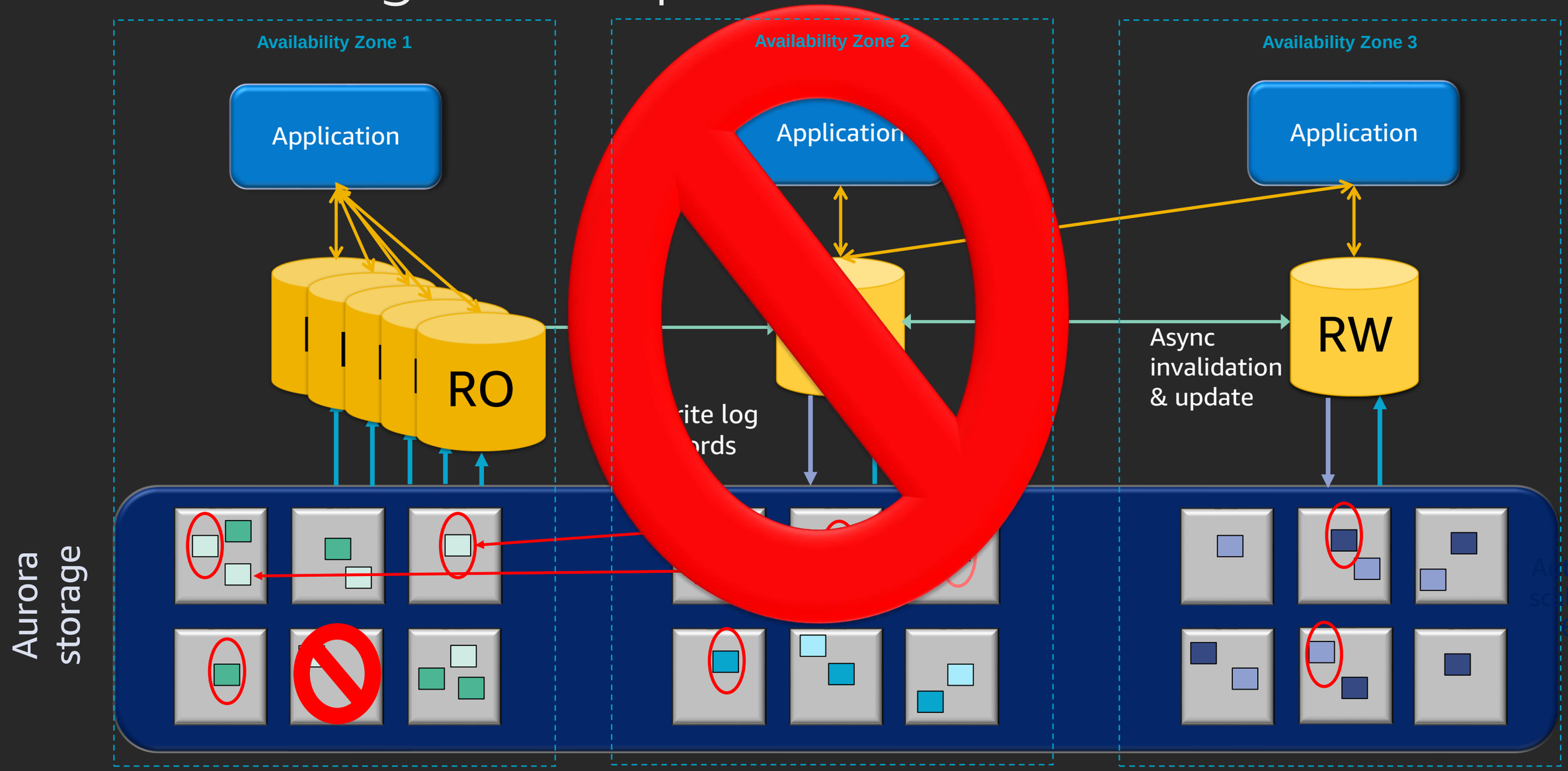
Data encryption at rest and in transit; industry compliance and assurance programs

