ORACLE

MySQL HeatWave

Integrated, automated, and secure generative AI and machine learning in one fully managed cloud service for transactions and lakehouse scale analytics—available in multiple clouds.

Copyright © 2025, Oracle and/or its affiliates
Public

Purpose statement

This document provides an overview of features and enhancements included in MySQL HeatWave. It is intended solely to help you assess the benefits of MySQL HeatWave and to plan your I.T. 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 and TPC-DS benchmark, but results are not comparable to published TPC-H and TPC-DS benchmark results since they do not comply with the TPC-H TPC-DS specification.

Copyright © 2025, Oracle and/or its affiliates / Public

Table of contents

Copyright © 2025, Oracle and/or its affiliates / Public

Executive Summary

MySQL HeatWave provides automated, integrated, and secure generative AI and machine learning (ML) in one cloud service for transactions and lakehouse scale analytics—without the complexity, latency, risks, and cost of ETL duplication. The service overcomes the limitations of traditional data warehouse, analytics, lakehouse, machine learning, Generative AI and vector store environments that use periodic long-running ETL batch jobs to refresh the data. Users get faster insights from all their data with unmatched performance and price-performance, and can deploy applications in their choice of cloud providers.

Challenges of Building a Modern Data Platform Solution

Digital transformation enabled organizations to improve existing processes and customer experiences, and to reinvent how they serve their market through net-new products, services, or business models. As a result, organizations capture exponential amounts of data, from social media to SaaS applications, and want to process and analyze this valuable data to make data-driven decisions. Organizations building a modern data platform solution need to address the following challenges:

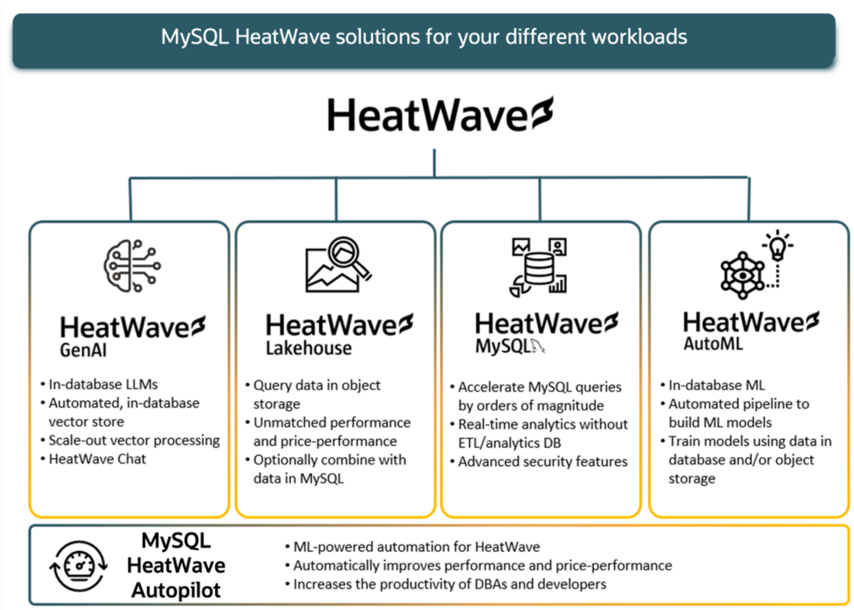
- **Data complexity:** The volume, data format (structured, semi-structured, and unstructured) and diversity of data (e.g. transactions, IoT, application logs) are exploding. The solution needs to provide an efficient way to store, integrate and cross-process this vast amount of data.
- **Disparate services:** Traditional solutions are designed to address a single use case, e.g. transactional database service, data warehouse, data lake analytics, machine learning, generative AI (GenAI) and vectore store. Configuring, securing, managing, integrating, and maintaining these separate services adds complexity.
- **Real-time data processing:** The traditional ETL process is complex and time consuming, often time resulting in stale data. Cusotmers need to ingest new data at high speed and be able to process massive amount of data for query processing, machine learning, and generative AI with high performance and scalability.
- **High cost:** Multiple services mean additional cost—for provisioning, configuration, maintenance, and the actual services costs. Additionally, for GenAI and vector store workloads, customers need to pay for external large language models (LLMs).
- **Higher security risks:** As data needs to move between different data stores, it's difficult to maintain the same security postures across different services.

Copyright © 2025, Oracle and/or its affiliates / Public

MySQL HeatWave

MySQL HeatWave is a fully managed cloud service that enables organizations to efficiently run transactions, real-time analytics across data warehouses and data lakes, plus automated generative AI and machine learning. The scale out design of MySQL HeatWave enables organizations to achieve very high performance and price performance for processing structured, semi-structured and unstructured data. With MySQL HeatWave GenAI, organizations can create a vector store and run large language models (LLMs) inside the database, providing them a secure, simple, efficient and low cost choice to build a new class of generative AI applications.

> "HeatWave's engineering innovation continues to deliver on the vision of a universal cloud database. The latest is generative AI done 'HeatWave style'—which includes the integration of an automated, in-database vector store and in-database LLMs directly into the HeatWave core. This enables developers to create new classes of applications as they combine HeatWave elements."
>
> Holger Mueller
> VP and Principal Analyst
> Constellation Research



MySQL HeatWave

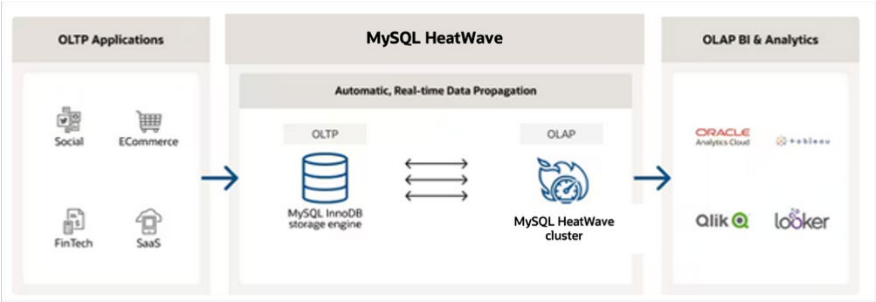
MySQL HeatWave targets use cases for data warehousing, real-time analytics, mixed workloads, transactional processing and machine learning workloads where the main data source is from MySQL databases, without the need for ETL.

A MySQL HeatWave instance consists of a MySQL instance and a MySQL HeatWave cluster. The MySQL HeatWave processing engine, separate from the MySQL InnoDB storage engine for OLTP workloads, is an innovative, in-memory, hybrid columnar analytics engine that is architected for scalability and performance. This results in a very performant solution for SQL analytics at a fraction of the cost of other solutions. Data is persisted in the MySQL InnoDB storage engine. This allows users to manage analytics data in the same way they manage transactional data in MySQL.

Users and applications interact with MySQL HeatWave through the MySQL Database instance. Users connect to MySQL HeatWave through standard tools and standard-based ODBC/JDBC connectors. MySQL HeatWave supports the same ANSI SQL standard and ACID properties as MySQL and

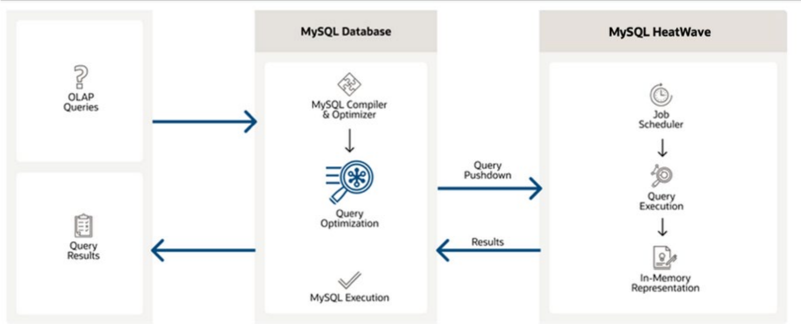
Copyright © 2025, Oracle and/or its affiliates / Public

supports diverse data types. This enables existing applications to take advantage of MySQL HeatWave without any changes, allowing easy and quick integration.



MySQL HeatWave integration with MySQL

Once users submit a query to the MySQL database, the MySQL query optimizer transparently decides if the query should be offloaded to the MySQL HeatWave cluster for accelerated execution. This is based on whether all operators and functions referenced in the query are supported by MySQL HeatWave and if the estimated time to process the query with the MySQL HeatWave engine is less than with MySQL. If both conditions are met, the query is pushed to the MySQL HeatWave nodes for processing. Once processed, the results are sent back to the MySQL database node and returned to users.



