



AWS
re:Invent

ANT416-R1

Performance and elasticity in Amazon Redshift

Thanos Papathanasiou

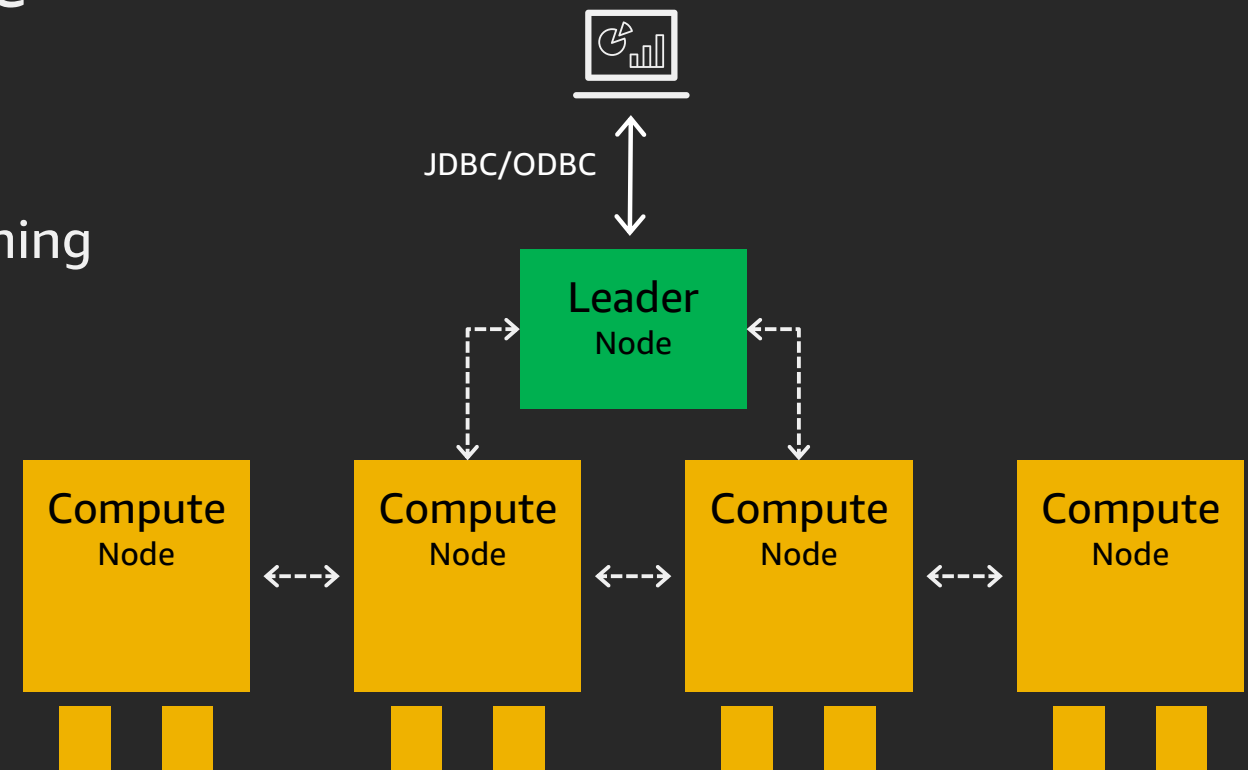
Principal Engineer
Amazon Web Services

Yuval Pemper

SVP Engineering
Innovid

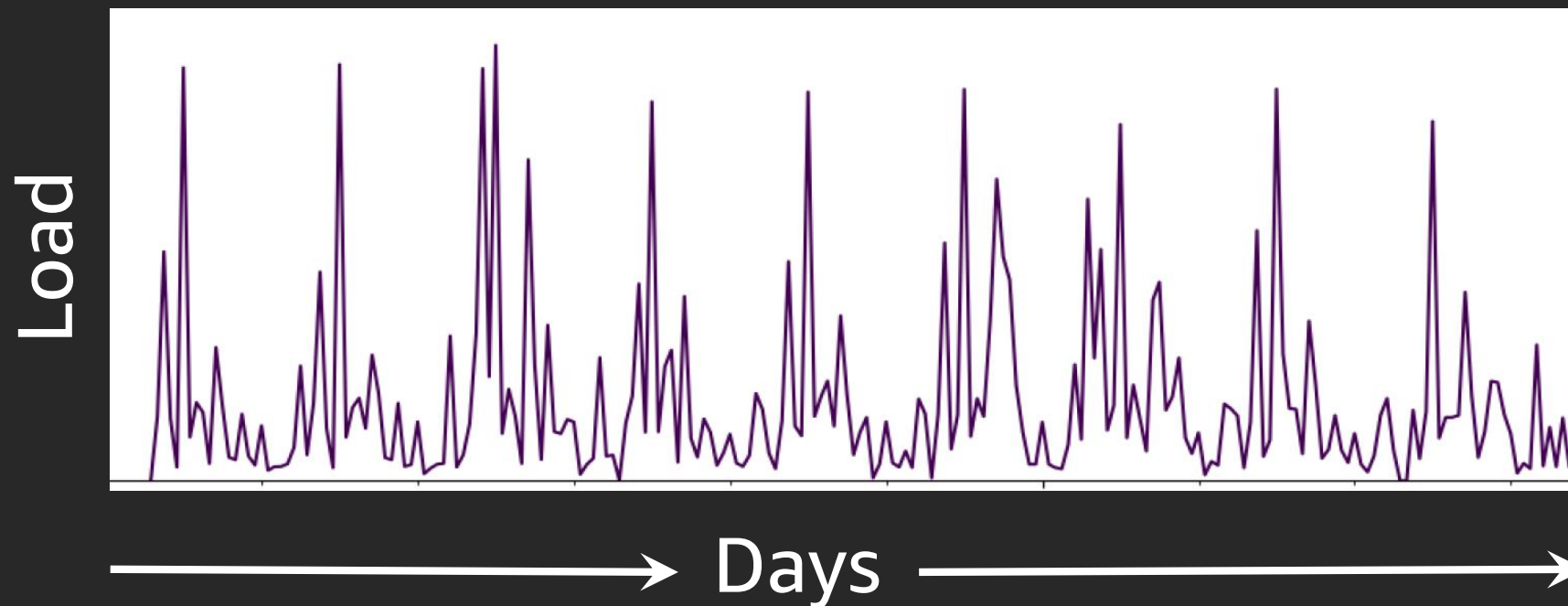
Amazon Redshift - Architecture

- Cloud-native distributed data warehouse
- Leader and compute nodes
 - Leader node manages connections, catalog, and planning
 - Compute node manages data and execution
- Distributed query processing
 - Table data partitioned across compute nodes
 - Table accesses are decentralized



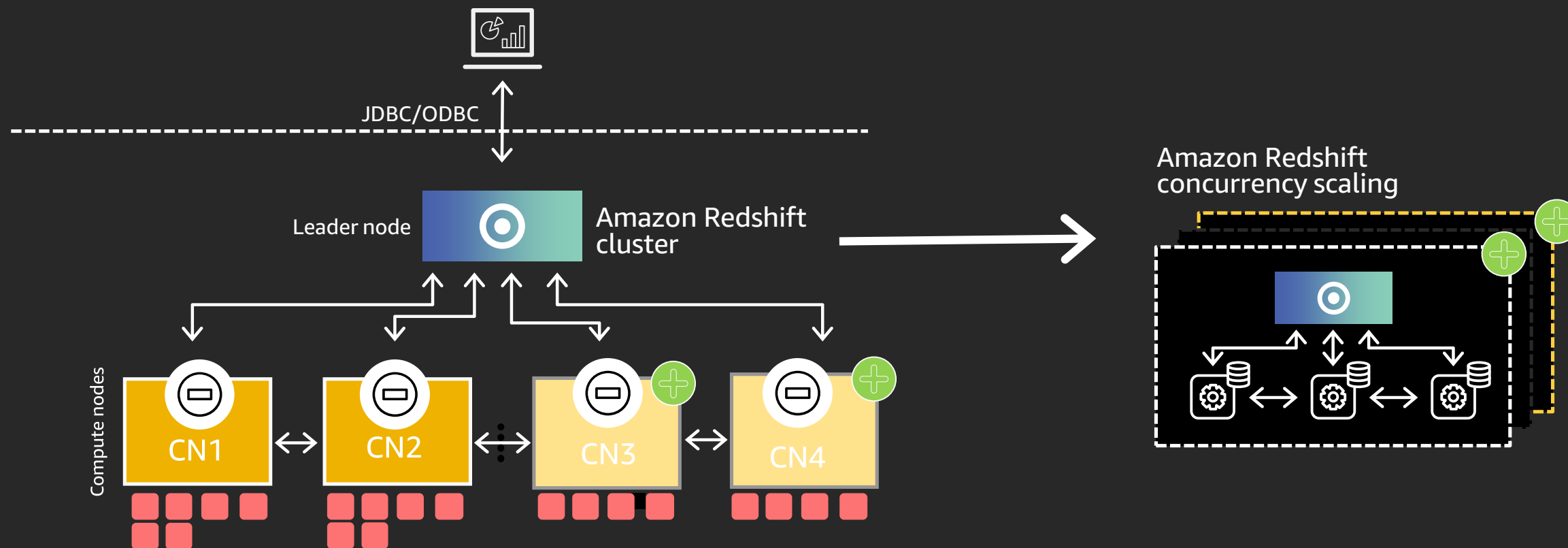
Need for Elasticity

- Non-uniform and unpredictable workload pattern
 - Example: Significant difference between peak and average usage
 - Increased costs if provisioning for peak usage
- Compute and storage needs vary independently
 - Compute scaling: To handle high query concurrency
 - Storage scaling: Retain (cold) data to comply with retention policies and auditing requirements
 - Decouple storage from compute and scale independently
- Users pay only for what they need