Amazon RDS PostgreSQL universe



Base architecture

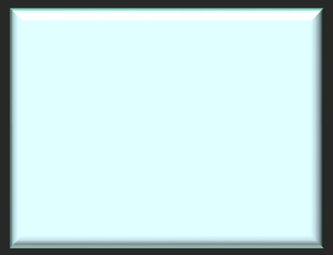
Aurora storage and replicas



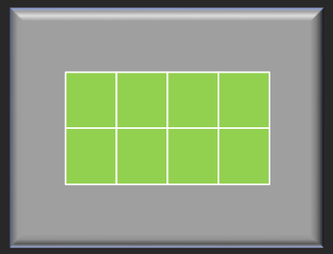
Log-based storage

Concurrency: Remove log buffer

Queued work



Log buffer



Storage

PostgreSQL

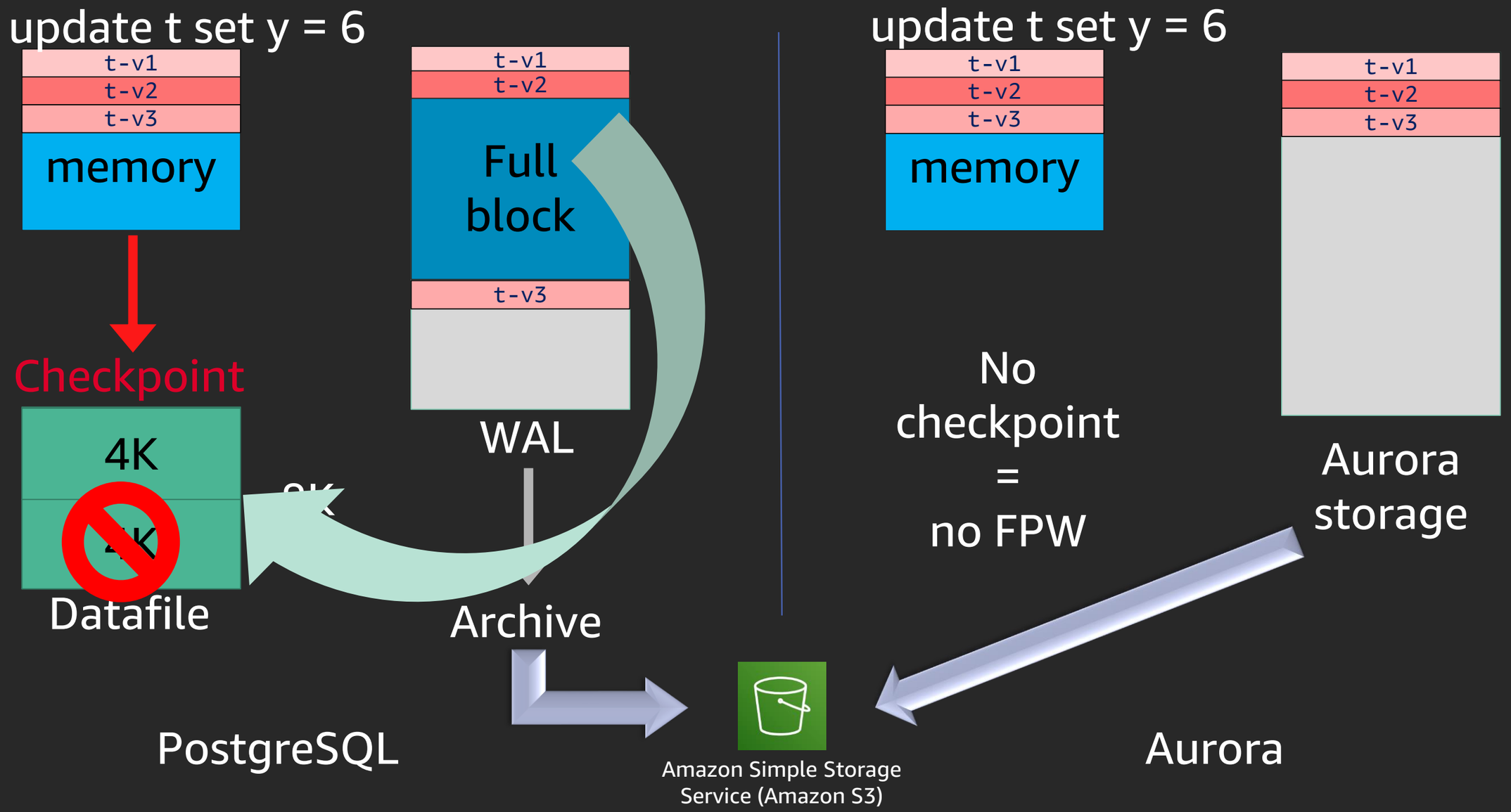
Queued work

Durability tracking

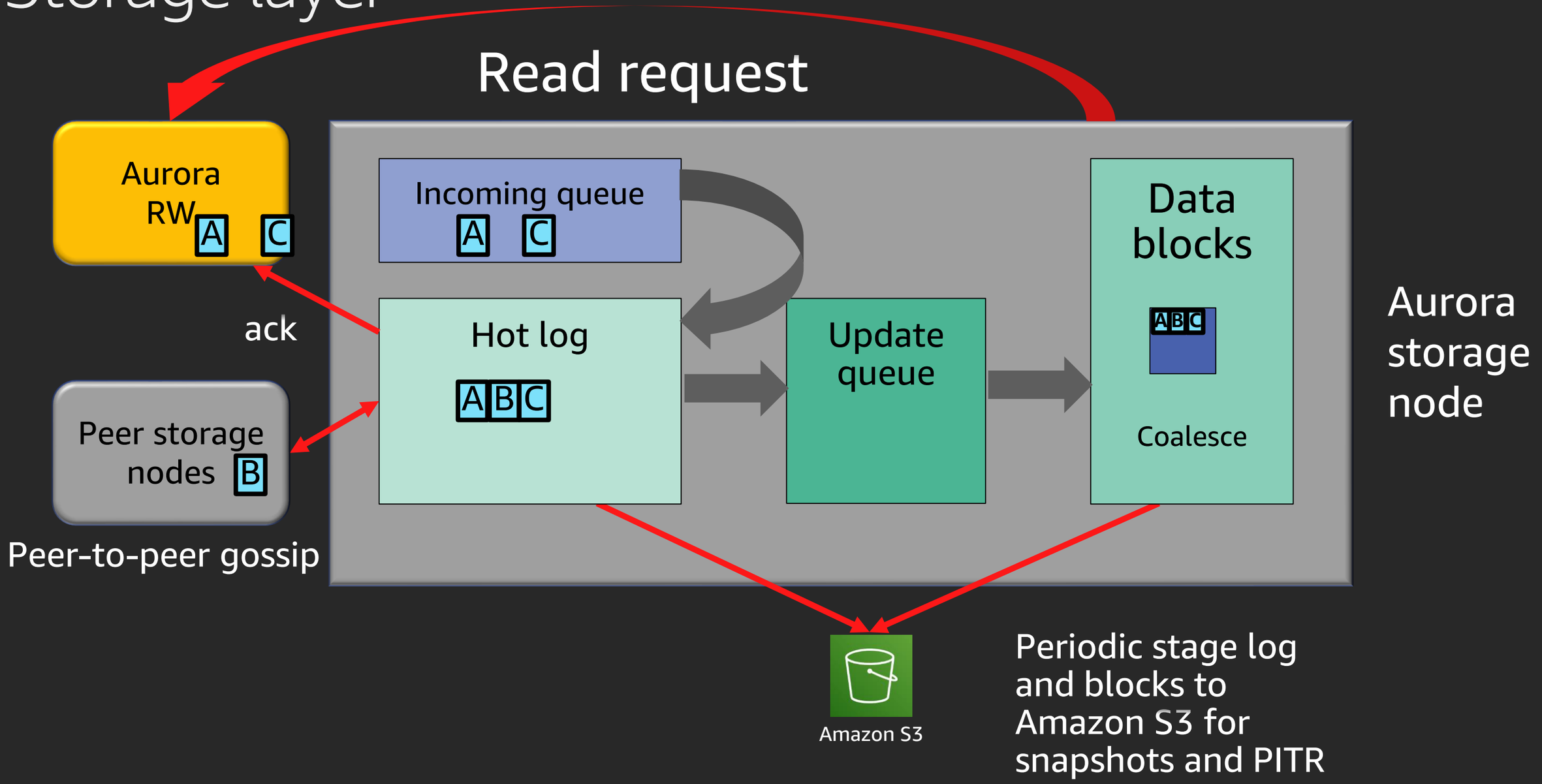
Storage

Aurora PostgreSQL

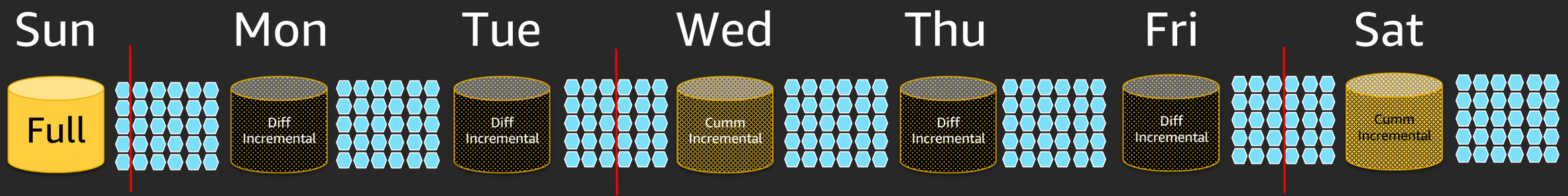
Aurora PostgreSQL: Writing less



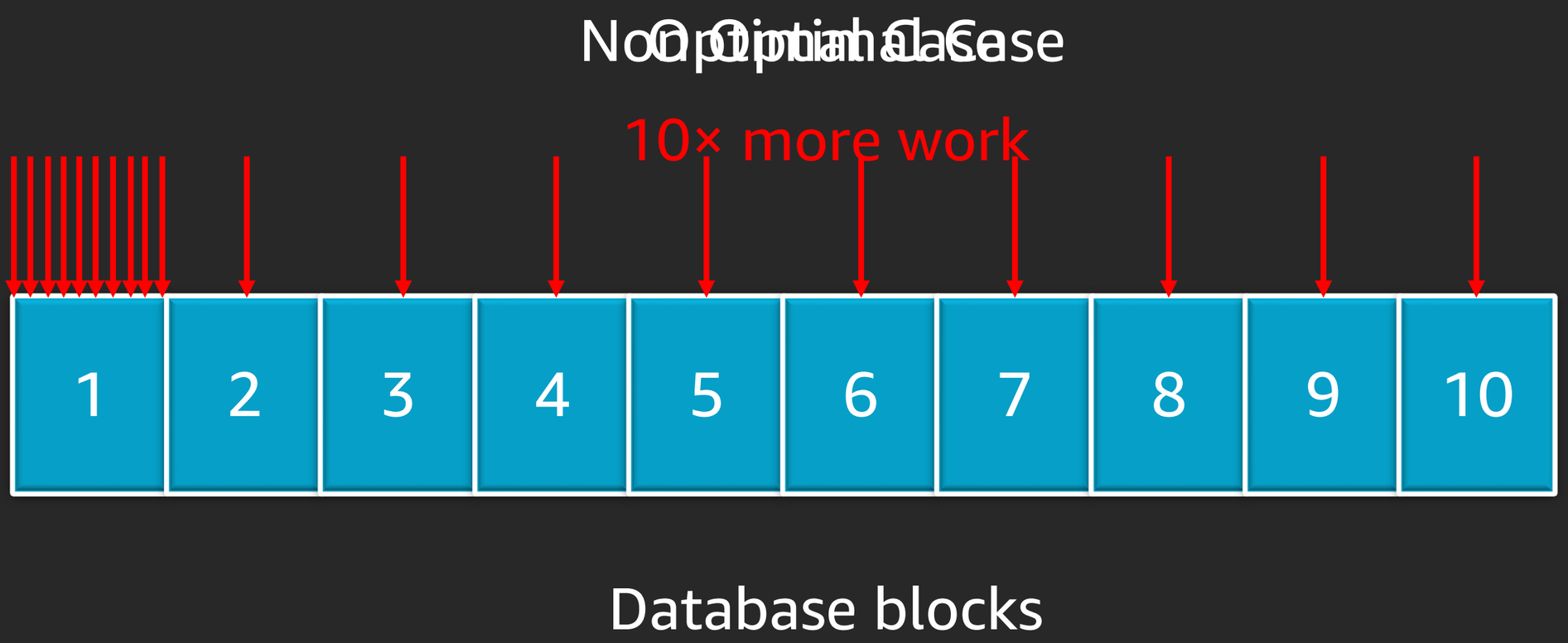
Storage layer



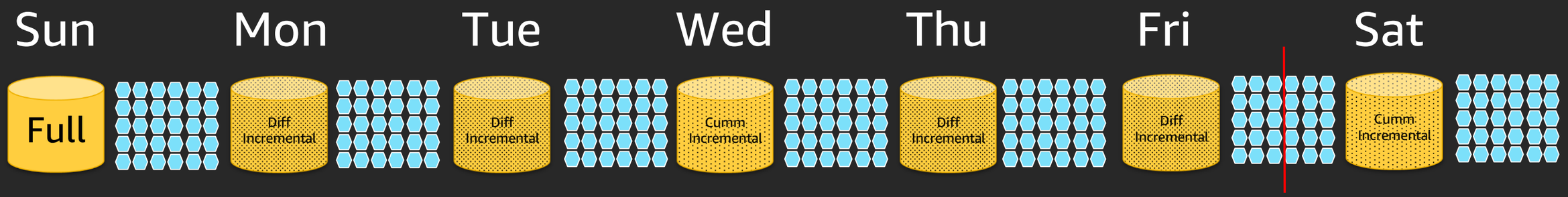
Conventional database backups



Incremental backups



Conventional database backups: Non-optimal



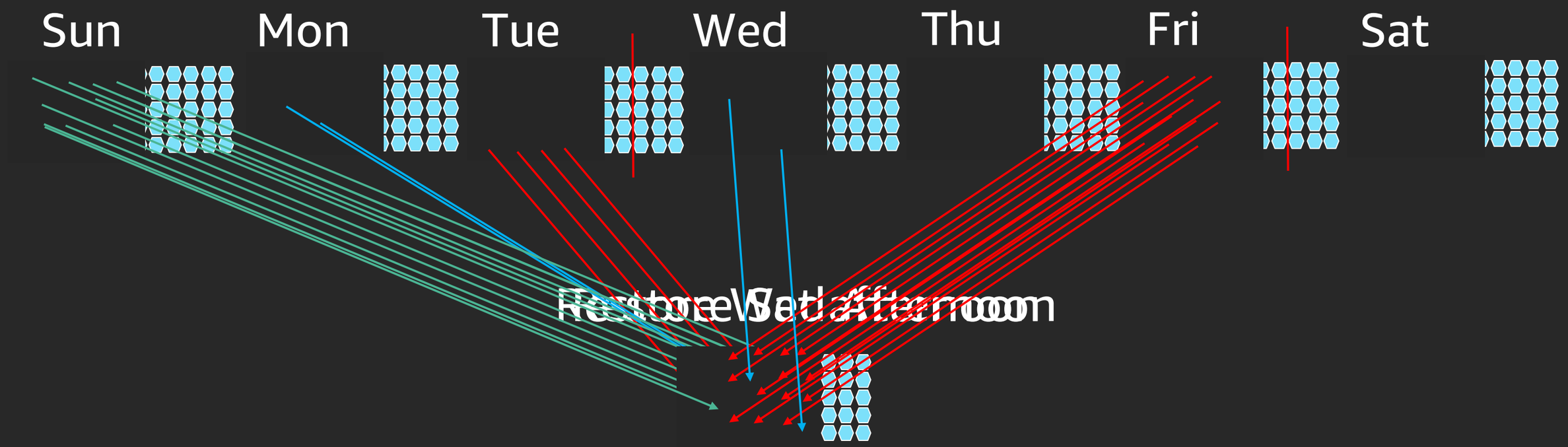
Restore Sat afternoon

3× more work

=

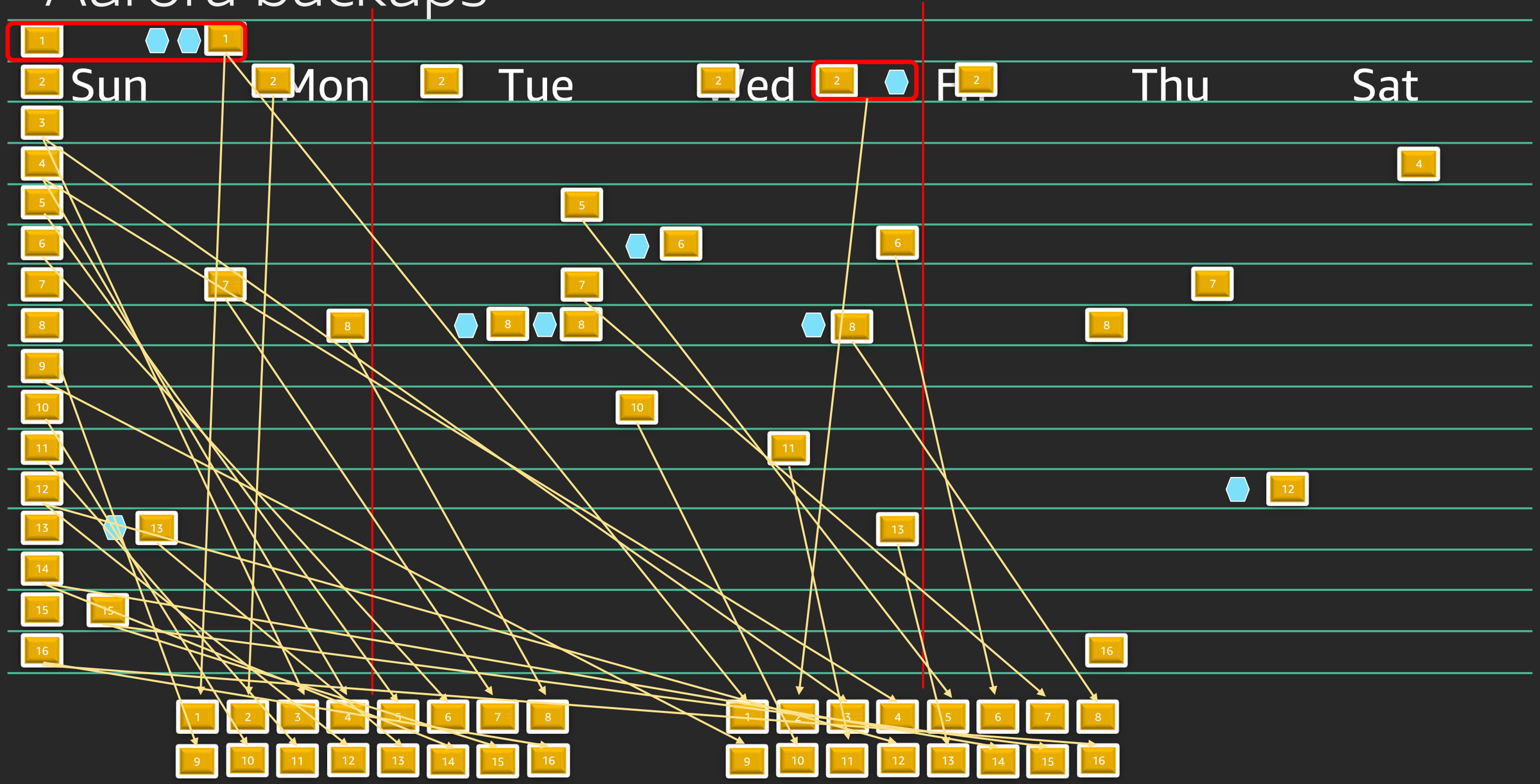
3× recovery time

Amazon RDS backups



Fixed time for EBS storage
Variable for WAL logs

Aurora backups

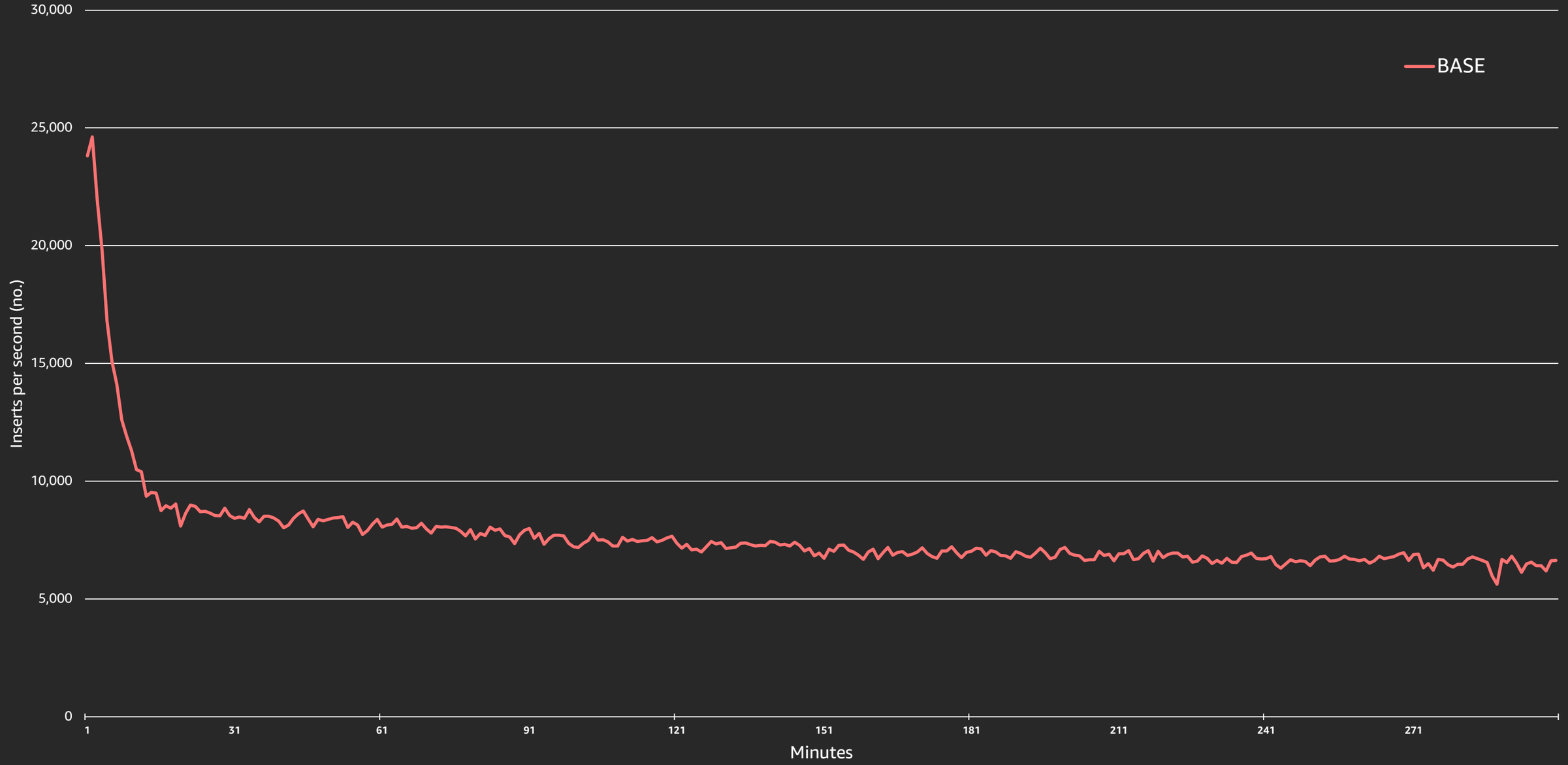


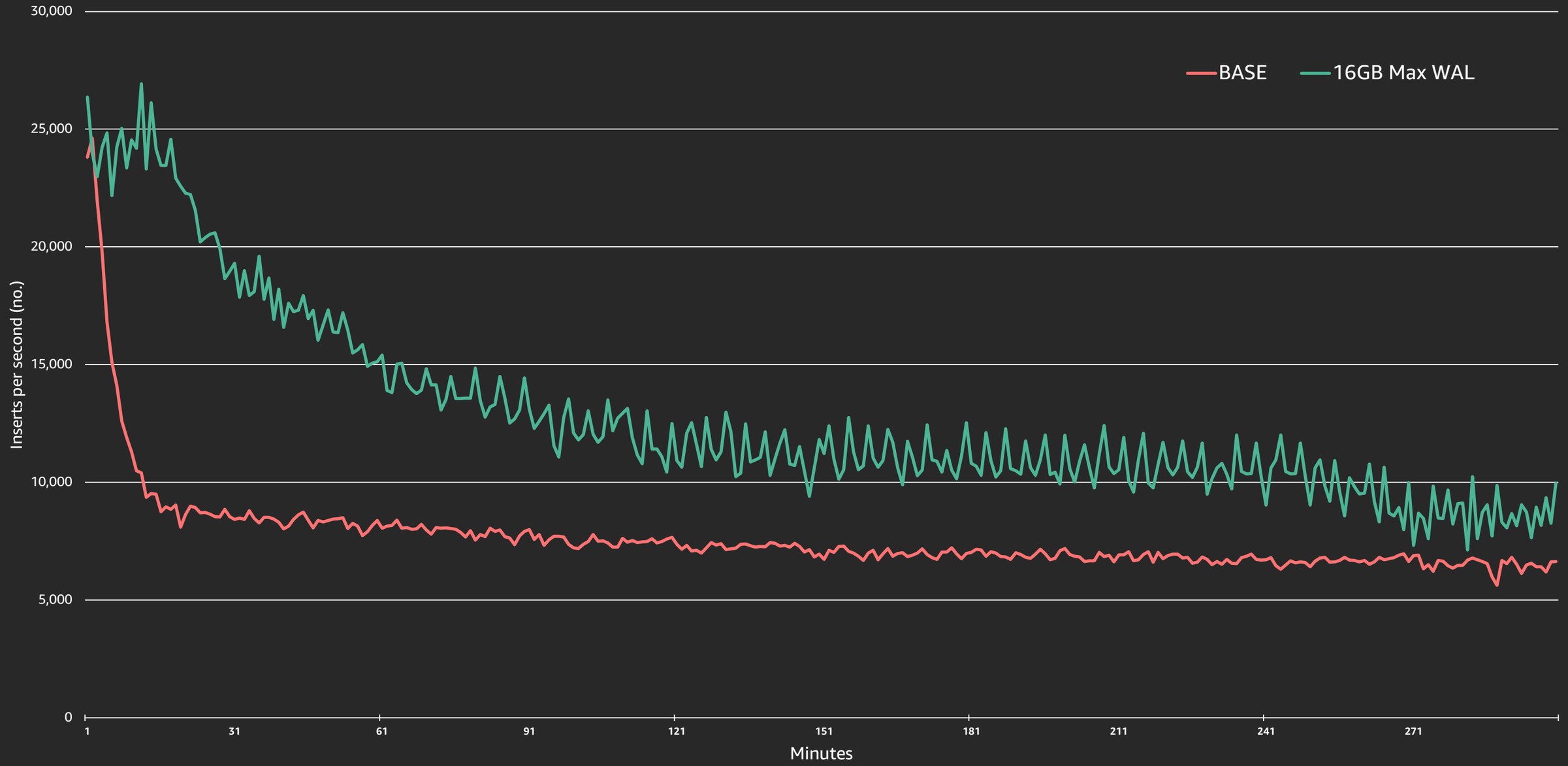
Insert test

Test table

- UUID PK—Random
- ID int—Right lean sequence
- VARCHAR(100)—Random
- VARCHAR(50)—Small set of words
- INT—Random
- INT—Random (smaller set)
- BOOLEAN—Random (50/50)
- BOOLEAN—Somewhat random (75/25)
- Timestamp—Right lean

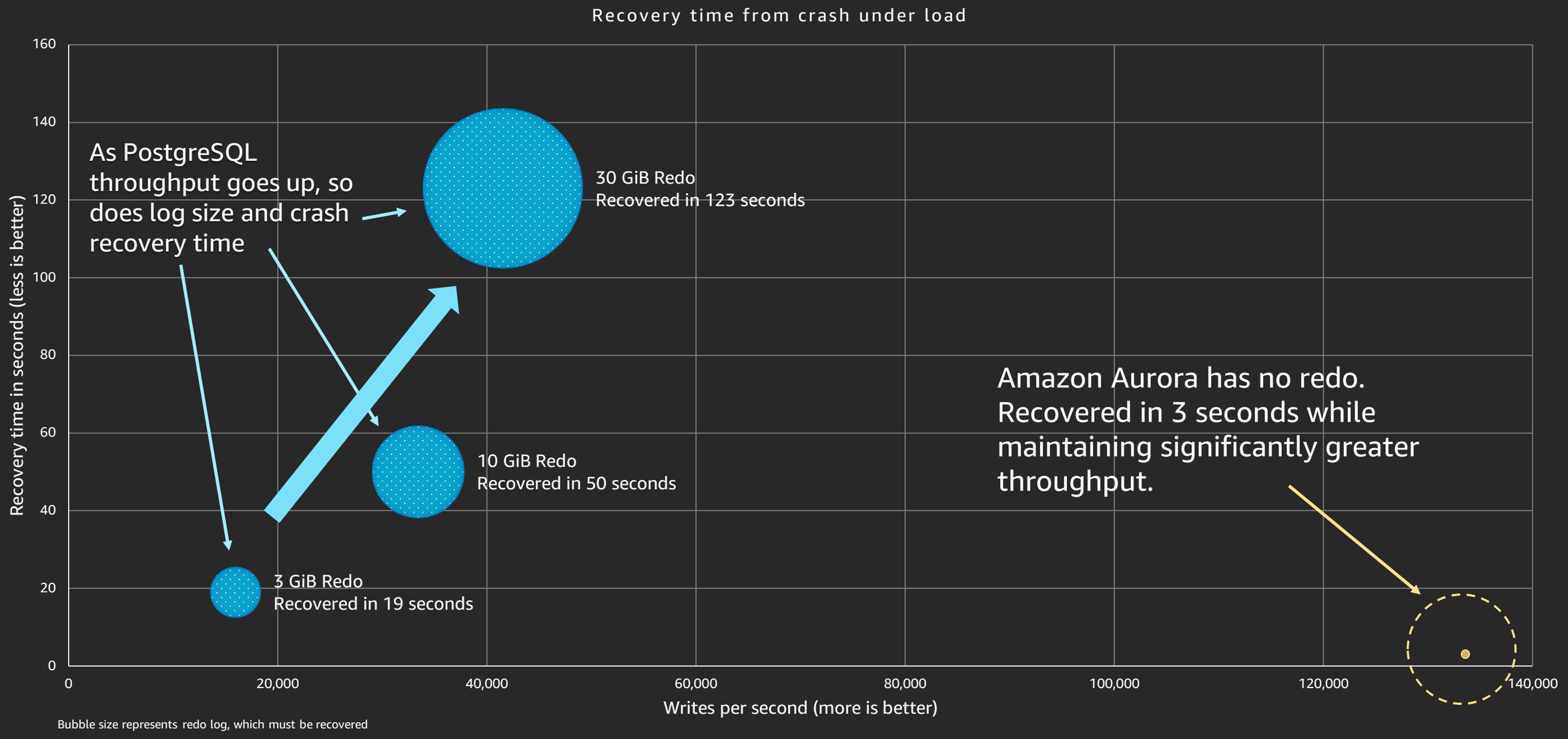
Index every column


Insert workload—PostgreSQL 9.6
30,000
25,000
20,000
15,000
10,000
5,000
0
Inserts per second (no.)
BASE
1
31
61
91
121
151
181
211
241
271
Minutes


