ORACLE

MySQL HeatWave Lakehouse —Technical overview

Querying hundreds of terabytes of data in object storage with unparalleled price-performance

Table of contents

Purpose statement

This document provides an overview of features and enhancements included in MySQL HeatWave Lakehouse. It is intended solely to help you assess the benefits of MySQL HeatWave Lakehouse and to plan your IT projects.

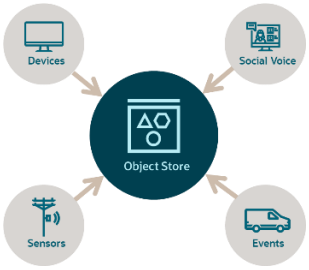
Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement, nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code. Benchmark queries are derived from the TPC-H benchmarks, but results are not comparable to published TPC-H benchmark results since these do not comply with the TPC-H specifications.

Executive Summary

MySQL HeatWave is a fully managed database service powered by the built-in HeatWave in-memory query accelerator. It delivers the best performance and price-performance in the industry for data warehouse workloads and offers fully automated in-database machine learning. It also uses machine learning to automate various aspects of the database services, which reduces the burden of tuning and database management. It is the only cloud database service that combines transactions, analytics, data lake querying, and automated machine learning into a single MySQL database system. It delivers real-time and secure analytics without the complexity, latency, and the extra cost of ETL duplication. MySQL HeatWave is available on OCI, AWS, and Azure.



HeatWave Lakehouse enables querying data in the object store in a variety of file formats, such as CSV, Parquet, Avro, and exports from databases (e.g., Aurora, Redshift, MySQL, Oracle). Customers can now query hundreds of terabytes of data in object storage and optionally combine it with transactional data in MySQL databases, *without* copying the data from the object store into the MySQL database. Users can also perform machine learning (ML) tasks, like training, predictions, and explanations on this data loaded in object storage. There is no need to load data into a database or move it to a machine-learning service, either. Querying data in the object store is as fast as querying the data in the database. MySQL HeatWave Lakehouse scales out to 512 nodes and allows customers to query up to half a petabyte of data. HeatWave Lakehouse is available on OCI and AWS.

As demonstrated by the TPC-H benchmark with 500 TB of data, the query performance of HeatWave Lakehouse is 17x faster than Snowflake, 9x faster than Amazon Redshift, 17x faster than Databricks, 36x faster than Google BigQuery. The load performance of HeatWave Lakehouse is 2x faster than Snowflake, 9x faster than Amazon Redshift, 6x faster than Databricks, and 8x faster than Google BigQuery.

Challenges Facing Lakehouse Solutions

The exponential growth of data creates several challenges that any viable lakehouse solution should meet:

- **Fast querying and scalability.** With unprecedented data growth, hundreds of terabytes of new data need to be query-ready in a very short amount of time. The ability to scale with large data volumes for both data ingestion and query performance is crucial for making informed business decisions.

- **Efficient mapping of file contents to database schema.** Users often face the burden of defining the schema for an external data source in formats like CSV and Parquet, which often evolve with the application that is generating them. The lack of automation, as well as manual processes, deteriorate the system usability and increase the tasks faced by database administrators.

- **Homogenous access to different file formats.** Data lake source files often include data generated by many applications, either in-house or external, with a variety of file formats, such as CSV, Parquet, and Avro, or even exports from other databases. A fundamental requirement of a lakehouse system is to provide uniform access to popular data file formats, with a common, SQL-like interface.

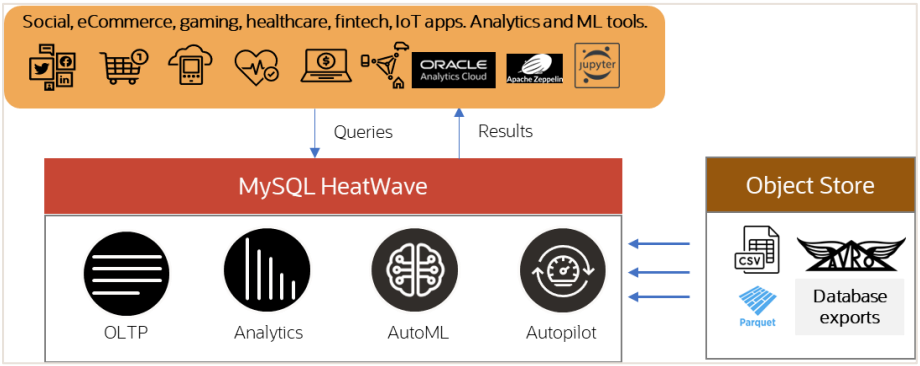
- **Converged & interoperable access to data sources.** Managing different database systems for different kinds of processing is a typical usability hindrance and requires extra data orchestration efforts across such systems. To illustrate, separate systems for OLTP, OLAP, machine learning, and yet another one for querying data in the object storage will push any post-processing across these data sources to the application level, further increasing complexity, maintainability, and data quality challenges.