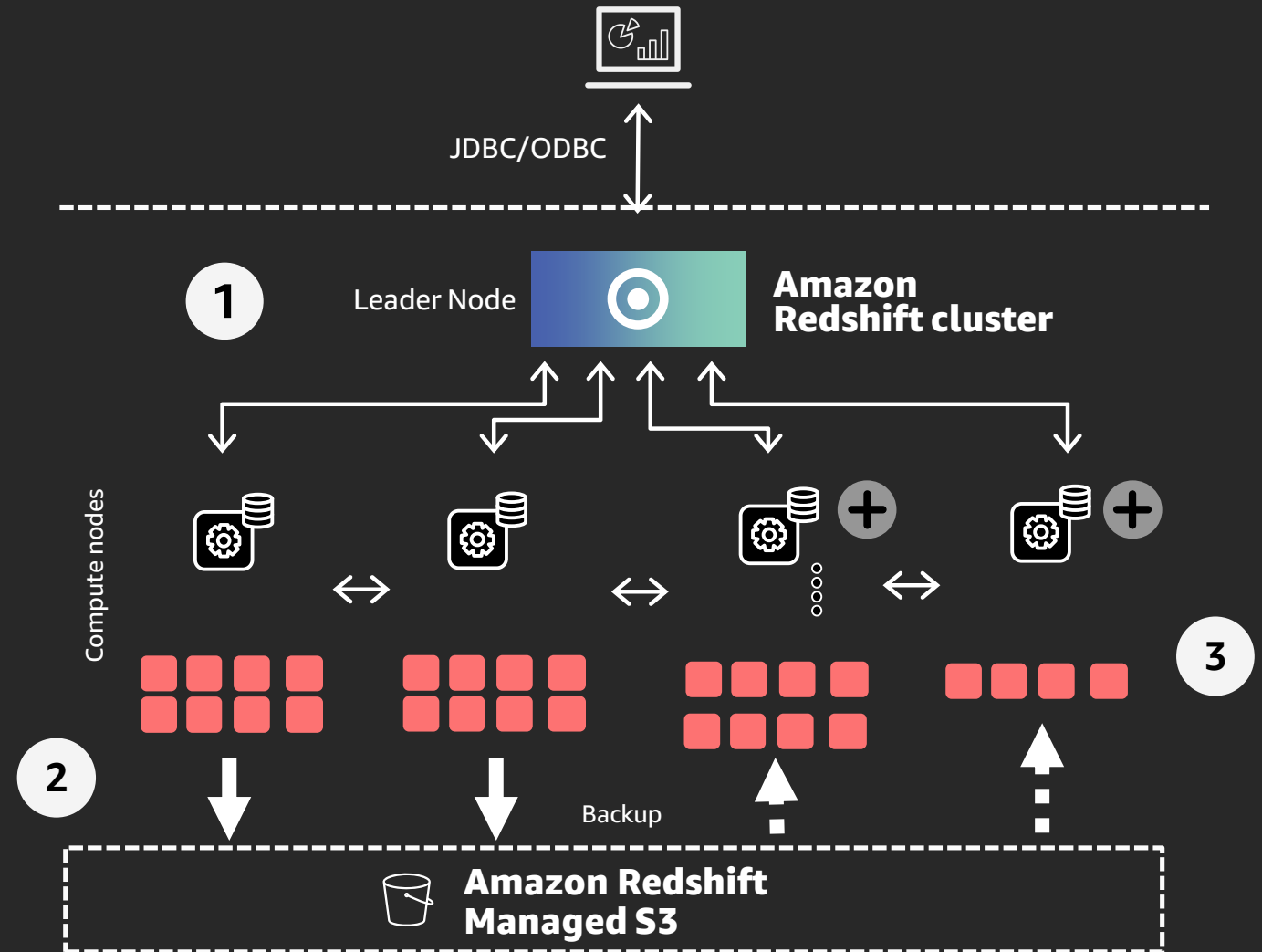
Two forms of **compute elasticity**

- Elastic resize: Scale the number of nodes up or down
- Concurrency scaling: Scale out to multiple clusters



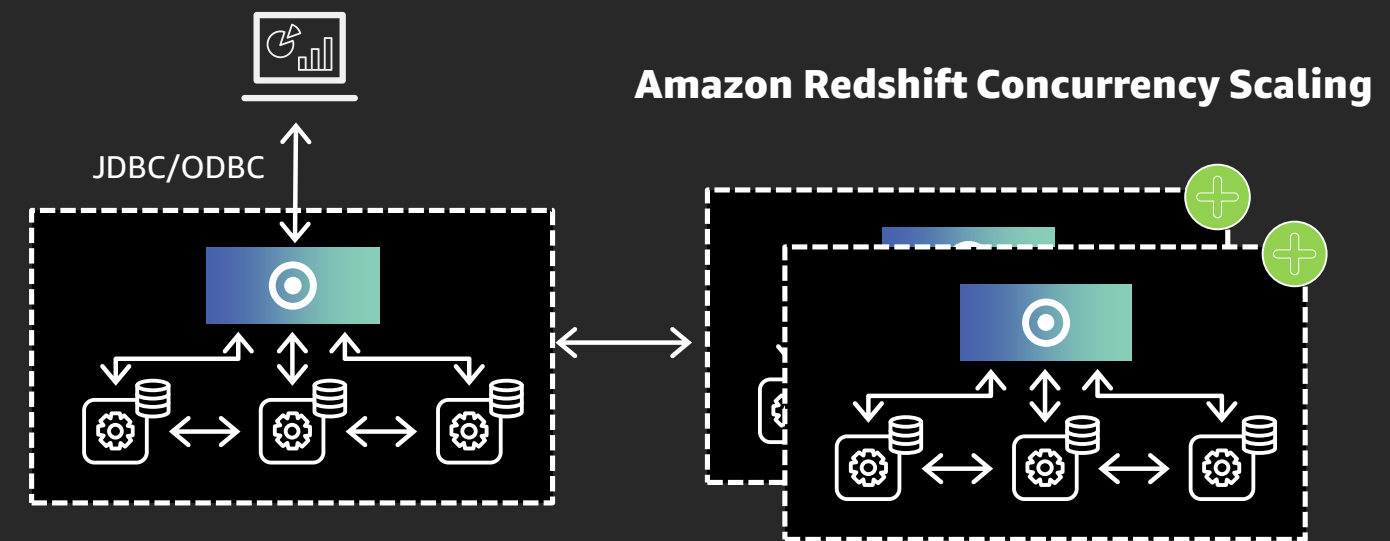
Compute Elasticity with Elastic Resize

- **In-place**
 - Add or remove nodes to/from existing cluster
- **Scale-out**
 - Performance scales proportionally
- **Fast**
 - Completes within a few minutes
 - Limited disruption to sessions and queries



Compute Elasticity with Concurrency Scaling

- Scale out to multiple Amazon Redshift clusters from a single endpoint
- Scale out in seconds
- Per-second billing for additional clusters used
- One hour free usage per day