Insert workload—PostgreSQL 9.6
BASE
16GB Max WAL
Inserts per second (no.)
30,000
25,000
20,000
15,000
10,000
5,000
0
1
31
61
91
121
151
181
211
241
271
Minutes


Insert Workload—PostgreSQL 9.6
30,000
25,000
20,000
15,000
10,000
5,000
0
Inserts per second (no.)
BASE
16GB Max WAL
Aurora PostgreSQL
1
31
61
91
121
151
181
211
241
271
Minutes

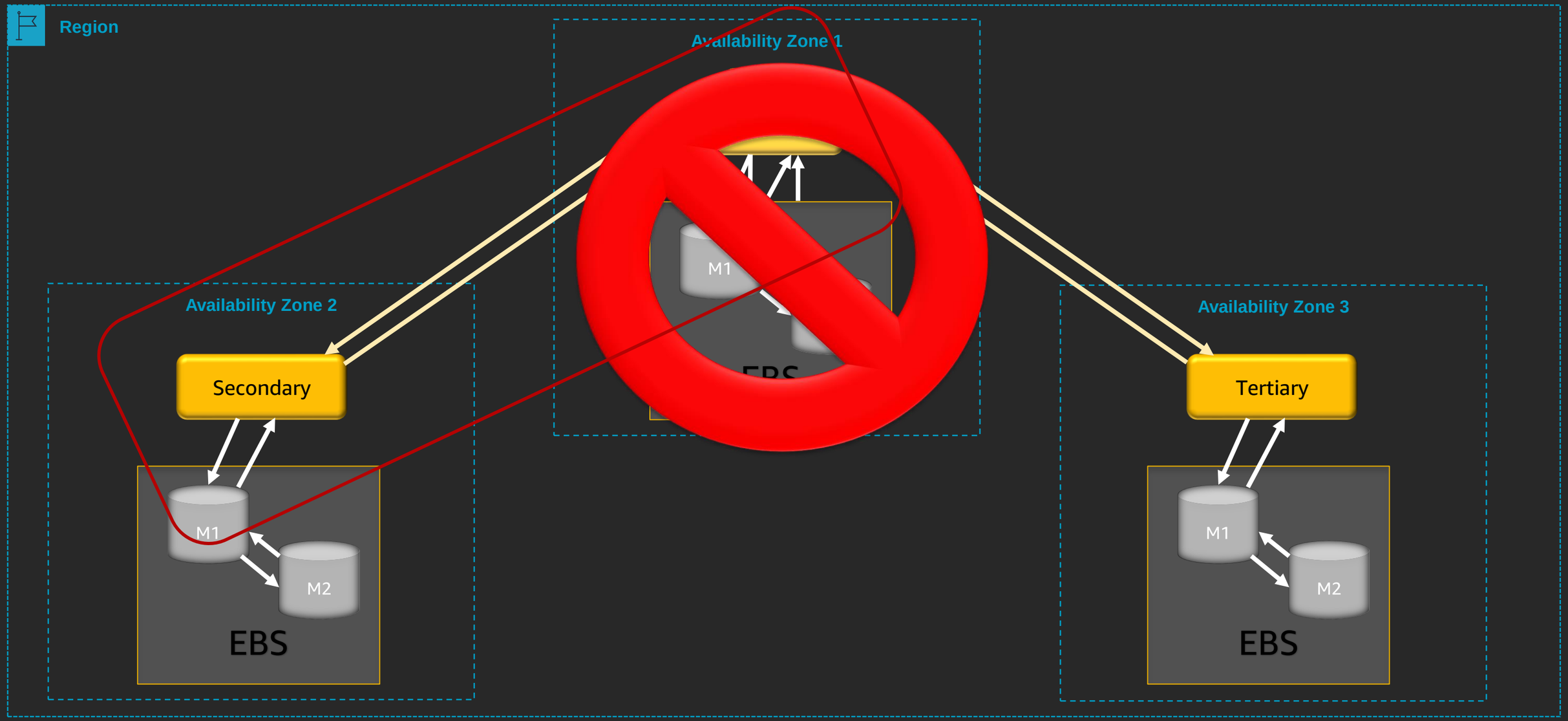
Amazon Aurora recovers up to 97% faster



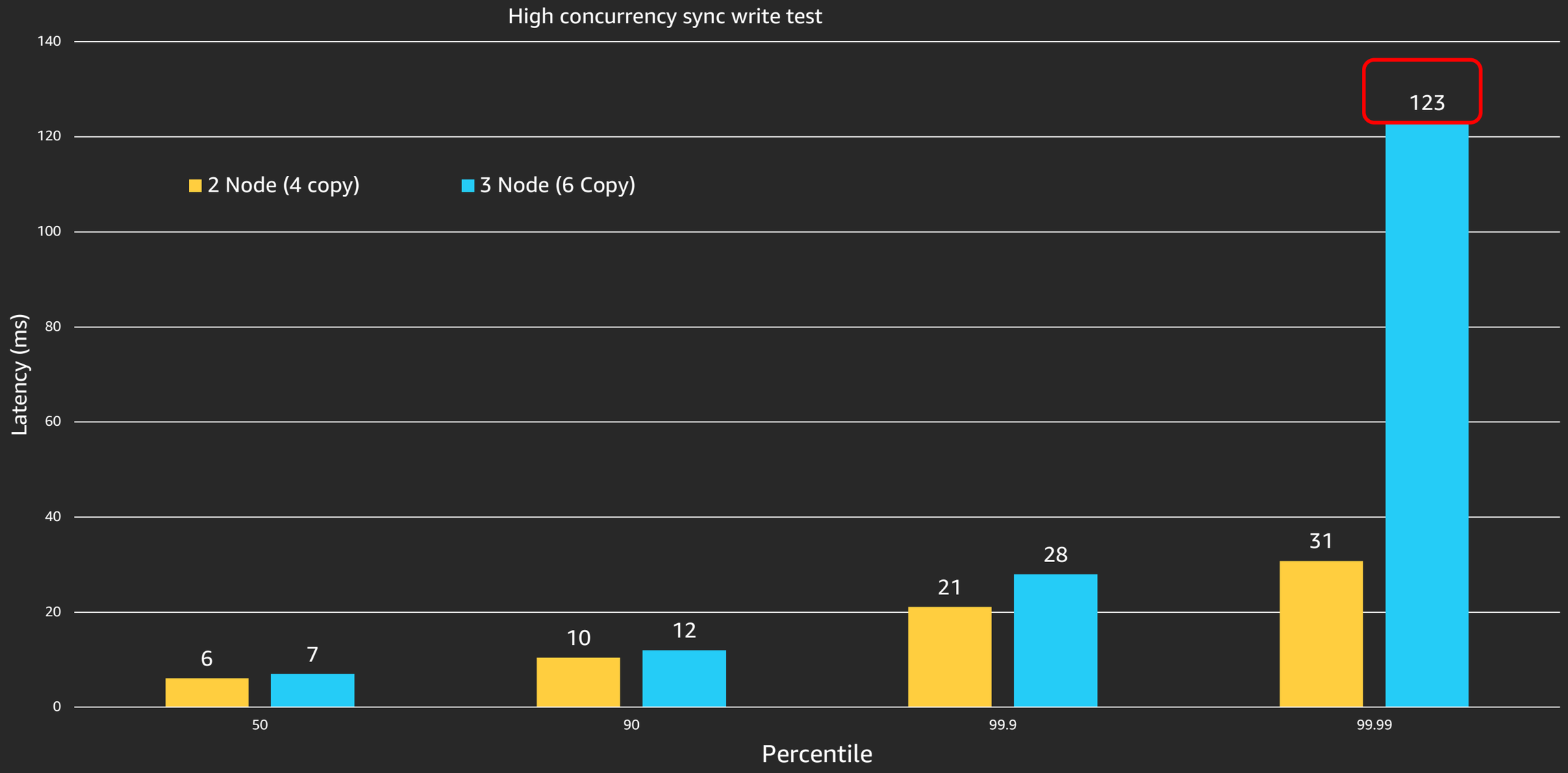
Bubble size represents redo log, which must be recovered

Durability: 4/6 quorum

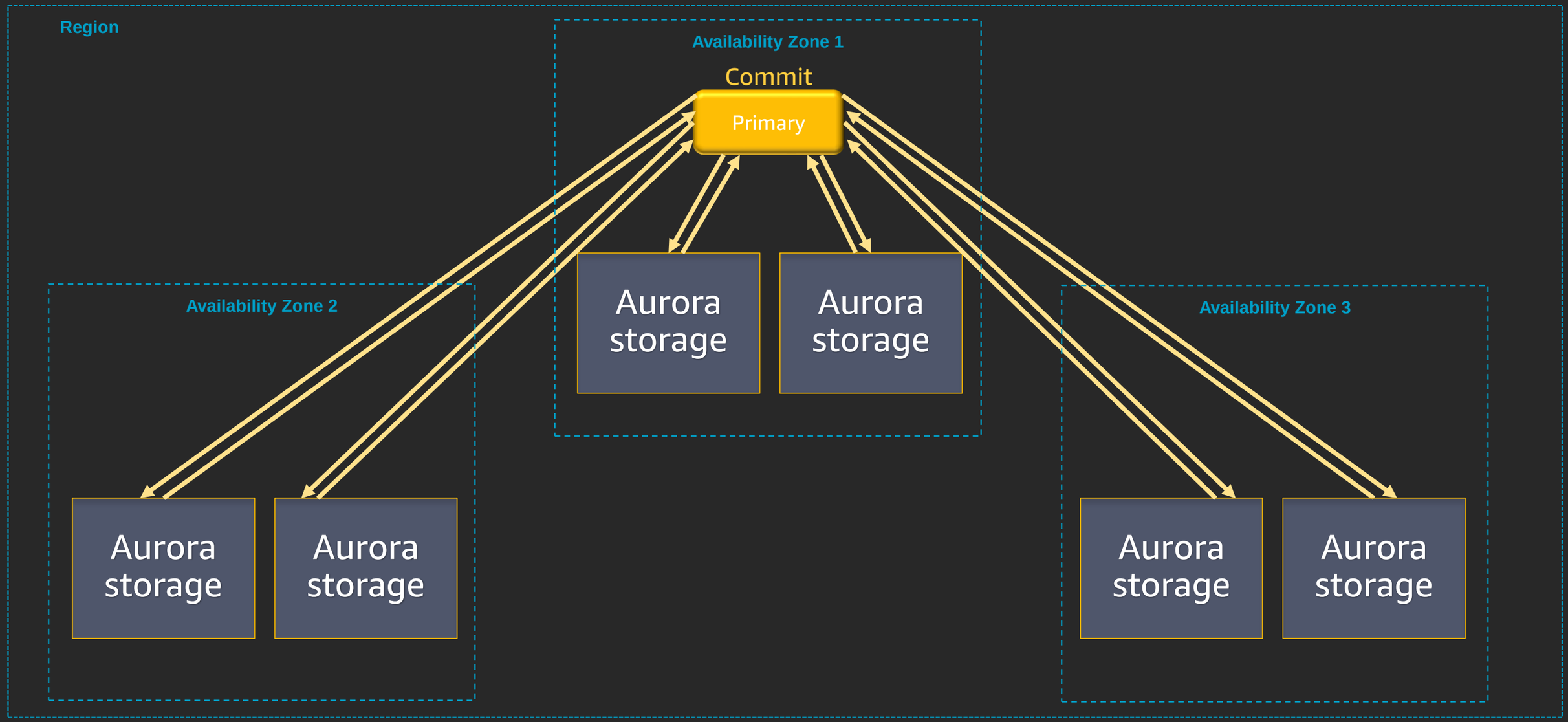
Typical synchronous replication: 3 locations