- **Predictable query performance across all data sources.** Any lakehouse solution supporting a variety of data sources should not expose the complexities or limitations of the underlying data sources. The representation and challenges of a text format like CSV, a binary columnar format like Parquet, and a row binary format like Avro should be abstracted out. Developers should not be forced to learn and adapt to performance limitations (e.g., manually rewriting queries), which curtails ease of use.

- **Machine learning on object store data.** Data gathered from telemetry, sensors, IoT devices, web applications, and other sources is used to derive insights and make predictions about usage patterns, identify anomalies, perform classification, generate predictions on possible churn, upsell opportunities, trial conversions, and more. This data often needs to be enriched with additional data about users that resides in OLTP systems. It is important that a lakehouse solution not only offers machine learning capabilities on object store data, but also supports the ability to combine it with OLTP data without having to move data from one system to the other, and without requiring specialized APIs.

> "It has been a given since Big Data has been around that Big Data / Lakehouse queries are substantially slower than transactional queries. MySQL Heatwave ends that once and forever, demonstrating that Lakehouse performance can be identical to transaction query performance—unheard of and even unthinkable."
>
> **Holger Mueller**
> Vice President & Principal Analyst
> Constellation Research

Introducing HeatWave Lakehouse



HeatWave Lakehouse is designed to address the challenges facing customers listed above through its built-in HeatWave in-memory query accelerator, which combines transactions, analytics across data warehouses and data lakes, and machine learning into one service. It delivers real-time, secure analytics without the complexity, latency, and extra cost of ETL duplication. HeatWave Lakehouse provides industry-leading performance and price-performance with the following important highlights:

- **A scale-out architecture** that can ingest, manage, and execute queries at record speeds on up to 500 TBs of data with a HeatWave cluster scaling to 512 nodes.

- **MySQL Autopilot** that automates common data management tasks, including automatic schema inference for semi-structured data and auto data loading.

- **A unified query engine** for cross-querying data in the database and in the data lake. MySQL HeatWave Lakehouse automatically transforms all data sources to a unique, highly optimized internal format. A tuned internal format facilitates the optimization and execution of queries independently from the data source (data in the InnoDB storage engine or in the data lake, e.g., in CSV or Parquet format)—and sustains high and consistent performance.

- **Built-in machine learning** that makes it fast and easy to build machine learning models on data in object storage (or the MySQL database) for predictions or explanations. The same set of APIs is used to train, predict, and explain a model, irrespective of the source of data—database or object storage; and independently from the underlying format of the object storage data—CSV, Avro, Parquet, or other database export.

- **No changes are required to MySQL** as MySQL HeatWave Lakehouse remains 100% compliant with the original MySQL syntax.

- **A highly available, managed database service** that can automatically recover data loaded into the HeatWave cluster in case of an unexpected compute node failure—without retransformation from external data formats.

- **Highly efficiently cluster memory usage** by automatically compressing relevant columns— ensuring customers get the most out of their provisioned HeatWave cluster.

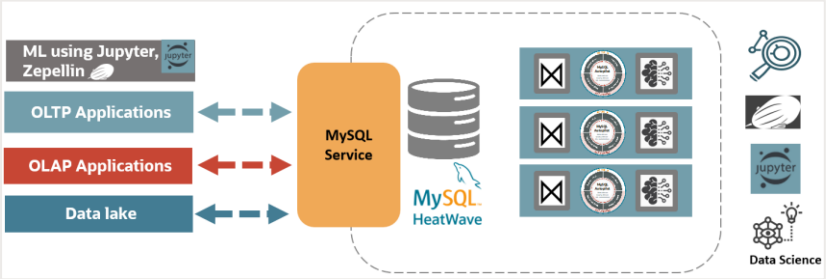
- **Full-control over access** to your data lake sources using access control mechanisms like OCI Resource Principal Authentication or Pre-Authenticated Requests (PARs). When running HeatWave Lakehouse in AWS, you define IAM roles and policies to grant access only to specific S3 data.

