

What's new in MySQL

On-premise and cloud

Nipun Agarwal

Senior Vice President, MySQL & HeatWave

First LTS Release!

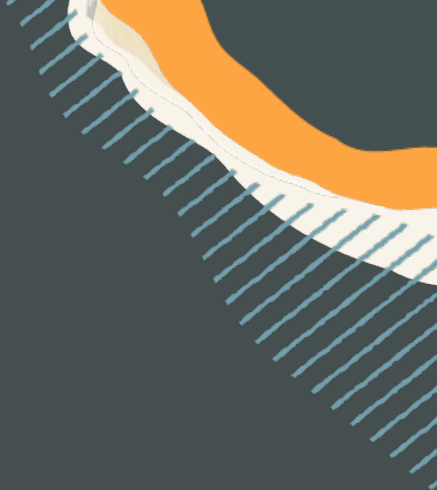
First LTS release – April 30 – 8.4.0 LTS

Innovation releases to start with 9.0.0

9.7.0 targeted LTS release - in 2 years

Focus areas

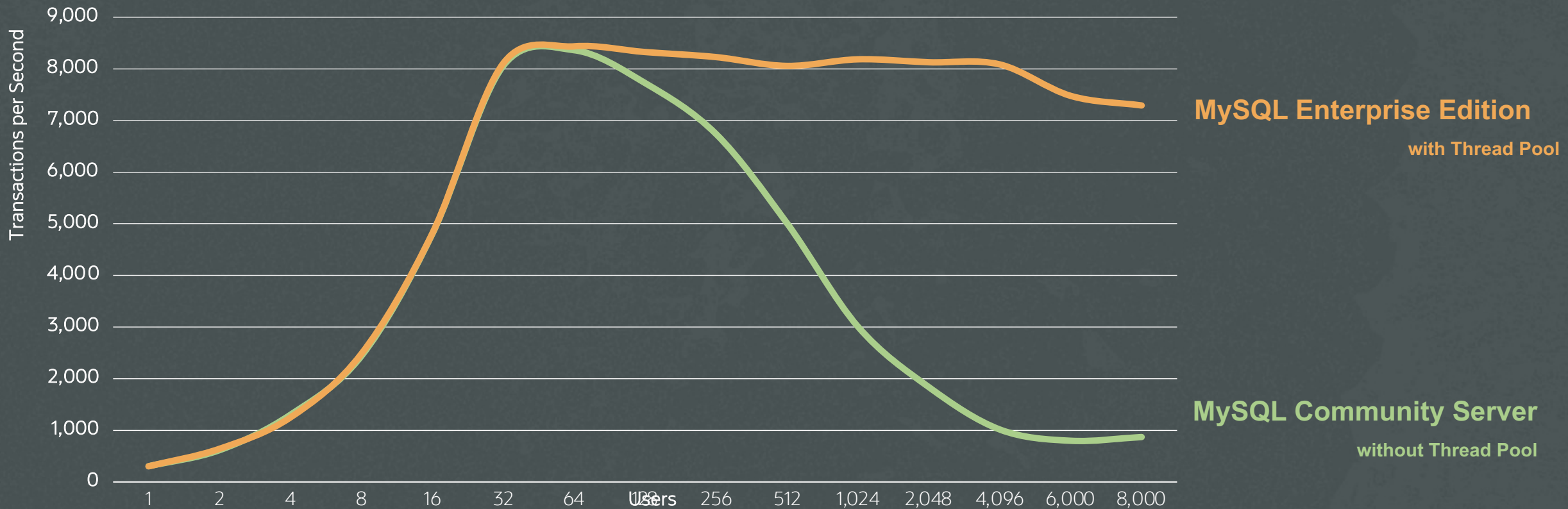
- Better performance
- Lower cost
- Developer productivity
- Security
- Higher availability



Thread Pool Improvements

Highly Scalable Thread-Handling Model

Sysbench OLTP Read/Write, "pareto" access pattern, 100GB data size



9x Better Scalability: Sysbench OLTP Read/Write

Eliminating double buffer writes

Atomics

Core	Config	InnoDB_DoubleWrite	Buffer_pool_pages_dirty	Time	Bo_sum
16	BP 128 GB --table = 10 --tablesize=40M	OFF	5356K	00:02:22	83.7 GB
		ON	5356K	00:34:44	172.5 GB
		ATOMIC_IO	5356K	00:02:22	83.7 GB
8	BP 128 GB --table = 10 --tablesize=20M	OFF	2679K	00:01:15	41.9 GB
		ON	2679K	00:21:51	86.3 GB
		ATOMIC_IO	2679K	00:01:17	41.9 GB
4	BP 128 GB --table = 5 --tablesize=10M	OFF	1339K	00:00:49	20.9 GB
		ON	1339K	00:11:06	43.2 GB
		ATOMIC_IO	1339K	00:00:49	20.9 GB

*

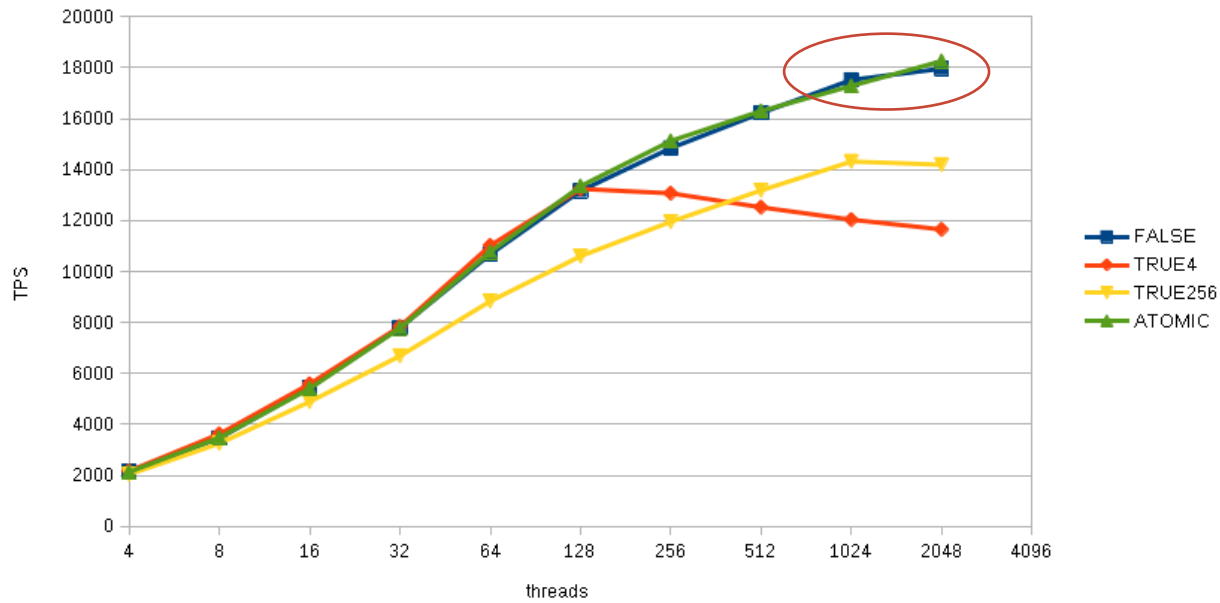
Experiments show:

- With ATOMIC I/O, **amount of data written** is similar to *dblwr=OFF* (~half) on disk
- With ATOMIC I/O, **time** to flush is similar to *dblwr=OFF*