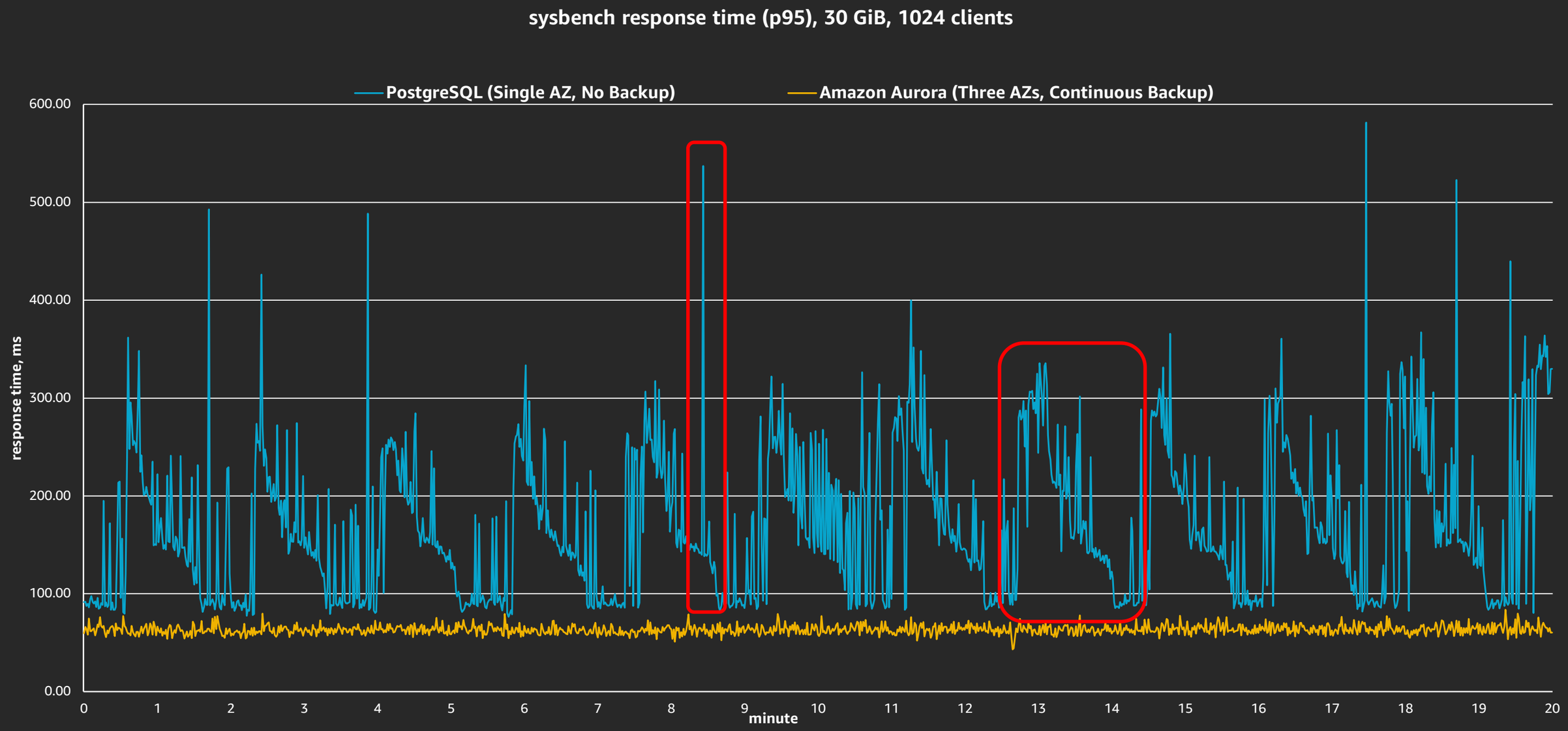
Cost of additional synchronous replicas



Amazon Aurora: 3 AZs, 6 copies

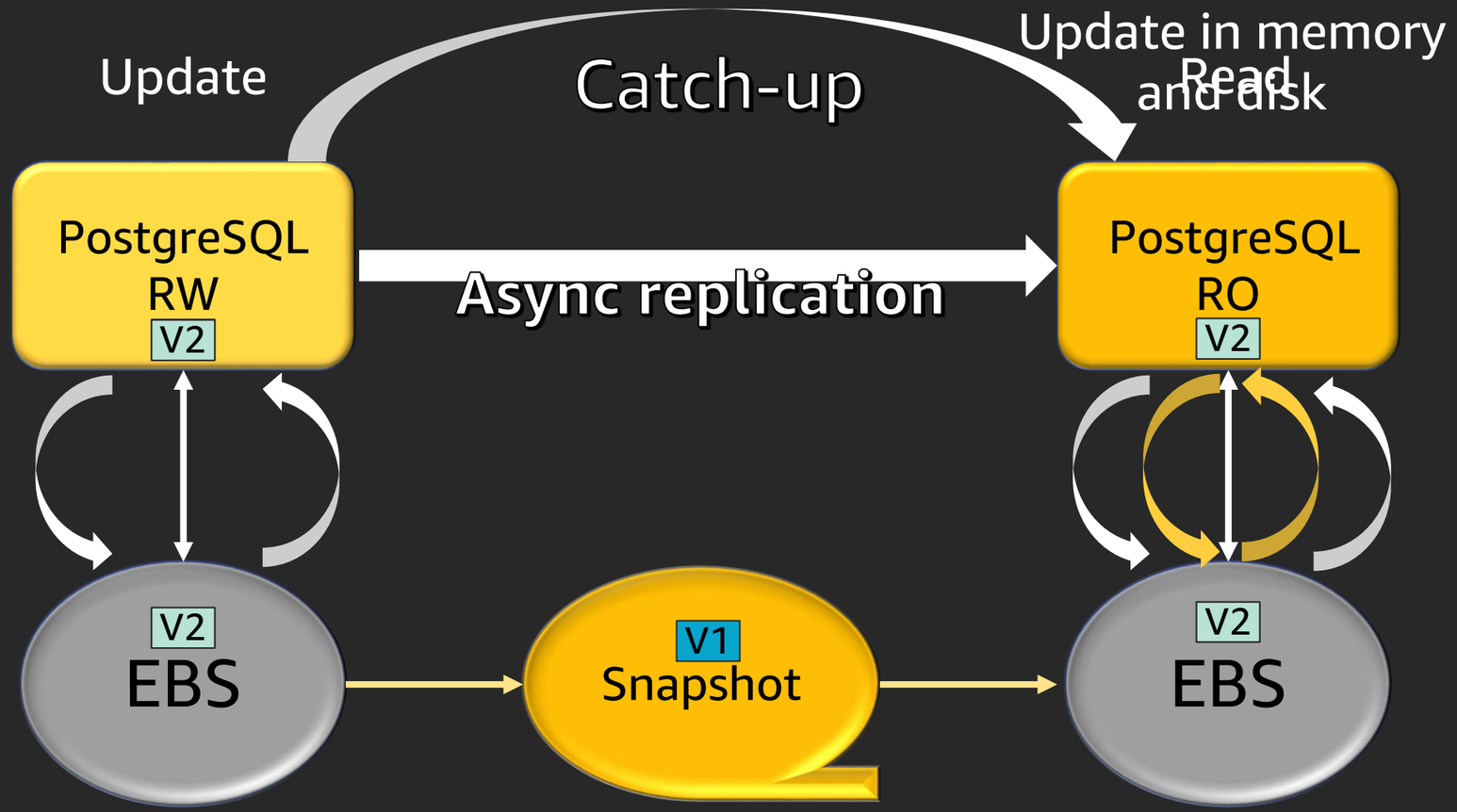


Amazon Aurora gives >2x lower response times

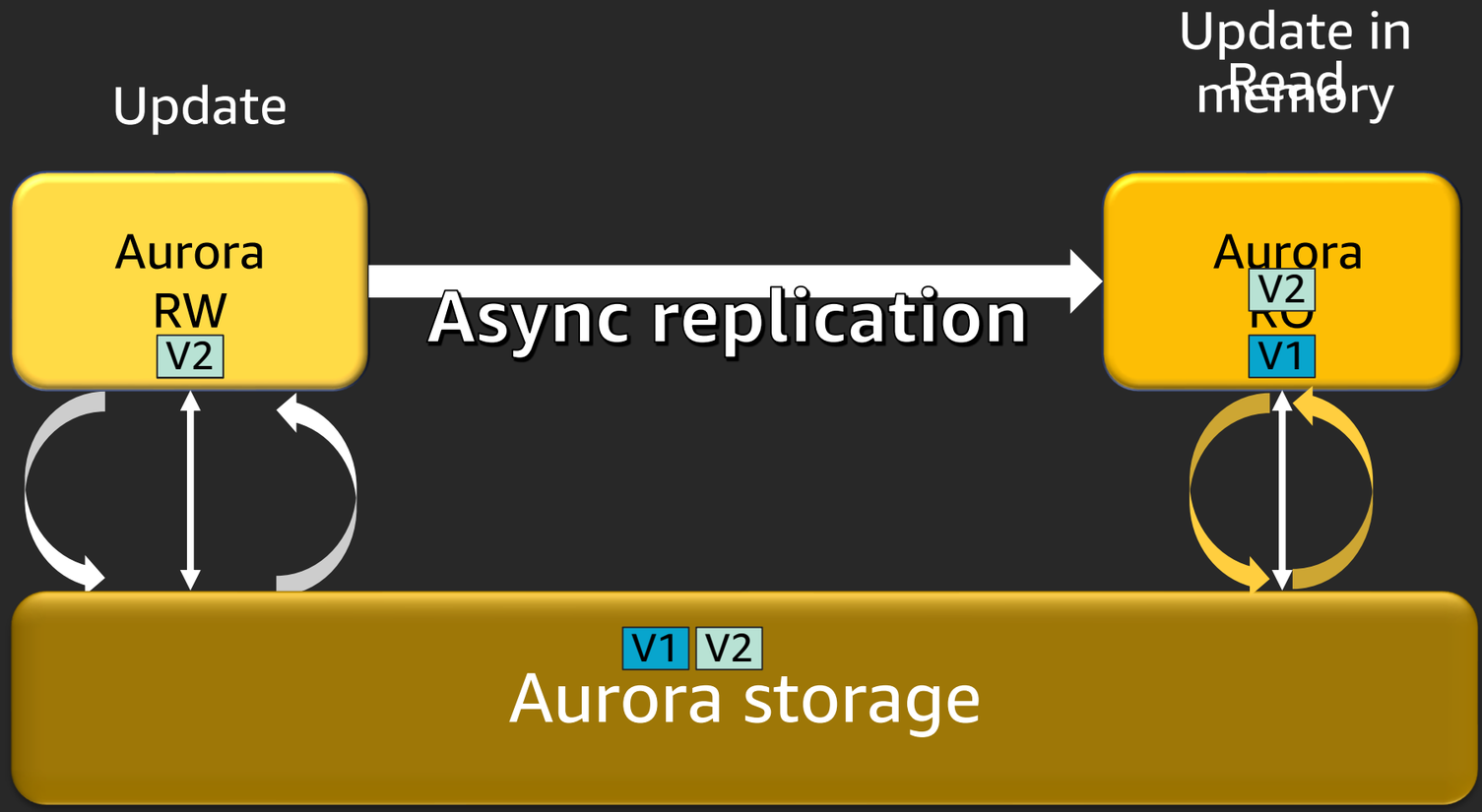


Replicas and clones

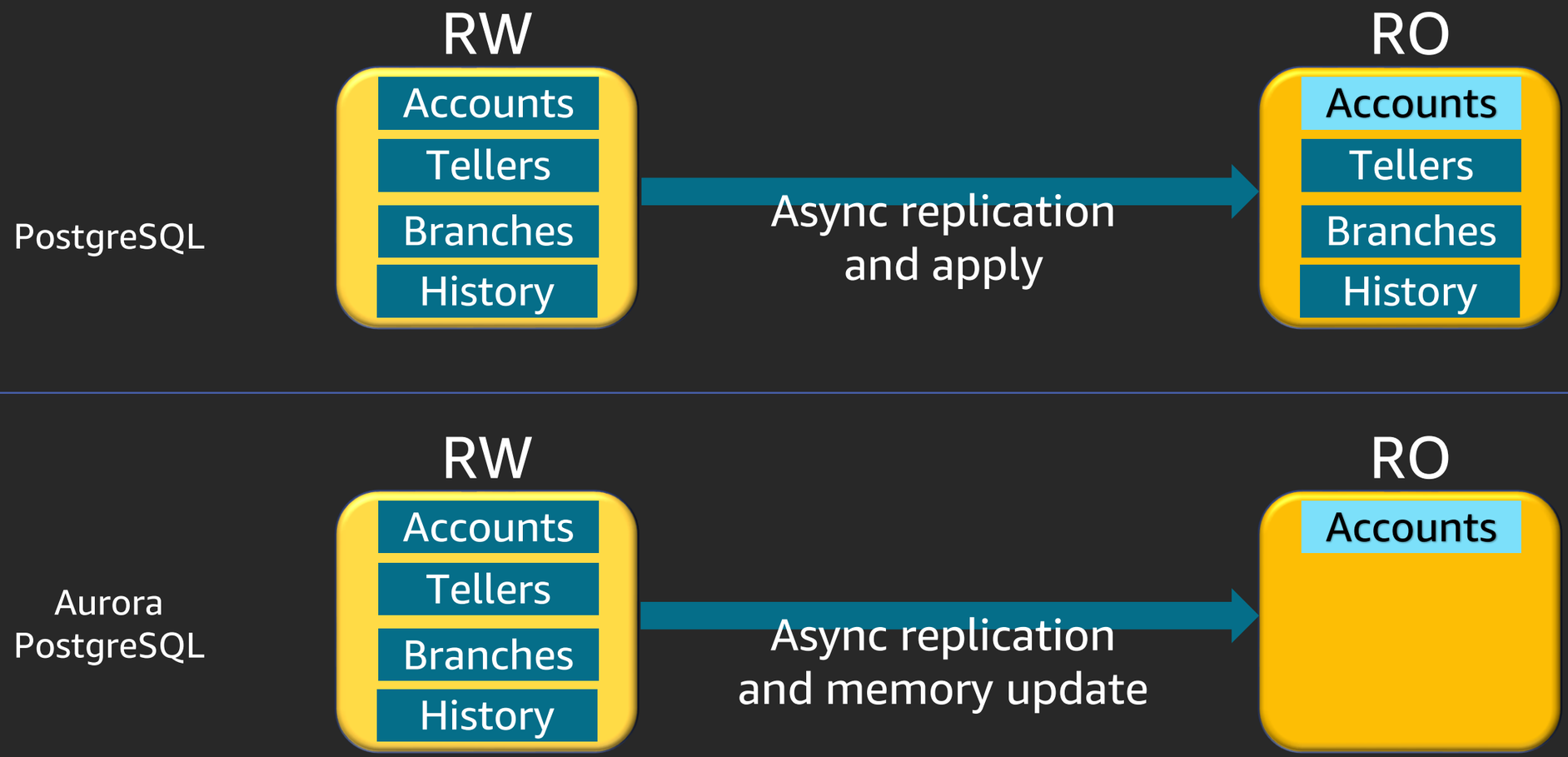
Replicas: PostgreSQL



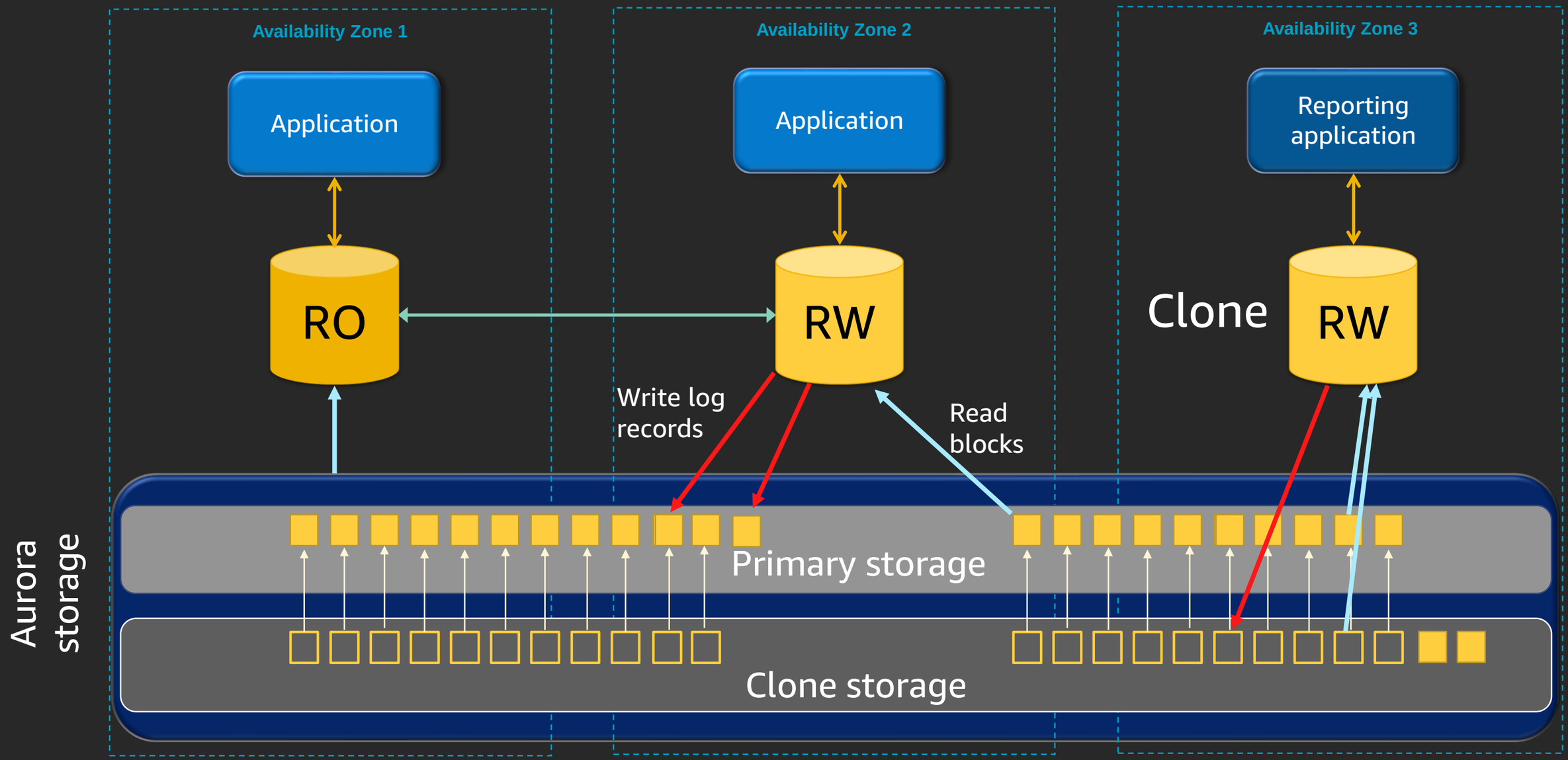
Replicas: Amazon Aurora



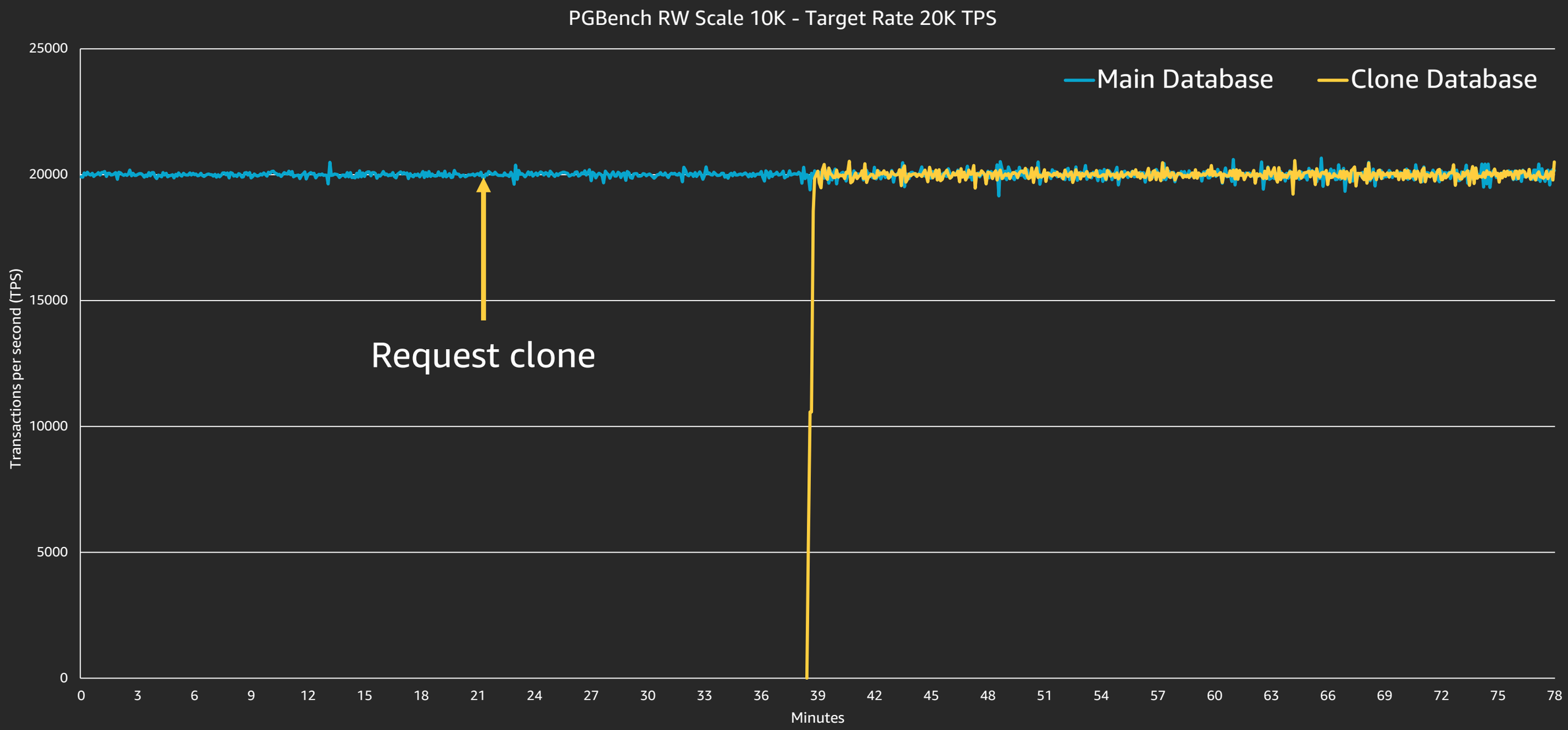
pgbench benchmark



Fast clones


Availability Zone 1
Availability Zone 2
Availability Zone 3
Application
Application
Reporting application
RO
RW
Clone
RW
Write log records
Read blocks
Aurora storage
Primary storage
Clone storage