"HeatWave Lakehouse scales out very well for loading data from object storage and for running queries on object store. The load time and the query times are nearly constant as the size of the data grows and the HeatWave cluster size grows correspondingly. This scale out characteristic of HeatWave Lakehouse for data management is key to efficiently processing very large amounts of data."

Henry Tullis
Leader
Cloud Infrastructure & Engineering
Deloitte Consulting

End-to-End Scale-out Architecture

MySQL HeatWave Lakehouse is powered by a massively parallel, high-performance, in-memory query processing engine optimized to manage half a petabyte of data across a cluster of hundreds of compute nodes. To design a scale-out lakehouse system, we not only require query processing to scale out, but also require efficient and fast transformation and loading of semi-structured data into the HeatWave cluster memory. Once transformed into the HeatWave internal format, data in object storage can be queried by the massively parallel HeatWave in-memory query processing engine. The remaining challenge is scaling the data ingestion along with an efficient transformation of multiple file formats into hybrid columnar in-memory data representation. HeatWave Lakehouse uses a massively parallel and scalable data transformation engine that fully utilizes all the compute nodes and the CPU cores in the cluster for a truly scale-out lakehouse architecture.



MySQL HeatWave Lakehouse is meticulously optimized to efficiently scale out with increasing nodes and data sizes in the following ways:

- Scaling the distribution of data scans and transformation tasks across the cluster can be challenging when performing data-driven partitioning. MySQL HeatWave Lakehouse is optimized for avoiding any synchronization issues across compute nodes with a novel technique called super-chunking that divides the source data into smaller units of work.
- Dynamic task load balancing across the cluster avoids stragglers by ensuring that no CPU core in the cluster is left idle by distributing tasks across the nodes adaptively while observing the CPU utilization in each.
- A novel adaptive data flow mechanism on each node in the cluster independently moderates its own rate of object store requests to match the maximum rate available at any given time. The presence of this novel technique avoids excessive read requests from just one node, which may otherwise result in poor performance and scalability degradation.

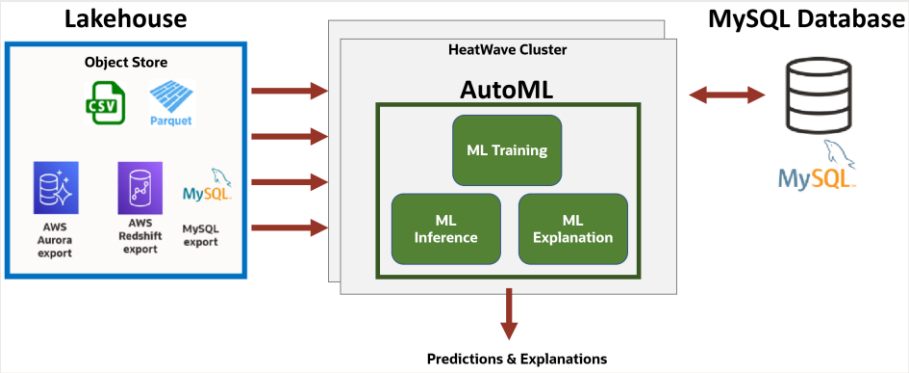
"Data is growing exponentially and so is the amount of data we store in our data lake. The ability to use standard MySQL syntax to query data across our database and object storage to get real-time insights is very important for Natura. This opens up new opportunities to explore and could represent new competitive advantages if we can analyze all this data faster than our competition."

Fabricio Rucci
Solution Architect Analyst
Natura&Co

Machine Learning and HeatWave Lakehouse

Customers can now train, predict, and explain their machine learning models on data loaded from object storage—in both OCI and AWS. HeatWave AutoML uses a common set of APIs to train, predict, and explain a model, irrespective of whether the data is in the lakehouse or the database. This simplifies tasks for the user, as they have a single, unified API to perform machine learning. Once loaded into HeatWave from object storage, users can create a model, train the model, and use this trained model to make predictions. The interactive console simplifies the process of creating models, explaining them, deriving inferences,