Continuous performance and scalability improvements

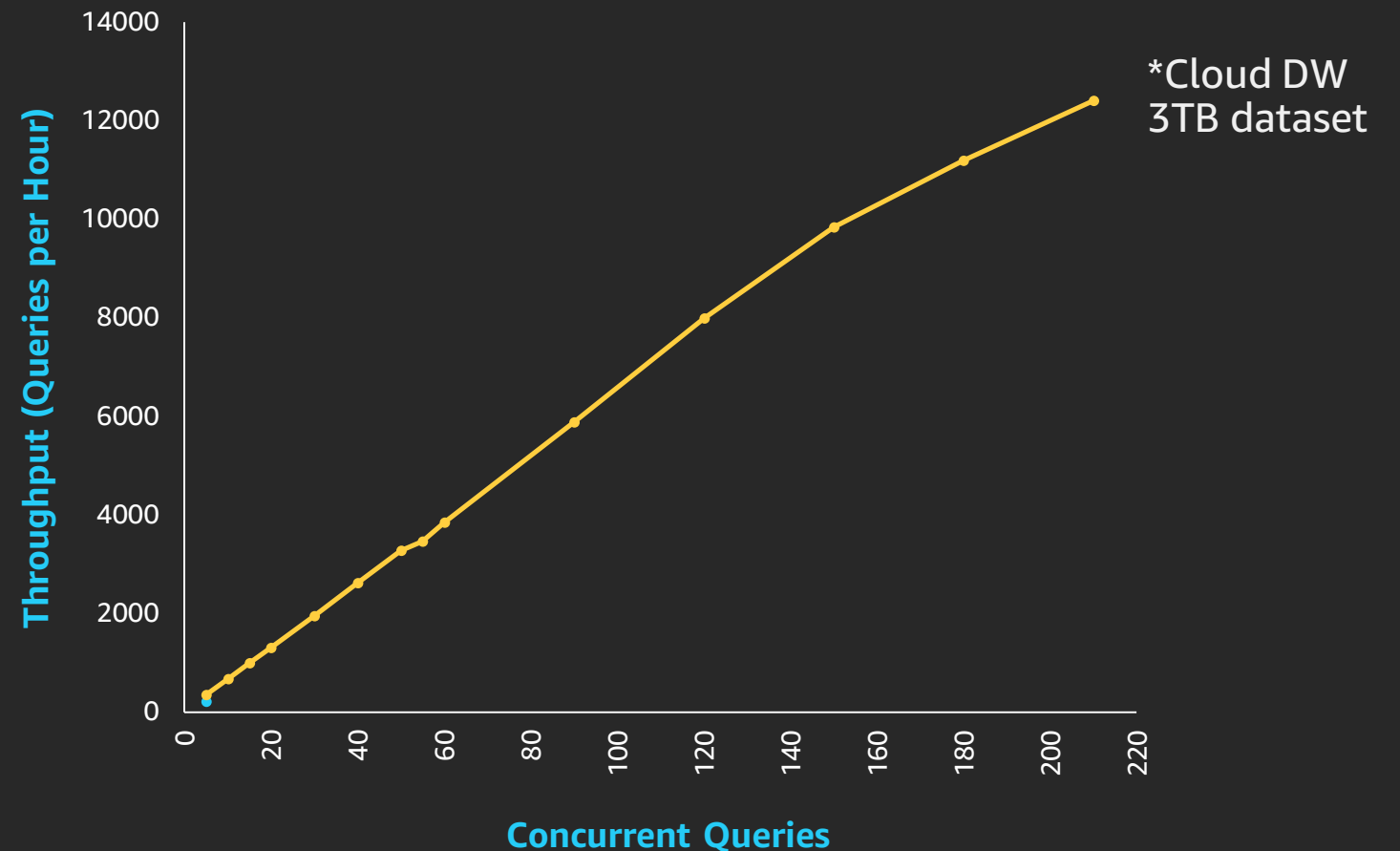
Maintain **performance** and **scalability** leadership through continuous telemetry and benchmarking

Within 2019:

- Concurrency Scaling
- AZ64 encoding
- Global compiled code cache
- AutoWLM & query priorities
- Auto Analyze & Auto Vacuum
- DistKey & SortKey Advisor
- Runtime filters (bloom filters)
- Materialized Views
- Unload to Parquet
- Commit Performance

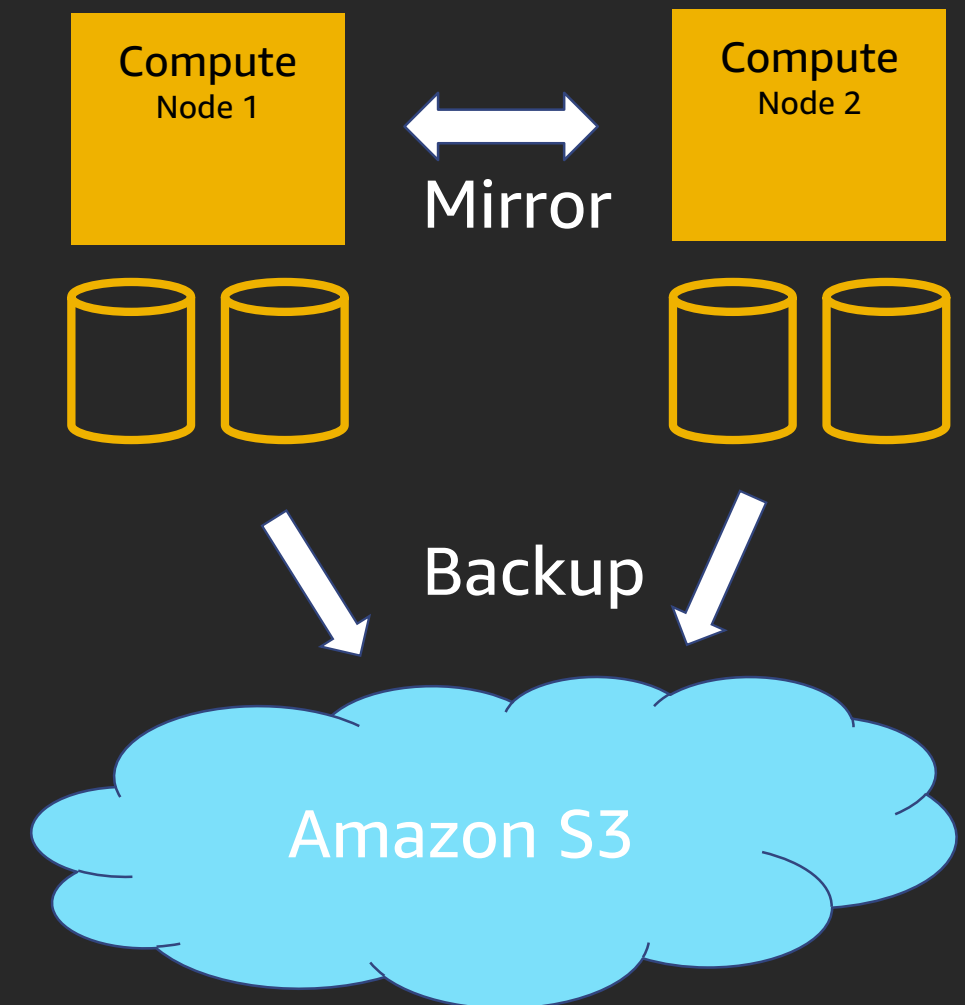
**>35x improvement
in throughput in 2019**