Fast clone example



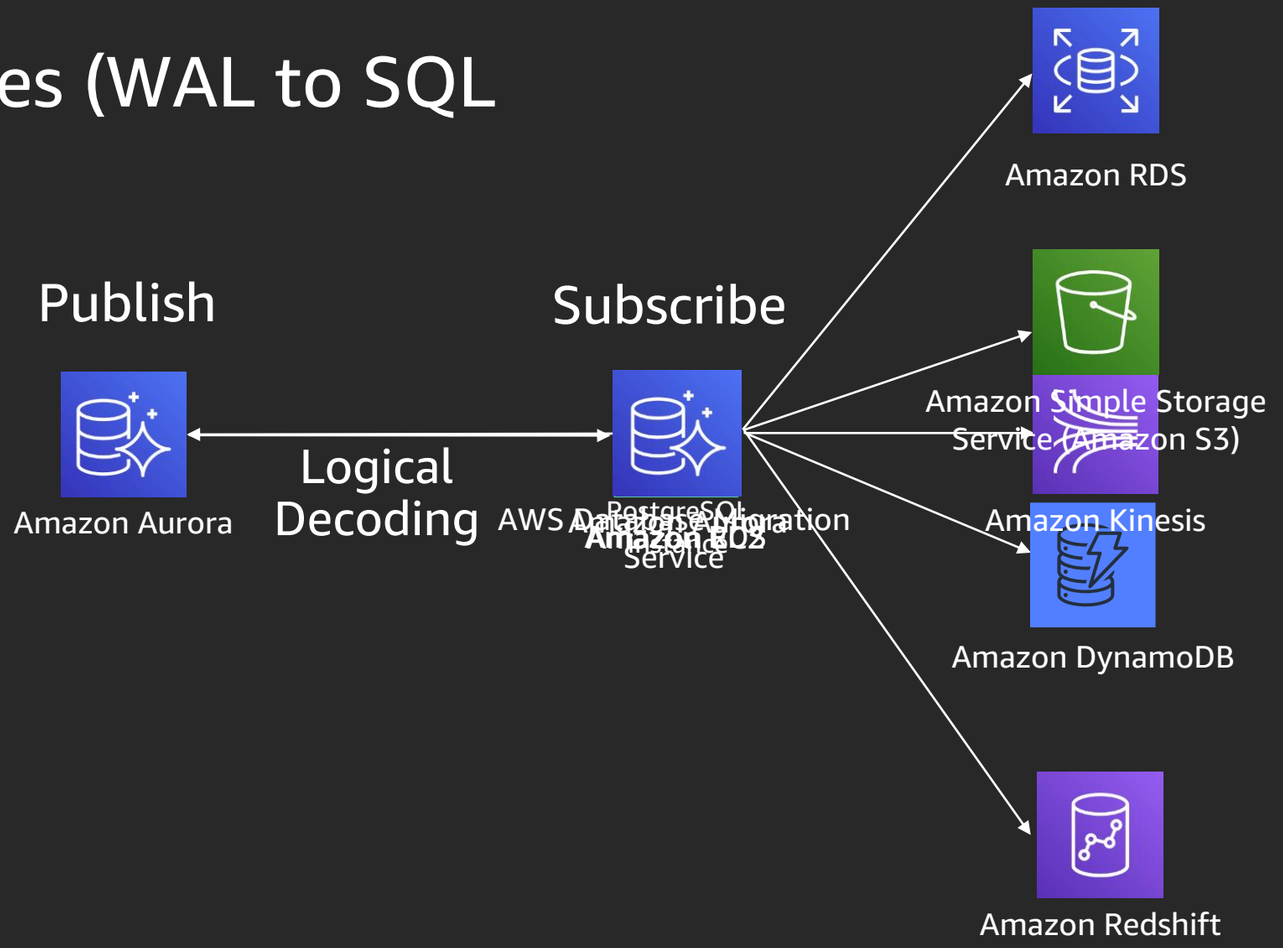
Replication

Logical replication support

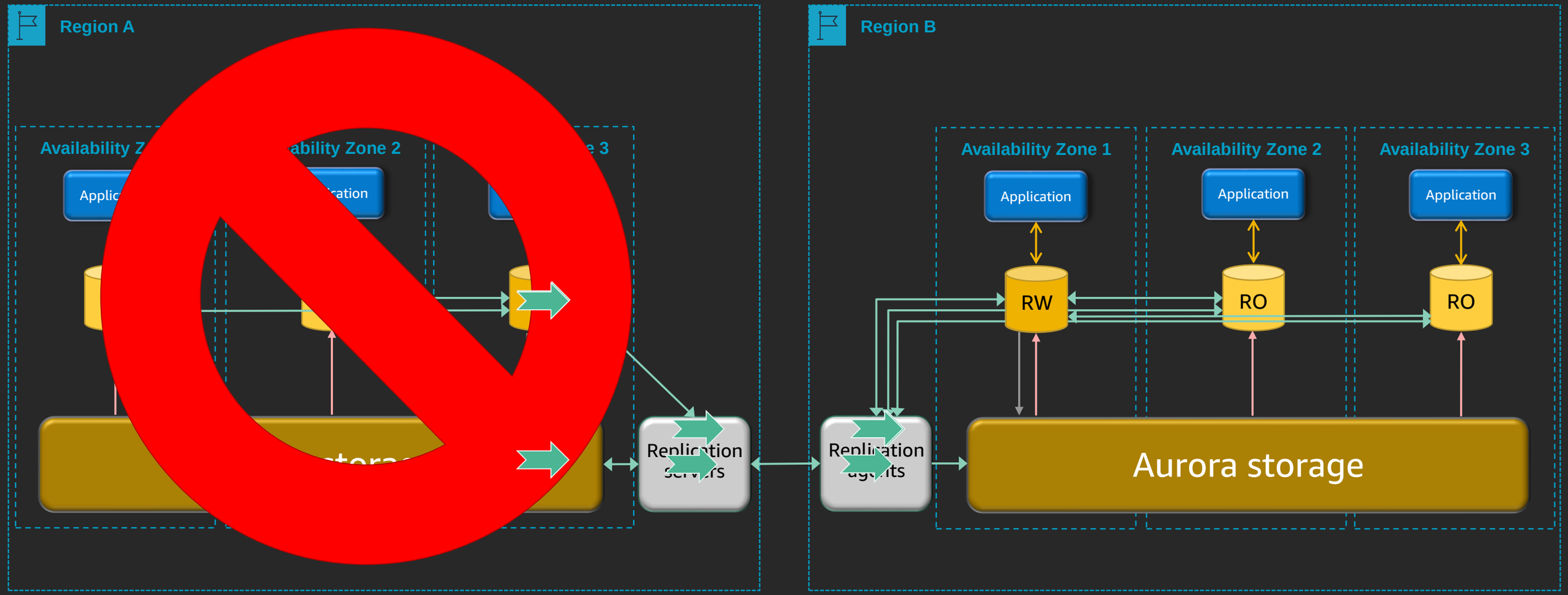
Converts physical changes (WAL to SQL statements (i.e., logical)

- Logical decoding plugin (test_decoder, decoder_raw, wal2jason)

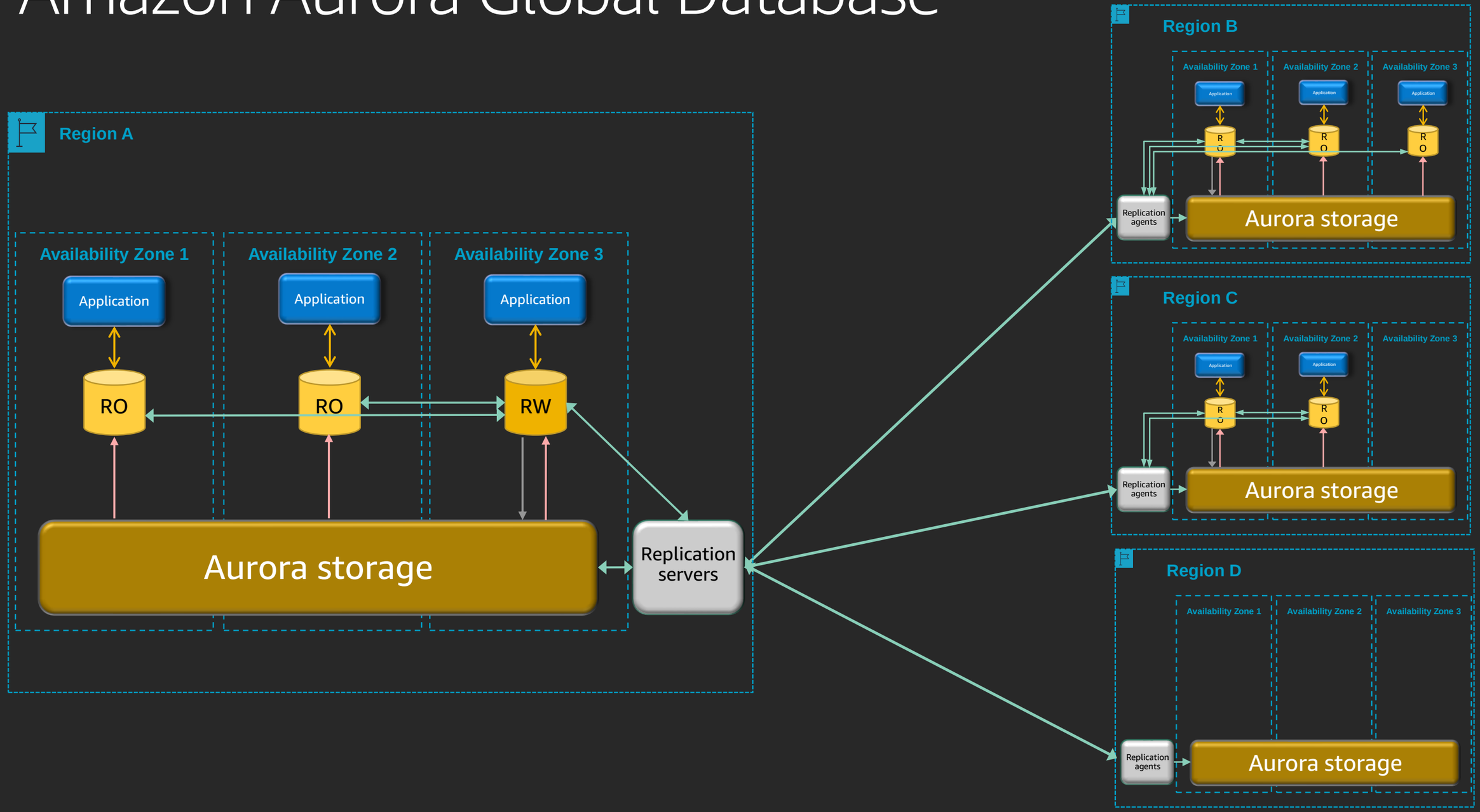
- V10/11: Publish/subscribe to another PostgreSQL instance