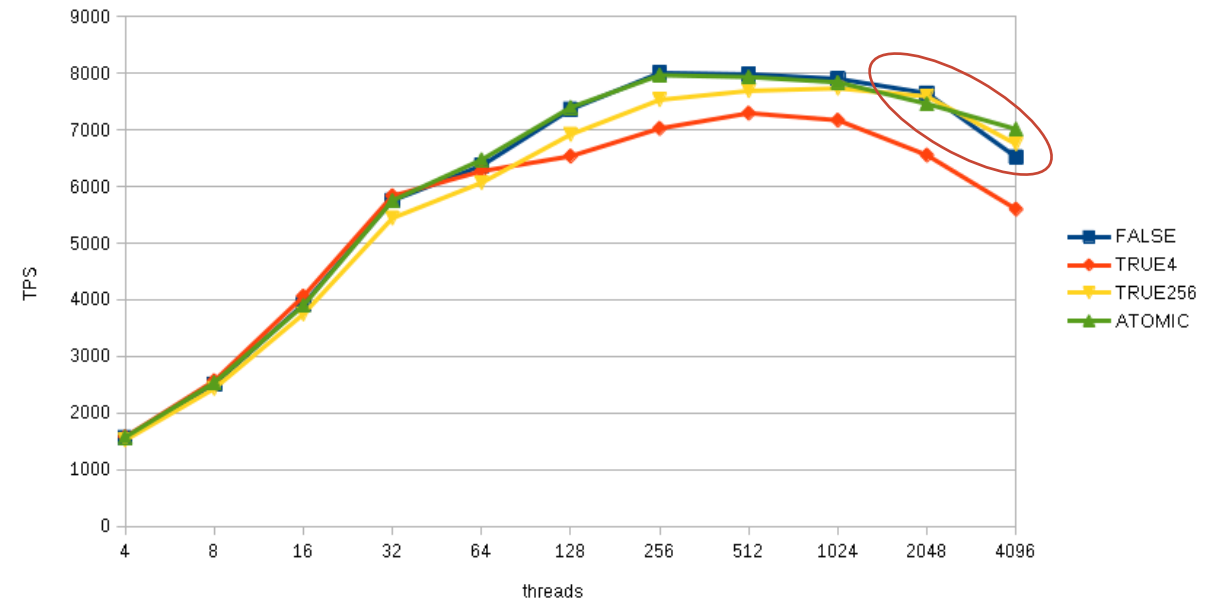
Eliminating double buffer writes

Write intensive workload

OLTP_RW++ (OCPU16; flush log at commit=1; log write ahead = 8K)



OLTP_RW++ (OCPU4; flush log at commit=1; log write ahead=8K)

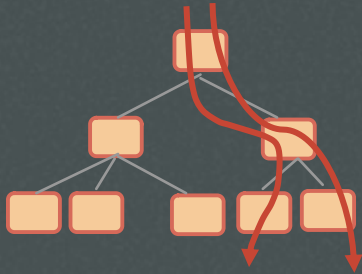


•ATOMIC I/O performance at par with *dblwr=OFF*

Bulk ingest support in MySQL HeatWave

In-Place Insertions
into B+ Tree

Clustered Index

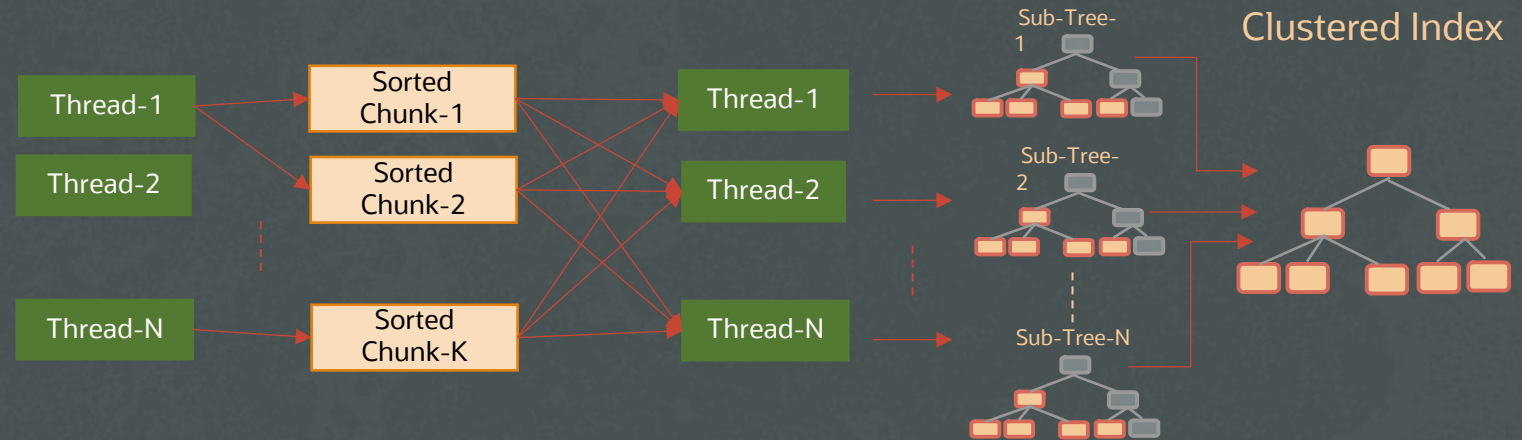


Earlier

Parallel, In-Memory
Sort and Merge

Parallel
Sub-Tree Build

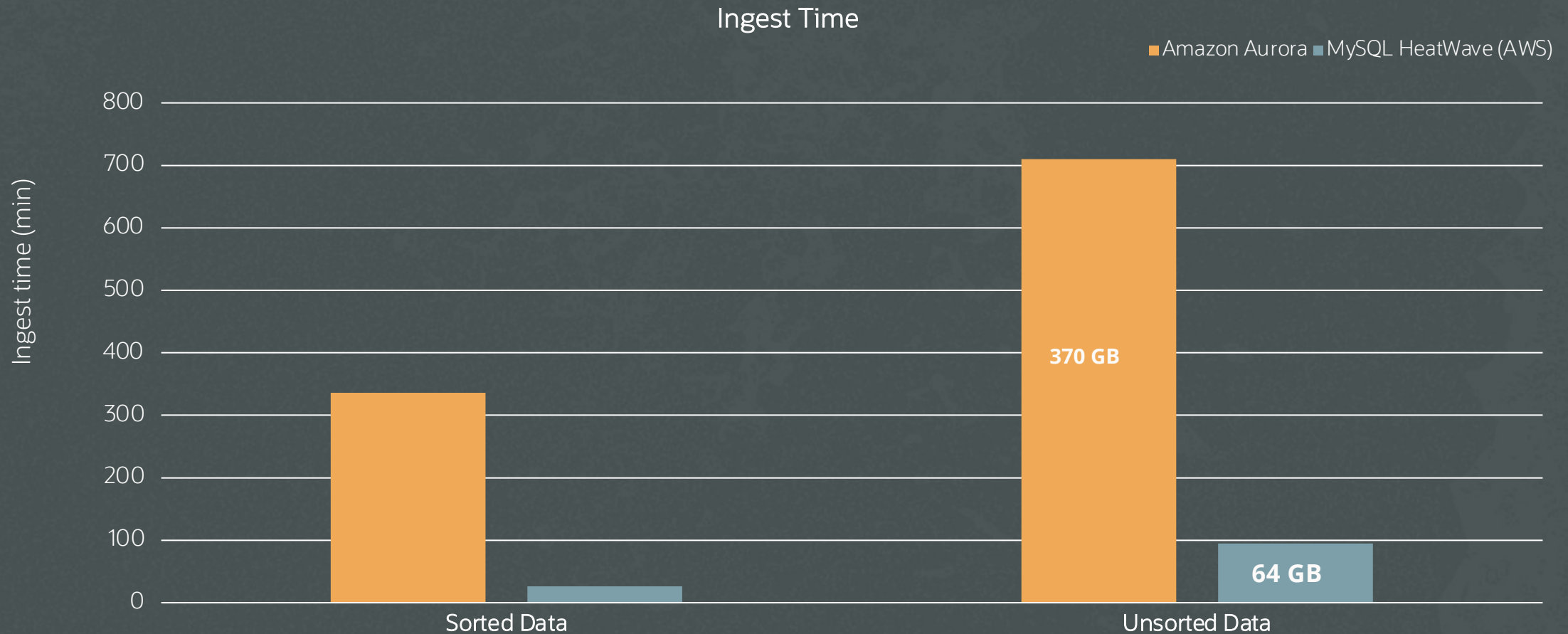
Constant-Time
Sub-Tree Merge



- Parallel sort & merge and parallel build of index sub-trees
- Sequential writes of sorted data into disk which eliminates random disk i/o
- Pipelining of internal stages which overlaps compute with disk i/o