MySQL Integration for Query Processing

> "HeatWave MySQL is the best database in the world and has given us more performance than we've ever had, translating to customer websites loading faster and more online gaming. With HeatWave and OCI we get the latest developments right out of the box, at the best price. What could be better?"
>
> **Fabrizio Farina**
> SEO
> Plax1

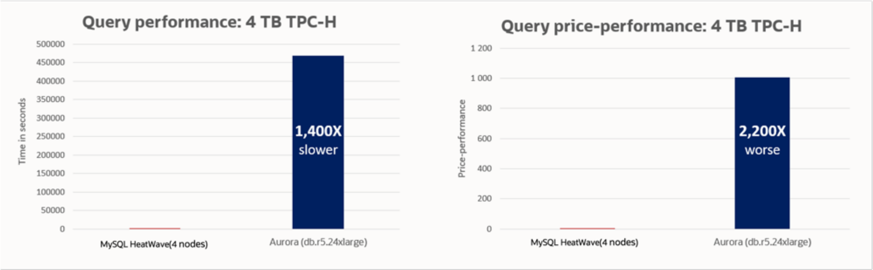
Real Time Analytics

Any updates to the InnoDB tables that are also loaded in MySQL HeatWave are automatically propagated to the memory of the MySQL HeatWave nodes in real-time. This allows subsequent queries to always have access to the latest data. This is completed behind the scenes by a light-weight change propagation algorithm that can keep up with the MySQL data update rates.

Compared to Amazon Aurora, using a 4TB TPC-H Benchmark workload, MySQL HeatWave offers significant improvements in performance for complex and analytic queries. MySQL HeatWave is 1,400x faster than Amazon Aurora, delivering 2,200X better price-performance. Furthermore, with MySQL HeatWave there is no need to create indexes on the base table,

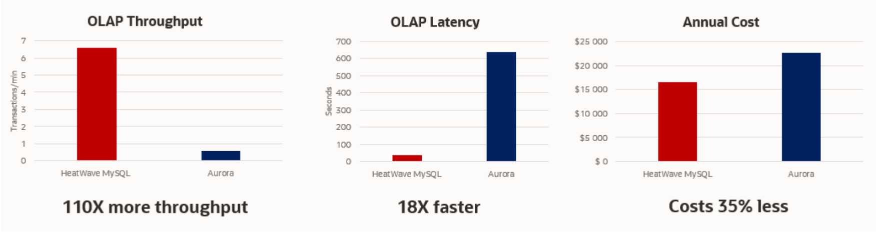
Copyright © 2025, Oracle and/or its affiliates / Public

which can take days with Amazon Aurora. As a result, the data is available to query much sooner with MySQL HeatWave than with Aurora.



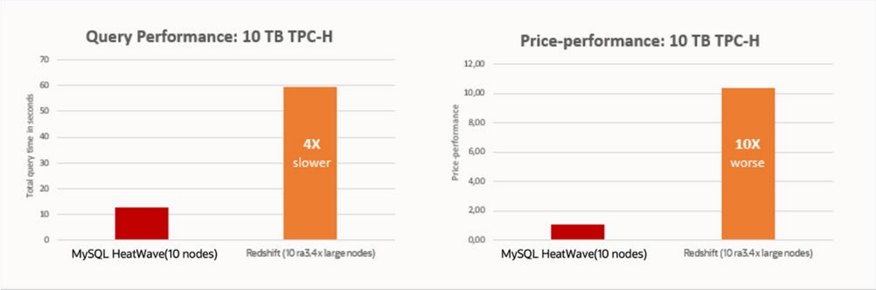
Mixed workloads

Most real-world applications have a mix of OLTP and complex OLAP queries. For such workloads, MySQL HeatWave is much faster and costs a fraction of Amazon Aurora. Using the industry standard CH-benCHmark on a 100 GB dataset for OLAP queries, Amazon Aurora is 18X slower, provides 110X less throughput, and is 35% more expensive than MySQL HeatWave. For OLTP queries, Amazon Aurora has the same performance as MySQL HeatWave and is 35% more expensive according to this benchmark.



Data Warehouse workloads

Amazon Redshift, which is designed for analytics, is offered in multiple shapes. With their instance shape (ra3.4xlarge) and advanced query accelerator (AQUA) turned on, MySQL HeatWave is 4X faster with 10X better price-performance [as demonstrated by a 10TB TPC-H benchmark].



Unlike Amazon Redshift, MySQL HeatWave is capable of both OLTP and OLAP, without the need for ETL.

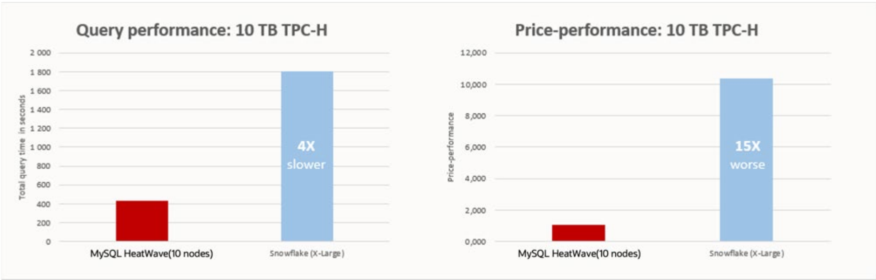
Customers who use MySQL HeatWave will benefit from significantly better performance and eliminate the need for ETL. They get real-time analytics, a

> "We recently migrated our production workload from another cloud solution to HeatWave MySQL. Doing so reduced our cost by 3x and it also significantly accelerated many of our queries which were taking a long time before."
>
> **Chien Hoang**
> Director of Engineering
> Tamara.co

Copyright © 2025, Oracle and/or its affiliates / Public

reduced monthly cost, and a single database for OLTP, OLAP, and machine learning.

Similarly, as demonstrated by a 10TB TPC-H benchmark, MySQL HeatWave is 4X faster than Snowflake with 15X better price-performance.



OLTP workloads

The MySQL node in a MySQL HeatWave instance is built on MySQL Enterprise Edition to help provide the highest levels of MySQL scalability, security, reliability, and availability. It is optimized for OLTP workloads.

Hypergraph-based optimizer for MySQL

The hypergraph-based optimizer uses true cost to guide the join search space and produce optimized join query plans. This improves the performance of complex queries, particularly for those involving joins. For complex queries in industry standard benchmark like TPC-H and TPC-DS, query performance was significantly improved.

Bulk ingestion

For customers who want to import a large amount of data upfront into their MySQL Database, we now enable "bulk ingest" at the instance creation time. This is done by parallelizing the insertion of the data into the table. Ingestion performance improves by up to 5X, regardless of whether the source data is sorted or unsorted. For example, for the TPC-H LINEITEM table, the time to load data is reduced from 90 minutes to 18 minutes.

JavaScript Support

JavaScript support enables rich procedural programming directly inside the database, allowing users to further reduce data movement complexity and costs in favor of server-side solutions. Native JavaScript support lets users write JavaScript stored functions and procedures in the server that are executed via GraalVM. The JavaScript functions and procedures can manipulate existing data, regardless of whether the data is in MySQL Database, MySQL HeatWave, or object storage.

Users can simplify application logic by moving data-intensive complex operations closer to their data. This reduces the cloud egress cost and the efforts required to maintain data pipelines. In addition, it improves end-to-end application performance and security by eliminating the need for client-server data movement.

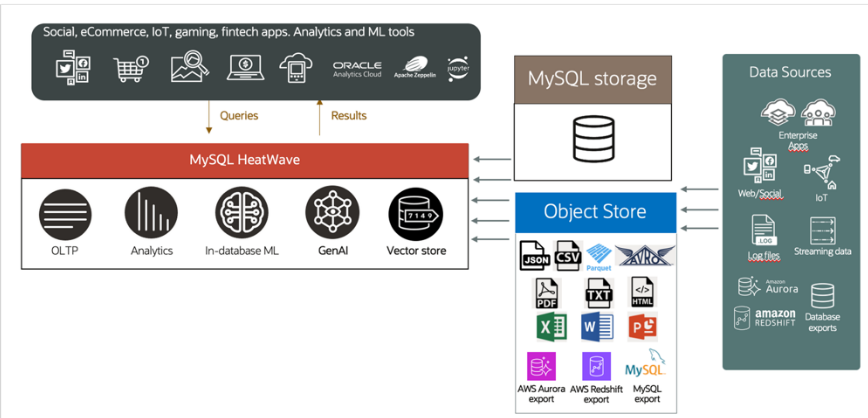
"We found HeatWave MySQL improved performance by 10 times and significantly dropped our costs after migrating from AWS Aurora. We also did not have to modify our application for a great experience.

Kanami Suzuki
Developer
Fan Communications

Copyright © 2025, Oracle and/or its affiliates / Public

MySQL HeatWave Lakehouse

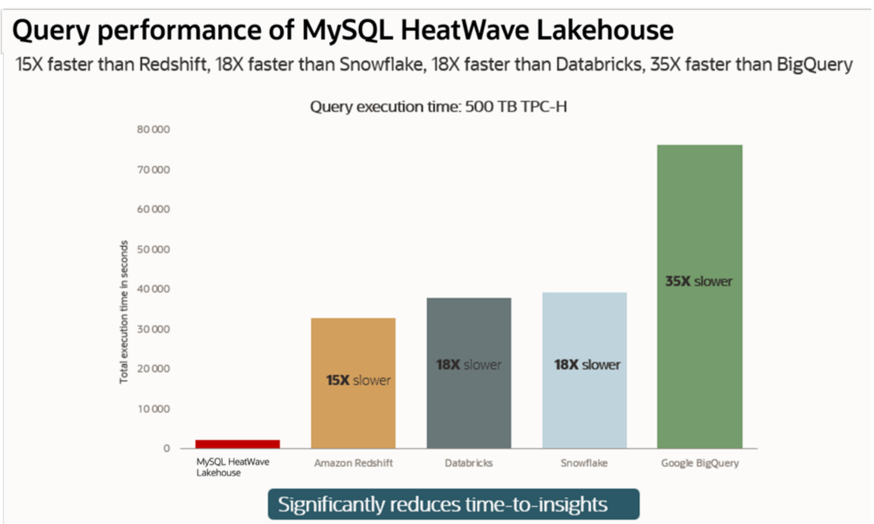
MySQL HeatWave includes Lakehouse, letting users process and query hundreds of terabytes of data in the object store—in a variety of file formats such as CSV, Parquet, Avro, JSON, Aurora/Redshift export files, and on-premises MySQL export files. Data in the object store remains in the object store and is not copied into the database. As a result, customers can leverage the benefits of MySQL HeatWave even when their data is not stored inside a MySQL database. The MySQL HeatWave cluster scales to 512 nodes to process data in object storage.



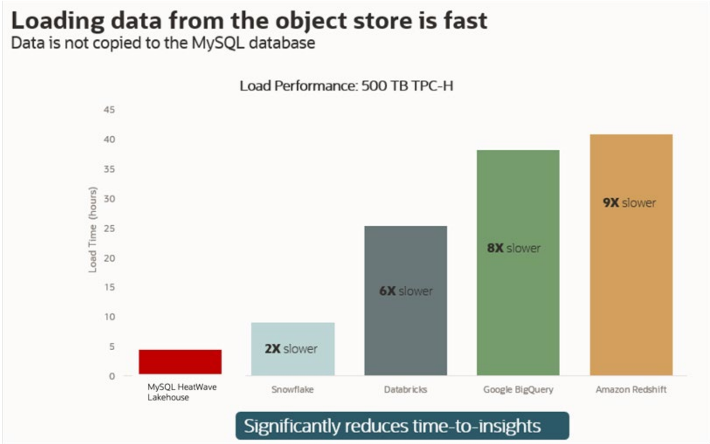
> "HeatWave is taking a big step in making generative AI and Retrieval-Augmented Generation (RAG) more accessible by pushing all the complexity of creating vector embeddings under the hood. Developers simply point to the source files sitting in cloud object storage, and HeatWave then handles the heavy lift. These are major steps toward democratizing AI."
>
> **Tony Baer**
> Founder and CEO
> Dbinsight

Faster than Snowflake, Amazon Redshift, Databricks, and Google BigQuery

As demonstrated by a [500 TB TPC-H benchmark](), the query performance of MySQL HeatWave Lakehouse is 15X faster than Amazon Redshift, 18X faster than Snowflake, 18X faster than Databricks, and 35X faster than Google BigQuery. The load performance of MySQL HeatWave Lakehouse is 9X faster than Amazon Redshift, 8X faster than Google BigQuery, 6X faster than Databricks, and 2X faster than Snowflake.



Copyright © 2025, Oracle and/or its affiliates / Public



Fast analytics on all data

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, as demonstrated by 10 TB and 30 TB TPC-H benchmarks.

Incremental load

MySQL HeatWave Lakehouse can uptake changes in user data incrementally without reloading the entire data set. It automatically determines which files have been added/changed/removed.

Ingestion of new data objects scales across the MySQL HeatWave cluster, delivering the same high load performance as for first time Lakehouse table load. This makes updating MySQL HeatWave Lakehouse data very efficient.

Machine learning on all data

Customers can train, predict, and explain their machine learning models on data loaded from object storage. MySQL HeatWave AutoML uses a common set of commands to train, predict, and explain a model, irrespective of whether the data is in object storage or in the database. This simplifies ML operations.

Learn more.

MySQL HeatWave AutoML

Current challenges of Machine Learning

Developing and using machine-learning models requires skill sets in topics such as:

- Candidate algorithms/models to select from
- Hyperparameters that need to be tuned per algorithm
- Features to engineer and select from
- Data preprocessing approach per data type
- Drift detection and retraining
- Knowledge of Python, as most ML algorithm frameworks are available only in Python

"A fully integrated GenAI platform that includes everything from a vector store and in-database LLMs to machine learning and natural language capabilities under the same hood. This approach eliminates unnecessary data movement and fragmentation, reduces the attack surface area, and helps to dramatically improve regulatory compliance posture."

Alexei Balaganski
Lead Analyst and CTO
KuppingerCole Analysts

Copyright © 2025, Oracle and/or its affiliates / Public

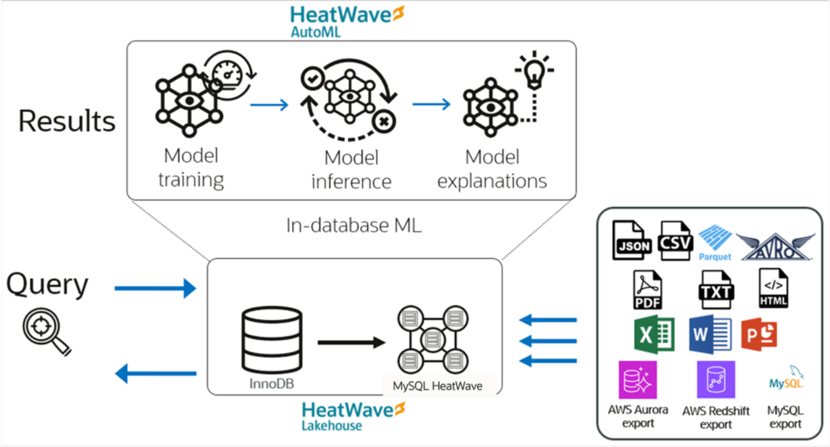
Even with the above expertise, users still need to extract data out of the database to train and test the model, which leads to trust and security issues.

MySQL HeatWave AutoML Approach

MySQL HeatWave AutoML is an in-database set of machine learning capabilities that enable users to train ML models, and generate inference and explanations across data stored in MySQL databases and object storage. All machine learning activities are performed inside the database, meaning there's no need to extract data out of the database to perform ML. It brings several advantages:

- **Fully Automated:** MySQL HeatWave AutoML fully automates the creation of tuned models, generating inferences and explanations, thus eliminating the need for the user to be an expert ML developer
- **SQL interface**: Provides the familiar MySQL interface for invoking machine learning capabilities
- **Security and Efficiency**: Data and models never leave the MySQL Database. Clients or any other services never see the data or models stored in the DB service
- **Explanations**: All models created by MySQL HeatWave AutoML can be explained. Enterprises have a growing need to explain the predictions of machine learning models to build trust, demonstrate fairness, and comply with regulatory requirements.
- **Performance and Scalability**: The performance of MySQL HeatWave AutoML is much better at a lower cost than competing services such as Redshift ML. Furthermore, MySQL HeatWave AutoML scales with the size of the cluster.
- **Easy Upgrades**: MySQL HeatWave AutoML leverages state-of-the-art open-source Python ML packages that enable continual and swift uptake of newer (and improved) versions.
- **Supported Models**: MySQL HeatWave AutoML supports multiple model types such as Classification, Regression, Time Series Forecasting, Anomaly Detection, Recommender System, etc. This enables organizations to use MySQL HeatWave AutoML for many different types of business use cases.

All these capabilities are available to MySQL HeatWave customers at no additional cost.



MySQL HeatWave AutoML is a native, in-database solution, eliminating the cost, complexity and risk of ETL

"HeatWave GenAI's automated in-database vector store and in-database LLMs greatly simplify how developers build AI applications and users interact with data. In contrast to many other vendors that require multiple steps to create vector stores and to use vector stores with LLMs, HeatWave GenAI requires only one SQL statement for each task and it's all automated from there."

Richard Winter
CEO
WinterCorp

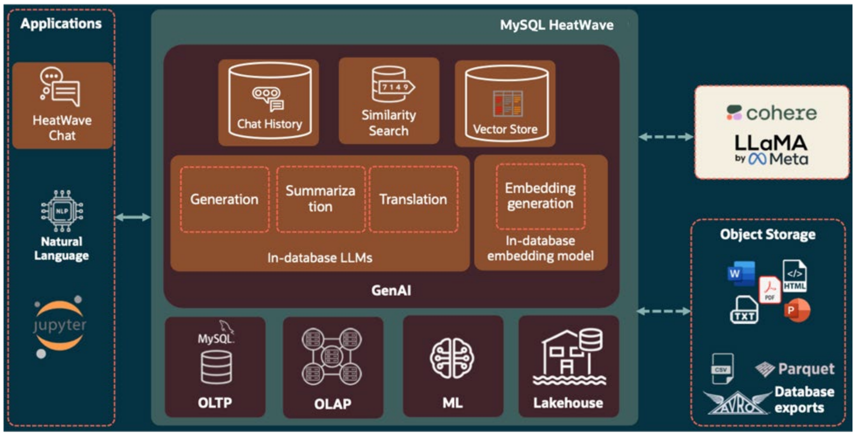
Copyright © 2025, Oracle and/or its affiliates / Public

[Learn more](#).

MySQL HeatWave GenAI

MySQL HeatWave GenAI enables developers to create a vector store for enterprise unstructured content with a single SQL command, using built-in embedding models. Users can perform natural language searches in a single step using either in-database or external LLMs. Data does not leave the database and, with MySQL HeatWave's very high performance and scalability, there is no need to provision GPUs. As a result, developers can reduce application complexity, increase performance, improve data security, and lower costs.

All the elements of the pipeline necessary to use MySQL HeatWave GenAI with proprietary data are built-in, integrated, and optimized to work with each other, enabling turnkey generative AI application development.



MySQL HeatWave GenAI features include:

In-database LLMs: simplify the development of generative AI applications at a lower cost. Customers can benefit from generative AI without the complexity of external LLM selection and integration, and without worrying about the availability of LLMs in various cloud providers' data centers. The in-database LLMs enable customers to search data, generate or summarize content, and perform retrieval-augmented generation (RAG) with MySQL HeatWave Vector Store. In addition, they can combine generative AI with other built-in MySQL HeatWave capabilities such as AutoML to build richer applications. MySQL HeatWave GenAI is also integrated with the OCI Generative AI service and Amazon Bedrock to access pre-trained, foundatio models from leading LLM providers.

Automated, In-database Vector Store: enables customers to use generative AI with their business documents without moving data to a separate vector database and without AI expertise. All the steps to create a vector store and vector embeddings are automated and executed inside the database, including discovering the documents in object storage, parsing them, generating embeddings in a highly parallel and optimized way, and

"With in-database LLMs that are ready to go and a fully automated vector store that's ready for vector processing on day one, HeatWave GenAI takes AI simplicity—and price performance—to a level that its competitors such as Snowflake, Google BigQuery and Databricks can't remotely begin to approach"

Steve McDowell
Chief Analyst
NAND Research

Copyright © 2025, Oracle and/or its affiliates / Public

inserting them into the vector store, making MySQL HeatWave Vector Store efficient and easy to use. Using a vector store for RAG helps solve the hallucination challenge of LLMs as the models can search proprietary data with appropriate context to provide more accurate and relevant answers.

Scale-out vector processing: delivers very fast semantic search results without any loss of accuracy. MySQL HeatWave supports a new, native VECTOR data type and an optimized implementation of the distance function**,** enabling customers to perform semantic queries with standard SQL. In-memory hybrid columnar representation and the scale-out architecture of MySQL HeatWave enable vector processing to execute at near-memory bandwidth and parallelize across up to 512 MySQL HeatWave nodes. As a result, customers get their questions answered rapidly. Users can also combine semantic search with other SQL operators to, for example, join several tables with different documents and perform similarity searches across all documents.

MySQL HeatWave Chat: is a Visual Studio code plug-in for MySQL shell which provides a graphical interface for MySQL HeatWave GenAI and enables developers to ask questions in natural language or SQL. The integrated Lakehouse Navigator enables users to select files from object storage and create a vector store. Users can search across the entire database or restrict the search to a folder. MySQL HeatWave maintains context with the history of questions asked, citations of the source documents, and the prompt to the LLM. This facilitates a contextual conversation and allows users to verify the source of answers generated by the LLM. This context is maintained in MySQL HeatWave and is available to any application using MySQL HeatWave.

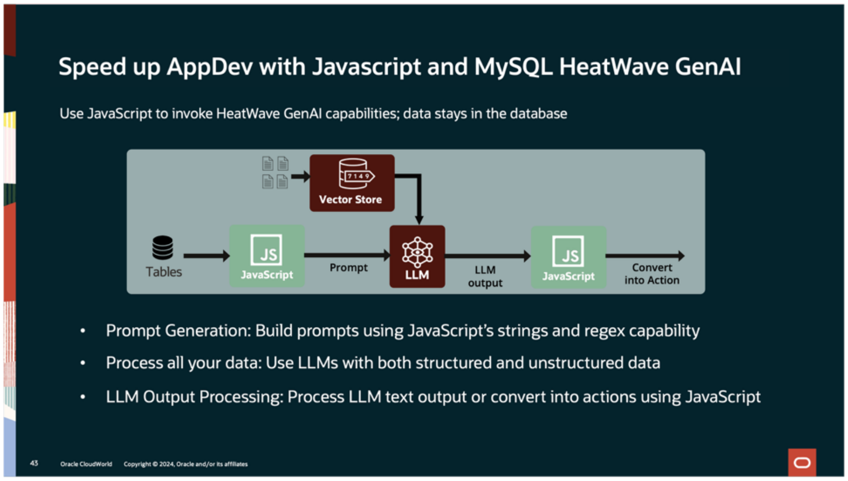
“We believe that Generative AI can enhance the efficiency of our client-facing teams through use of semantic search and summarization of documents by using HeatWave GenAI with the HeatWave Vector Store, which offers unique capabilities. We are working on this potential use case and we hope to productize the benefits to our teams.”

Ramesh Lakshminarayanan
CIO & Group Head-IT, HDFC Bank

In-database JavaScript support for GenAI

As generative AI and LLMs primarily handle textual and JSON data, JavaScript is a natural choice for manipulating this data. We’ve added native support for the vector data type in JavaScript and the ability to invoke MySQL HeatWave GenAI capabilities from a JavaScript program. Developers can seamlessly use MySQL HeatWave GenAI APIs in their JavaScript functions, which are executed directly in the database. Developers can pre-process the prompt or post-process the generated response using JavaScript’s powerful string and JSON processing capabilities.

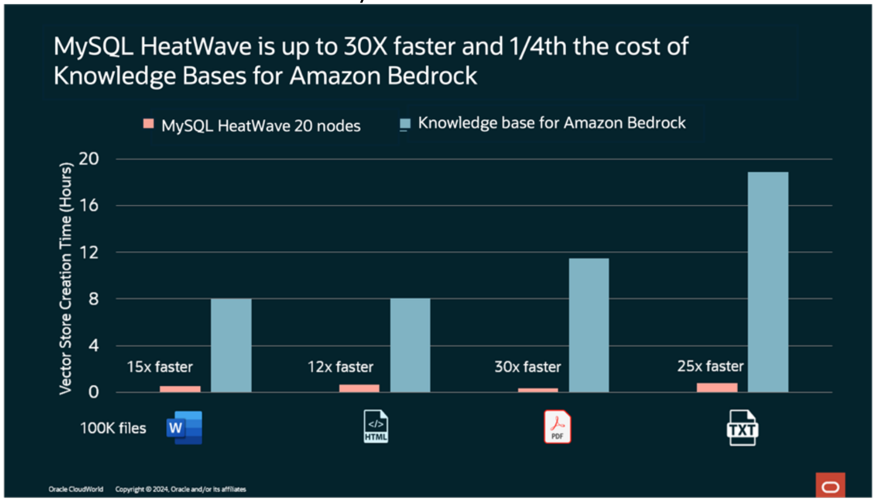
Copyright © 2025, Oracle and/or its affiliates / Public



Performance

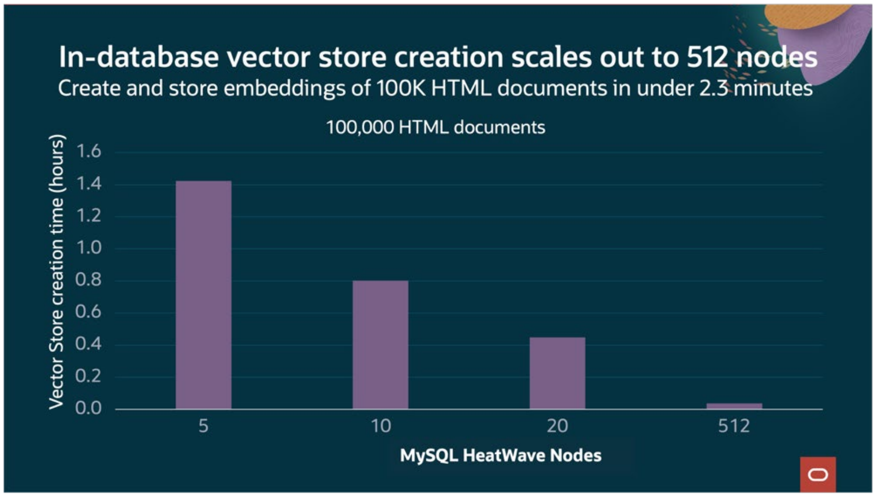
Each stage of MySQL HeatWave Vector Store creation has been optimized for performance and scalability. Different stages from document parsing to vector embedding generation have very different characteristics, resource requirements, and parallelism opportunities. These have been integrated and tuned to work efficiently with each other, enabling MySQL HeatWave Vector Store creation to scale to thousands of cores.

MySQL HeatWave Vector Store creation is up to 30X faster than Knowledge Bases for Amazon Bedrock at 1/4th the cost.



MySQL HeatWave Vector Store creation performance scales with larger cluster sizes, giving users a simple way to reduce their vector store creation time. Users have the flexibility to choose the required performance and associated cost according to their needs for each workload.

Copyright © 2025, Oracle and/or its affiliates / Public



All these capabilities are available to MySQL HeatWave customers at no additional cost. Learn more.

Deployment Scenarios

A MySQL HeatWave instance is a cluster composed of a MySQL Database instance and multiple MySQL HeatWave nodes. When the MySQL HeatWave cluster is enabled, a MySQL HeatWave plugin is installed on the MySQL node. It is responsible for cluster management, loading data into the memory of the MySQL HeatWave nodes, query scheduling, and query execution. Applications written in Java, PHP, Ruby, etc. work seamlessly with MySQL HeatWave using standard MySQL ODBC/JDBC connectors. The MySQL HeatWave cluster is an in-memory query accelerator and supports all MySQL syntax. Hence, all existing tools and applications built using standard SQL will work without requiring any modification to queries.

Data that is needed for analytic processing is stored in the memory of the MySQL HeatWave nodes, in a hybrid columnar compressed format. The number of nodes needed to run a workload depends on the amount of data present for analytic processing, the compression factor that is achieved on the dataset, and the query characteristics. The number of nodes needed can be automatically determined by using MySQL HeatWave Autopilot - Auto Provisioning advisor, which uses machine learning to predict the MySQL HeatWave cluster memory required for the workload.

MySQL HeatWave supports 2 shapes and up to 64 nodes per cluster for workloads up to 64TB of data from the MySQL database or up to 512TB of data from object storage:

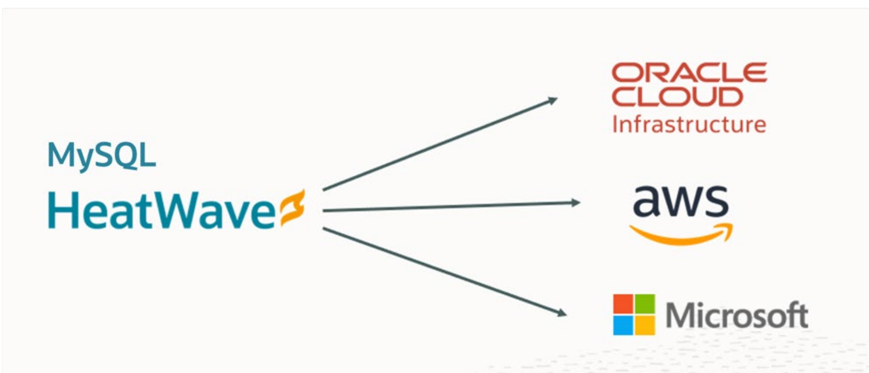
1. HeatWave.32GB – This shape has 32GB of memory and can process up to 50GB of data per node
2. HeatWave.512GB – This shape has 512GB memory and can process up to 1TB of data per node

Copyright © 2025, Oracle and/or its affiliates / Public

The maximum amount of data which can be populated in the memory of the MySQL HeatWave nodes at a given moment depends on the data characterisitics and workload. There is no limit to the amount of data which can be stored in MySQL Database or object storage. Customers can choose which tables or columns from the MySQL database schema or files from object storage to load into the memory of MySQL HeatWave nodes. If the tables are no longer needed by queries, they can be removed from memory to make room for other data.

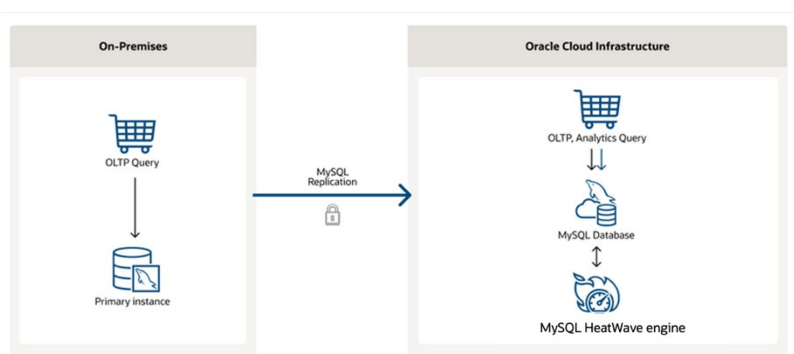
MySQL HeatWave provides a great solution for customers who need to run transactional, analytical, machine learning, and generative AI workloads. While transactional queries are run in the MySQL node, data updated in MySQL InnoDB or object store is transparently propagated to the MySQL HeatWave cluster in real-time. This enables customers to always process the latest data.

For customers with a multi-cloud strategy, MySQL HeatWave is available on OCI, AWS, and Azure, as well as in customers' data centers with OCI Dedicated Region and Oracle Alloy. Customers can take advantage of MySQL HeatWave on the cloud platform that they prefer.



MySQL HeatWave is optimized to deliver best price performance on OCI, AWS, and Azure respectively

On-premises customers who cannot move their MySQL deployment to a cloud due to compliance or regulatory requirements can still leverage MySQL HeatWave by using the hybrid deployment model. In such a hybrid deployment, customers can leverage MySQL replication to replicate on-premises MySQL data to MySQL HeatWave without the need for ETL.



Copyright © 2025, Oracle and/or its affiliates / Public

Hybrid deployment for enabling analytics on data stored on premise

MySQL HeatWave Architecture

There are four key architectural design charateristics that lead to a compelling performance and cost advantage with MySQL HeatWave:

1. Innovative in-memory hybrid columnar analytics engine designed for scalability and performance, which implements state of the art algorithms.

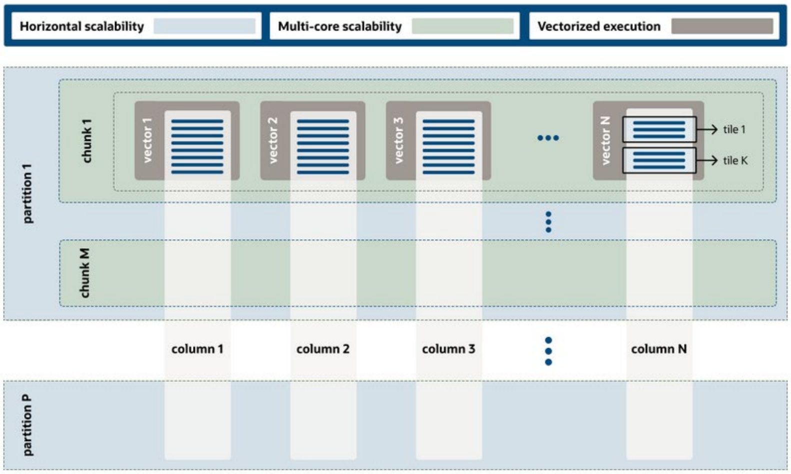
2. Optimized for the cloud to provide the best price performance database service based on commodity hardware.

3. MySQL HeatWave scale out data management layer enabling the MySQL HeatWave representation of data to be stored in OCI object storage (or Amazon S3), allowing fast data reload for operations like error recovery, maintenance, and system restart to increase service uptime.

4. MySQL HeatWave Autopilot provides machine learning-powered automation that improves the performance, scalability, and ease of use of MySQL HeatWave. It automates the database lifecycle operations including provisioning, data loading, query processing, and error handling. It also provides capabilities for OLTP workloads.

In-memory hybrid columnar analytics engine

The MySQL HeatWave engine uses a columnar in-memory representation that facilitates vectorized processing, leading to very good query performance. The data is encoded and compressed prior to being loaded in memory. This compressed and optimized in-memory representation is used for both numeric and string data. This results in significant performance speed-up and reduced memory footprint, which translates into reduced cost for customers.

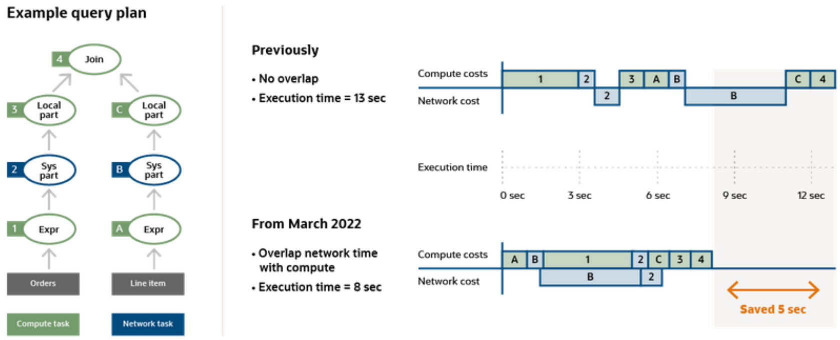


Vectorized in-memory columnar representation for analytic processing

Copyright © 2025, Oracle and/or its affiliates / Public

One of the key design points of the MySQL HeatWave engine is to massively partition data across a cluster of MySQL HeatWave nodes, which can be operated in parallel in each node. This enables high cache hits for analytic operations and provides very good inter-node scalability. Each MySQL HeatWave node within a cluster and each core within a node can process partitioned data in parallel, including parallel scans, joins, group-by, aggregation and top-k processing.

MySQL HeatWave has implemented high performance algorithms for distributed in-memory analytic processing. Joins within a partition are processed fast by using vectorized build and probe join kernels. The highly-optimized network communication between analytics nodes is achieved by using asynchronous batch I/Os. The algorithms are designed to overlap compute time with the communication of data across nodes, which helps achieve good scalability.



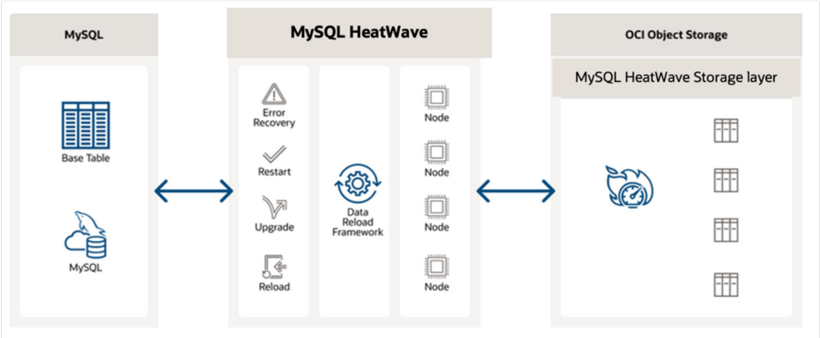
MySQL HeatWave implements state of art algorithms for distributed in-memory analytic processing.

Scale-out Data Management

The MySQL InnoDB storage engine stores data in a row-based format, while MySQL HeatWave stores data in memory in a hybrid columnar format. Loading data from MySQL to the MySQL HeatWave cluster involves transforming data into the MySQL HeatWave columnar format, which could take time depending on data size. Since data is stored in-memory in MySQL HeatWave, operations like error recovery, maintenance, and system restart could take a long time as data needs to be re-transformed and reloaded after the cluster is ready.

To improve service uptime, MySQL HeatWave introduced a new storage layer that is built on OCI Object Storage, or Amazon S3 when using MySQL HeatWave on AWS. This new architecture enables storing MySQL HeatWave formatted data in a persistent storage, allowing reload of data in constant time regardless of data size.

Copyright © 2025, Oracle and/or its affiliates / Public



MySQL HeatWave scale-out storage layer

In the MySQL HeatWave storage layer, persisted data is organized in the same way as that of in-memory data. Each MySQL HeatWave node can restore data independently and in parallel, allowing a very fast and near-constant time data reload.

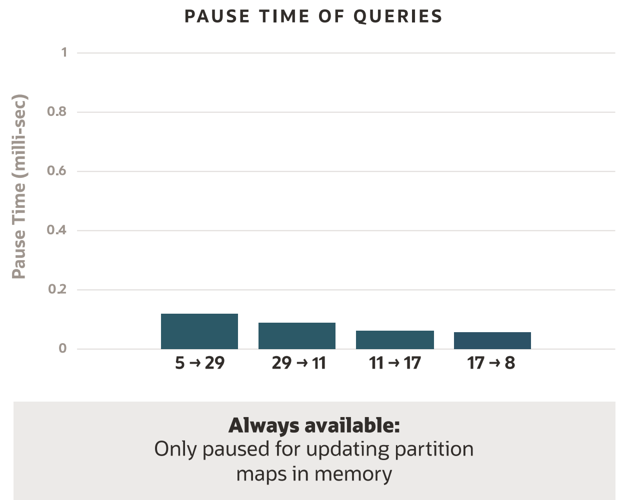
Real-time Elasticity

With the MySQL HeatWave storage layer, each MySQL HeatWave node can load data independently and in parallel based on different data sizes and different cluster sizes. This enables MySQL HeatWave to provide a flexible, predictable, and performant online cluster scaling capability that allows users to resize their MySQL HeatWave cluster when their workload and data demand change.

The MySQL HeatWave cluster can be scaled up or down to any number of nodes. During the scaling operation, the cluster continues to work without interruption in client connections for read or write and there is minimal effect on query performance in MySQL HeatWave. For sizing up, data is loaded from the MySQL HeatWave storage layer to the newly created nodes. Once the data is loaded, MySQL HeatWave updates its metadata and queries. Queries will then be processed using the increased cluster size. For sizing down, additional data is loaded from the MySQL HeatWave storage layer to the MySQL HeatWave nodes that will be kept for the new reduced cluster size. Once the additional data is loaded, MySQL HeatWave updates its metadata and queries, and queries will be processed using the reduced cluster size. For both operations, data is balanced across all nodes automatically without user intervention to ensure optimal query performance.

The below chart shows the paused time in microseconds that the MySQL HeatWave cluster needs to update its metadata.

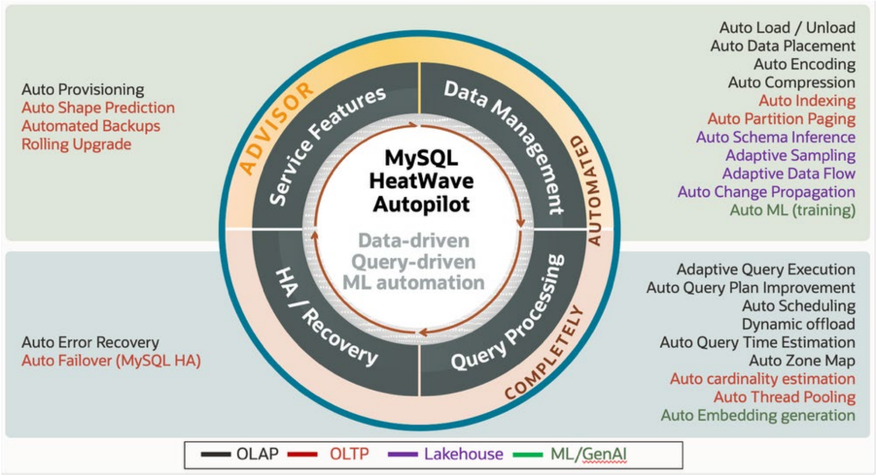
Copyright © 2025, Oracle and/or its affiliates / Public



MySQL HeatWave real-time elasticity performance

MySQL HeatWave Autopilot

MySQL HeatWave Autopilot automates many of the most important and often challenging aspects of achieving high query performance at scale - including system setup, data loading, query execution and failure handling. It uses advanced techniques to sample data, collect statistics on data and queries, and build machine learning models to model memory usage, network load and execution time. MySQL HeatWave Autopilot uses these ML models to intelligently learn from queries executed in the system, resulting in continually improving system performance over time. MySQL HeatWave Autopilot also provides capabilities to improve the performance and price-performance of OLTP workloads.



MySQL HeatWave Autopilot automates different aspects of the service to improve performance, scalability and usability of the system

System Setup

1. ***Auto provisioning*** predicts the number of MySQL HeatWave nodes required for running a workload by adaptive sampling of table data on

Copyright © 2025, Oracle and/or its affiliates / Public

which analytics is required. This means that customers no longer need to manually estimate the optimal size of their cluster.

2. ***Auto shape prediction*** continuously monitors the OLTP workload, including throughput and buffer pool hit rate, to recommend the right compute shape at any given time—allowing customers to always get the best price-performance.
3. ***Auto schema inference*** automatically infers the mapping of file data to data types 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, saving time and effort.
4. ***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 HeatWave Autopilot can scan and propose schema predictions on a set of data files totalling 500 TBs in under one minute.

Data Load

1. ***Autopilot indexing*** recommends the right set of indexes for columns in order to improve OLTP query performance. It balances cost, storage space, and performance by adding or removing indexes.
2. ***Auto parallel load*** monitors the MySQL workload, identifies queries that can be accelerated by MySQL HeatWave, and automatically loads relevant tables into MySQL HeatWave to maximize query acceleration. It also optimizes load time and memory usage by predicting the optimal degree of parallelism for each table being loaded into MySQL HeatWave.
3. ***Auto encoding*** determines the optimal representation of columns being loaded into MySQL HeatWave taking queries into consideration. This optimal representation provides the best query performance and minimizes the size of the cluster to minimize the cost.
4. ***Auto data placement*** predicts the column on which tables should be partitioned in-memory to achieve the best performance for queries. It also predicts the expected gain in query performance with the new column recommendation.
5. ***Auto unload*** automatically unloads unused or rarely used tables in MySQL HeatWave, and predicts the memory saved from unloading these tables.
6. ***Auto compression*** dynamically determines which compression algorithms to use for each column based on its data characteristics. This enables it to provide the optimal memory usage and query performance.
7. ***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.

Query Execution

Copyright © 2025, Oracle and/or its affiliates / Public

8. ***Auto query plan improvement*** learns various statistics from the execution of queries and improves the execution plan of future queries. This improves the performance of the system as more queries are run.
9. ***Auto query time estimation*** estimates the execution time of a query prior to executing the query, allowing quick tryout and testing on different queries
10. ***Auto change propagation*** intelligently determines the optimal time when changes in MySQL Database should be propagated to the MySQL HeatWave storage layer. This ensures that changes are being propagated at the right optimal cadence.
11. ***Auto scheduling*** determines which queries in the queue are short-running and prioritizes them over long-running queries in an intelligent way to reduce overall wait time.
12. ***Adaptive Query Execution*** dynamically adjusts the query execution plan based on runtime statistics to improve query execution time.
13. ***Auto thread pooling*** lets the database service process more transactions for a given hardware configuration, delivering higher throughput for OLTP workloads and preventing it from dropping at high levels of transactions and concurrency.

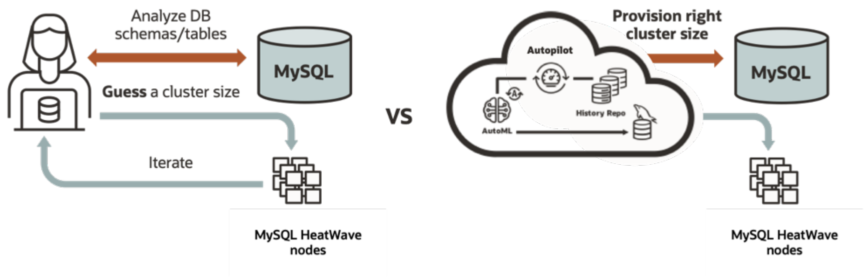
Failure Handling

14. ***Auto error recovery:*** Provisions new nodes and reloads necessary data from the MySQL HeatWave storage layer if one or more MySQL HeatWave nodes are unresponsive due to software or hardware failure

Auto Provisioning

Auto Provisioning provides a recommendation on how many MySQL HeatWave nodes are needed to run a workload. When the service is started, database tables on which analytics queries are run need to be loaded to the MySQL HeatWave cluster memory. The size of the cluster needed depends on the tables and columns required to load, and the compression achieved in memory for this data. The diagram below compares the traditional (i.e., manual) approach to estimating the cluster size with Auto Provisioning. In traditional provisioning, users need to guess a cluster size. Underestimation results in data load or query execution failure due to space limitations. Overestimation results in additional costs for unneeded resources. As a result, users iterate until they determine the right cluster size, and this size estimate becomes inaccurate when tables are updated.

Copyright © 2025, Oracle and/or its affiliates / Public

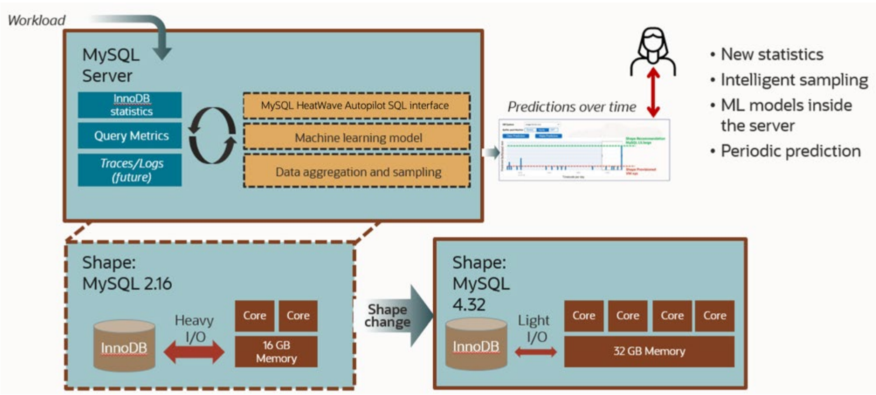


Comparison of manual provisioning vs Auto provisioning

The right side of the above diagram shows how auto provisioning, a ML-based cluster size estimation advisor, solves this problem. By leveraging well trained and accurate ML models, the user consults the auto provisioning advisor to obtain the right cluster size for their dataset. As a result, users do not need to guess the cluster size. Later, if the customer data grows or additional tables are added, the users can again take advantage of the auto provisioning advisor.

Auto Shape Prediction

To alleviate the burden of experimenting with different MySQL shapes to determine the most performant shape for a given workload, Auto Shape Prediction provides suggestions for the right MySQL server shape, based on highly accurate predictions from machine-learning models inside the MySQL server and the most recent query execution metrics and traces. Since Auto Shape Prediction continuously collects workload execution statistics, it can adapt to the evolving workload patterns and hence provide suggestions based on the most recent workload.



Auto Schema Inference

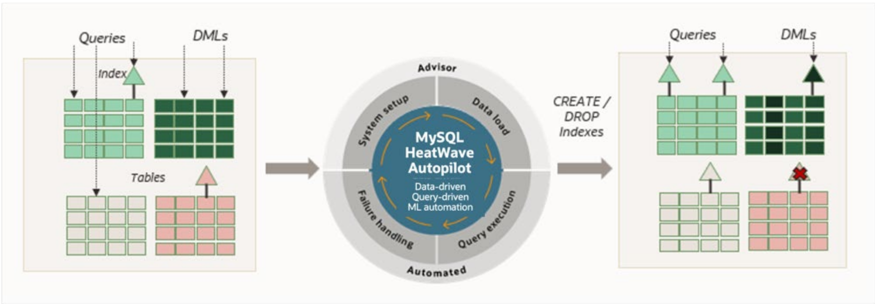
Auto Schema Inference automatically predicts the table definitions for the data stored in object storage. This is valuable, because the schema for the data stored in object storage needs to be defined in order for MySQL HeatWave to perform data loading and query processing. It accurately infers the data type and its precision, for structured or semi-structured data, even

Copyright © 2025, Oracle and/or its affiliates / Public

when metadata is not present in the files (for example, CSV). Auto Schema Inference generates the DDL for the table definition, which can be easily reviewed and modified. It makes it easy to load data into MySQL HeatWave from various data sources where the schema is not known in advance.

Autopilot Indexing

MySQL HeatWave Autopilot Indexing is an ML-based technology designed to optimize OLTP workloads for better cost and performance.



Autopilot Indexing automatically generates secondary index recommendations to create or drop indexes based on the current OTLP workload. It considers both the query performance and the cost of maintaining the indexes when generating recommendations. It provides performance and storage estimations, as well as explanations for the recommendations it generates. Autopilot Indexing consists of a simple and intuitive console that customers can use to view and analyze the projected performance and storage impact of recommended index suggestions. This makes it easy to foresee the impact of changes to the database systems before applying the suggestions.

For more information, read the Autopilot Indexing Technical Brief

Auto Unload

Users can save costs by unloading tables that are never or rarely queried in MySQL HeatWave to lower the total memory required and decrease the MySQL HeatWave cluster size. However, this requires understanding all queries and access patterns of the tables. Auto unload tracks table usage and automatically unloads unused tables, while providing an explanation to support each action taken.

Auto Compression

Data stored in MySQL HeatWave is compressed using different compression algorithms. To minimize memory usage while providing the best query performance, auto compression dynamically determines the optimal compression algorithm to use for each column based on its data characteristics.