Amazon Aurora Global Database

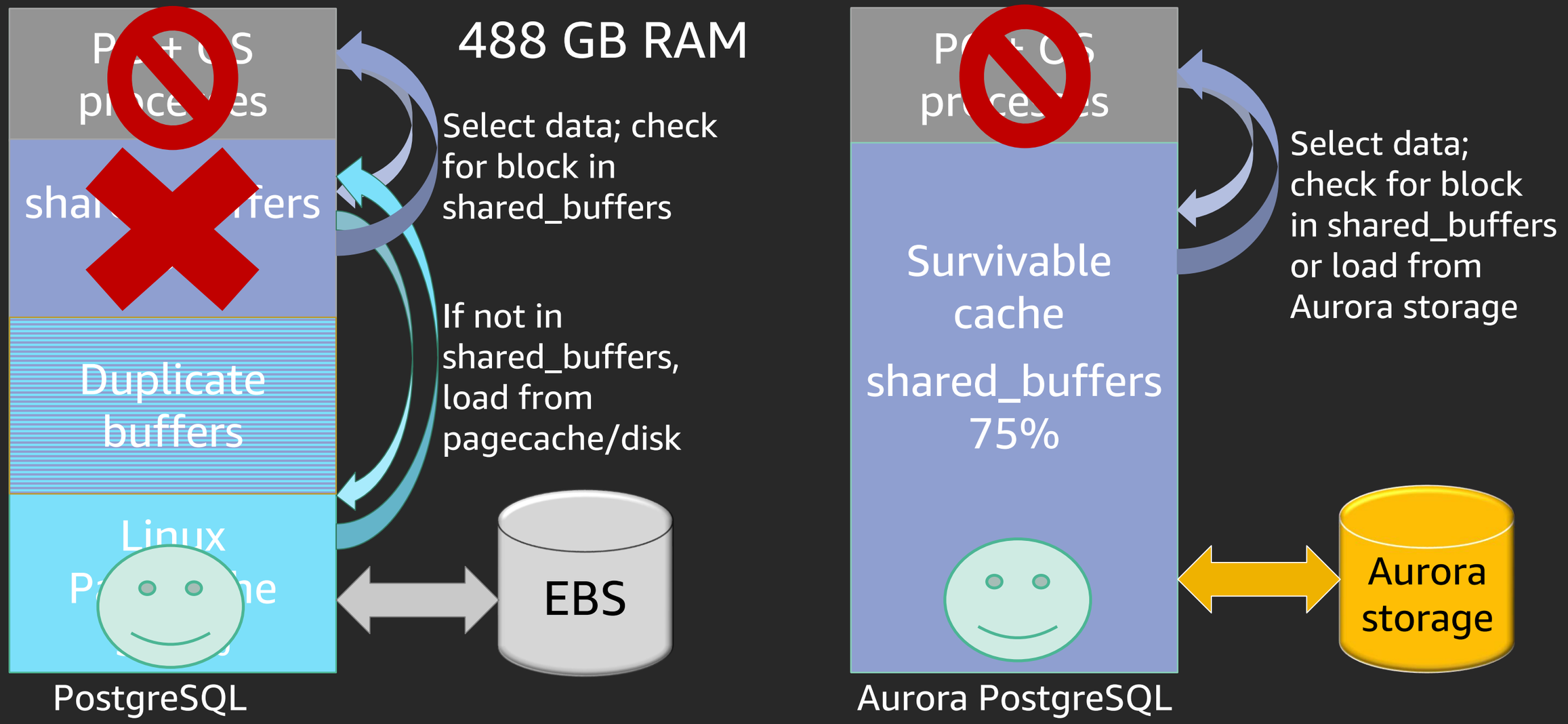


Amazon Aurora Global Database

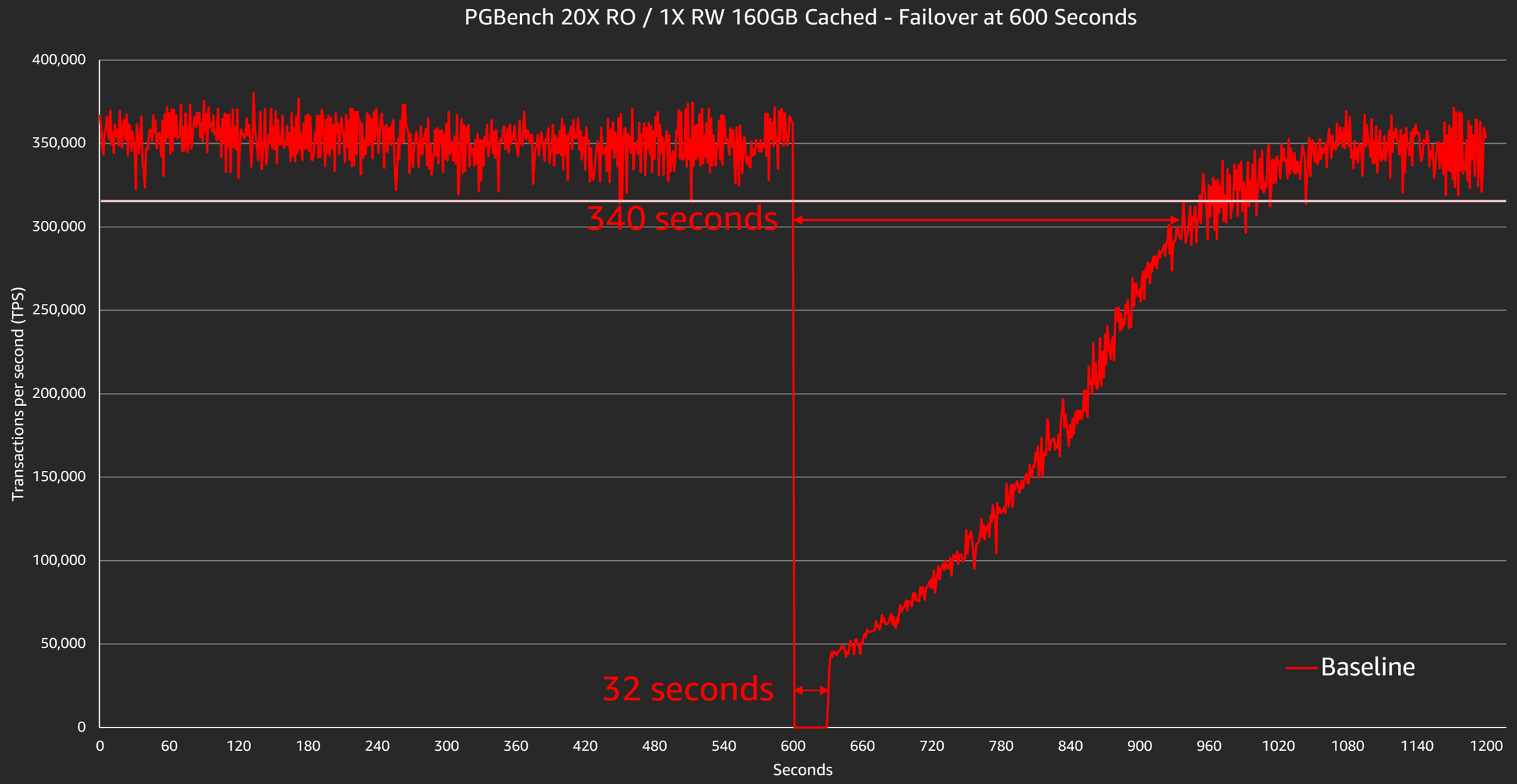


Caching

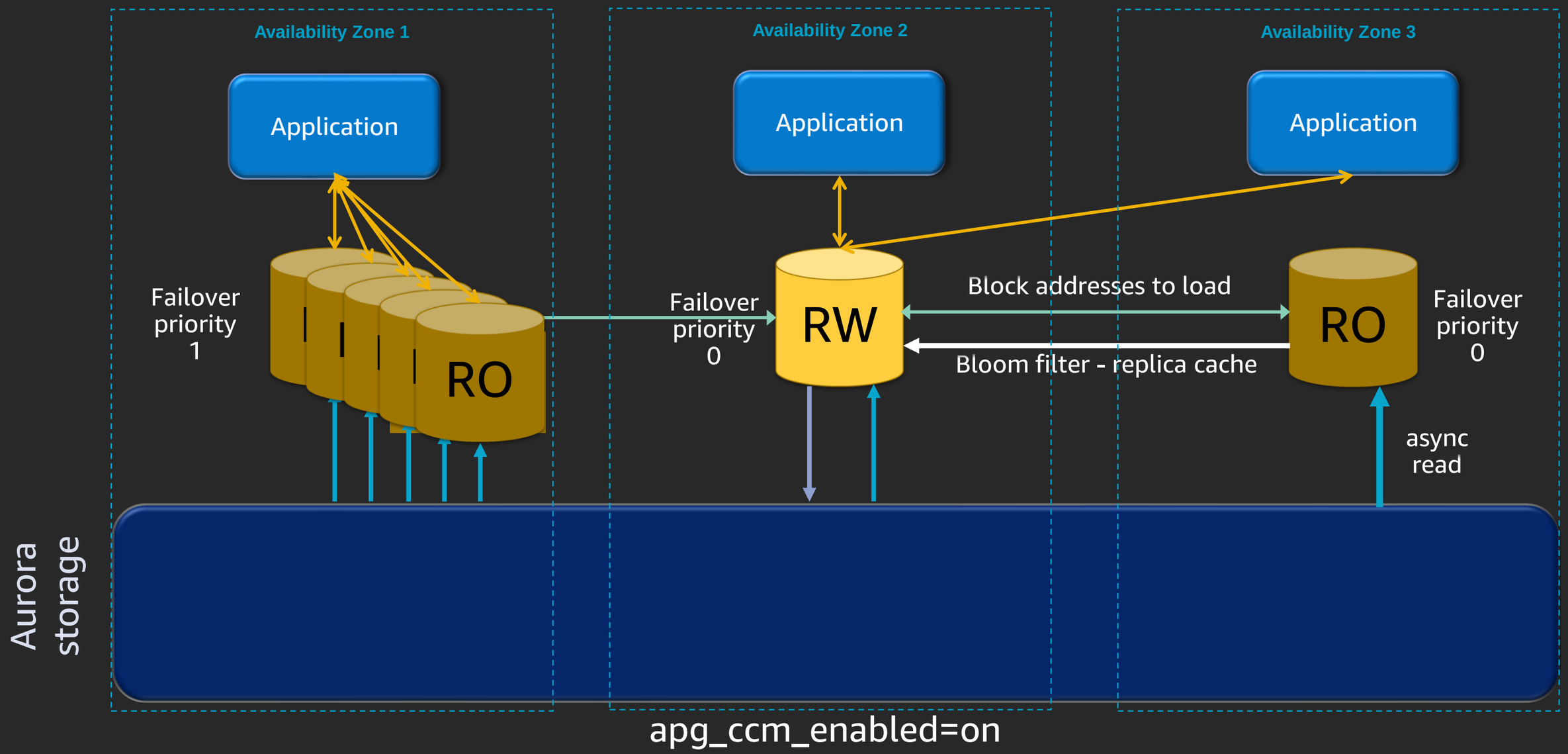
Caching changes: No double buffering



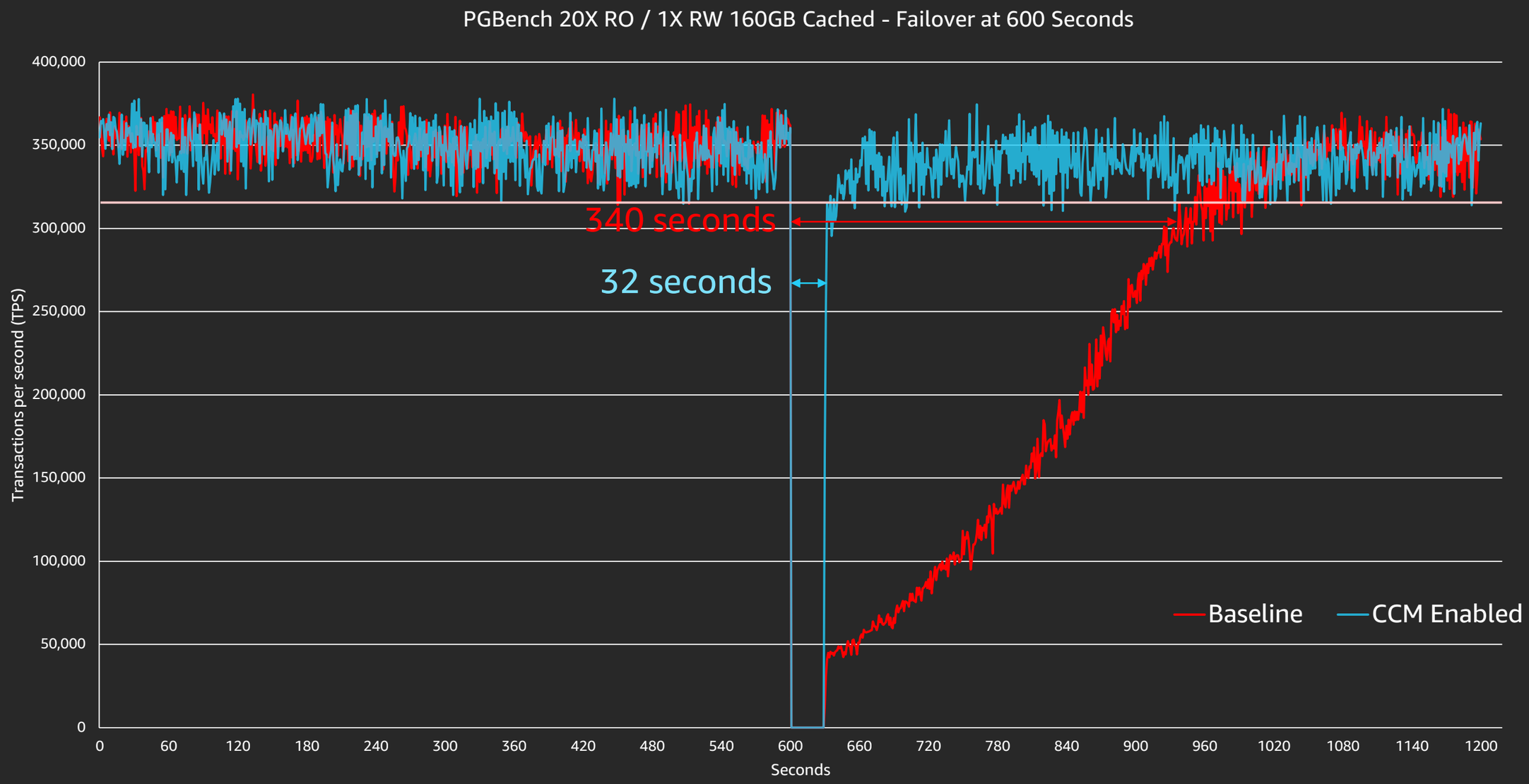
Cluster Cache Management (CCM): Failover



CCM feature


Availability Zone 1
Availability Zone 2
Availability Zone 3
Application
Application
Application
Failover priority 1
RO
Failover priority 0
RW
Block addresses to load
Bloom filter - replica cache
RO
Failover priority 0
async read
Aurora storage
apg_ccm_enabled=on

CCM


PGBench 20X RO / 1X RW 160GB Cached - Failover at 600 Seconds
Transactions per second (TPS)
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
0
60
120
180
240
300
360
420
480
540
600
660
720
780
840
900
960
1020
1080
1140
1200
Seconds
340 seconds
32 seconds
Baseline
CCM Enabled

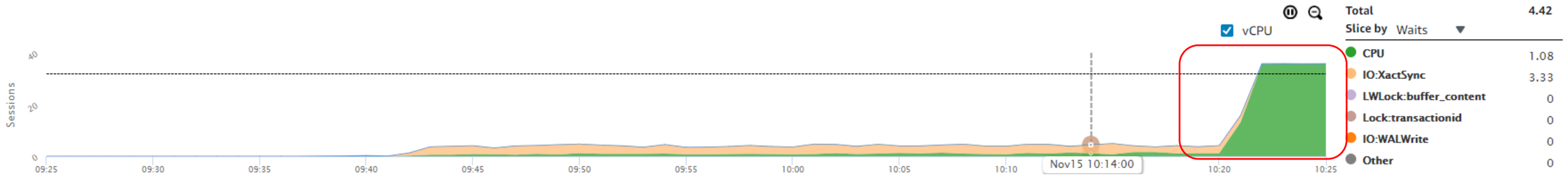
Performance

Performance Insights

Database load

Average active sessions (AAS)

5m | 1h | 5h | 24h | 1w | All

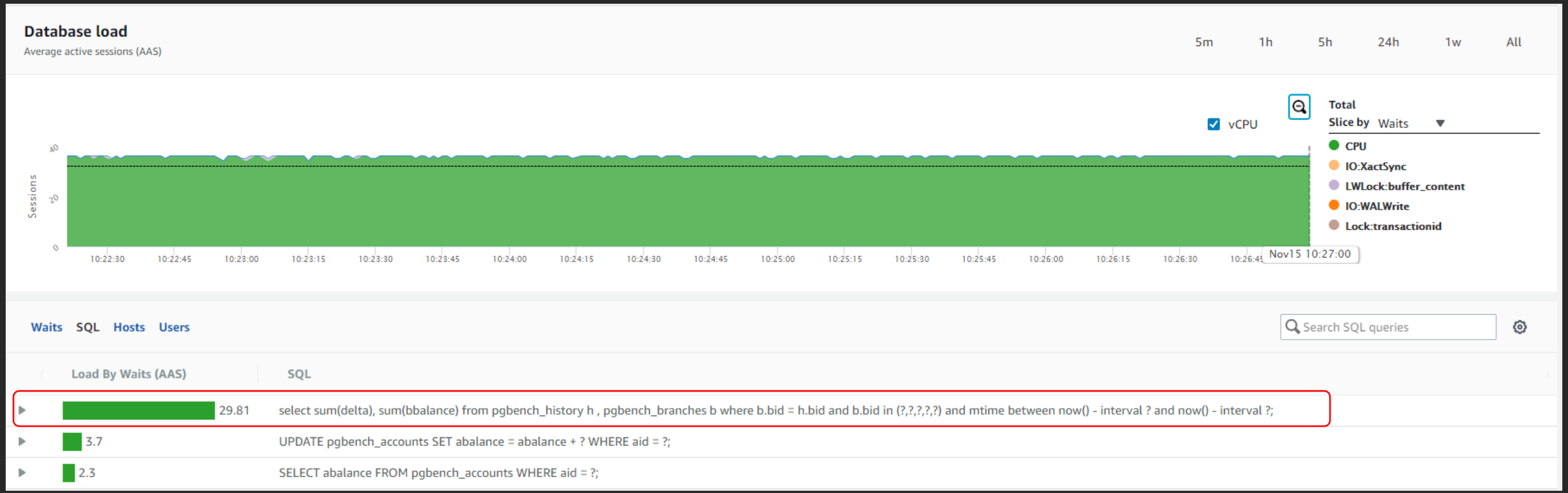


Waits | SQL | Hosts | Users

Search SQL queries

Load By Waits (AAS)	SQL
2.52	select sum(delta), sum(bbalance) from pgbench_history h , pgbench_branches b where b.bid = h.bid and b.bid in (?,?,?,?,?) and mtime between now() - interval ? and now() - interval ?;
2.03	END;
0.41	UPDATE pgbench_accounts SET abalance = abalance + ? WHERE aid = ?;
0.17	SELECT abalance FROM pgbench_accounts WHERE aid = ?;
5.7e-2	UPDATE pgbench_branches SET bbalance = bbalance + ? WHERE bid = ?;
5.3e-2	UPDATE pgbench_tellers SET tbalance = tbalance + ? WHERE tid = ?;
3.1e-2	select sum(delta), sum(bbalance) from pgbench_history h , pgbench_branches b where b.bid = h.bid and b.bid in (?,?,?,?,?) and mtime between now() - interval ? and now() - interval ?;
2.3e-2	UPDATE pgbench_accounts SET abalance = abalance + ? WHERE aid = ?;
2e-2	UPDATE pgbench_tellers SET tbalance = tbalance + ? WHERE tid = ?;
2e-2	UPDATE pgbench_branches SET bbalance = bbalance + ? WHERE bid = ?;

Performance Insights



Performance Insights

Plan change

Before

```
Aggregate  (cost=3804.15..3804.16 rows=1 width=16)
   ->  Nested Loop  (cost=12.67..3802.61 rows=307 width=8)
         ->  Index Scan using pgbench_branches_pkey on pgbench_branches b  (cost=0.29..16.60 rows=2 width=8)
               Index Cond: (bid = ANY ('{1,4}'::integer[]))
         ->  Bitmap Heap Scan on pgbench_history h  (cost=12.39..1891.47 rows=154 width=8)
               Recheck Cond: (bid = b.bid)
               Filter: ((mtime >= (now() - '01:00:00'::interval)) AND (mtime <= (now() - '00:30:00'::interval)))
               ->  Bitmap Index Scan on i_p_bid  (cost=0.00..12.35 rows=522 width=0)
                     Index Cond: (bid = b.bid)
```

- Stats change?
- Config change?
- Index change?