Bulk ingest performance

10x faster than Amazon Aurora, uses less memory



Many Performance Fixes in 8.4

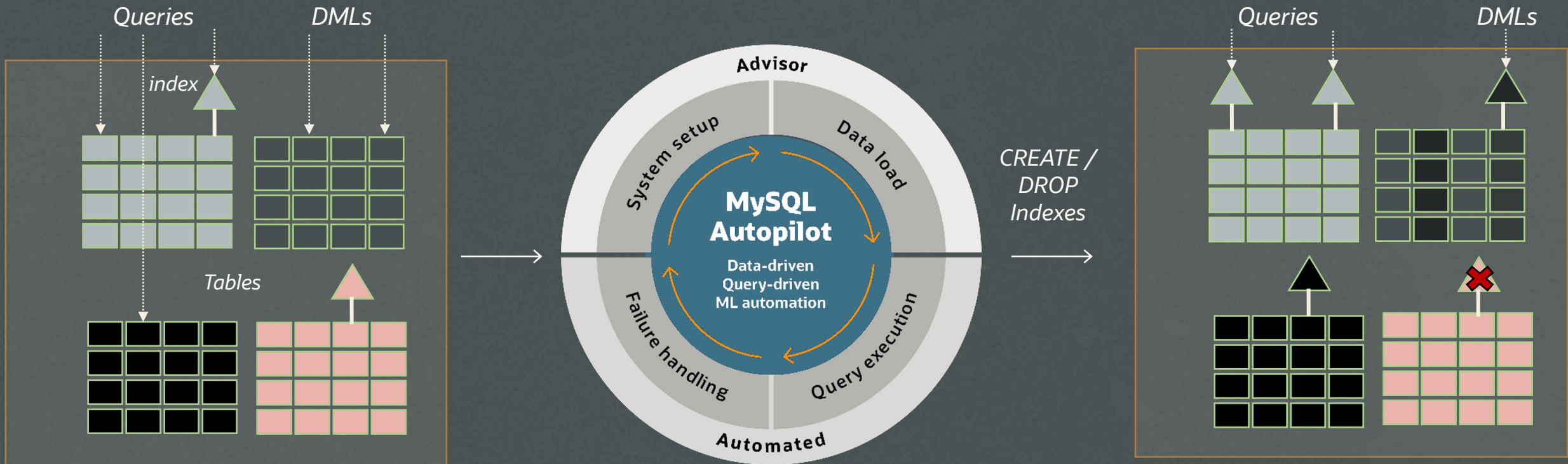
Number	Description
Bug#25903274	PERFORMANCE REGRESSION WITH PREPARED STATEMENTS. Fixed in 8.0.3.
Bug#36154818	Redolog files on Windows are not using Overlapped mode. Fixed in 8.0.37.
Bug#36142806	InnoDB_parallel_read_threads >1 makes simple select count more expensive. Fixed in 8.0.37.
Bug#102238	log_writer uses too much CPU on small servers. Closed as "not a bug".
Bug#32511973	SLOW PERFORMANCE OF MYSQL 8 COMPARED WITH 5.7 ON GEOGRAPHICAL SELECT.
Bug#34951273	Performance of scanning data_lock_waits worse than expected with read-only trx. Fixed in 8.4.
Bug#35936316	Regexp/rlike function in routines extreme regression mysql 5.7 to mysql 8.
Bug#93684	mysql innodb dump restore slows down after upgrade mysql 5.7 to 8.0.
Bug#33684069	SHOW SLAVE STATUS became expensive for Replica->Slave column name conversions. Fixed in 8.0.31.
Bug#35712638	Materializing performance_schema.data_locks can lead to excessive mem usage. Fixed in 8.0.37.
Bug#27877386	MySQL Upgrade from 5.7 to 8.0: performance improvements.
Bug#89963	Slowdown in creating new SSL connection. Closed as "not a bug".

Many Performance Fixes in 8.4

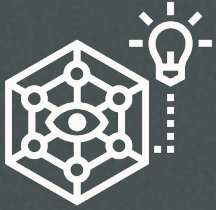
Number	Description
Bug#28857534	MySQL 8.0 performance degradation on INSERT with foreign_key_checks=0. Fixed in 8.0.14.
Bug#35916912	Performance degradation from 8.0.30 onwards related to performance_schema. Fixed in 8.0.36.
Bug#33840573	Regression on DDL statements with big stage/sql/checking permissions.
Bug#30837086	SELECT from I_S.INNODB_COLUMNS is slow in 8.0. Fixed in 8.0.21.
Bug#34959356	Poor performance when using HASH field to check unique. Fixed in 8.0.36.
Bug#27934653	60% overhead from performance schema instrumentation of stored functions. Fixed in 8.0.36.

MySQL Autopilot indexing

Recommends secondary indexes for OLTP workloads



Why ML-based automation?



Works for individual workloads

- No guess work
- Interpretable



ML models are adaptable

- Ever-changing env
- New server releases

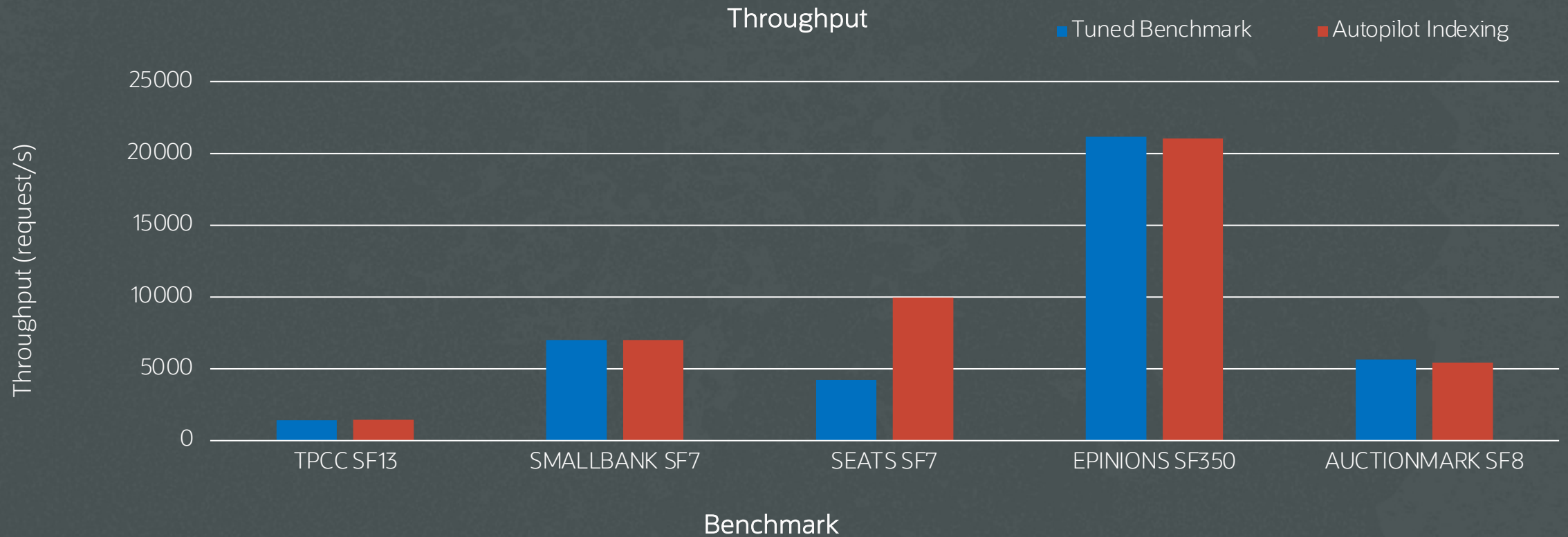


Can predict improvement

- Helps plan recommendation

Results

Throughput at Par or Better Even on Benchmarks Which are Tuned



- Autopilot recommends indexes whose performance is at par or better than manually tuned benchmarks
- In some cases, Autopilot recommends fewer indexes which saves storage











Stored Procedures Inside the Database

Handle data-intensive app functionality

- Minimize data movement
- Reduce cost
- Improve Security
- Simplify complex ETL → ELT



MySQL Stored Programs - SQL vs JavaScript

	SQL Stored Procedures	JavaScript Stored Programs
Expressiveness	 Hard to use, lacks basic constructs like containers (arrays, maps)	 Highly expressive and robust
Efficiency	 Challenging to optimize due to interpreted code	 Many JS code analysis tools. JavaScript apps are fast and optimized by GraalVM
Ecosystem	 Insufficient: Lacks support from IDEs, debuggers, testing frameworks, ...	 Large ecosystem of tools for developers of JavaScript applications
Availability of developers	 Few experienced programmers Especially with MySQL Ecosystem	 13.8 M Developers The most popular developer language
Reusable 3rd Party libraries	 Few, mostly code examples	 Thousands

Oracle GraalVM

High Performance JDK

The advanced optimizing Graal compiler provides ahead of time compilation, and polyglot language execution



High-performance optimizing
Just-in-Time (JIT) compiler



Ahead-of-Time (AOT)
“Native Image” generator



Multi-language support

JavaScript inside MySQL

On premise, OTN, MySQL HeatWave

SELECT

- Use anywhere where SQL stored functions can be used
- Expressions, Projection, WHERE clause, GROUP-BY, JOIN, ORDER BY, HAVING etc.