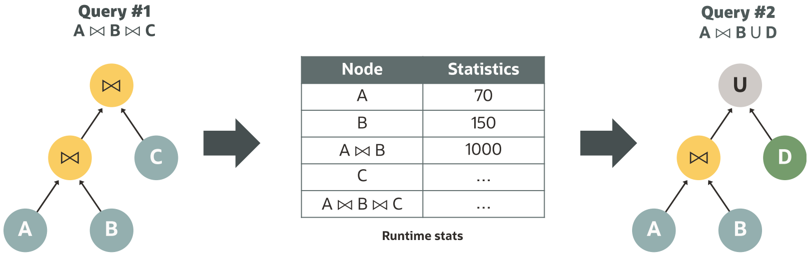
Auto compression employs an adaptive sampling technique during the data loading process. The algorithm selected is based on the compression ratio

Copyright © 2025, Oracle and/or its affiliates / Public

and the compression and decompression rates, which balance the memory needed to store the data in MySQL HeatWave with query execution time.

Auto Query Plan Improvement

Auto query plan improvement enhances query performance by improving query plan statistics based on previous query executions. By maintaining more accurate query statistics, MySQL HeatWave creates better query plans and makes better decisions on the underlying physical operators; consequently improving the overall query performance.

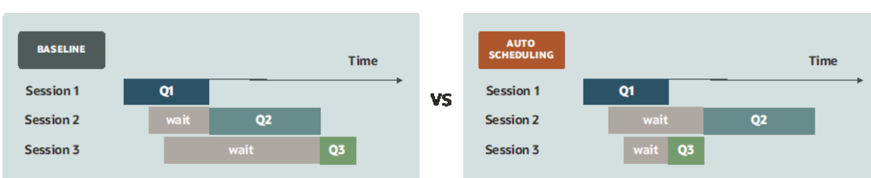


Query 2 benefits from statistics of a previous similar query (query 1) with auto query plan improvement

The above shows how auto query plan improvement works without user intervention. After a query (Q1) executes on MySQL HeatWave, auto query plan improvement collects and stores the cardinalities of all operations in the query execution plan (e.g., scan, join, group by). When a similar (or identical) query arrives (Q2), the system checks whether it can take advantage of the previously collected statistics information for Q2. If the system determines a similarity between the two query plans, a better query plan is generated based on statistics information from Q1. In doing so, it improves query performance and cluster memory usage significantly.

Auto Scheduling

Traditional database systems process queries based on their arrival time, which can result in long-running queries starving short-running queries.



Traditional database system vs MySQL HeatWave auto scheduling

On the left, is a sub-optimal case where three queries (Q1, Q2, Q3) from three user sessions are scheduled in the FIFO order. After the execution completes, one can identify that waiting time for Q3 could be reduced significantly with minimal impact on Q2 latency.

On the right, it shows how auto scheduling improves the user experience for short running queries in a multi-session application. Auto scheduling identifies and prioritizes short-running queries by automatically classifying

Copyright © 2025, Oracle and/or its affiliates / Public

queries into short or long queries. Therefore, Q3 is prioritized before Q2 as Q3 is identified as a short-running query.

Auto Scheduling reduces the elapsed time for short-running queries significantly when the multi-session applications consist of a mix of short and long running queries. It also ensures long-running queries are not penalized and are not postponed indefinitely.

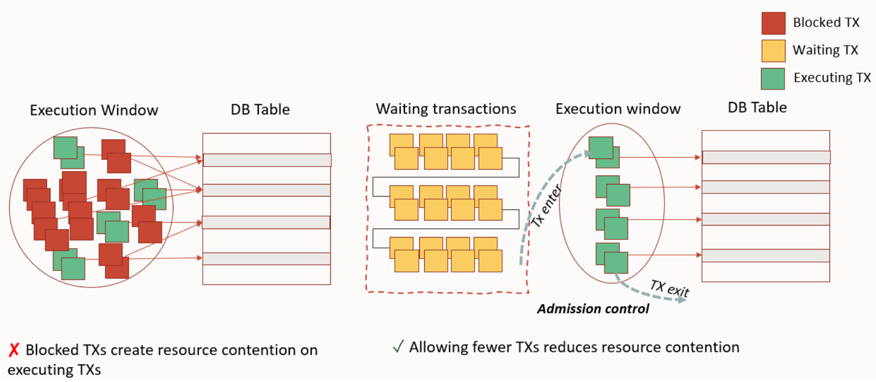
Adaptive Query Optimization

Adaptive query optimization automatically improves query performance and memory consumption, and mitigates skew-related performance issues as well as “out of memory” issues. It uses various statistics to adjust data structures and system resources after query execution has started—independently optimizing query execution for each node based on actual data distribution at runtime. This helps improve the performance of ad hoc queries by up to 25%.

The MySQL HeatWave optimizer generates a physical query plan based on statistics collected by Autopilot. During query execution, each MySQL HeatWave node executes the same physical plan. With adaptive query execution, each individual MySQL HeatWave node adjusts the local physical query plan based on statistics such as cardinality and distinct value counts of intermediate relations collected locally in real-time. This allows each MySQL HeatWave node to tailor the data structures it needs, resulting in better query execution time, lower memory usage, and improved data skew-related performance.

Auto Thread Pooling

With Auto Thread Pooling, MySQL HeatWave MySQL prioritizes not only peak single-thread performance but also high throughput in the presence of concurrent clients running concurrent queries on MySQL server. MySQL server performs workload-aware admission control of the incoming transactions. It eliminates the resource contention created by too many awaiting transactions, automatically queuing them to maximize performance while sustaining the throughput in the face of high concurrency.



Copyright © 2025, Oracle and/or its affiliates / Public

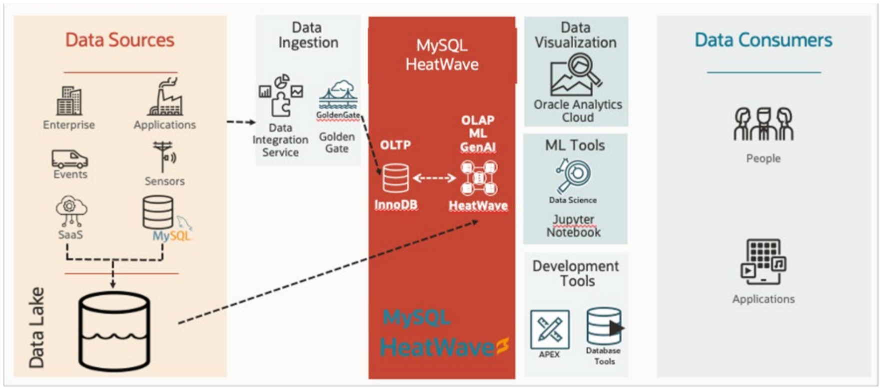
Auto Error Recovery

MySQL HeatWave automatically provisions new MySQL HeatWave node(s) when a hardware or software failure is detected on a node. When the cluster is restored, auto error recovery automatically reloads the data only to the re-provisioned node(s), allowing a very fast recovery.

Note: Auto thread pooling and auto shape prediction in MySQL HeatWave Autopilot are available in MySQL HeatWave on AWS, and will be available soon on OCI.

Integration with Oracle Cloud Services

OCI offers a wide range of services for data analytics, machine learning, and data lake. Native integration with these services makes it easier for existing applications to use MySQL HeatWave.



End to end integration with other Oracle Cloud Services from Data Ingestion to Data Visualization

Oracle Analytics Cloud (OAC) provides the industry's most comprehensive cloud analytics in a single unified platform, including self-service visualization and inline data preparation to enterprise reporting, advanced analytics, and self-learning analytics that deliver proactive insights. Integration with OAC provides a BI visualization platform for users to analyze their MySQL data.

OCI Data Integration Service provides extract, transform and load (ETL) capabilities to target data warehousing scenarios on the OCI platform. It supports various data sources, starting with relational, cloud, and Hadoop. Integration with OCI Data Integration allows users to easily transform and import data from data sources other than MySQL to MySQL HeatWave, expanding the scope of data that can take advantage of MySQL HeatWave.

OCI Database Management and Ops Insight provides comprehensive database performance diagnostics and management capabilities for MySQL HeatWave. It also provides information about resource use and capacity of databases and hosts for users to analyze CPU and storage resources, forecast capacity issues, and proactively identify SQL performance issues across a MySQL HeatWave database fleet.

Copyright © 2025, Oracle and/or its affiliates / Public

Conclusion

MySQL HeatWave is the only fully managed cloud service that provides integrated, automated, and secure generative AI and machine learning (ML) in one cloud service for transactions and lakehouse scale analytics—without the complexity, latency, risks, and cost of ETL duplication. Users get faster insights from all their data with unmatched performance and price-performance. MySQL HeatWave can be deployed in OCI, AWS, Azure, in a hybrid environment, and in customers' data centers with OCI Dedicated Region and Oracle Alloy.

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

Copyright © 2025, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Benchmark queries are derived from TPC-H benchmark, but results are not comparable to published TPC-H benchmark results since they do not comply with TPC-H specification.