After

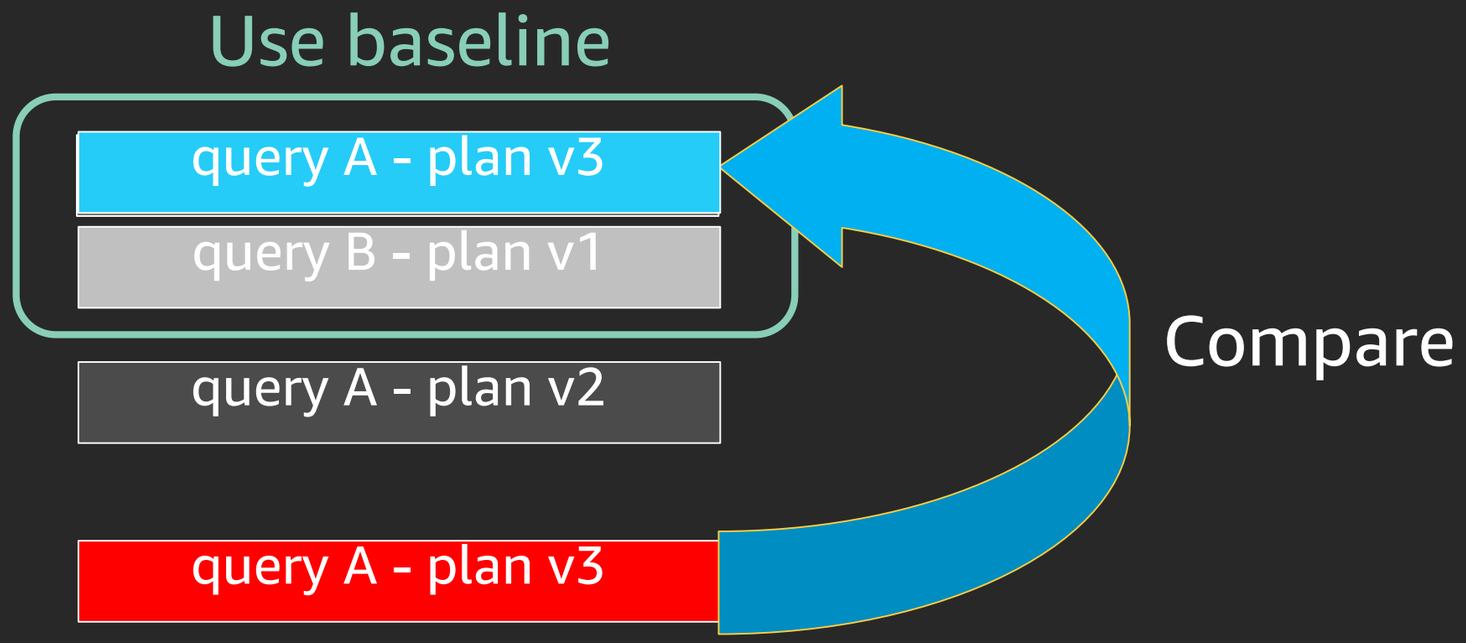
```
 Aggregate  (cost=171092.96..171092.97 rows=1 width=16)
   ->  Hash Join  (cost=329.02..171091.42 rows=307 width=8)
         Hash Cond: (h.bid = b.bid)
         ->  Seq Scan on pgbench_history h  (cost=0.00..166712.20 rows=1542280 width=8)
               Filter: ((mtime >= (now() - '01:00:00'::interval)) AND (mtime <= (now() - '00:30:00'::interval)))
         ->  Hash  (cost=329.00..329.00 rows=2 width=8)
               ->  Seq Scan on pgbench_branches b  (cost=0.00..329.00 rows=2 width=8)
                     Filter: (bid = ANY ('{1,4}'::integer[]))
```

- enable_bitmapscan=off
- enable_indexscan=off

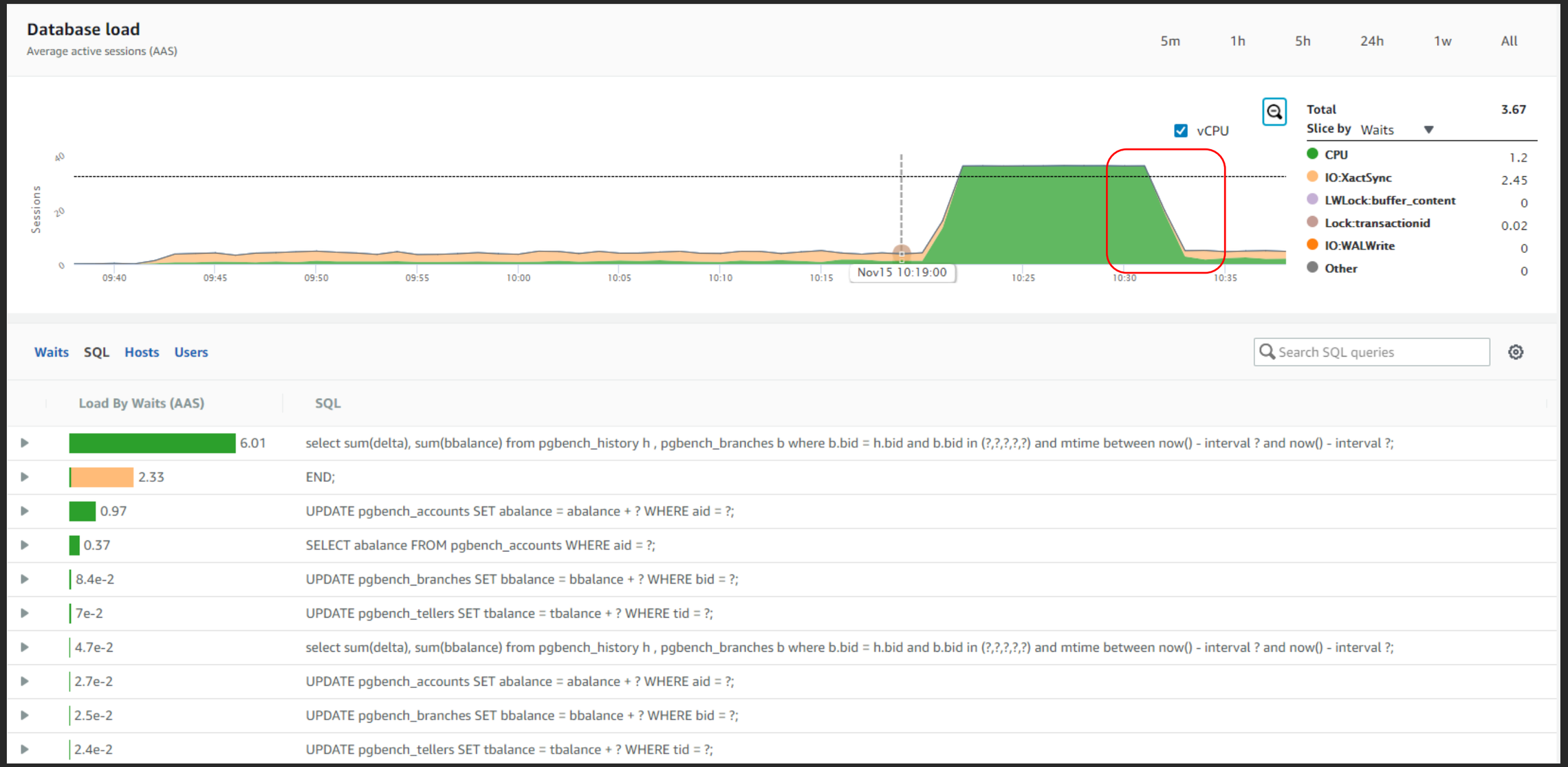
Query Plan Management (QPM)