DMLs, DDLs, VIEWS

- Support inside DMLs (INSERT, UPDATE, DELETE, ...)
- DDLs including CREATE TABLE AS SELECT
- Support inside VIEWS

Interoperability

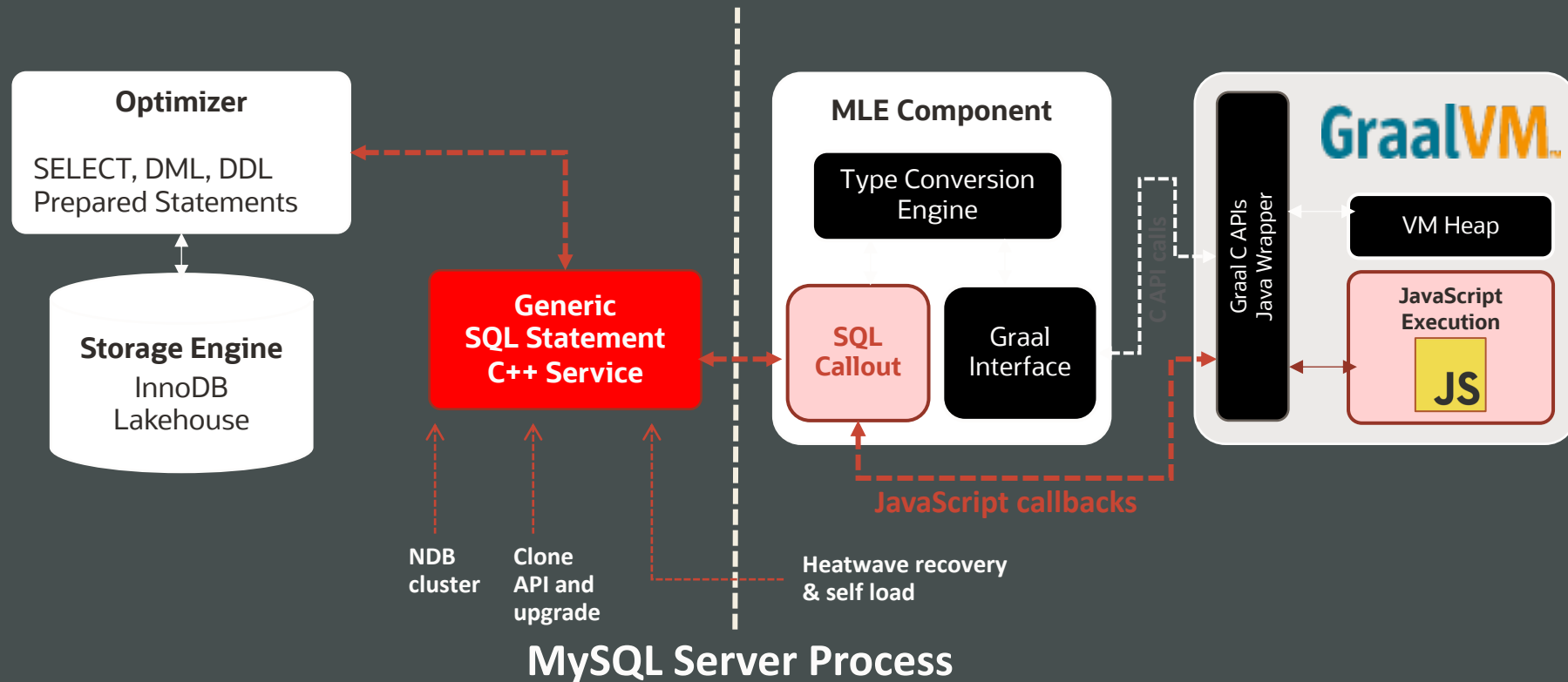
- Invoke JavaScript & SQL functions and Programs inside existing SQL stored functions or procedures

```
SELECT col1, col2, gcd_js(col1,col2)
FROM my_table
WHERE gcd_js(col1, col2) > 1
ORDER BY gcd_js(col1, col2);
CREATE TABLE gcd_table
AS SELECT gcd_js(col1,col2) FROM
my_table;
```

```
CREATE TABLE gcd_table
AS SELECT gcd_js(col1,col2) FROM
my_table;
```

MySQL-JavaScript

Server extended to support ability for SQL callout from JavaScript



SQL inside JavaScript

Statement Types

- Simple SQL statements
- Prepared statements with bind parameters

Data Access API

- Execute SQL inside JavaScript using XDevAPI
- Seamless MySQL ↔ JavaScript type conversion for query results

Session State

- Continue transactions inside JavaScript
- Access all session state inside JavaScript such as session variables & temporary tables

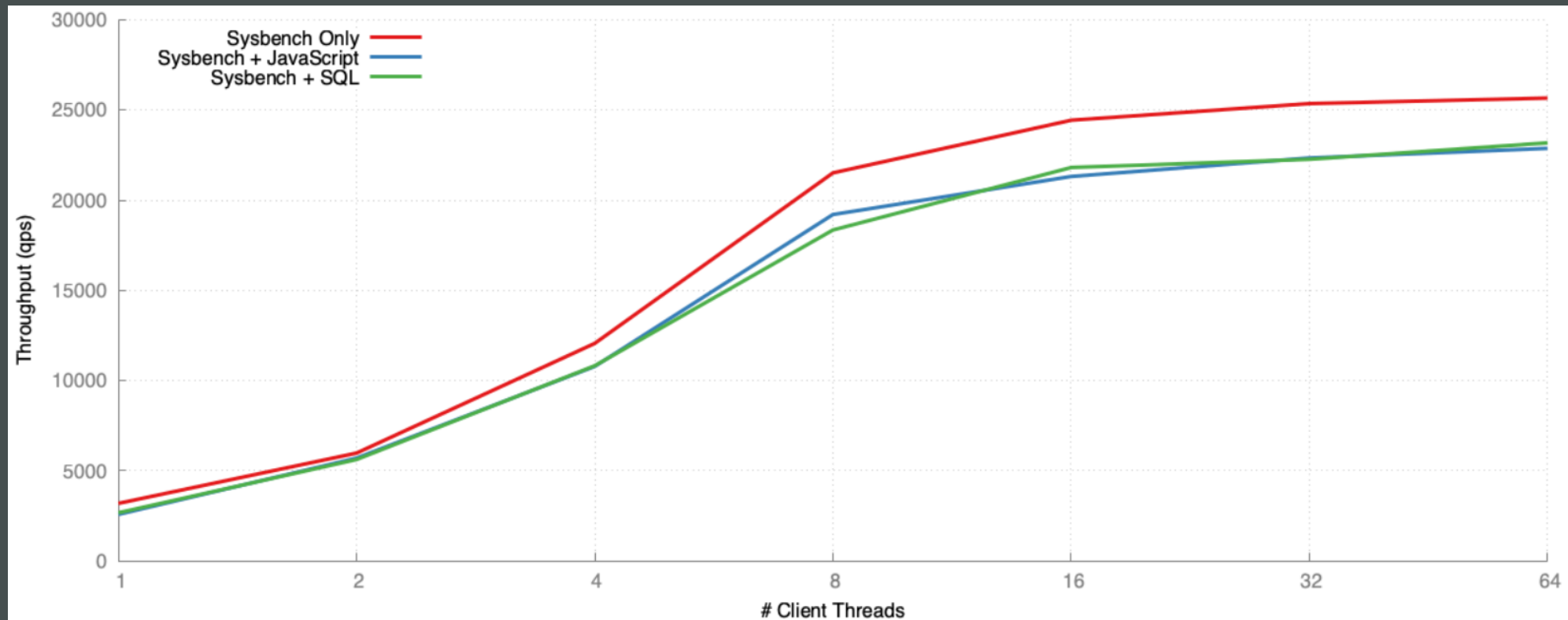
```
CREATE PROCEDURE gen_random_age (IN row_count INT) LANGUAGE
JAVASCRIPT AS $$
let insertStatement = session.prepare( "INSERT INTO
my_table(age) VALUES ( ? )" ); for (let j = 0; j < row_count;
j++) {
let random_age = Math.trunc(Math.random() * 100);
insertStatement.bind(random_age).execute();
}
$$
```

```
CREATE PROCEDURE average_age (OUT avg_age FLOAT) LANGUAGE
JAVASCRIPT AS $$
let age_sum = 0, count = 0;
let selectStatement = session.sql(
"SELECT age FROM my_table");
let result = selectStatement.execute(), row = null;
while(row = result.fetchOne()) {
age_sum += row[0]; count++;
}
avg_age = age_sum / count;
$$
```

SQL-Callout

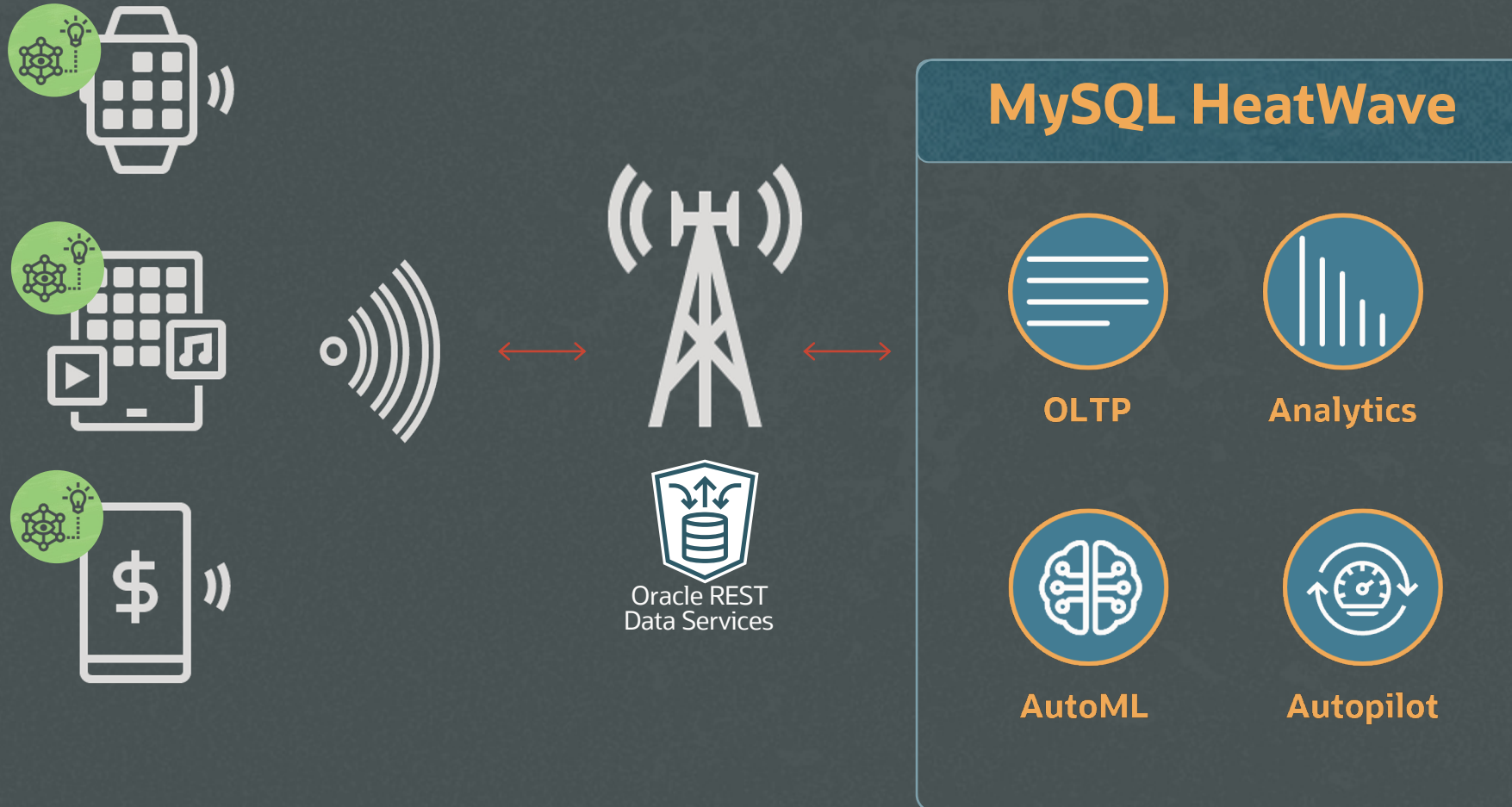
OLTP Noise test

Benchmark: Sysbench E4.1.8Gb Shape



Impact on background OLTP same as SQL procedures

REST access to MySQL HeatWave



Scalable and flexible access to MySQL HeatWave

MySQL Shell for VS Code enhanced to support new HeatWave features

The screenshot displays the MySQL Shell for VS Code interface. The left sidebar shows a tree view of databases and infrastructure. The main editor shows a SQL query and its results. A pie chart visualizes the data from the query.

Execute directly with MySQL HeatWave

Connect directly from customers' on-premises network via Bastion

MySQL HeatWave management

- Start the HeatWave Cluster
- Stop the HeatWave Cluster
- Restart the HeatWave Cluster
- Rescale the HeatWave Cluster
- Delete the HeatWave Cluster

HeatWave Cluster tuning

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINNESS	2021-09-28 22:18:53
2	NICK	WAHLBERG	2006-02-15 04:34:33
	ED	CHASE	2006-02-15 04:34:33
	JENNIFER	DAVIS	2006-02-15 04:34:33
	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
	BETTE	NICHOLSON	2006-02-15 04:34:33
	GRACE	MOSTEL	2006-02-15 04:34:33
	MATTHEW	JOHANSSON	2006-02-15 04:34:33
	JOE	SWANK	2006-02-15 04:34:33
	CHRISTIAN	GABLE	2006-02-15 04:34:33

SELECT l.name as label, count(f.film_id) as value FROM sakila.language l, sakila.film f " +
WHERE f.language_id = l.language_id GROUP BY f.language_id", (res: ResultSetRows) => {
graph = new PieGraph(PieGraph.layout.mediumPie, res);

German English
French Italian
Mandarin
Japanese



OpenTelemetry and MySQL

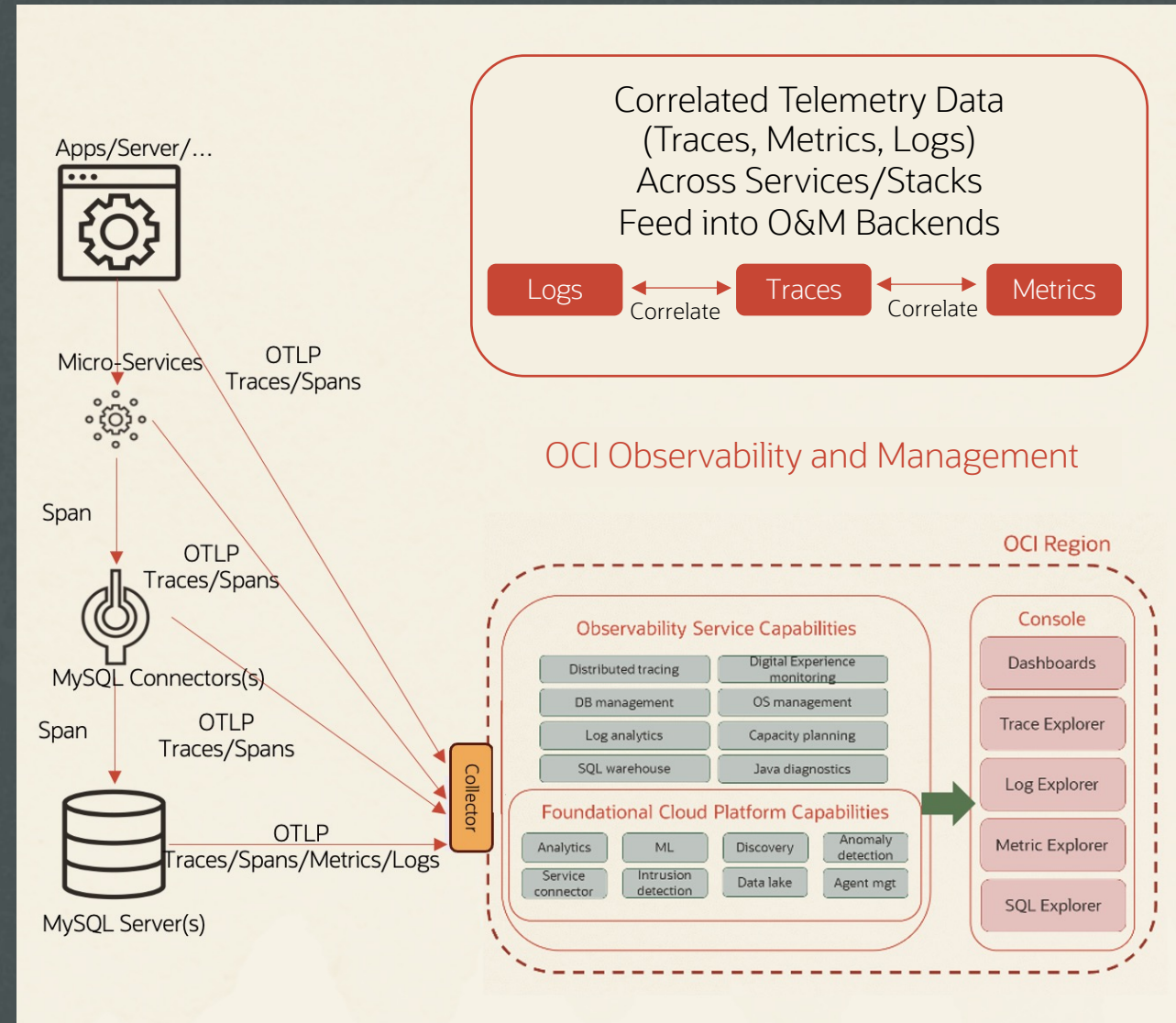
Correlates across technologies
Automates problem identification and diagnosis

OpenTelemetry (Otel)

- Open standard for telemetry data
 - Cloud Native Compute Foundation (CNCF) project
 - Oracle OCI is CNCF Platinum member
- Provides technology to collect and export telemetry
 - APIs, libraries, agents, and instrumentation
- Very Popular – second most active project

MySQL and OpenTelemetry

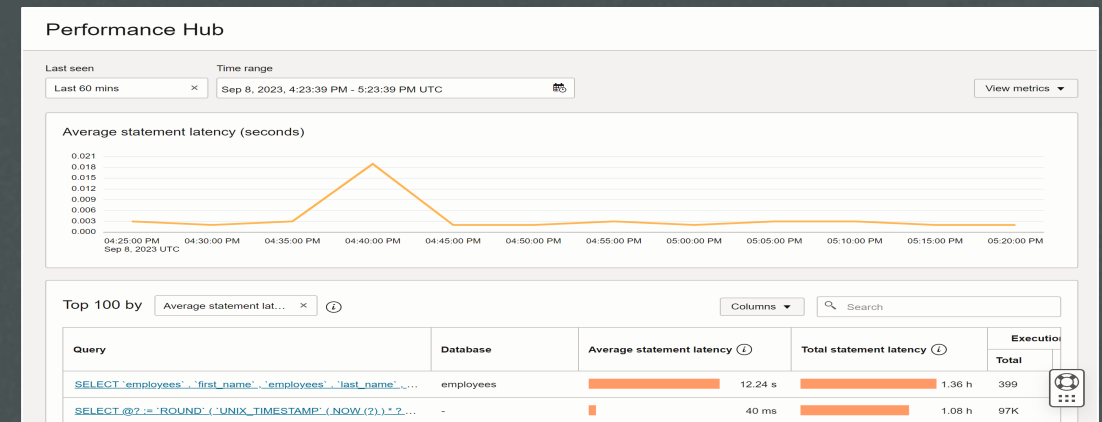
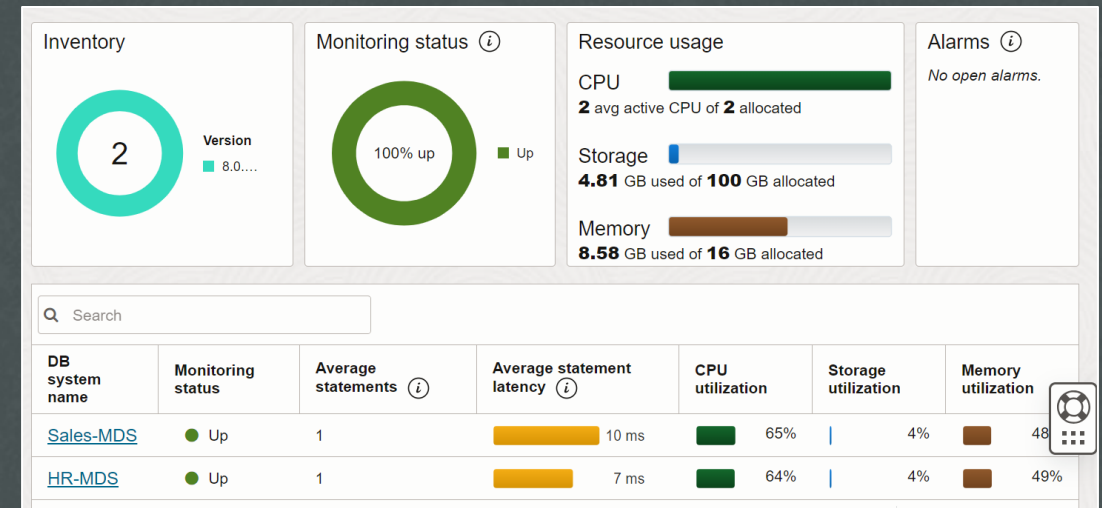
- Includes Otel libraries to emit Traces, Metrics, Logs
- Traces include Spans (unit of work context)
- Metrics – choose from 400+ metrics (meters and gauges)
- Logs - in progress for 9.0



Database Management Service

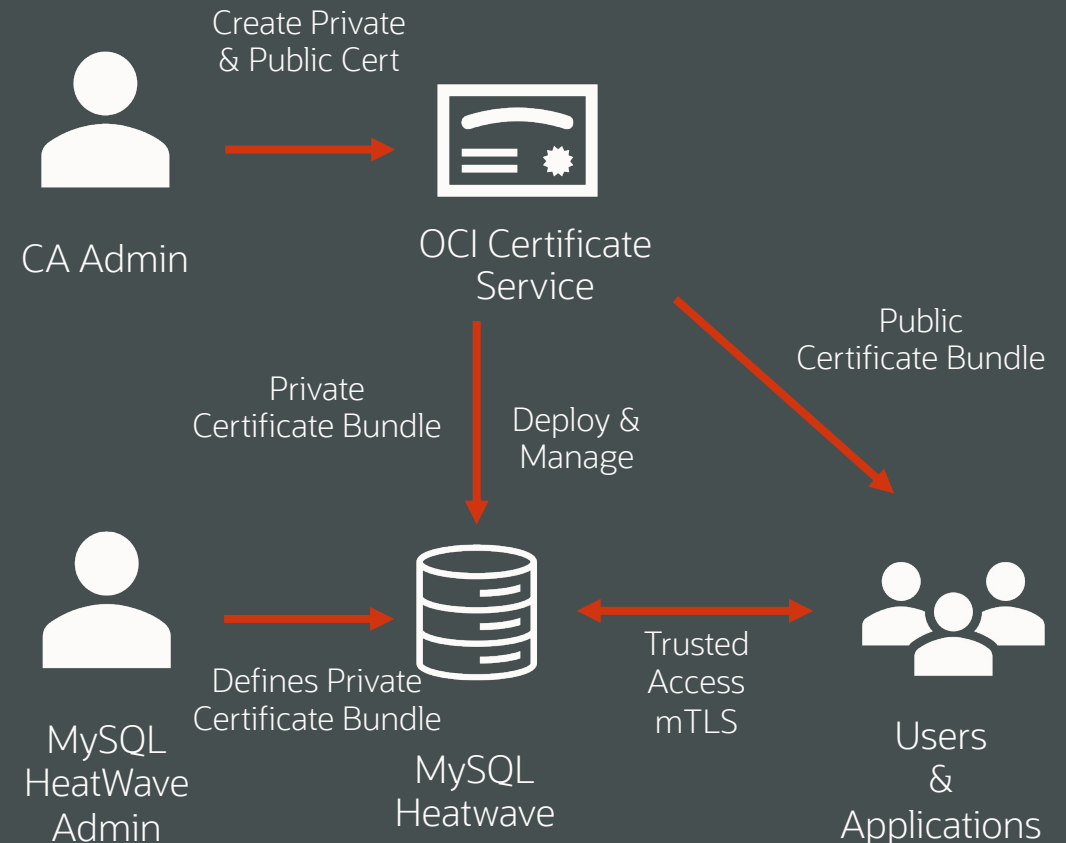
Monitoring, Diagnostics & Predictive Insights

- On-demand subscription-based cloud service
- Unified fleet monitoring and management for on-prem and cloud (MySQL Heatwave) databases
- Fleet summary page
- Visualization-driven load and performance analysis
- Forecast demand for changing workloads
- ML-driven SQL insights



Security: Bring Your Own Certificate with MySQL HeatWave

- MySQL HeatWave allows users to point to the OCI Certificate Services so users can create, provision, manage, deploy, and automatically rotate PKI certs.
- OCI Certificates Service
 - Generates and stores private keys in a **FIPS Level 3 HSM**.
 - Removes the error prone, manual process of purchasing, uploading, and renewing certs.
 - Provides a Private Certificate Authority option for customers managing their own certificate chains.
 - Provides automatic cert deployment and renewal.



Security - Open ID

Multi-Cloud, On-Prem, Hybrid Authentication

Adding support for Authentication using the OpenID Connect Standard

What is OpenID Connect

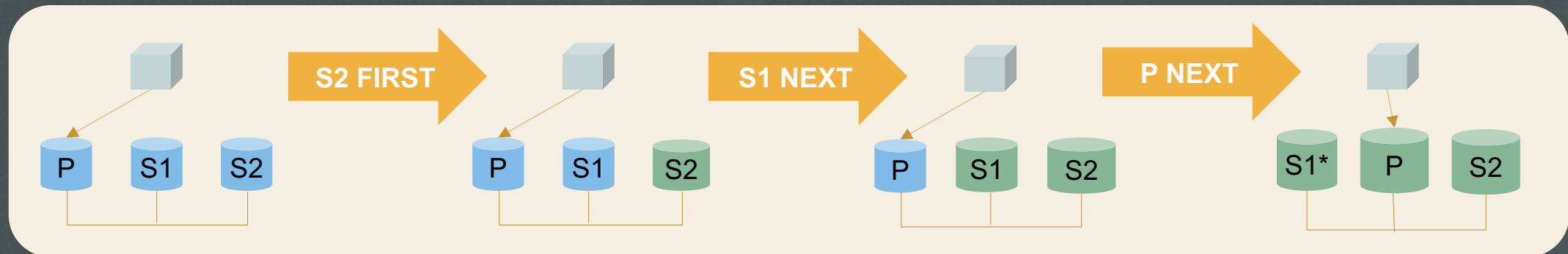
- Industry Standard
- Security protocol for verifying a users identity.
- Uses Oauth 2.0 (IETF RFC 6749 and 6750)
- Cloud application focused

Benefits

- Supports Single Sign-On
- Token based
- Secure – works with MFA, Bioauthn, ...
- Simple to configure
- Broad use cases
 - Cloud to Cloud
 - Cloud to OnPrem
 - OnPrem to Cloud
- Many providers, supporters
 - Oracle Access Management, OCI Identity
 - Oracle Fusion
 - Microsoft ADFS, Ping, Okta, ...

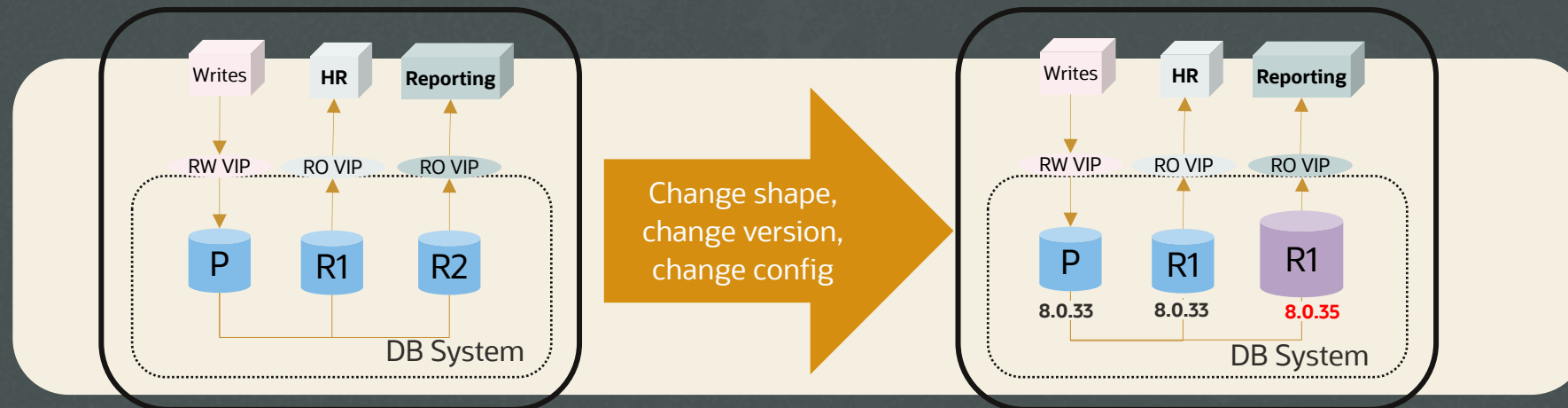
High availability improvements for OLTP workloads in MySQL HeatWave

- Improvements in downtime:
 - Planned Upgrade: **~60s → ~15s**
 - Planned Switchover: **~60s → ~7s**
- Automatically replace uncoverable instances
- Online changes to shape and configuration of a HA cluster



Read replica enhancements for OLTP workloads

Description	Summary
Change Shape	Customer can specify the shape for a specific read replica and can make it handle a specific read traffic profile that is more demanding resource-wise
Change Configuration	Customer can specify the configuration for a specific read replica and can adjust it to handle a specific read traffic profile.
Change Version	Customer can specify the version for a specific read replica and try out new features with minimal disruption. Or they may want to pin a given replica to a specific version.