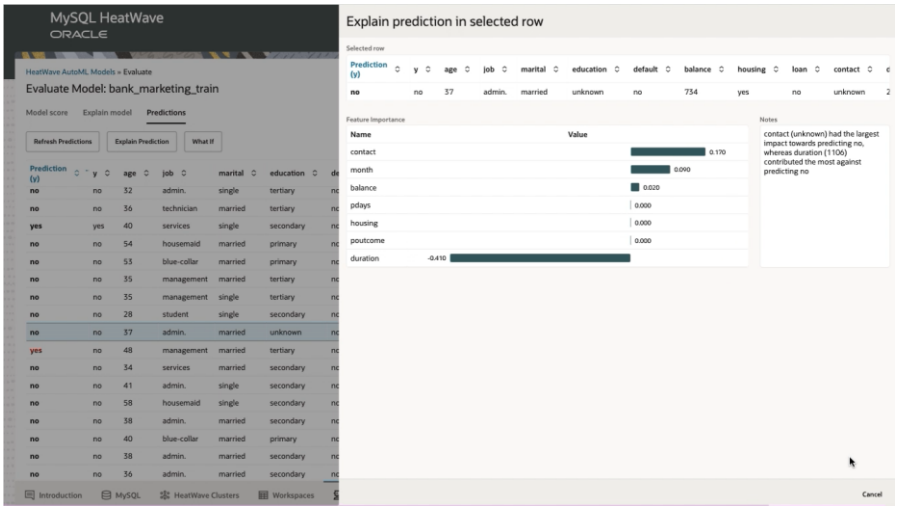
and performing scenario analysis. The console enables non-technical users to perform machine learning with ease.



In the screenshot below, a bank's marketing dataset has been trained on a classification model to predict whether the bank's marketing calls successfully led to term deposit subscriptions. Users can also run explanations on these predictions.

Both plain-text explanations and an array of attributions are displayed to aid in determining which attributes were the most impactful in making the prediction.

In the example below, while we can see that the 'duration' of the call had the largest impact against predicting a 'no', and 'contact' had the largest impact towards predicting 'no,' we also notice that the 'month' in which the call was made significantly impacted this customers decision. Calling in a different month may have swayed this customer to a 'yes' in agreeing to sign-up for a term deposit.



Explaining predictions for a selected result on object store data

"For HeatWave Lakehouse to deliver record performance for both loading data and querying data is an unprecedented innovation in cloud data services."

Ron Westfall
Senior Analyst and Research Director
Futurum Research

New MySQL Autopilot Capabilities for MySQL HeatWave Lakehouse

MySQL Autopilot provides machine learning-powered automation for MySQL HeatWave. Several existing MySQL Autopilot features have been enhanced to support HeatWave Lakehouse and new capabilities have been introduced. **Auto-**

provisioning predicts the number of required HeatWave compute nodes for running a workload and has been enhanced to support and consume files directly from the object store. **Auto query plan improvement** learns various run-time statistics from the execution of queries to further improve the execution plan of unique queries in the future. **Auto parallel loading** analyzes data to predict the load time into HeatWave and loads data efficiently from the object store with a high degree of parallelism.

New and enhanced capabilities introduced in MySQL Autopilot for MySQL HeatWave Lakehouse include:

- **Auto-schema inference** samples a small fraction of data in object storage and infers the number of columns, the data types, and the precision of these columns. This is particularly advantageous when working with CSV files that do not contain any metadata.
- **Adaptive data sampling** intelligently samples files to derive information needed for automation and the nature of the data in question. Using these novel techniques, MySQL Autopilot can scan and propose schema predictions on a set of data files totalling 500 TBs in under one minute.
- **Adaptive data flow** learns and coordinates network bandwidth utilization to the object store across a large cluster of nodes, dynamically adapting to the performance of the underlying object store, resulting in optimal performance and availability.
- **Auto query plan improvement**: MySQL Autopilot learns query and data statistics from previously executed queries, which improves the optimizer statistics, and, therefore, subsequent query execution plans.

> "Simply put: MySQL HeatWave Lakehouse enables you to stay ahead of the competition by taking swift action on meaningful business insights."
>
> **Steve McDowell**
> Principal Analyst & Founding Partner
> NAND Research

When it comes to data lakes, common file formats may not be structured, and often it is not trivial to define strict data models for such data sources. Specifically, CSV is a good example of a semi-structured file format where the column types are not pre-defined in the file. Without prior knowledge or insight from the data, users often choose conservative data types and sizes that would be wasteful or lead to sub-optimal query performance (e.g., using varchar for all types). With MySQL Autopilot, this process is now fully automated and data-driven, eliminating user guesswork.