- Capture statements
- Approve statements

- Evolve better plans



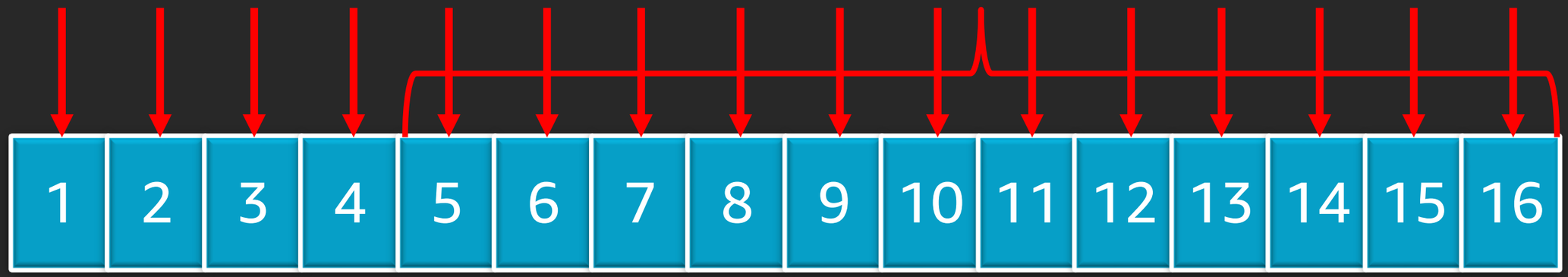
QPM: Use plan baselines



Performance: prefetch

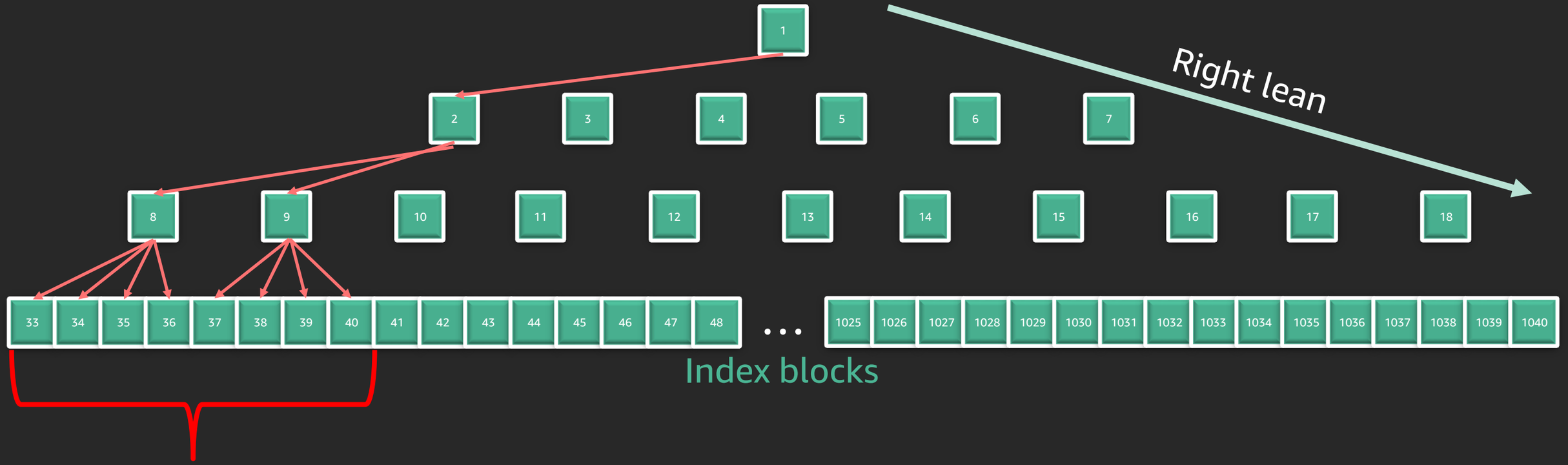
Basic heap prefetch

Table scan

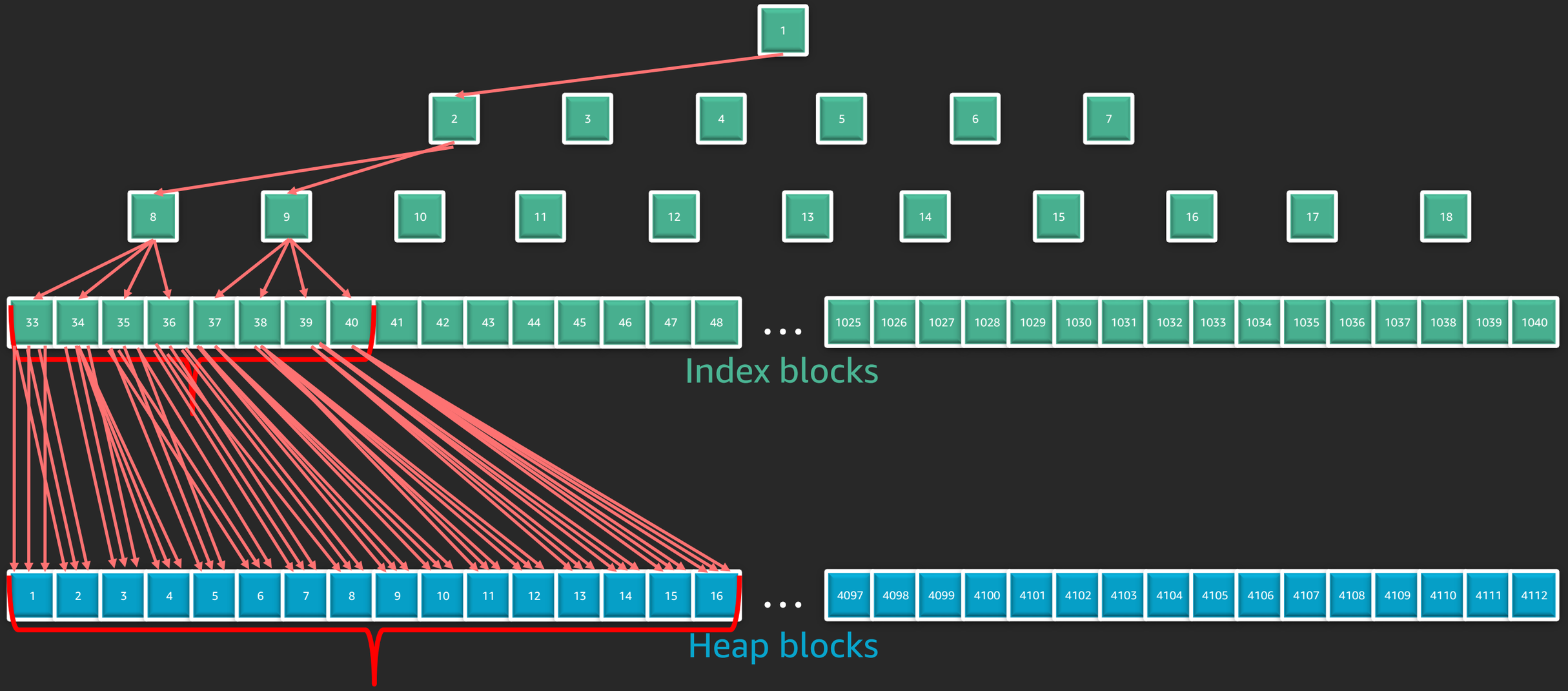


8K heap blocks

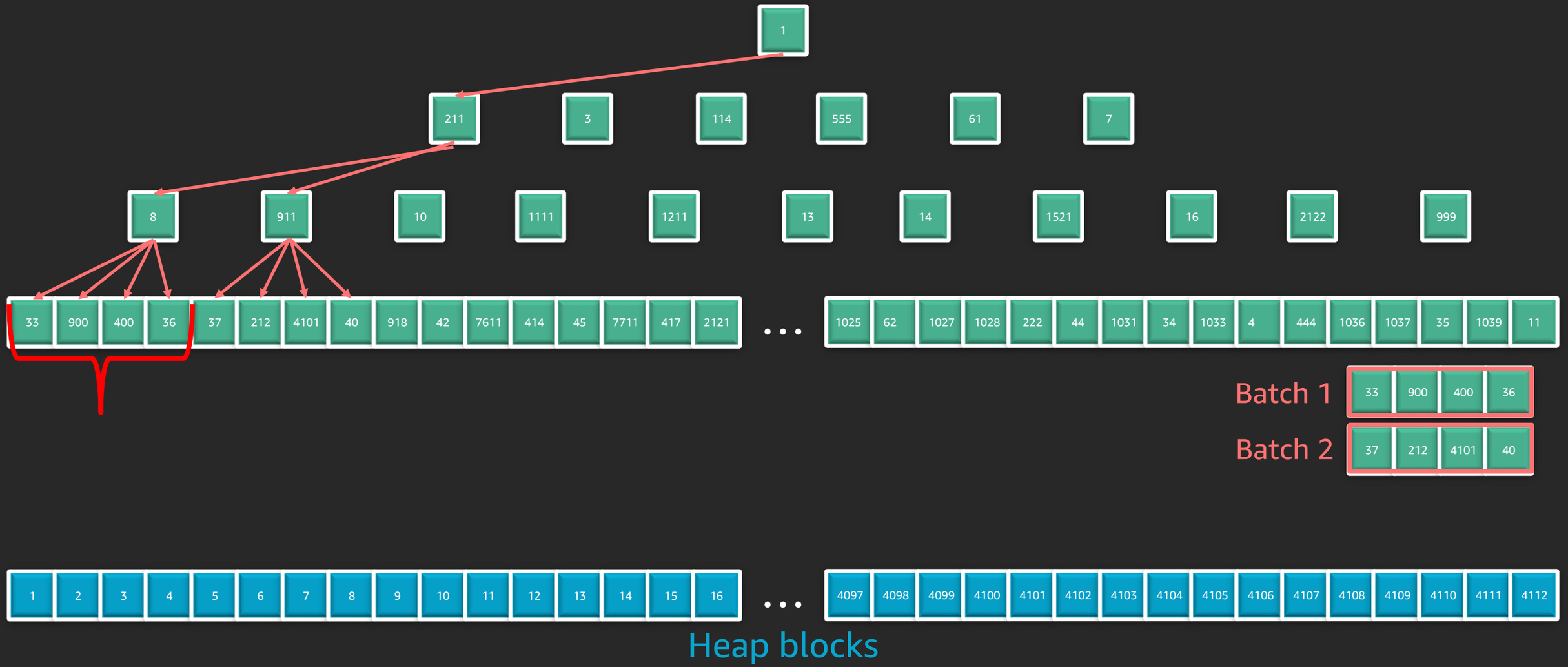
Index-only range scan: Basic prefetch



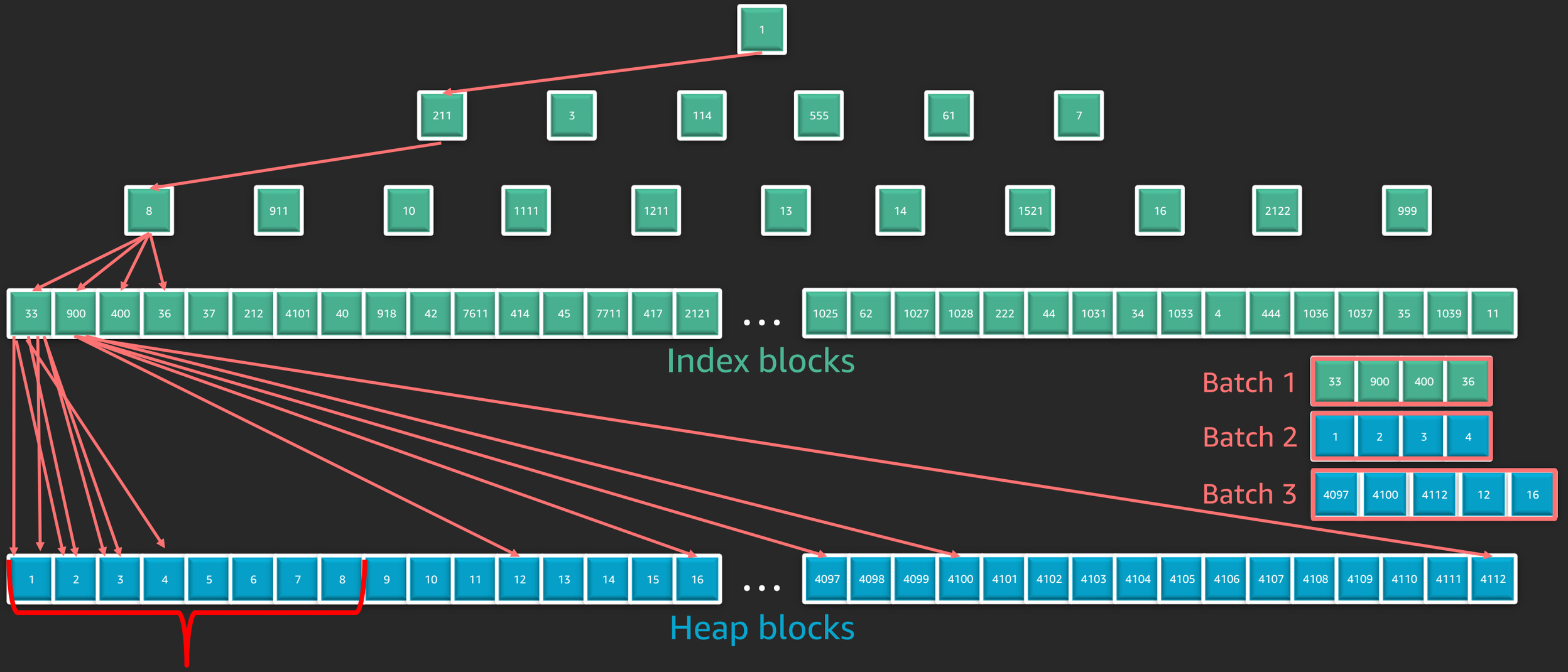
Index range scan: Basic prefetch



Index-only range scan: Intelligent prefetch



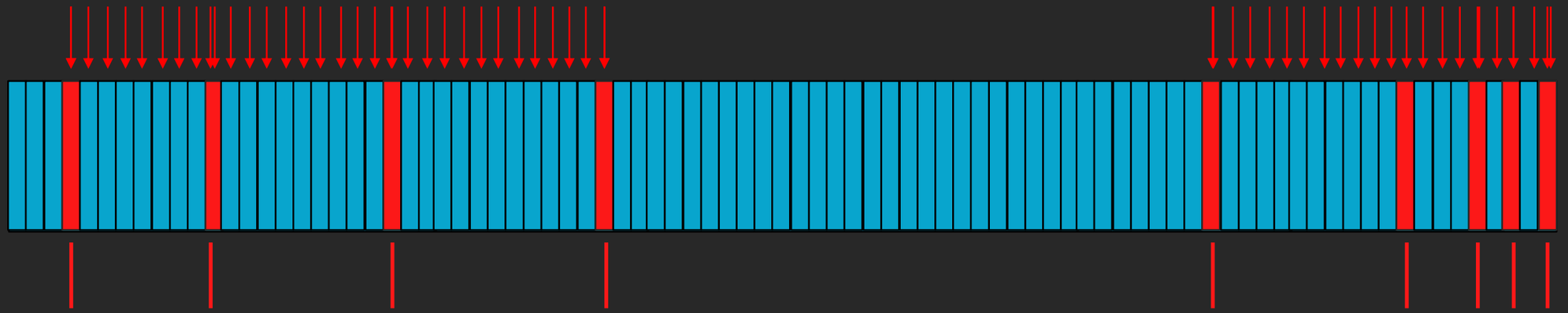
Index range scan: Intelligent prefetch



Intelligent vacuum prefetch

PostgreSQL 402 seconds

Visibility & frozen map



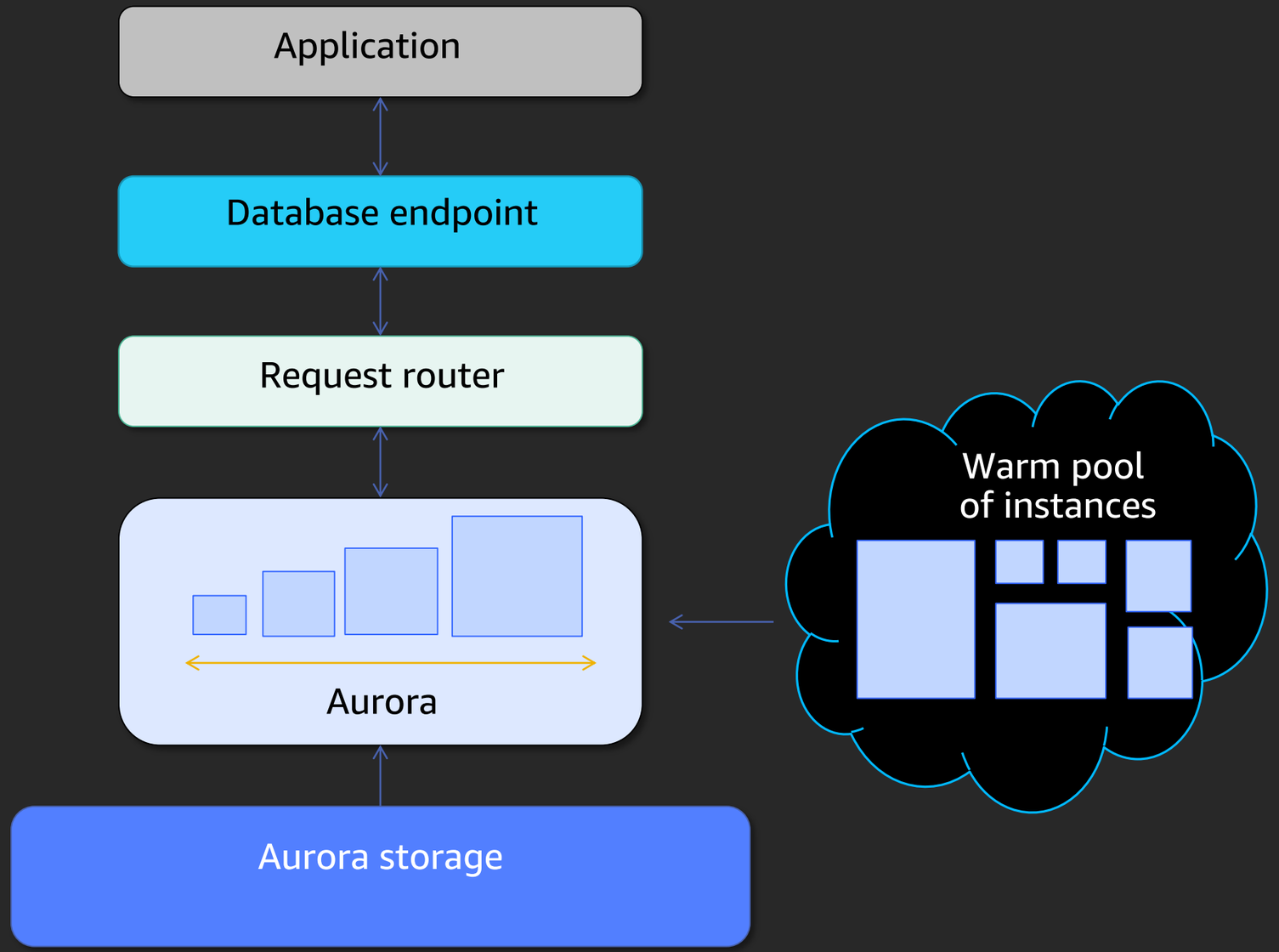
Aurora PostgreSQL 163 seconds

Submit batch I/O up to 256 blocks

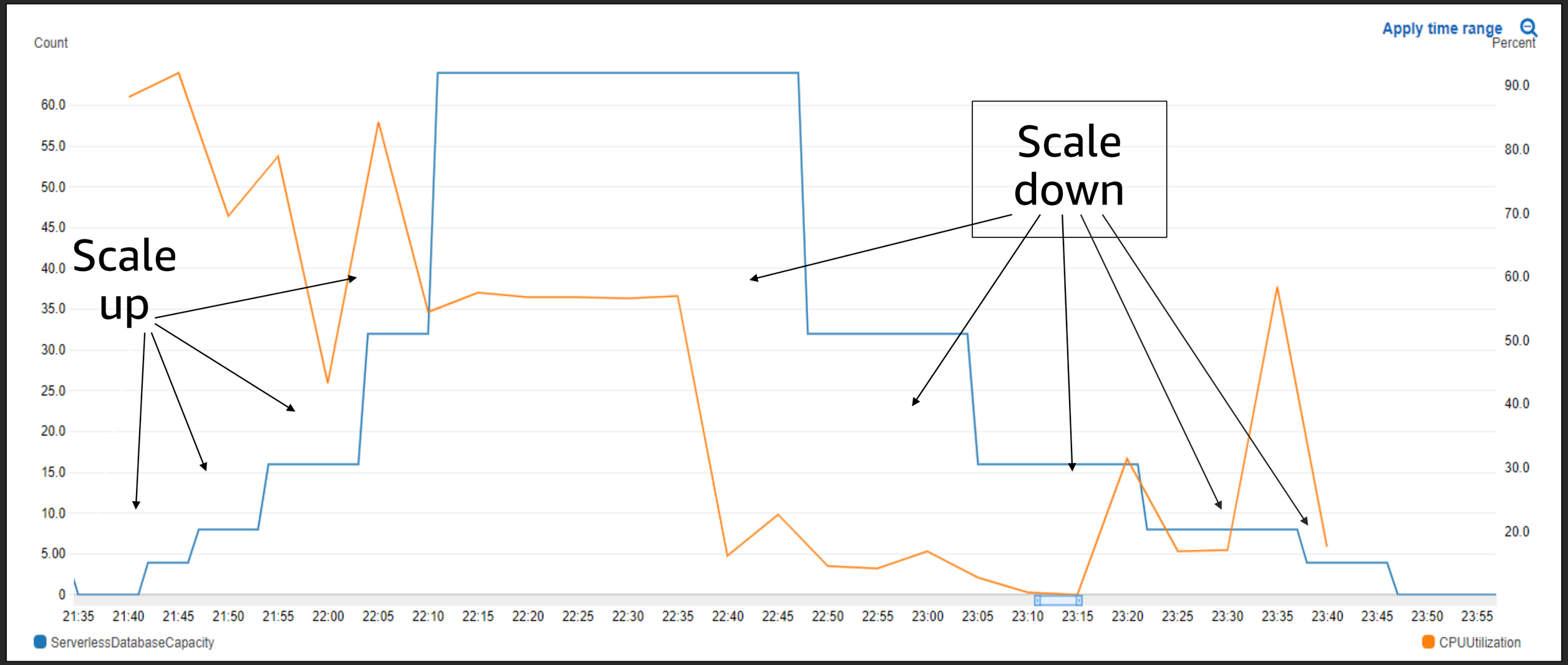
Serverless

Aurora Serverless

- Starts up on demand; shuts down when not in use
- Scales up/down automatically
- No application impact when scaling
- Pay per second, 1-minute minimum
- Data API



Scaling up & down



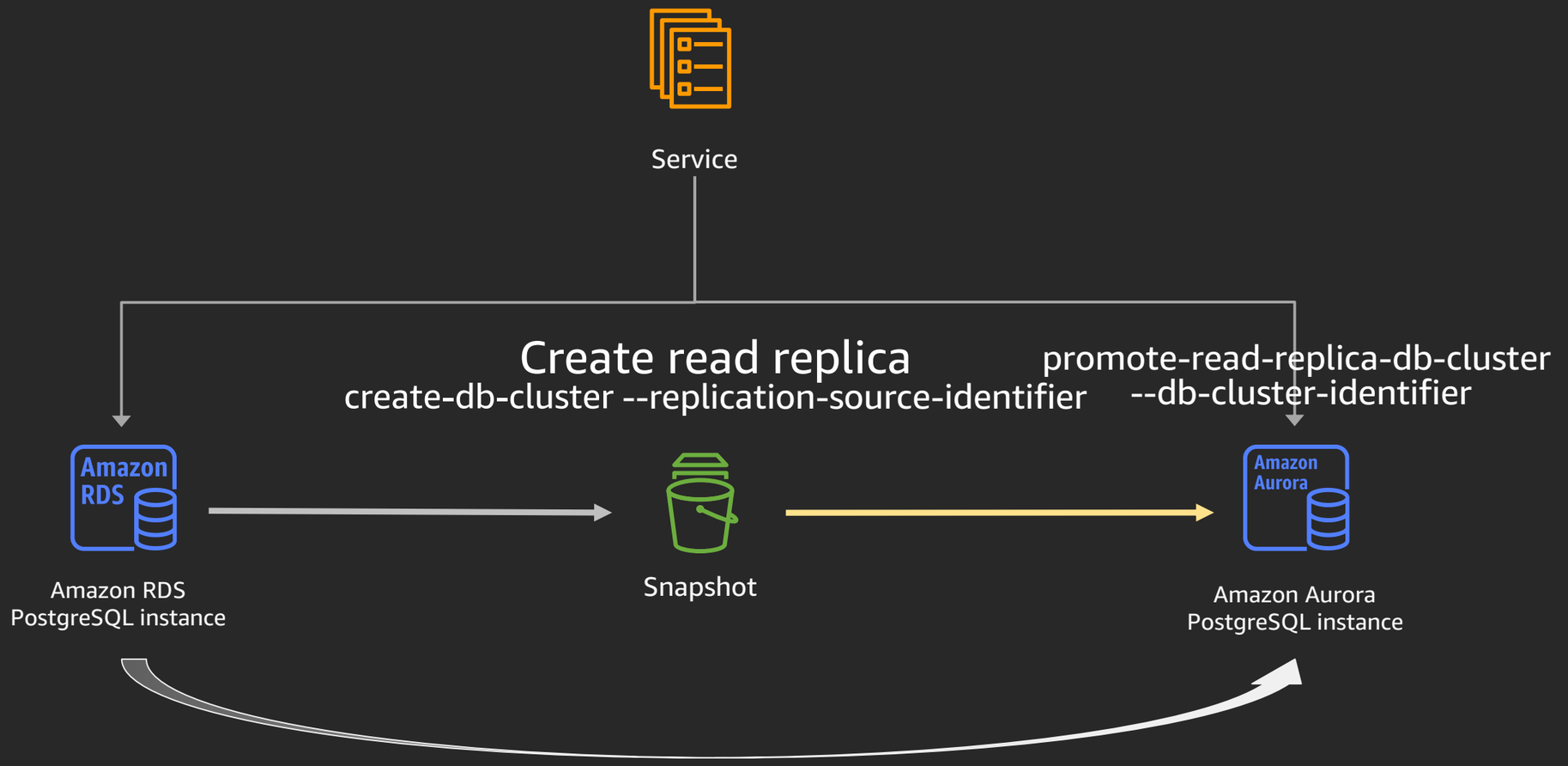
Migration

Migration to Aurora PostgreSQL

Methods

- PostgreSQL: pg_dump/pg_restore
- AWS Database Migration Service (AWS DMS)
- Amazon RDS PostgreSQL: Snapshot import
- Amazon RDS PostgreSQL: Read replica

Migration: Read replica

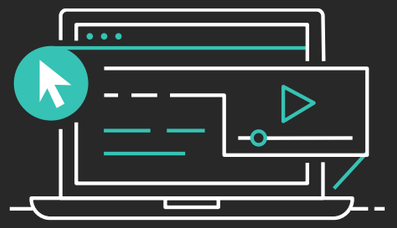


Catchup via PostgreSQL asynchronous replication

Training

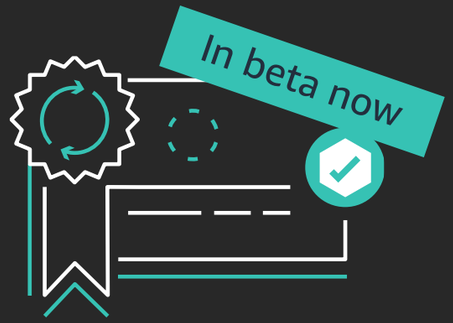
Learn databases with AWS Training and Certification

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

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.