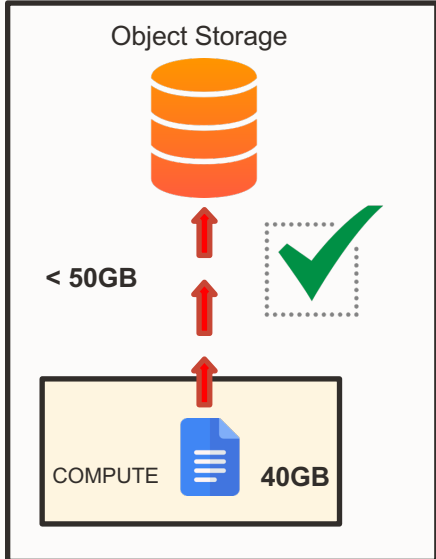


Faster Binlog Upload for PITR

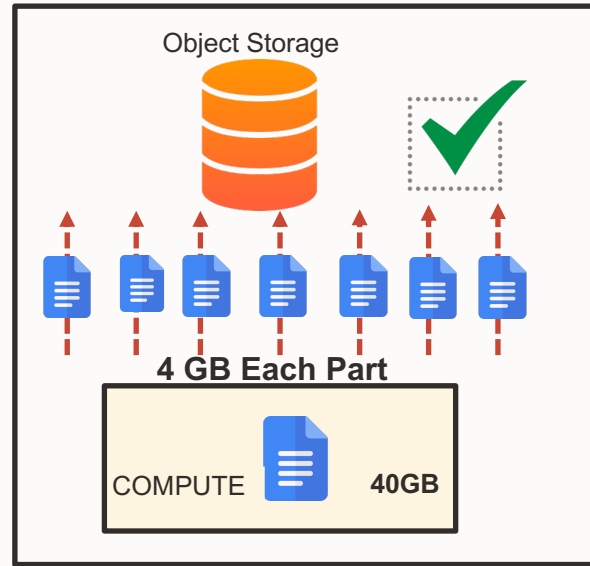


- Slow performance **increases vulnerability window** during large transactions/binlogs
- Designed multi-part upload where binlog split into multiple files and uploaded via multiple threads
 - Reduces compute bottleneck and doesn't run into the object store file size limitation

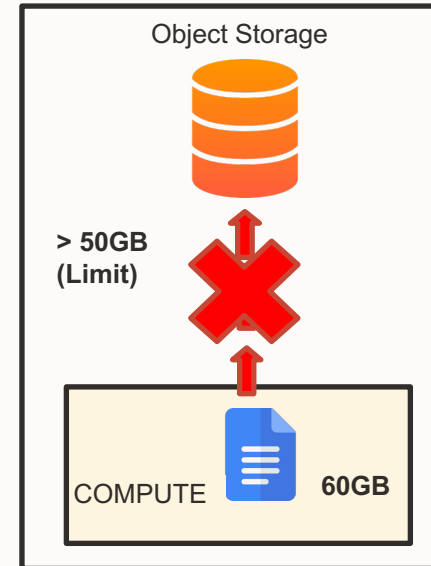
Binlog Upload Process



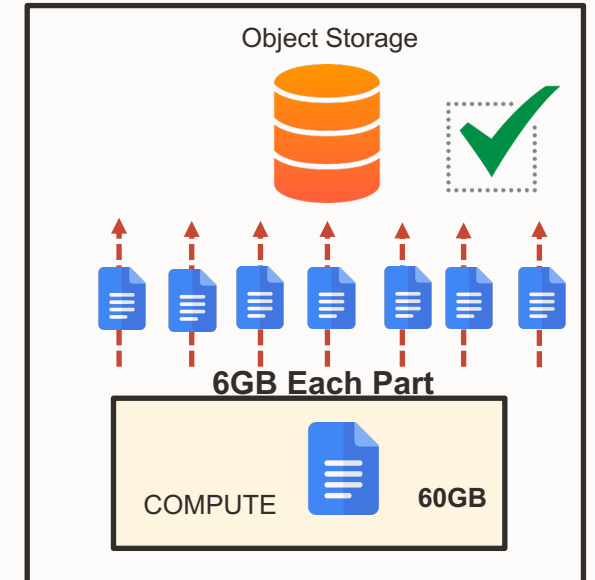
Multi-Part Binlog Upload Process



Binlog Upload Process



Multi-Part Binlog Upload Process



Binlog < 50 GB

Binlog > 50 GB



Faster Binlog Upload for PITR : upto 5x faster

- The size of each object can range from 100 MB to 25 GB based on overall binlog size
- Max 10 threads for parallel upload of smaller objects to object storage bucket
- Not limited to 50GB binlog upload

Binlog Size (GB)	Single File Upload	Multi-Part Upload
1	24 seconds	11 seconds
4	52 seconds	36 seconds
16	10 min, 2 seconds	2 min, 42 seconds
44	15 min, 10 seconds	2 min, 59 sec
800	Not possible	29 minutes

5x faster, smaller vulnerability window

Point in time recovery improvements

1

Multi-Threaded PITR: 2x better performance

VM	Applying 500GB binlog	
	Earlier	Now
4 core VM	22 hours	10 hours
8 core VM	14 hours	7 hours

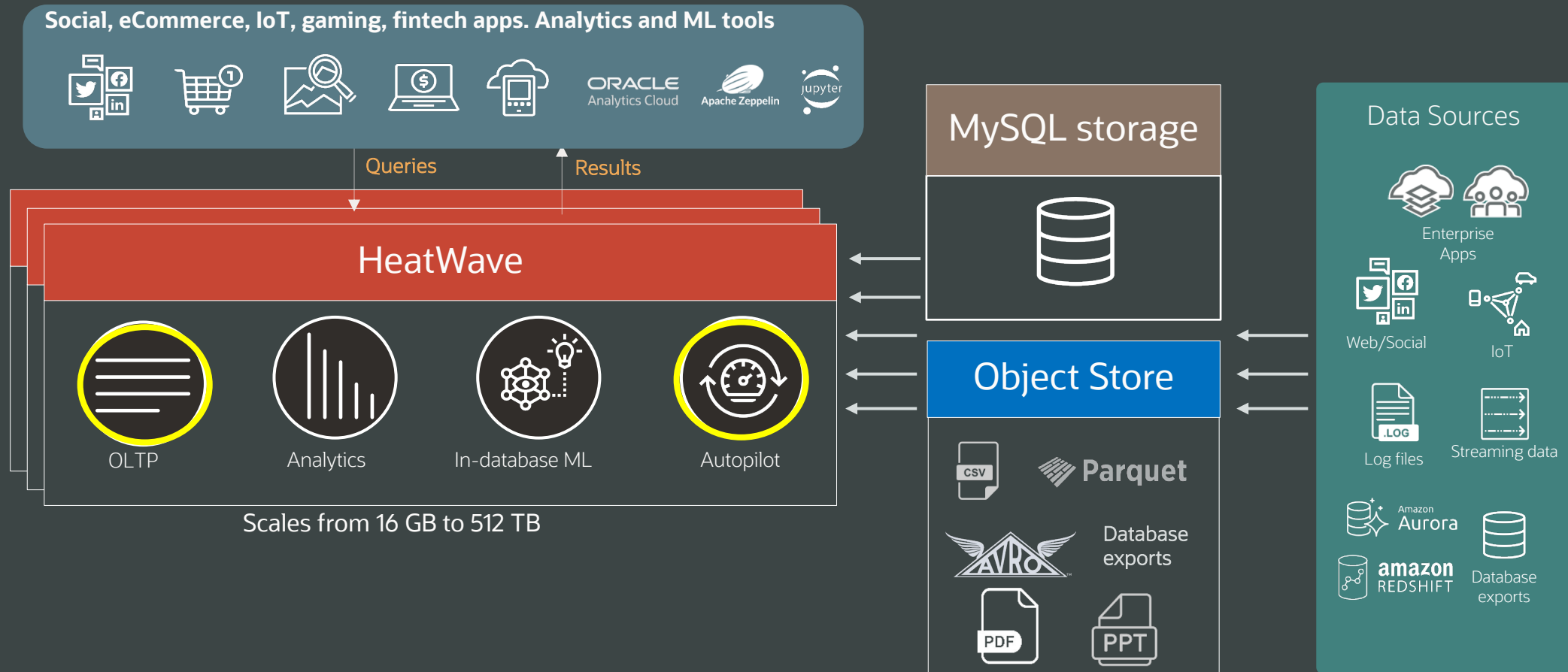
2

Improved disable PITR performance

- 34min to 3min
- Does not block other DB System operations

MySQL HeatWave – Eliminates the need for multiple services

Lakehouse, Real-Time Analytics, Machine Learning, Transaction Processing in one Database Service



ORACLE