All these intelligent optimizations by MySQL Autopilot are interactive, even for large data sizes (as large as 500TB), and use an efficient adaptive sampling algorithm on a relevant subset of the underlying data to make suggestions.

Deployment and Use Case Scenarios

To best understand the capabilities and usability of our managed service, we will walk through a deployment scenario that is uniquely possible with MySQL HeatWave Lakehouse. The deployment goal here is to have the following tables managed and be query-ready in MySQL HeatWave Lakehouse:

- *Table inside database:* **Sales** is a traditional MySQL transactional table managed by the InnoDB engine and loaded into the HeatWave cluster. This

table is frequently updated by many cloud applications. Any change done to this table through InnoDB is propagated in real-time and is readily available in the HeatWave cluster for queries.

- *Object store files:* **Sensor** is a CSV file generated by an application. **SensorInventory** contains the data exported from an Amazon Aurora database as a Parquet file which has been uploaded to the object store.

Let us assume that all OLTP tables are already managed by MySQL HeatWave, with the Lakehouse feature enabled. You will provide MySQL HeatWave Lakehouse access to the objects in the object storage. This can be done with two access control methods: OCI Resource Principal mechanism or PAR.

To start using these this external data as external tables, users need to:

- Define the schema of the external tables. To do this, run MySQL Autopilot on data in the Object Store.

```
mysql> CALL
sys.heatwave_load(<db_names>,<info_about_file_in_ObjectStore>);
```

MySQL Autopilot runs and provides the DDL for the Sensor table.

- The table is created by running the DDLs returned by MySQL Autopilot:

```
mysql> CREATE TABLE Sensor
            (`id` INT NOT NULL,
             `date` DATE NOT NULL,
             `temperature` INT NOT NULL)
             ENGINE=lakehouse
             SECONDARY_ENGINE=RAPID
             ENGINE_ATTRIBUTE='{"file": [{"par": "<PAR URL>"}],
                    "dialect": {"format": "csv"…}}';
```

- Load the data from Object Store into HeatWave.

```
mysql> ALTER TABLE Sensor SECONDARY_LOAD;
```

Just as the **Sensor** table was loaded into HeatWave, the **SensorInventory** Amazon Aurora table exported to and copied over to object storage can also be loaded into HeatWave. With the two new external tables now loaded, users and developers can use the familiar MySQL syntax to construct queries:

```
mysql> SELECT count(*) FROM Sensor, SALES
            WHERE Sensor.degrees> 30 AND Sensor.id= SALES.id;
```

- Such queries are not only limited to InnoDB and external tables but also work across different external tables in different file formats, e.g., a join between the **Sensor** and **SensorInventory.**

In all the above scenarios, customers do not need any lengthy ETL processes between disparate systems, nor do they require the cloud application to be aware of the different data sources.

"MySQL HeatWave, now with Lakehouse, may be the most significant open-source cloud database innovation in the last decade."

Marc Staimer
Senior Analyst
Wikibon

HeatWave Lakehouse Performance and Price-Performance

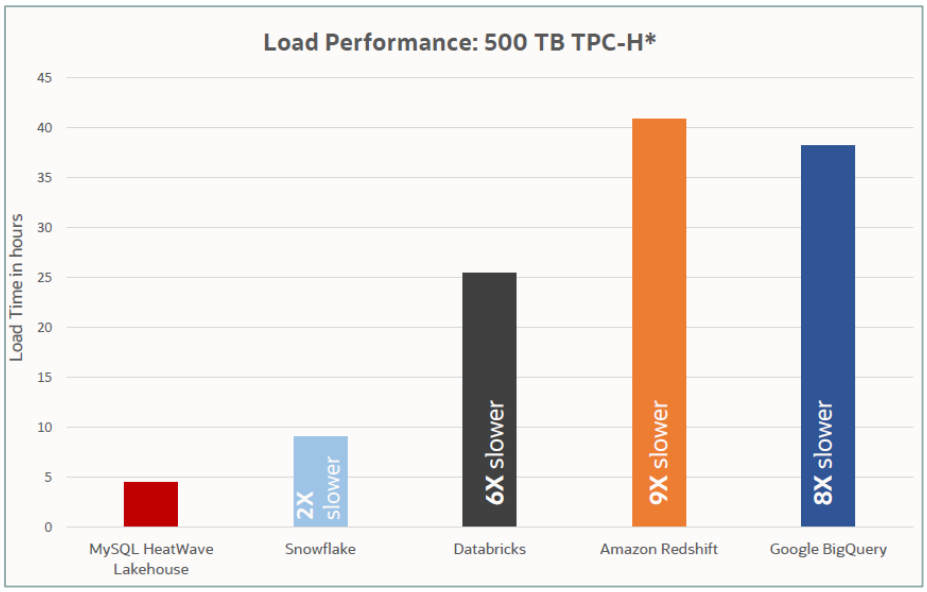
A MySQL HeatWave whitepaper would be incomplete without published benchmark results. The benchmark is designed to answer common questions customers face when switching to a new service:

- How fast can we ingest data-lake scale data (e.g., 500TB)?
- Is it fast enough to load new data every day?
- How does query performance and price-performance compare to other services?
- Is the query engine truly unified? Do the query runtimes vary based on the data source (data warehouse vs. data lake)?

Load Performance

The following MySQL HeatWave Lakehouse benchmark results answer these questions:

Load 500TB of data in the object store in 4 hours



**Benchmark queries are derived from the TPC-H benchmarks, but results are not comparable to published TPC-H benchmark results since these do not comply with the TPC-H specifications.*

Configuration: MySQL HeatWave Lakehouse: 512 nodes; Snowflake: 4X-Large Cluster; Databricks: 3X-Large Cluster; Amazon Redshift: 20-ra3.16xlarge; Google BigQuery: 6400 slots

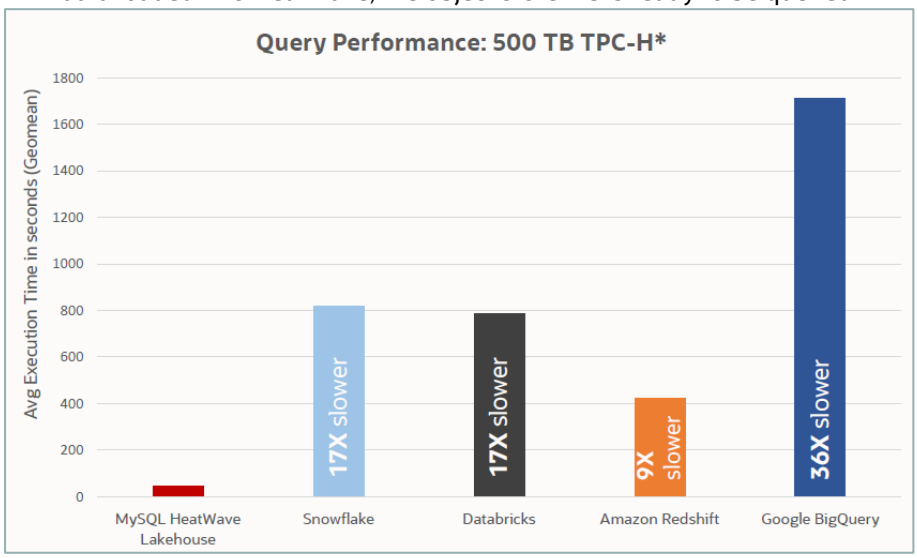
The load performance of MySQL HeatWave Lakehouse is:

- 9x faster than Redshift
- 2x faster than Snowflake
- 6x faster than Databricks
- 8x faster than Google BigQuery

Such record speed is possible because of the scale-out architecture of our processes that perfectly partition and balance tasks and utilize all the available CPU cores to get external files query-ready, guaranteeing that all the 512 nodes in the cluster are used in-tandem, ensuring massive scalability.

Query Performance

With data loaded into HeatWave, the object store file is ready to be queried.



**Benchmark queries are derived from the TPC-H benchmarks, but results are not comparable to published TPC-H benchmark results since these do not comply with the TPC-H specifications.*

Configuration: MySQL HeatWave Lakehouse: 512 nodes; Snowflake: 4X-Large Cluster; Databricks: 3X-Large Cluster; Amazon Redshift: 20-ra3.16xlarge; Google BigQuery: 6400 slots

As demonstrated by the 500 TB TPC-H benchmark, the query performance of MySQL HeatWave Lakehouse is:

- 9x faster than Amazon Redshift, delivering 8x better price-performance
- 17x faster than Snowflake, delivering 22x better price performance
- 17x faster than Databricks, delivering 18x better price performance
- 36xfaster than Google BigQuery, delivering 30x better price performance

