ORACLE

MySQL HeatWave Lakehouse – Technical overview

Process and query hundreds of terabytes of data in the object store

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

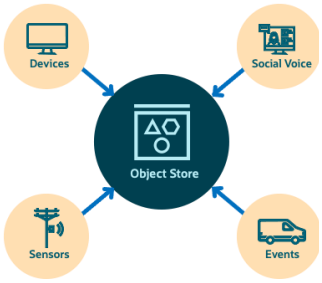
This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed 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 integrated HeatWave in-memory query accelerator. It's the only cloud database service that combines transactions, analytics, and machine learning services into one MySQL Database, delivering real-time, secure analytics without the complexity, latency, and cost of ETL duplication.

MySQL HeatWave provides excellent performance and price performance for analytics, machine learning, and OLTP processing for data stored inside the database. However, we're facing an unprecedented growth of data in files that reside outside of the database. IDC predicted that generated data will more than double, from 79 ZB in 2021 to 180 ZB in 2025, and 99.5% of all data remains unused (Grow.com) due to the unavailability of efficient services to process it.



MySQL HeatWave expands to include MySQL HeatWave Lakehouse, letting users process and query hundreds of terabytes of data in the object store—in a variety of file formats, such as CSV, Parquet, and Aurora/Redshift backups. Customers can query transactional data in MySQL databases, data in various formats in object storage, or a combination of both using standard MySQL commands. Querying the data in the database is as fast as querying data in the object store. The HeatWave cluster scales to 512 nodes, letting customers query up to 400 TB of data with MySQL HeatWave Lakehouse. As demonstrated by 400 TB TPC-H benchmarks, the query performance of MySQL HeatWave Lakehouse is 17X faster than Snowflake and 6X faster than Amazon Redshift; the load performance is 8X faster than Amazon Redshift and 2.7X faster than Snowflake.

MySQL HeatWave Lakehouse is now available in Beta for customers to try and is slated for general availability in 1HCY23.

Challenges Facing Lakehouse Solutions

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

1. **Fast query readiness and scalability.** With unprecedented data growth, it is expected that 100s of terabytes of new data need to be query-ready in a very short time, without requiring a day-long (or more) ETL process. The ability to scale with large data sizes for both data ingestion and query performance is crucial to rapidly identify and react to evolving market trends.

2. **Efficient transformation of semi-structured data to SQL schema definitions.** Users often face the burden of defining the schema for an external data source in semi-structured formats like CSV, which often evolve with the application that is generating them. Without proper insights into the data, developers often make conservative choices that might degrade the performance of downstream query processing.

3. **Heterogeneous data file formats.** External data lake sources often include data generated by many applications, either in-house or external, with a variety of file formats, such as CSV, Parquet, and Avro. Any lakehouse platform should be able to support popular formats and provide a common SQL-like interface over them.

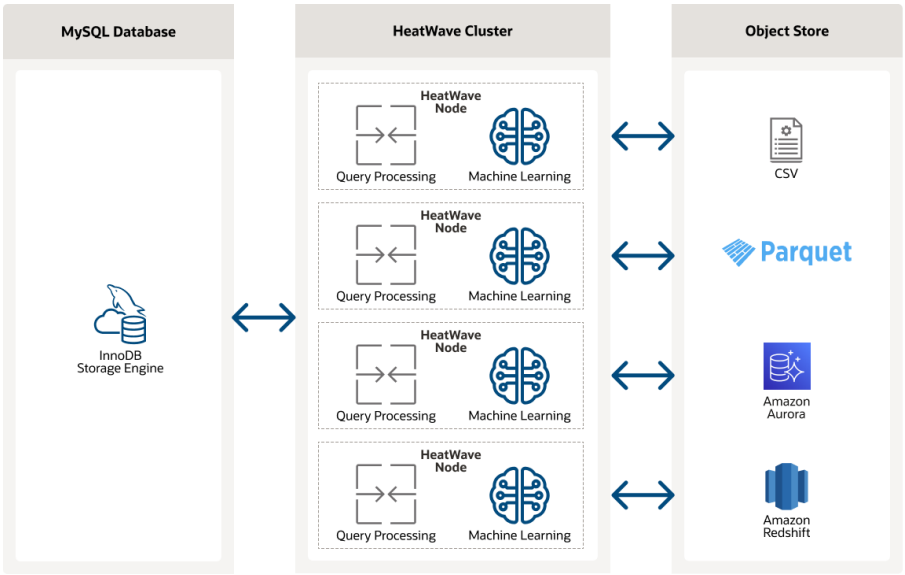
4. **Interoperability across data sources.** Managing different database systems for different data sources will severely limit usability and require data orchestration across such systems. For example, separate systems for OLTP and OLAP plus yet another one for querying data in object storage will push any post-processing across these data sources to the application level.

5. **Predictable query performance across all data sources.** Any Lakehouse solution supporting a variety of data sources should not expose complexities or limitations for specific data sources. Without proper abstractions, developers are forced to learn and adapt to such performance limitations, including by rewriting queries.

"MySQL HeatWave Lakehouse sets the competition on fire by blazing the trail to the previously uncharted territory of 400TB cloud database benchmarks at breakneck speeds."

Ron Westfall
Senior Analyst and Research Director
Futurum Research

Introducing MySQL HeatWave Lakehouse



MySQL HeatWave Lakehouse is designed to address the challenges typically facing customers. In addition to all the MySQL HeatWave benefits, MySQL HeatWave Lakehouse provides the following:

1. **A scale-out architecture** that can ingest, manage, and execute queries at record speeds for up to 400 TB of data with a HeatWave cluster scaling to 512 nodes.

2. **MySQL Autopilot** automates common data management tasks, including automatic schema inference for semi-structured data and auto loading, which we will review later in this document.

3. **A unified query engine** for data in the database and in the data lake. MySQL HeatWave Lakehouse automatically transforms all data sources to a single optimized internal format. This provides the ability to optimize and execute queries regardless of the data source (data in the InnoDB storage engine or in the data lake, for example in CSV and Parquet format)—and get high, consistent performance.

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

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

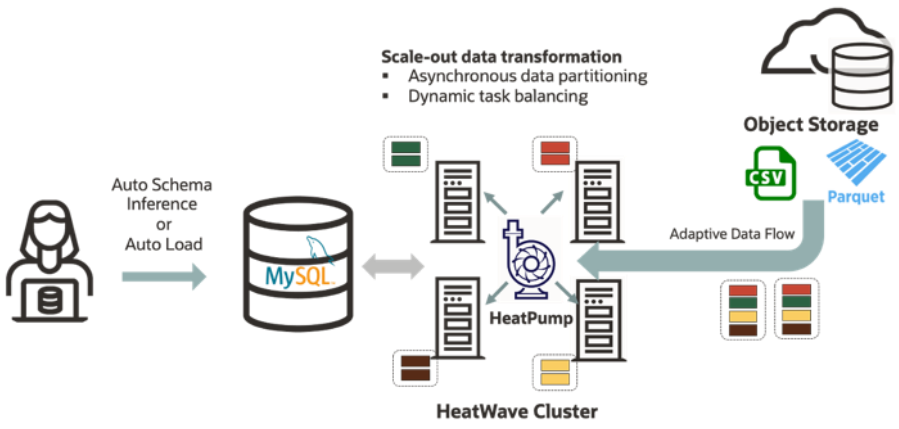
6. **Highly efficiently cluster memory usage** by automatically compressing relevant columns, providing a compression ratio of up to 2x— ensuring customers get the most out of their provisioned HeatWave cluster.

7. **Full-control over access** to your data lake sources using secure access control methods like Pre-Authenticated Request (PAR) or OCI Resource Principal mechanism.

> "MySQL HeatWave Lakehouse supports cloud databases in excess of 400TB, demonstrating continued innovation at a compelling price point for customers worldwide."
>
> **Marc Staimer**
> Senior Analyst
> Wikibon

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-scale data size across a cluster of nodes. To design a scale-out lakehouse system, we not only require query processing to scale out but also the loading and transformation of semi-structured data into HeatWave's hybrid columnar format. Once transformed into the HeatWave internal format, external data can be used by the HeatWave massively parallel in-memory query processing engine. The remaining challenge is scaling data ingestion and efficient transformation of multiple file formats into hybrid columnar in-memory data representation. To this end, we designed HeatPump, a massively parallel and scalable data transformation engine that fully utilizes all the nodes and cores in the cluster to provide a truly scale-out lakehouse architecture.



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

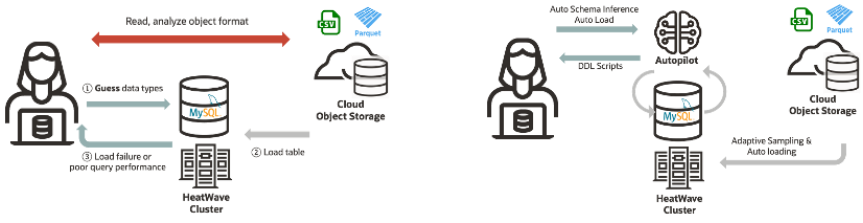
- Scaling distribution of data read and transformation tasks across the cluster can be challenging when doing data-driven partitioning. HeatPump is optimized to avoid any synchronizing across nodes with a process called super chunking that introduces virtual partitions.
- Dynamic task load balancing across the cluster to avoid stragglers by making sure that no core in the cluster is left idle, stealing tasks from nodes lagging behind.
- Adaptive data flow control that coordinates network bandwidth utilization to the object store across a large cluster of nodes. Without adaptive data flow control, excessive read requests from just one node result in poor scalability.

> "MySQL HeatWave has become a very competitive cloud database, in little time, offering CxOs a powerful universal MySQL database for their next generation applications."
>
> **Holger Mueller**
> Vice President and Principal Analyst
> Constellation Research

New MySQL Autopilot Capabilities for MySQL HeatWave Lakehouse

MySQL Autopilot provides machine learning-based automation for MySQL HeatWave. Existing MySQL Autopilot capabilities such as auto provisioning and auto query plan improvement have been enhanced for MySQL HeatWave Lakehouse, which further reduce database administration overhead and improve performance.

When it comes to data lakes, common data lake 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 where the column types are not pre-defined in the file. Without prior knowledge or insights from the data, users often choose conservative data types and sizes that would be wasteful or provide sub-optimal query performance (e.g. using varchar for all types). With MySQL Autopilot, this process is now fully automated and data-driven, eliminating guesses from users (described below).



A number of new MySQL Autopilot capabilities are now available for MySQL HeatWave Lakehouse.

1. **Auto schema inference**: Autopilot automatically infers the mapping of the file data to datatypes in the database. As a result, customers don't need to manually specify the mapping for each new file to be queried by MySQL HeatWave Lakehouse—thereby saving time and effort.

2. **Adaptive data sampling**: Autopilot intelligently samples portions of files in object storage, collecting accurate statistics with minimal data access. MySQL HeatWave uses these statistics to generate and improve query plans, determine the optimal schema mapping, and other purposes.

3. **Auto load**: Autopilot analyzes the data to predict the load time into MySQL HeatWave, determines the mapping of the datatypes, and automatically generates the loading scripts. Users don't have to manually specify the mapping of files to database schemas and tables.

4. **Adaptive data flow**: MySQL HeatWave Lakehouse dynamically adapts to the performance of the underlying object store. As a result, MySQL HeatWave can get the maximum available performance from the underlying cloud infrastructure which improves overall performance, price performance, and availability.

All these intelligent optimizations by MySQL Autopilot are interactive, even for large data sizes, using an efficient adaptive sampling of a relevant subset of data to make suggestions.

Deployment and Use Case Scenarios

To best understand the capabilities and ease of use 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:

- *OLTP Tables: LineItem, Orders* are traditional MySQL transactional tables managed by the InnoDB engine and loaded into the HeatWave cluster. The table is frequently updated by many cloud applications. Any change done through InnoDB is propagated in real-time and is readily available in the HeatWave cluster for queries.
- *External Table: Sales* is an external data lake table that is constructed from a collection of CSV files generated by a different application. The size of these files ranges from a few tens of TB to hundreds of TB in size. These files are all placed within the OCI Object Store.
- *External Table: AdsClickTraffic* contains the data exported from an Amazon Aurora database as multiple Parquet files on OCI Object Store.

Let's assume that all OLTP tables are already managed by MySQL HeatWave, which is Lakehouse-enabled. To start using these external and exported tables, users need to:

- Give MySQL HeatWave Lakehouse access to the objects in object storage. This can be done with two secure access control methods: OCI Resource Principal mechanism or PAR.
- Define the schema of the external tables. Here we seek the help of MySQL Autopilot, which returns a fully defined CREATE TABLE statement like this:

```
mysql> CREATE TABLE Sales(`col_0` bigint NOT NULL, `col_1`
decimal(8,2) NOT NULL, `col_2` date NOT NULL, `col_3`
varchar(43) NOT NULL) ENGINE=datalake SECONDARY_ENGINE=RAPID
ENGINE_ATTRIBUTE='{"file": [{"par": "https://objectstorage.us-
ashburn-1.oraclecloud.com/.../o/sales"}], "dialect": {"format":
"csv", "skip_rows": 0, "field_delimiter": "|",
"quotation_marks": "\\"", "record_delimiter": "|\\n"}}';
```

- Get the external tables query ready. Again, MySQL Autopilot helps by suggesting an expanded cluster size for the new tables, and automatically loads them. This is not limited to external tables, and includes existing tables in InnoDB.

With the two new external tables now loaded, users and developers can use the familiar MySQL syntax to construct queries like (but not limited to):

- Joins between OLTP tables and any of the two External Tables. Note that with OLTP tables all changes are readily available to query. For example, queries to get real-time sales recommendations based on recent customer orders and historic *Sales* activities generated by a 3rd party application (in CSV). An example simple query:

```
mysql> SELECT count(*) FROM ORDER, SALES WHERE ORDER.date =
SALES.col_2;
```

- 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 *Sales* and *AdsClickTraffic*.

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

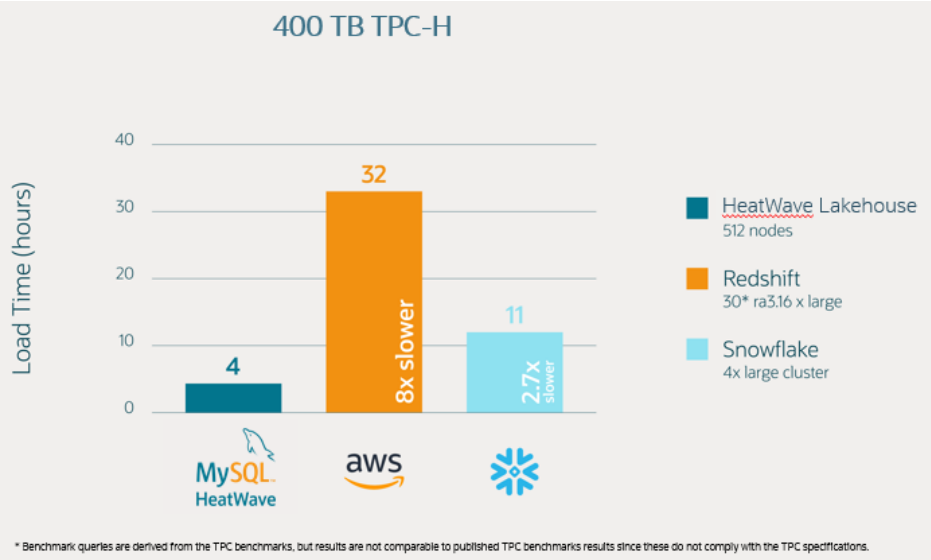
Record MySQL HeatWave Lakehouse Performance

A MySQL HeatWave whitepaper would not be complete 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. 400TB)?
- Is it fast enough to load new data every day?
- How does query performance compare to other services?
- Is the query engine truly unified? Do the query runtimes vary based on the data source (data warehouse vs. lake)?

The following MySQL HeatWave Lakehouse benchmark results answer those questions:

Load 400TBs of data in the object store in 4 hours



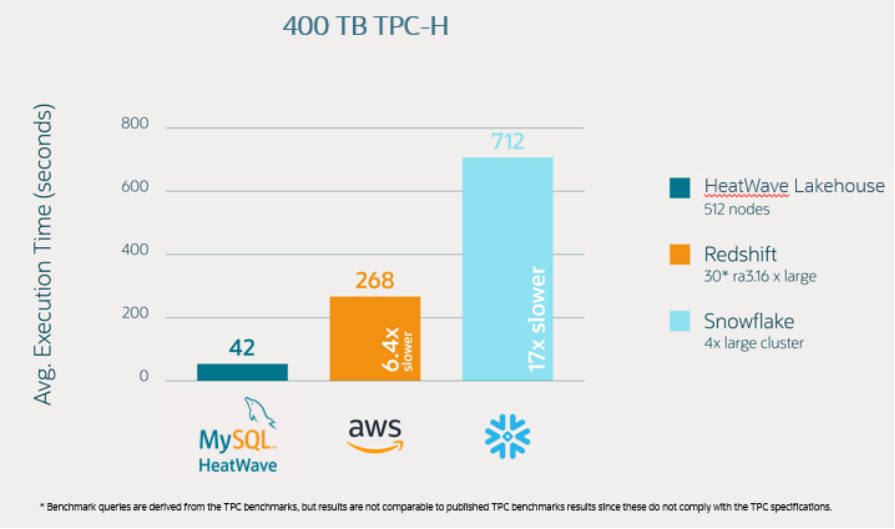
As demonstrated by a fully transparent, publicly available 400 TB TPC-H* benchmark, the load performance of MySQL HeatWave Lakehouse is:

- 8X faster than Amazon Redshift
- 2.7X faster than Snowflake

Such record speed is possible because of the scale-out architecture of our HeatPump process that perfectly partitions and balances tasks, and utilizes every available core to get external files query-ready. HeatPump guarantees that all the 512 nodes in the cluster are used in-tandem ensuring strong scalability.

Run queries on 400TB - on average in 42 seconds

With data converted to our proprietary hybrid-columnar format, the external tables are ready to be queried.



As demonstrated by the 400 TB TPC-H benchmark, the query performance of MySQL HeatWave Lakehouse is:

- 17X faster than Snowflake
- 6X faster than Amazon Redshift

We observe orders of magnitude faster query performance, even for large-scale data-lakes, for three main reasons:

1. The MySQL HeatWave query engine is massively parallel and highly scalable, fully utilizing each core in the cluster.
2. With assistance from MySQL Autopilot, we earlier had accurately identified the data type for each column in the semi-structured dataset, which in turn improves query processing performance.
3. Even though we maintain all the data in memory on a large cluster, we significantly compress our data and rely on highly optimized software to use commodity hardware. This significantly reduces the annual cost of ownership of the cluster compared to other existing solutions.

Get a truly unified query execution engine

Querying the data in the database is as fast as querying data in the object store, as demonstrated by 10 TB and 30 TB TPC-H benchmarks.

TPC-H/ Query Times	Warehouse	Data lake
10TB	16 secs	16 secs
30TB	18 secs	18 secs

Conclusion

We are facing a huge growth in data stored outside of databases (social media files, data from IoT sensors…etc.) that businesses want to use to rapidly generate new insights. With MySQL HeatWave Lakehouse, customers can leverage all the benefits of HeatWave on data residing in object store. As demonstrated by a 400 TB TPC-H benchmark, the query performance of MySQL HeatWave Lakehouse is 17X faster than Snowflake and 6X faster than Amazon Redshift. Loading data into MySQL HeatWave Lakehouse is also significantly faster. The load performance of MySQL HeatWave Lakehouse is 8X faster than Amazon Redshift and 2.7X faster than Snowflake. MySQL HeatWave now provides one service for transaction processing, analytics across data warehouses and data lakes, and machine learning—without ETL across cloud services.

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