Scalability improvements



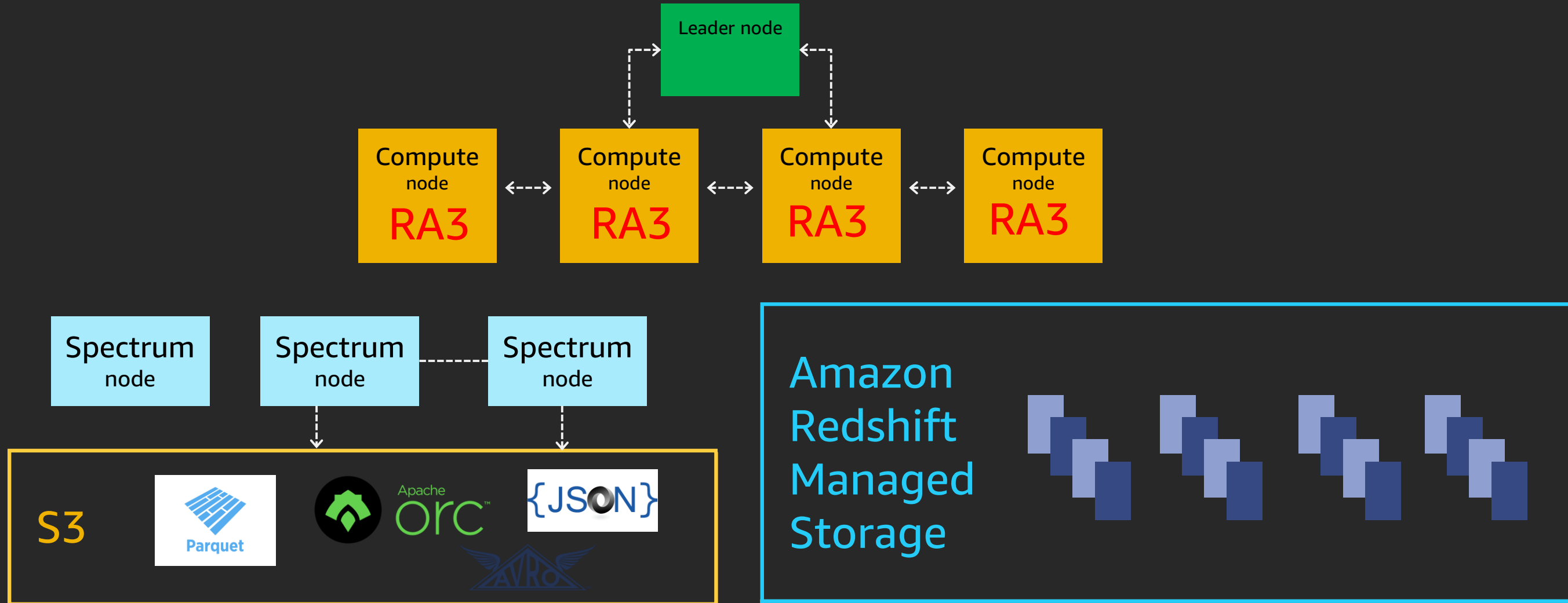
Data persistence

- Replication factor of 2 architecture
 - Primary copies stored on local disks
 - Replicas mirrored on to neighboring nodes
- Data backed up to Amazon Simple Storage Service (Amazon S3) periodically
- Data re-replication and rehydration system
 - Increases replicas when needed
 - Fetches data from Amazon S3 or mirror
 - Activated during disk failures, node replace, restore and elastic resize



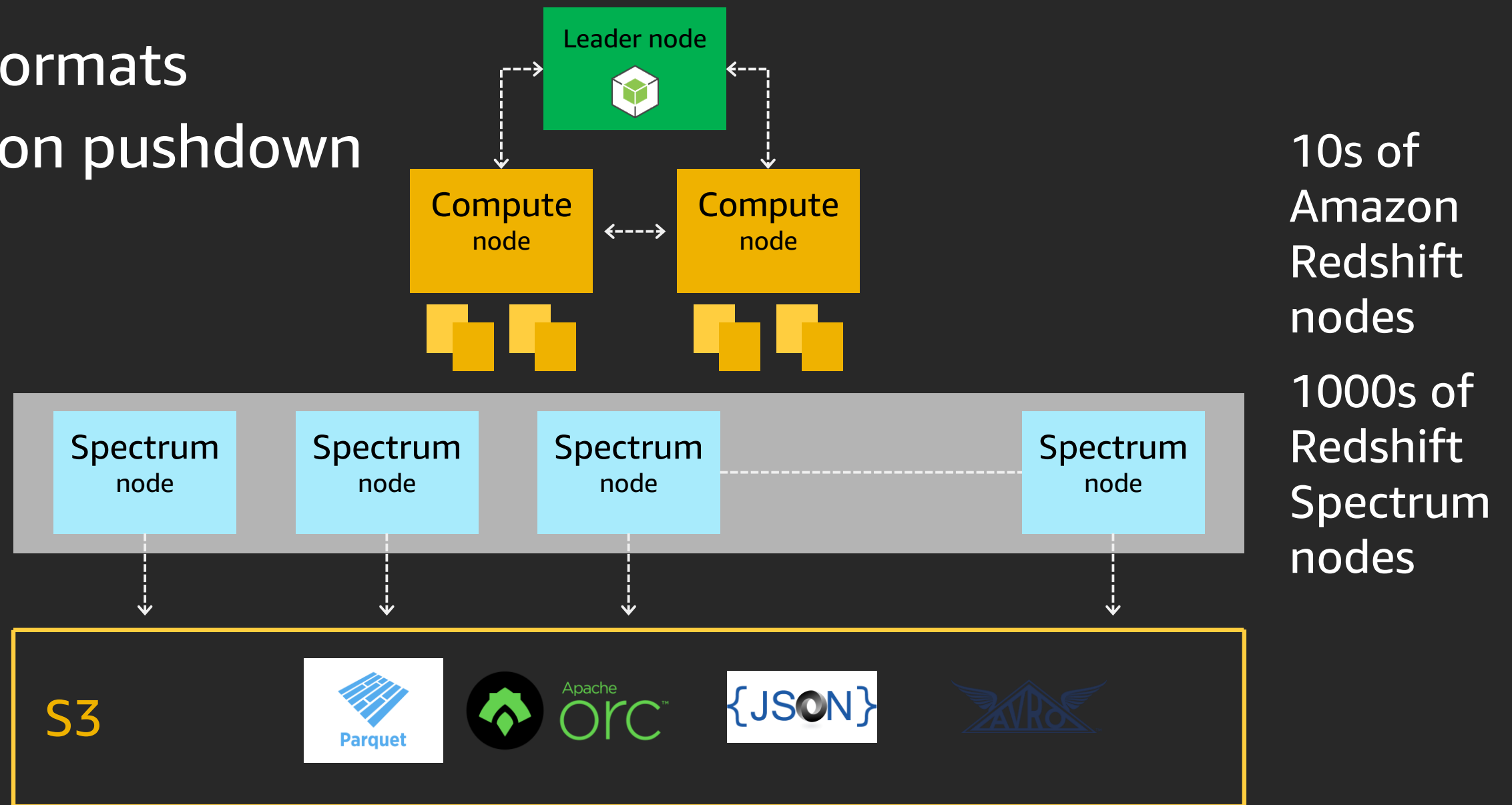
Two forms of storage elasticity

- **Spectrum**: Query data in open file formats in Amazon S3 data lake
- **Amazon Redshift managed storage**: Analytics-optimized storage layer **New**



Storage elasticity with Amazon Redshift Spectrum

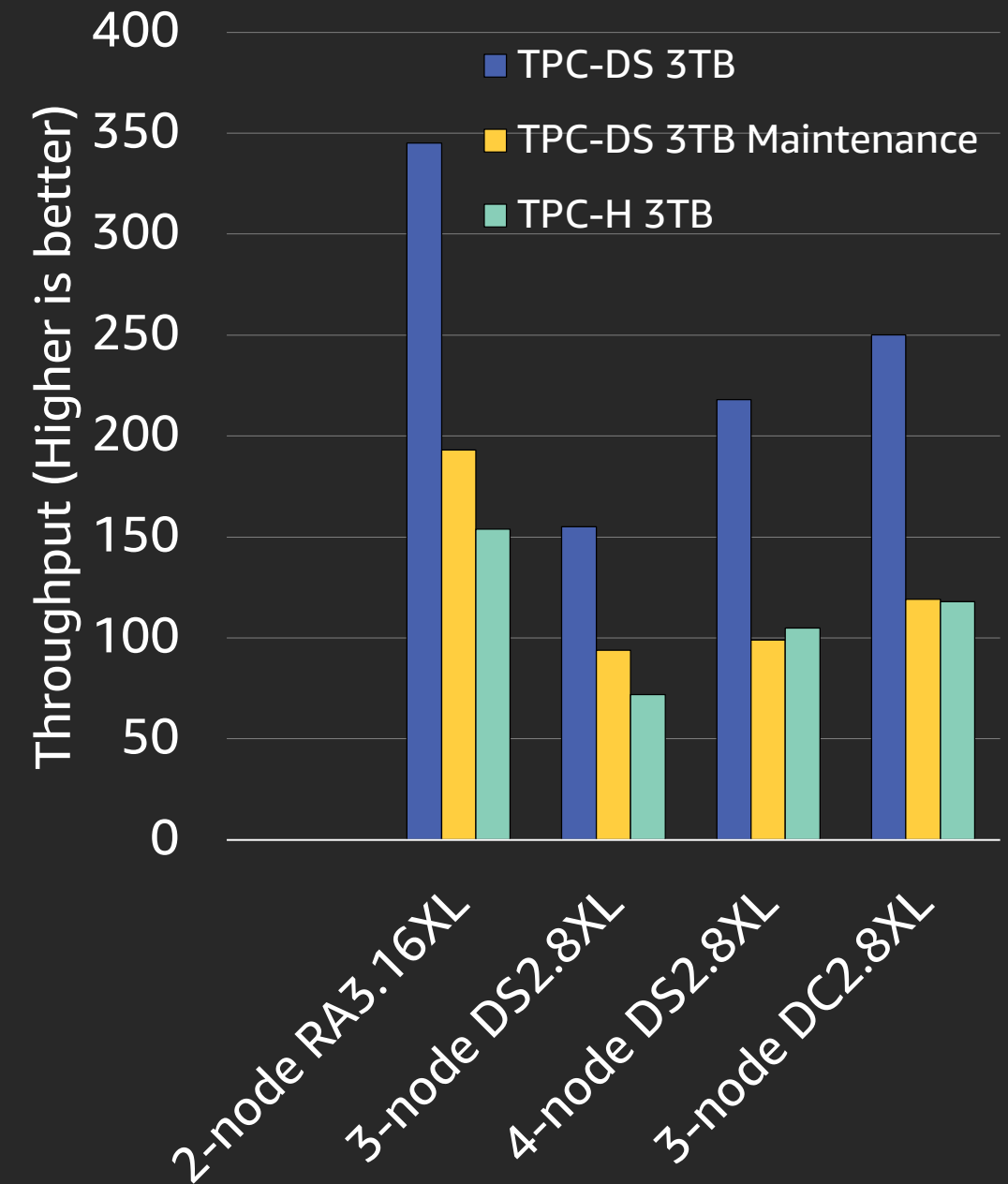
- Queries directly against data in Amazon S3 using thousands of nodes
- Open file formats
- Computation pushdown



Storage Elasticity with Amazon Redshift Managed Storage

- Transactional capability and large datasets in efficient Amazon Redshift format
- Relaxed storage provisioning
 - Ingest & analyze more data than stored on local storage
- Performance benefit of locally attached storage
 - System stores hottest data locally
- Continuous data temperature tracking
 - System ranks hot data based on frequency and recency of accesses

RA3 is the first instance that supports Redshift Managed Storage



RA3: Unmatched performance at unbeatable price

RA3.16XL

- Compute price
 - On demand price - \$13.04/node/hour
- Storage price
 - Up to 64TB in managed storage per node
 - Price- \$0.024/GB-Month

- Up to 2x performance and 2x storage capacity compared to DS2.8XL
- The minimum sized RA3.16XL cluster scales up to 128TB of (compressed) data
- An Amazon Redshift RA3 cluster can have up to 8PB of (compressed) data in managed storage
















































RA3.4XL – coming soon

Innovid Redshift Concurrency Scaling



The only independent
omni-channel advertising
and analytics platform
built for television.

Established Customer Base

							
							
							
							
							
							
TOP AGENCIES		OVER 1,000 LEADING BRANDS				TOP BROADCASTERS	

Backed by

Goldman
Sachs

SEQUOIA

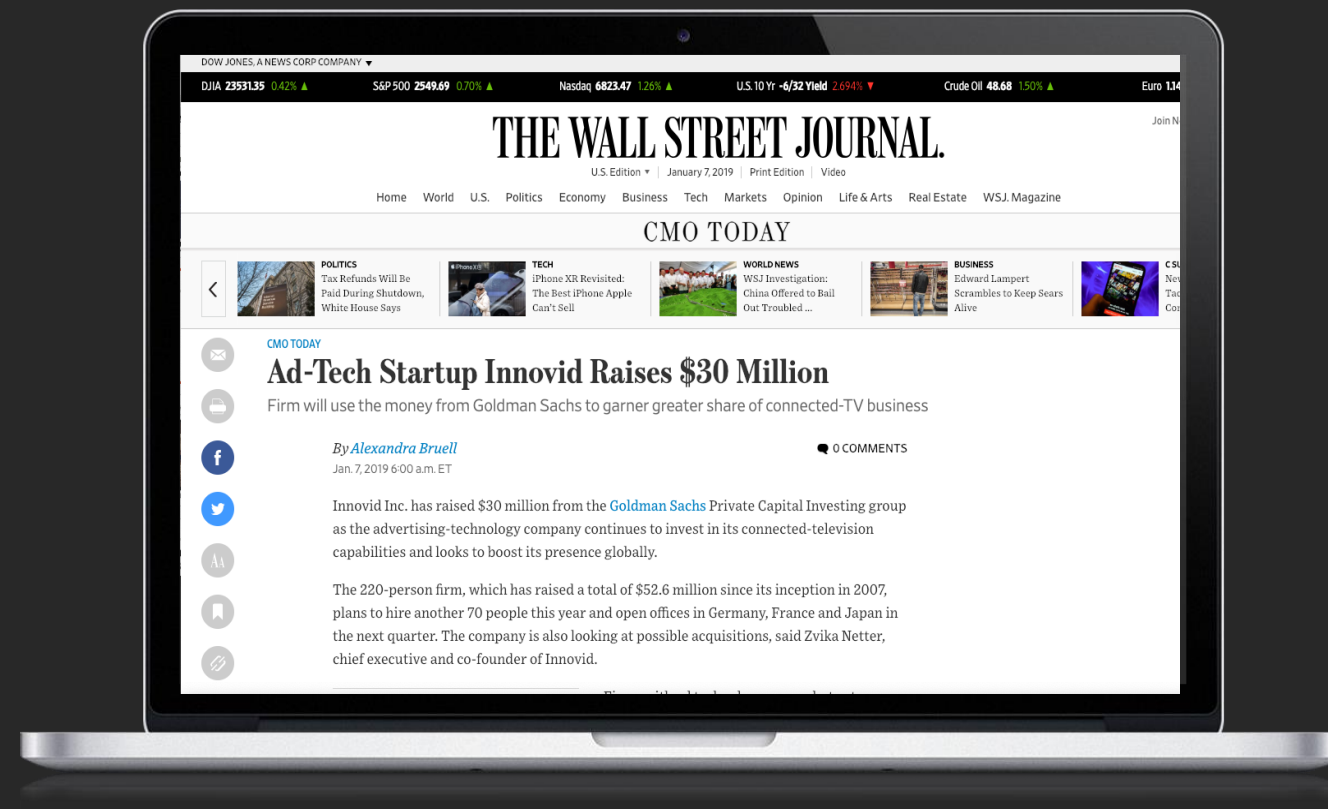
CISCO

GENESIS
PARTNERS

Vintage
EUROPE ISRAEL US
Investment
Partners

Deutsche
Telekom

NEWSPRING
CAPITAL



\$30,000,000
INVESTMENT FROM GOLDMAN SACHS IN Q1 2019

300+
Employees Globally

16
Offices Globally

500MM
Videos Delivered Per Day

171 Years
Hour of Videos Viewed Per Day

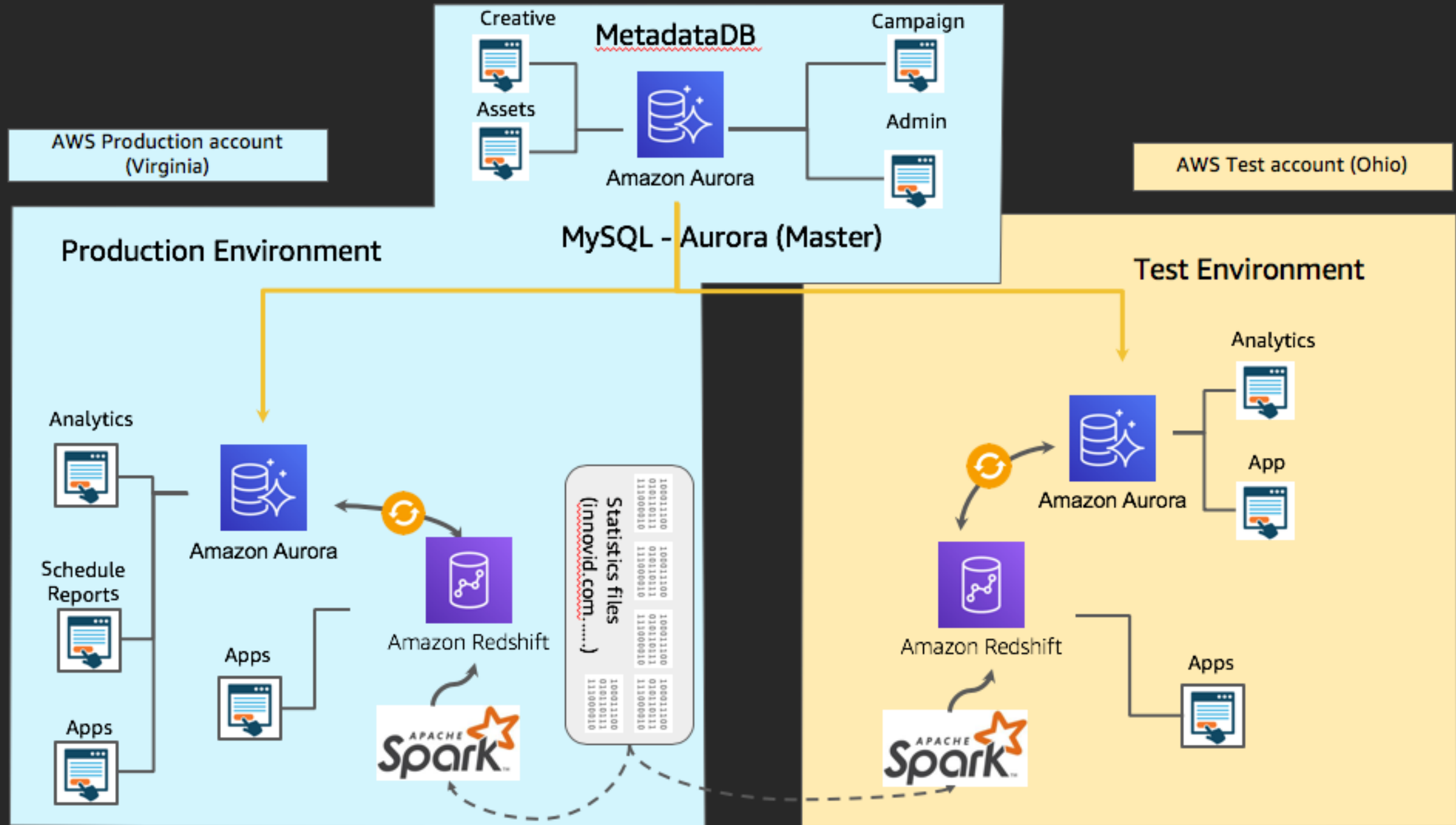


We process more than **5 billion** events per day

Basic processing using *Spark EMR cluster*

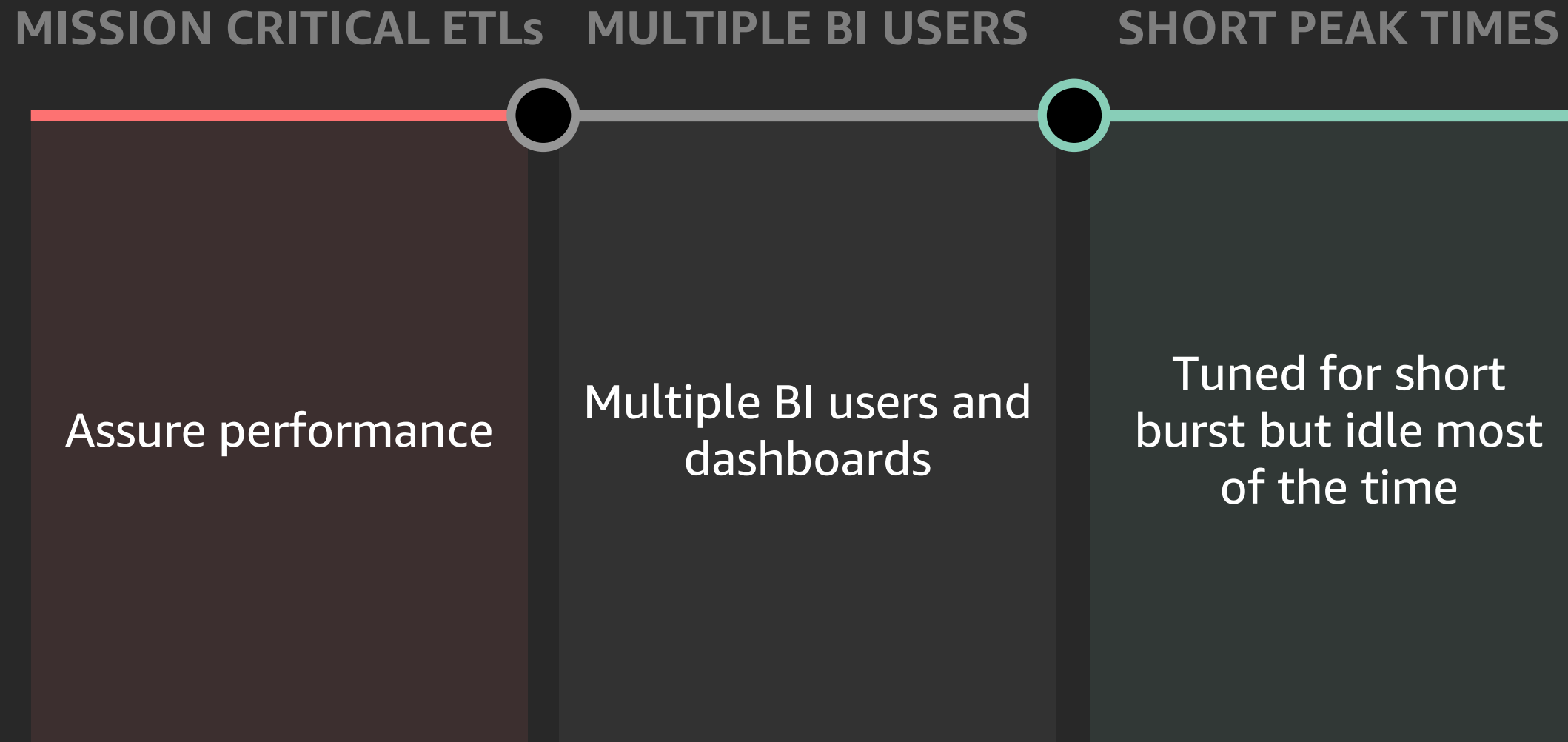
Business logic layer is implemented on *Amazon Redshift*

Production Environment



Scaling out on demand – why?

How did we get to Concurrency Scaling?



Testing Variants

- 1 **Concurrency
Scaling On/Off**
- 2 **Queue
Concurrency**
- 3 **SQA**
- 4 **Dedicated
Queues**

Testing Invariants

- 1 **Query Types**
- 2 **Test Machine
(Client)**
- 3 **Dedicated
Cluster**
- 4 **Data**

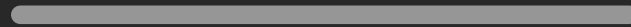
Query Types

SHORT



< 10 secs

MEDIUM



10 secs – 10 mins

LONG

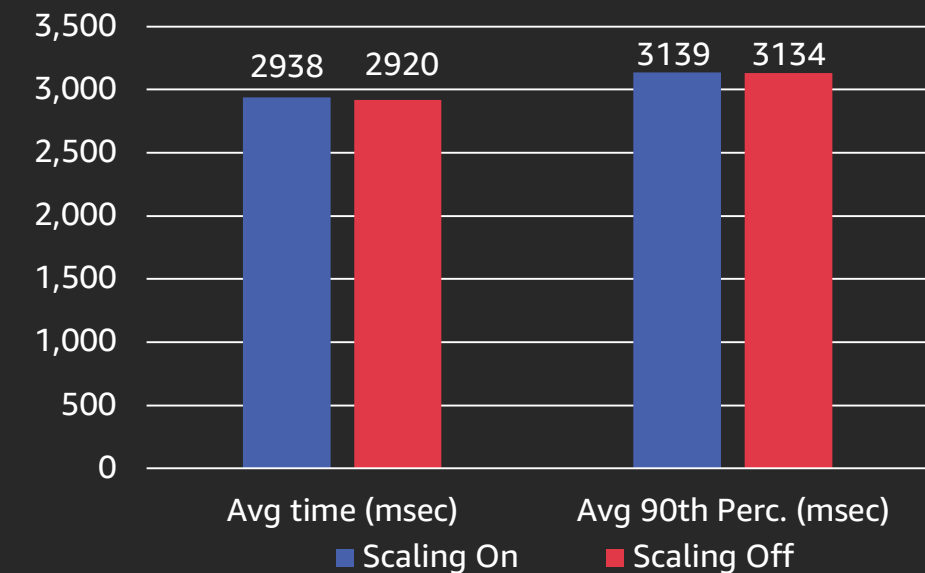
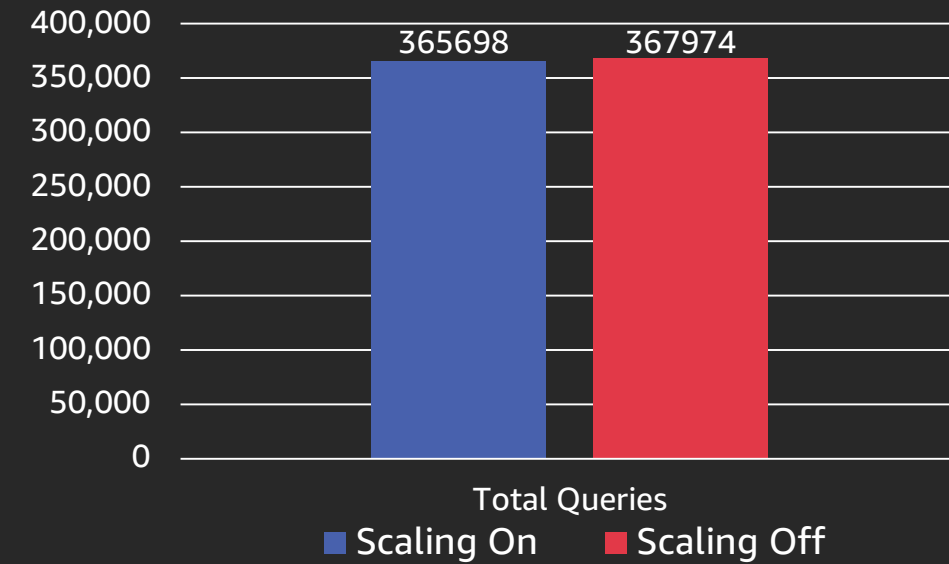


> 10 mins

Short Queries Test

TEST SETUP:

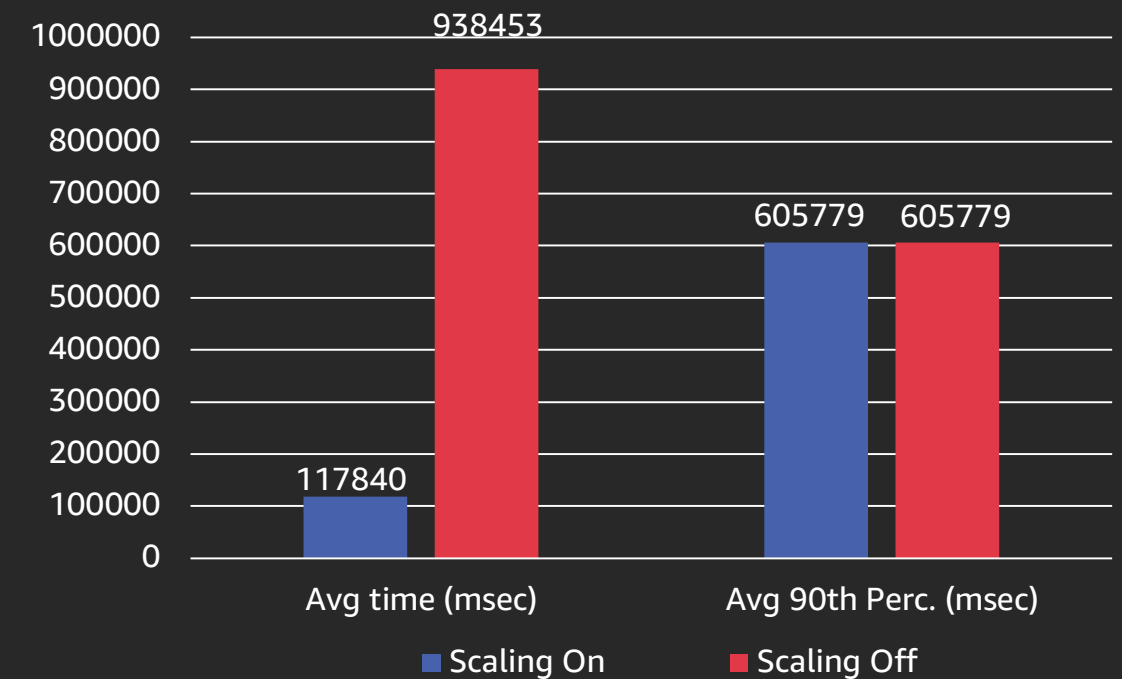
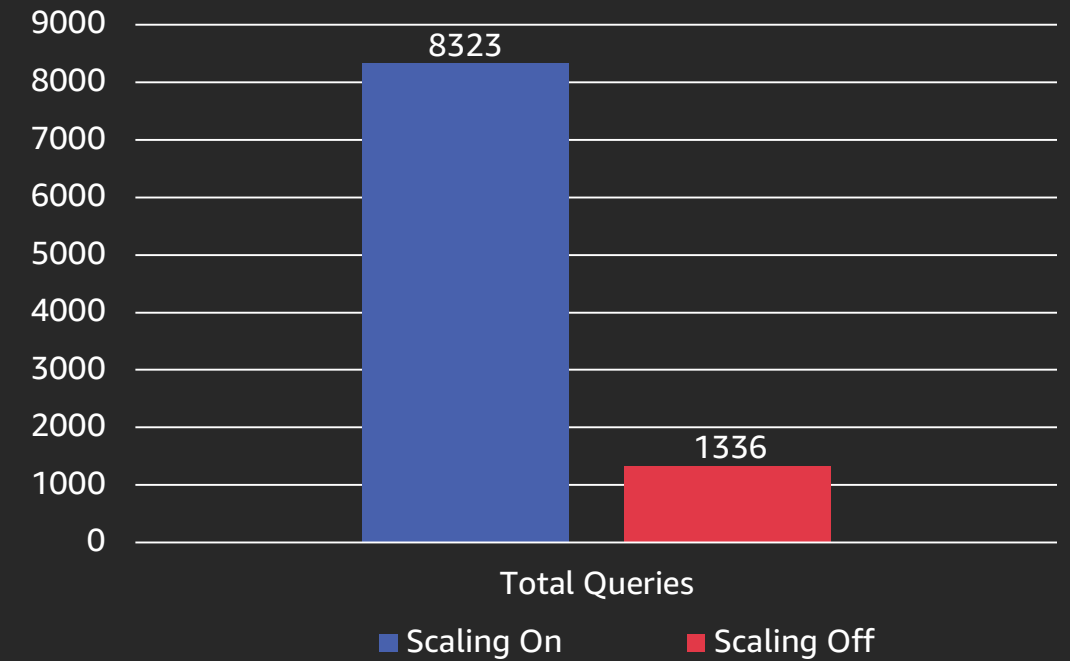
- 300 concurrent queries
- SQA = Dynamic
- Queue Concurrency = 15



Medium Queries Test

TEST SETUP:

- 300 concurrent queries
- SQA = Off
- Queue Concurrency = 35

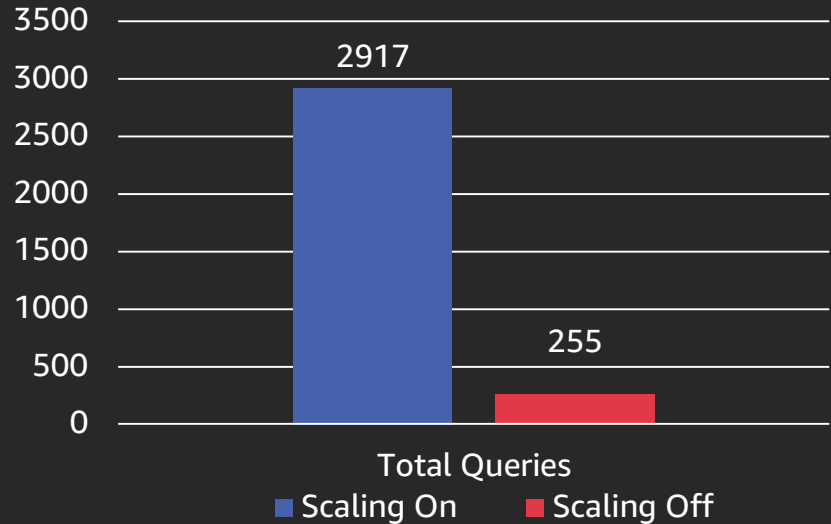


Mix of long & short queries

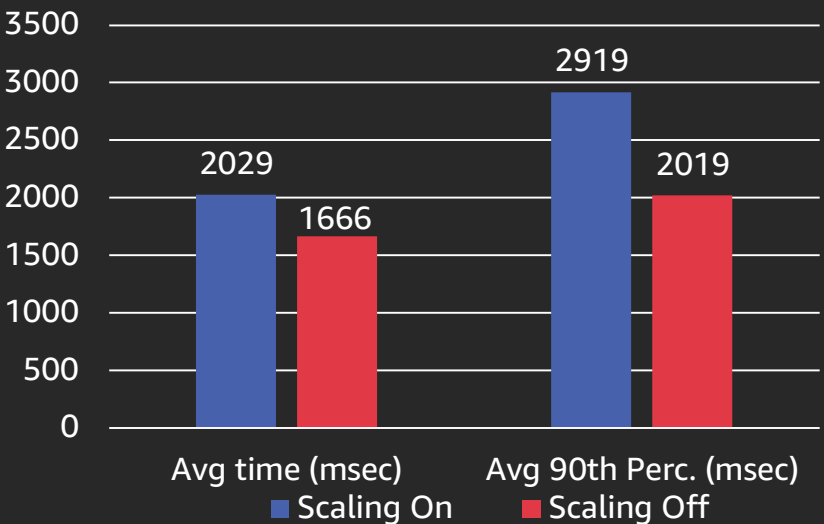
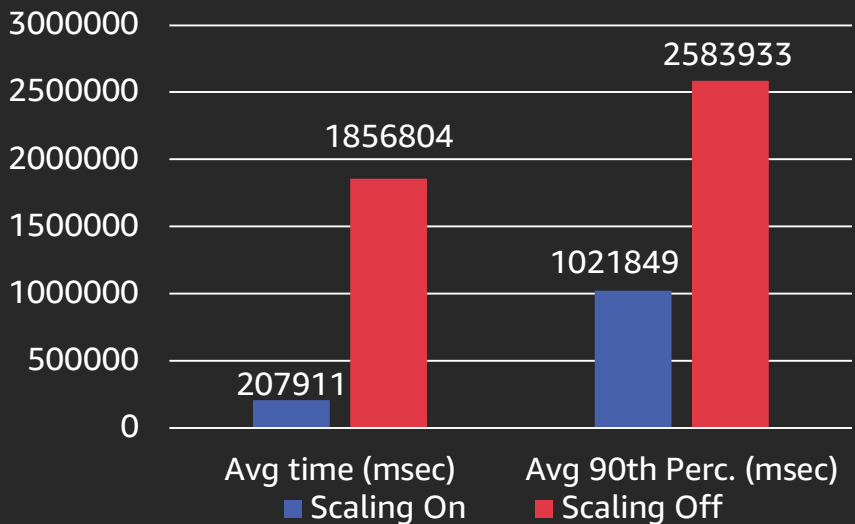
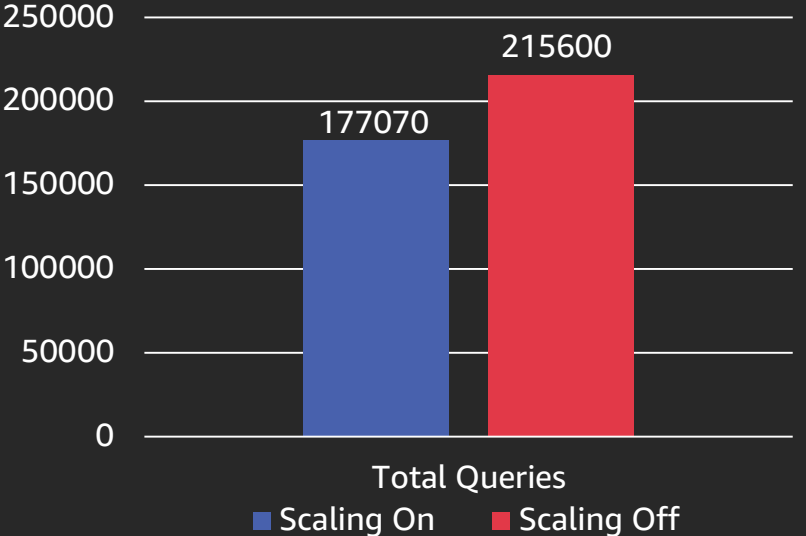
TEST SETUP:

- 300 concurrent queries
 - 200 long
 - 100 short
- SQA = Dynamic
- Queue Concurrency = 5

Long queries



Short queries



Database performance metrics

Concurrency Scaling Activity

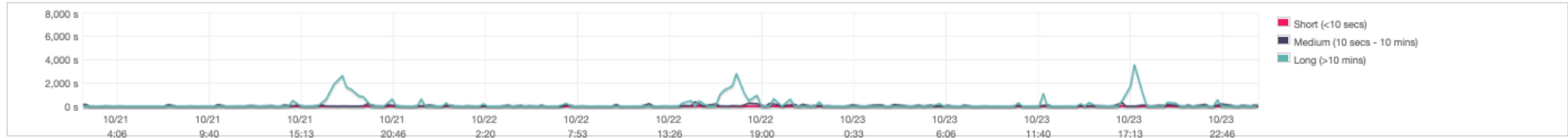


Concurrency Scaling Usage In Minutes (Sum)

Total usage: 196.35 minutes ⓘ



Query Duration



ⓘ Short, medium, and long time ranges are based on the total time it takes to process a query.

Key takeaways

Scales out well for medium/long queries

Best for:

- Peak times
- Ensure level of service
- Expected users growth

Not recommended when:

- Most of the load is writes
- Mostly short queries usage
- Constant load

Thank you!

Thanos Papathanasiou

Contact: thanos@amazon.com

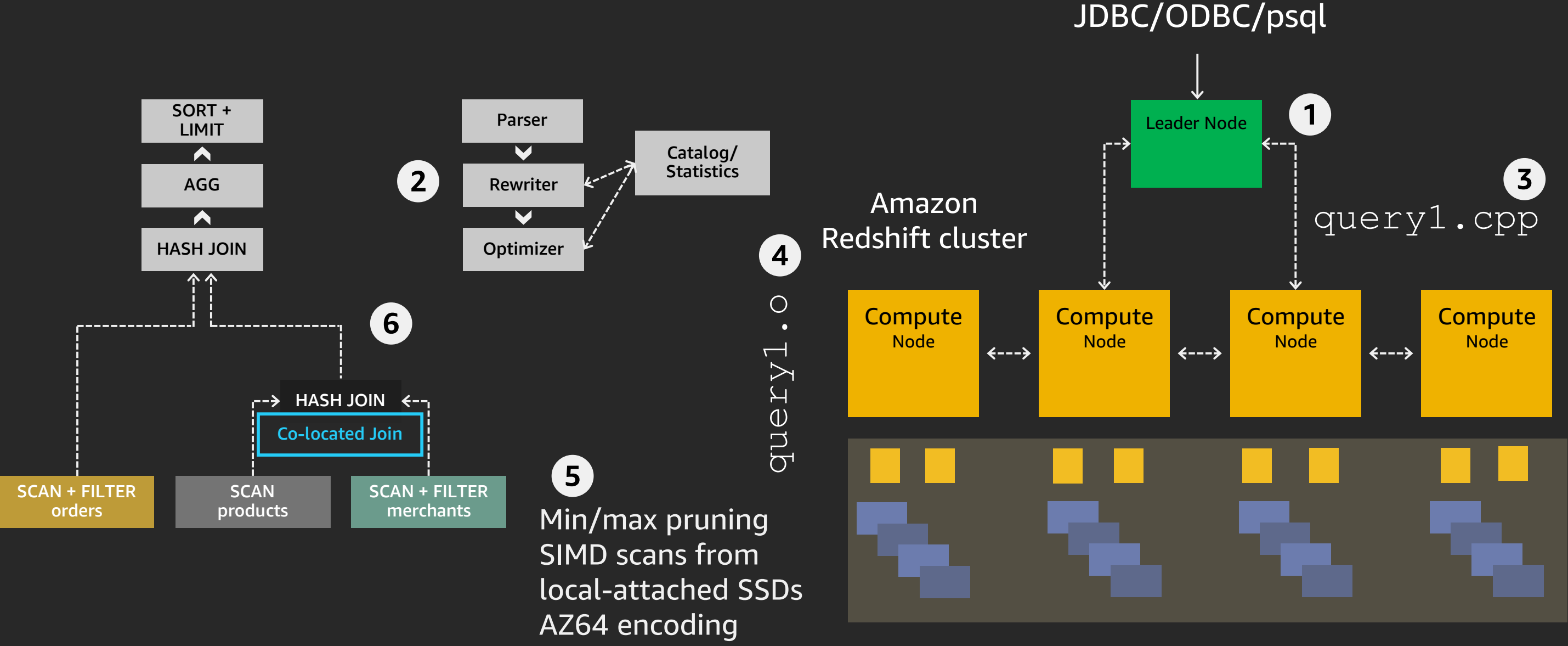
Yuval Pemper

Contact: yuval@innovid.com



Please complete the session
survey in the mobile app.

Executing this query in Redshift



Top-of the line data warehousing performance