HeatWave offers over an order of magnitude faster query performance compared to other analytic databases due to the following reasons:

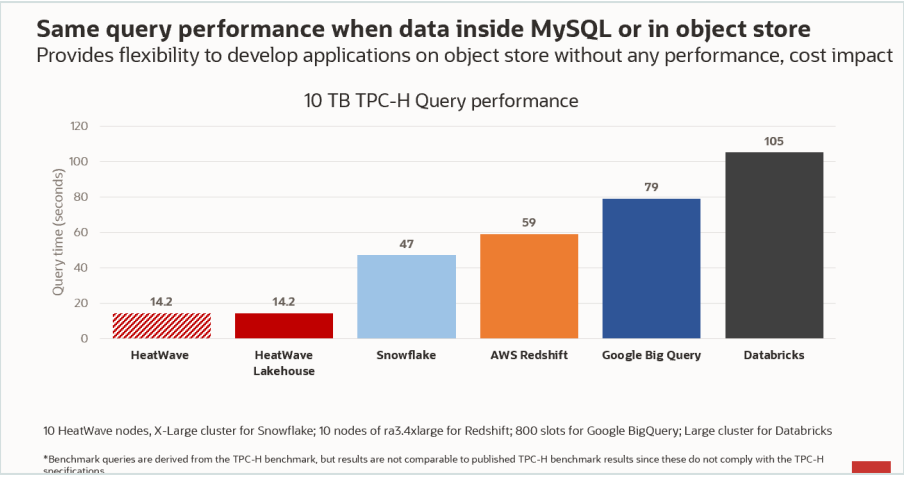
- The MySQL HeatWave query engine is massively parallel and highly scalable, and fully utilizes each CPU core in the cluster.

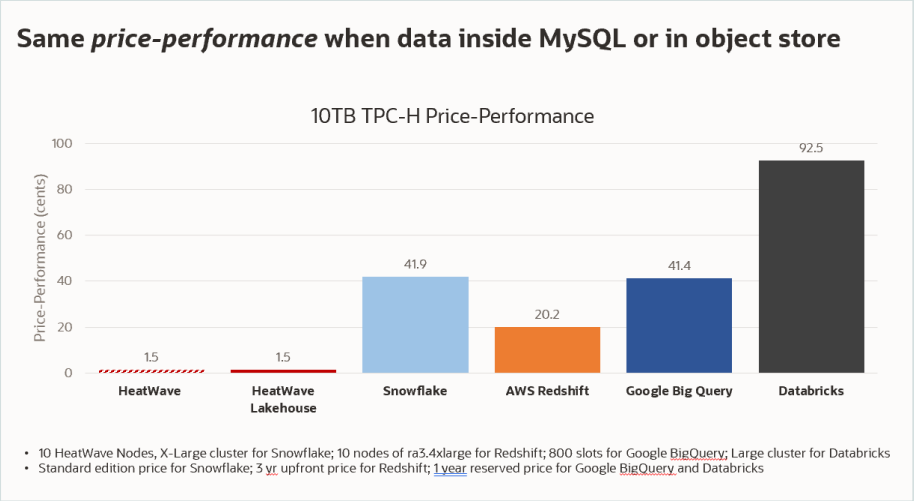
- With assistance from MySQL Autopilot, the system accurately identifies the data type for each column in the semi-structured dataset, which in turn improves the query processing performance.

- MySQL Autopilot learns various run-time statistics from previous queries that have been executed and improves the execution time for new queries.

Identical performance and price-performance for querying from object storage and database

The performance and cost of querying data in the object store is identical to the performance and cost of querying data inside the database. This is demonstrated by the performance and price-performance of a 10TB TPCH workload.





Conclusion

With the data deluge outside of databases (social media files, data from IoT sensors, connected devices, web application telemetry, and other sources) businesses want to rapidly generate new insights and apply machine learning operations to train their data, make predictions, and explain results. With HeatWave Lakehouse, customers can leverage all the benefits of HeatWave and the convenience of familiar MySQL commands on data residing in object storage. As demonstrated by the 500 TB TPC-H benchmark, MySQL HeatWave Lakehouse delivers superior query performance, price-performance, and load performance compared to other available offerings. MySQL HeatWave now provides a single, fully-managed service for transaction processing, analytics across data warehouses and data lakes, and machine learning—without ETL across cloud services. It is available on OCI and AWS.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle