MySQL HeatWave

One MySQL Database for OLTP, OLAP, and Machine Learning

March 2023

Public

Purpose statement

This document provides an overview of features and enhancements included in HeatWave. It is intended solely to help you assess the benefits of 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.

Table of contents

Executive Summary

MySQL is the world's most popular open source database because of its reliability, high-performance, and ease of use. It powers the world's most trafficked web sites including Facebook, Twitter, LinkedIn and Booking.com.

The companies that will thrive in the evolving digital landscape, will be those that make data, analytics and machine learning a core part of their strategy. According to McKinsey, 92% of company leaders surveyed believed that their business model would not remain viable at the current rate of digitization. This fear of disruption is the leading driver behind the investment in modern data, analytics and machine learning platforms.

MySQL HeatWave is a fully managed database service that lets developers quickly develop and deploy secure cloud native applications using the world's most popular open source database. MySQL HeatWave is the only MySQL service with a massively-scalable, integrated, real-time query accelerator, and a fully automated in-database machine learning engine. This service overcomes the limitations of traditional data warehouse, analytics and machine learning environments that use periodic long-running ETL batch jobs to refresh the data. MySQL HeatWave provides a unified MySQL database platform for OLTP, OLAP and machine learning.

> "We successfully migrated our 6TB database and in-house digital marketing and media management applications from AWS Aurora to MySQL HeatWave on OCI that reduced our costs by 60 percent and improved performance for complex queries by more than 1000x and overall workloads improved 85 percent."

Amit Palshikar
Co-Founder, CTO
Red3i

Challenges of Existing Data & Analytics Solutions

MySQL is optimized for OLTP, but it is not designed for analytic processing (OLAP). As a result, organizations that need to efficiently run analytics need to move their data to another database.

This approach of moving data to another database introduces complexity and additional cost to customers in multiple ways:

1. **Applications need to define complex logic** for extracting relevant data from MySQL.
2. **The extracted data needs to be transported to another database** across networks securely, consuming network bandwidth and incurring latency.
3. **Data in the other database needs to be manually kept in sync** with the MySQL database and as a result the data on which analytics is performed can become stale.
4. **Additional cost and overhead of managing multiple databases** for running OLTP and analytics applications.

Performance: Real Time Analytics

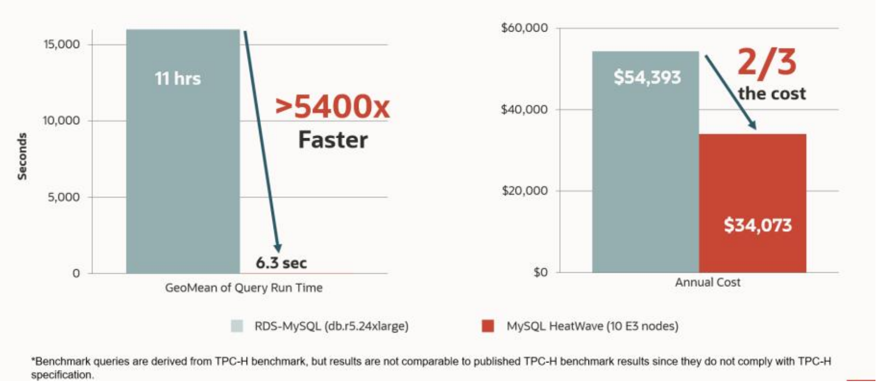
HeatWave is designed to enable customers to run analytics on data which is stored in MySQL databases, without the need for ETL. This service is built on an innovative, in-memory analytics engine which is architected for scalability and performance. This results in a very performant solution for SQL analytics at a fraction of the cost compared to other industry solutions.

Compared to other MySQL solutions, with a 4TB TPC-H Benchmark workload, HeatWave is much faster and cheaper:

HeatWave advantages on 4TB TPC-H			
Other MySQL solution	**Speedup**	**Price Performance**	**Cost**
Amazon Aurora	1400x	2800x	1/2
Amazon RDS for MySQL	5400x	8100x	2/3

Figure 1. Summary of HeatWave performance and cost advantage over other MySQL solutions on 4TB TPC-H workload

Customers with HeatWave will experience a 5400x speedup over Amazon RDS for MySQL and pay 1/3 less.



*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.

Figure 2. HeatWave accelerates Amazon RDS for MySQL queries by 5400x

Compared to Amazon Aurora, MySQL HeatWave offers dramatic improvements in performance for complex and analytic queries. MySQL HeatWave is more than 1400x faster than Amazon Aurora and costs half of Aurora. Furthermore, with HeatWave there is no need to create indexes on the base table which takes days with Amazon Aurora. As a result, the data is available to query much sooner with HeatWave than with Aurora.

> “We found MySQL HeatWave improved performance by 10 timesand 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

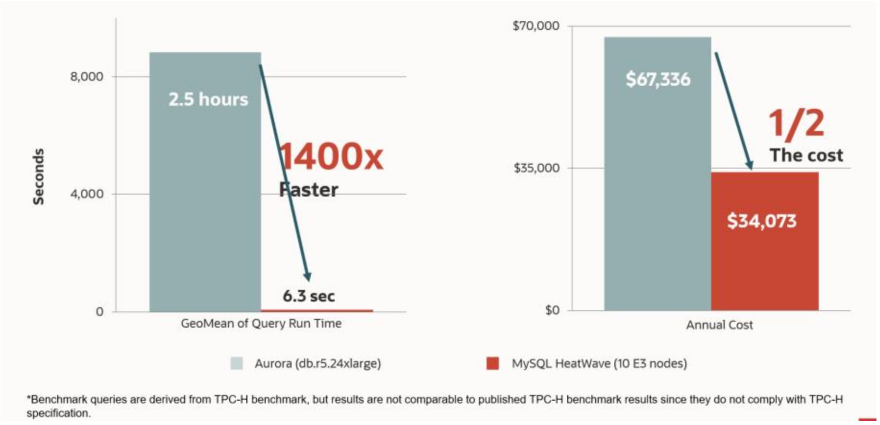


Figure 3. HeatWave is 1400x faster than Amazon Aurora

The performance improvement of MySQL HeatWave over Aurora increases with the size of data.

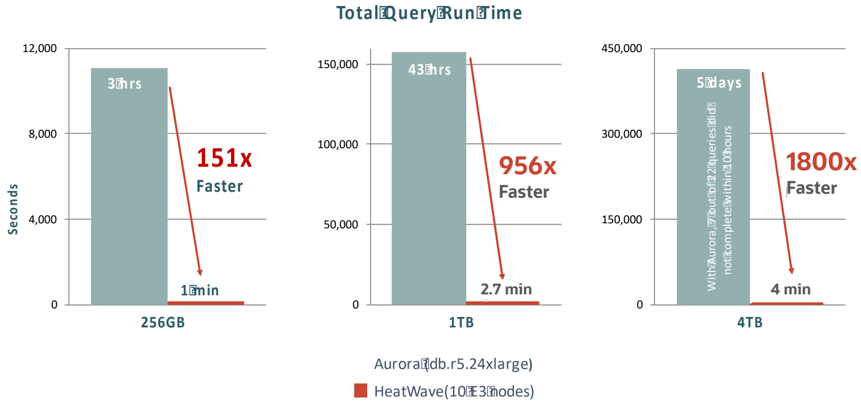


Figure 4. The performance advantage of HeatWave increases with data size vs. Amazon Aurora

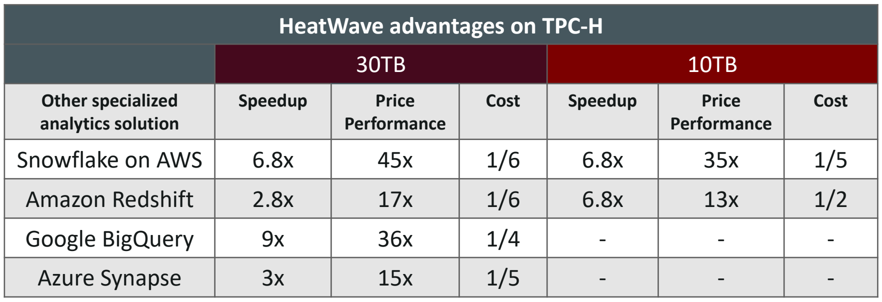
MySQL HeatWave is faster than all the competing analytics services at a fraction of the cost.

HeatWave advantages on 10TB TPC-DS			
Other specialized analytics solution	**Speedup**	**Price Performance**	**Cost**
Snowflake on AWS	2.5x	14.4x	1/5
Amazon Redshift	1.6x	4.8x	1/3
Google BigQuery	3.9x	12.9x	1/3
Azure Synapse	4.5x	14.9x	1/3

Figure 5. Summary of HeatWave performance and cost advantages over other analytics specific solutions on 10TB TPC-DS workload

> “MySQL HeatWave dramatically reduced our AWS Aurora and Redshift cost by more than 50 percent, we are no longer moving data around so now we have blazing fast, real-real-time insights with no effort. More importantly, scalability has made our expansion plan possible, allowing us to onboard more data and new clients without impact to costs. It's a dream come true.”

Pablo Lemos
Co-Founder, CTO
Tetris.co



HeatWave advantages on TPC-H						
	30TB			10TB		
Other specialized analytics solution	Speedup	Price Performance	Cost	Speedup	Price Performance	Cost
Snowflake on AWS	6.8x	45x	1/6	6.8x	35x	1/5
Amazon Redshift	2.8x	17x	1/6	6.8x	13x	1/2
Google BigQuery	9x	36x	1/4	-	-	-
Azure Synapse	3x	15x	1/5	-	-	-

Figure 6. Summary of HeatWave performance and cost advantages over other analytics specific solutions on 30TB and 10TB TPC-H workload

Amazon Redshift, which is designed for analytics, is offered in multiple shapes. With their latest instance shape (ra3.4xlarge) and latest advanced query accelerator (AQUA) turned on, HeatWave is 6.8x faster at half the cost for 10TB TPC-H data.

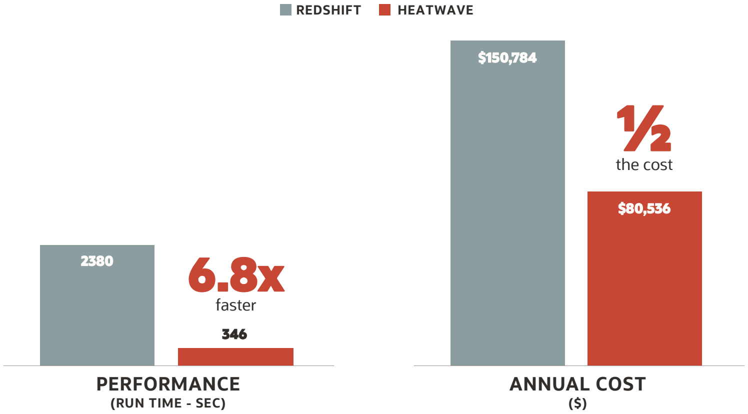


Figure 7. HeatWave performance and cost advantages Amazon RedShift AQUA on 10TB and 30TB TPC-H workload

Also, 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, eliminating the need for ETL, support for real-time analytics, reduced monthly cost and a single database for OLTP, OLAP, and machine learning.

Mixed workload

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 industry standard CH-benCHmark on a 100GB dataset:

For OLAP queries: HeatWave is 18x faster, provides 110x better throughput and is 2.4x cheaper than Aurora for OLAP queries resulting in 42x better price performance.

> “We recently migrated our production workload from another cloud solution to MySQL HeatWave. 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

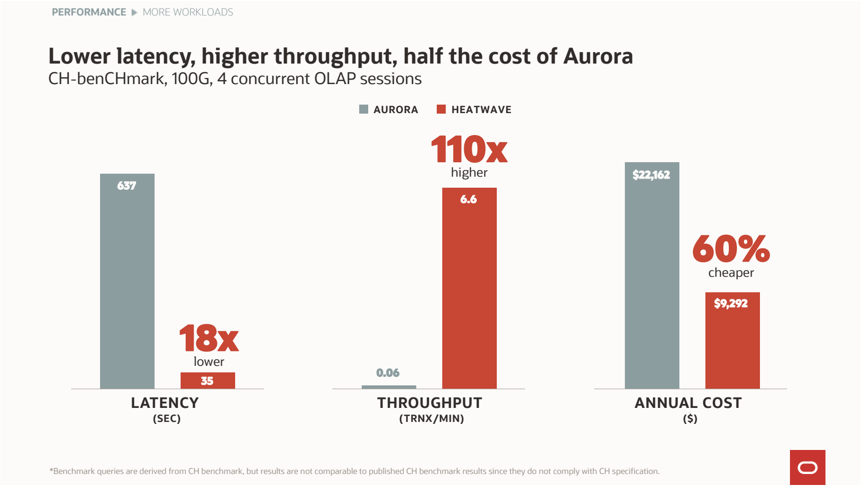


Figure 8. HeatWave advantages over Amazon Aurora for OLAP queries on 100GB mixed workload

For OLTP queries, MySQL HeatWave has the same performance as Aurora and costs 2.4x less resulting in 2.4x better price performance.

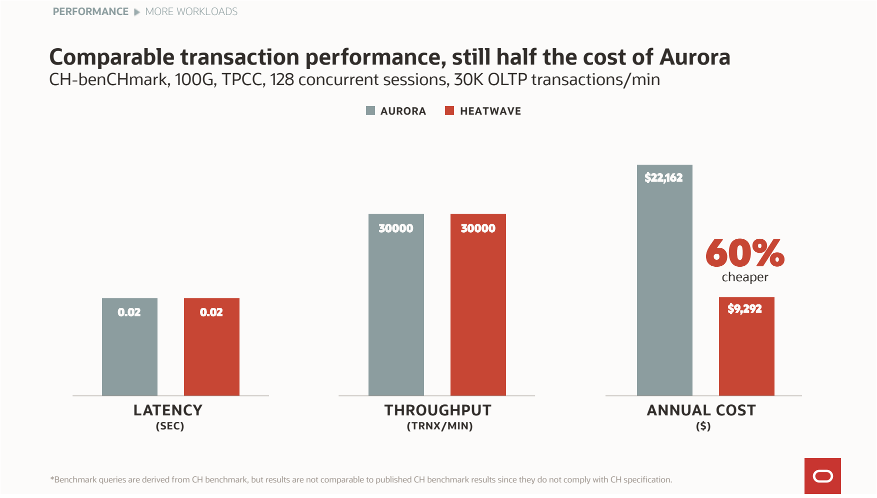


Figure 9. HeatWave advantages over Amazon Aurora for OLTP queries on 100GB mixed workload

Deployment Scenarios

MySQL HeatWave is a fully managed service. The MySQL database has been enhanced to natively integrate HeatWave, as a result, customers who store data in MySQL can seamlessly run analytics and machine learning by enabling this service.

A MySQL HeatWave instance is a cluster composed of a MySQL instance and multiple HeatWave nodes. When HeatWave is enabled, the HeatWave server is installed on the MySQL node. It is responsible for cluster management, loading data into the memory of the HeatWave nodes, query scheduling and query execution. MySQL applications written in Java, PHP, Ruby, etc. work seamlessly with HeatWave using standard MySQL ODBC/JDBC connectors. HeatWave is an in-memory accelerator and supports all MySQL syntax. Hence, all existing tools and applications built using standard SQL will work without requiring any modification to queries (Figure 10).

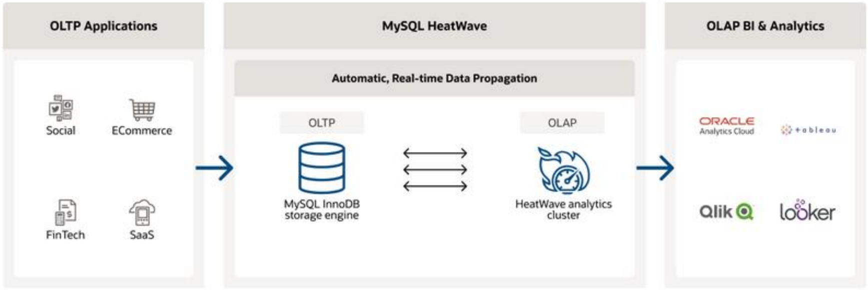


Figure 10. Existing tools and applications work seamlessly with HeatWave

Data which is needed for analytic processing is stored in memory of the 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 which is achieved on the dataset, and the query characteristics. The number of nodes needed can be automatically derived by using the Auto Provisioning advisor which is available with HeatWave.

HeatWave supports 2 shapes and up to 64 nodes per cluster for workload less than 50GB and up to 64TB of data:

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

64TB is the approximate maximum amount of data which can be populated in the memory of the HeatWave nodes at a given moment. There is no limit to the amount of data which can be stored in the MySQL database and customers can choose which tables or columns from MySQL database schema to load into the memory of HeatWave nodes. If the tables are no longer needed by queries, user can remove the tables from the memory to make room for other data.

HeatWave provides a great solution for customers who need to run transactional and analytical workloads. While transactional queries are run in the MySQL node, data updated in MySQL InnoDB is transparently propagated to the HeatWave cluster in real-time for accelerated analytical processing. This enables customers to run OLTP and real-time analytics workloads simultaneously within a single database platform (Figure 11).

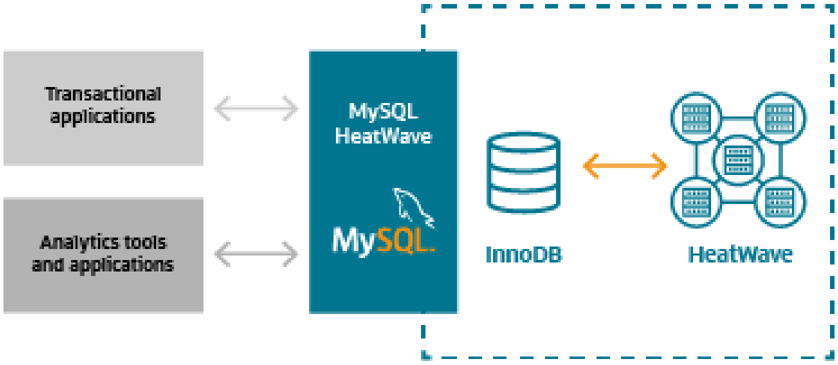


Figure 11. Single MySQL database for OLTP and OLAP applications, no ETL required.

For customers with multi-cloud strategy, MySQL HeatWave is available on OCI, AWS and Azure (Fig 12), as well as in customers' data centers with OCI Dedicated Region. Customers can take advantage of MySQL HeatWave on the cloud platform that they prefer.

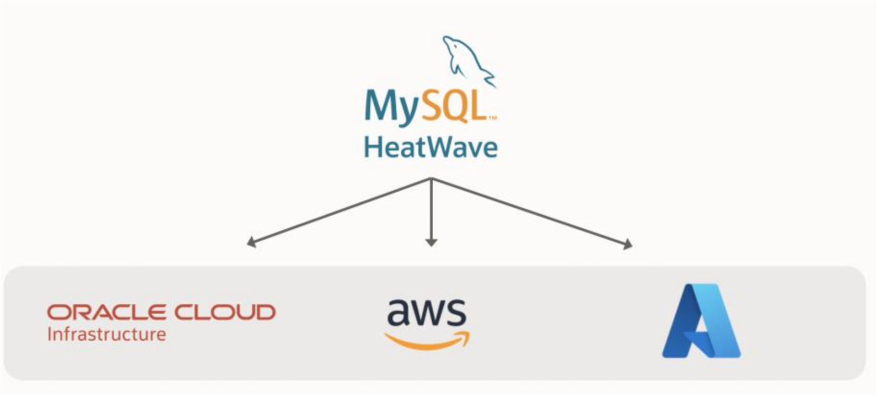


Figure 12. MySQL HeatWave is optimized to deliver best price performance on OCI, AWS ans Azure respectly

On-premises customers who cannot move their MySQL deployment to a cloud due to compliance or regulatory requirements, can still leverage HeatWave by using the hybrid deployment model as shown in Figure 13. In such a hybrid deployment, customers can leverage MySQL replication to replicate on-premises MySQL data to HeatWave without the need for ETL.

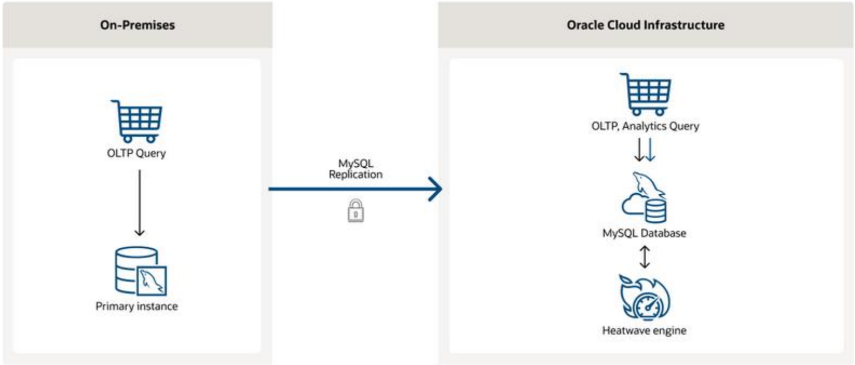


Figure 13. Hybrid deployment for enabling analytics on data stored on premise

HeatWave Architecture

There are five key architectural choices that lead to a compelling performance and cost advantage with 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. HeatWave scale out data management layer enables 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 Autopilot provides machine learning automation that improves the performance, scalability, and ease of use of HeatWave. It automates the database lifecycle operations including provisioning, data loading, query processing, and error handling. It also provides capabilities for OLTP workloads.

5. HeatWave AutoML provides familiar MySQL interfaces which enables MySQL users to build and train machine learning models, generate inferences and explanations without the need to have extensive ML expertise.

The Heatwave engine uses a columnar in-memory representation that facilitates vectorized processing, leading to very good query performance (Figure 14). 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 to reduced cost for customer.

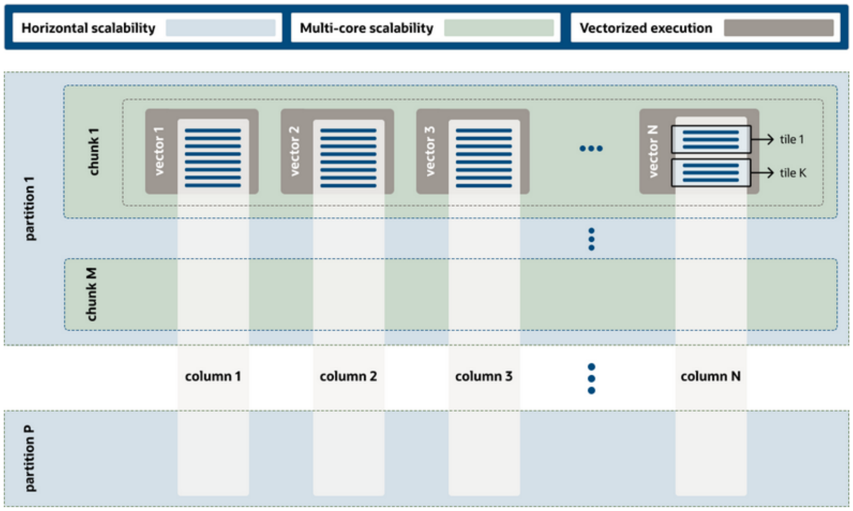


Figure 14. Vectorized in-memory columnar representation for analytic processing

One of the key design points of the HeatWave engine is to massively partition data across a cluster of 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 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.

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 communication of data across nodes which helps achieve good scalability (Figure 15).

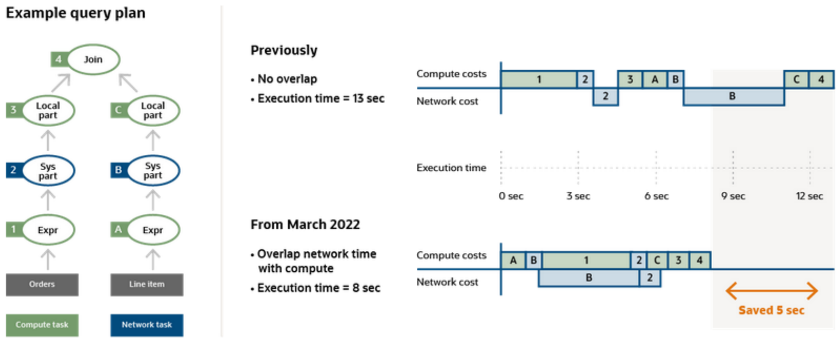


Figure 15. HeatWave implements state of art algorithms for distributed in-memory analytic processing.

Native MySQL Analytics

The integration of HeatWave with MySQL provides a single data management platform for OLTP, OLAP, and ML. HeatWave is designed as a MySQL pluggable storage engine, which completely shields all the low-level implementation details at the storage level from the end users. As a result, users can manage both HeatWave and MySQL with the same management tools, including the OCI console, REST APIs, and the command line interface.

Since HeatWave is an in-memory processing engine, data is persisted in the MySQL InnoDB storage engine. This allows users to manage analytics data the same way they manage transactional data in MySQL.

Users and applications interact with HeatWave through the MySQL database node in the cluster. Users connects to HeatWave through standard tools and standard-based ODBC/JDBC connectors. HeatWave supports the same ANSI SQL standard and ACID properties as MySQL and supports diverse data types. This enables existing applications to take advantage of HeatWave without any changes to their application, allowing easy and quick integration.

Once users submit a query to the MySQL database, the MySQL query optimizer transparently decides if the query should be offloaded to the HeatWave cluster for accelerated execution. This is based on whether all operators and functions referenced in the query are supported by HeatWave and if the estimated time to process the query with the HeatWave engine is less than with MySQL. If both conditions are met, the query is pushed to HeatWave nodes for processing. Once processed, the results are sent back to the MySQL database node and returned to users (Figure 16).

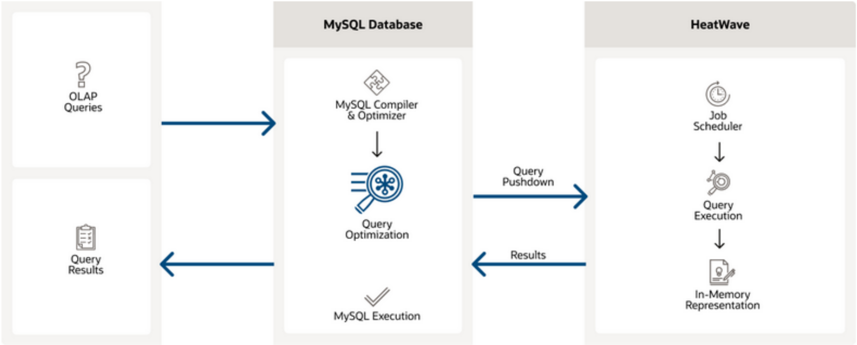


Figure 16. MySQL Integration for Query Processing

Any updates to the tables in InnoDB that are also loaded in HeatWave, are automatically propagated to the memory of the HeatWave nodes in real time. This allows subsequent queries to always have access to the latest data, as shown in Figure 17. This is done behind the scene by a light-weight change propagation algorithm that can keep up with MySQL data update rates.

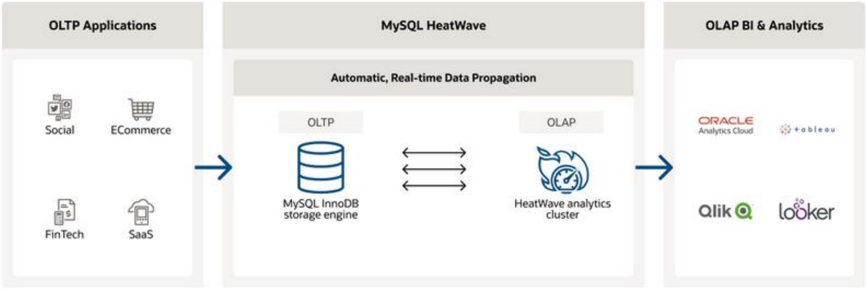


Figure 17. HeatWave integration with MySQL

Scale-out Data Management

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

To improve service uptime, 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 HeatWave formatted data in a persistent storage, allowing reload of data in constant time regardless of data size.

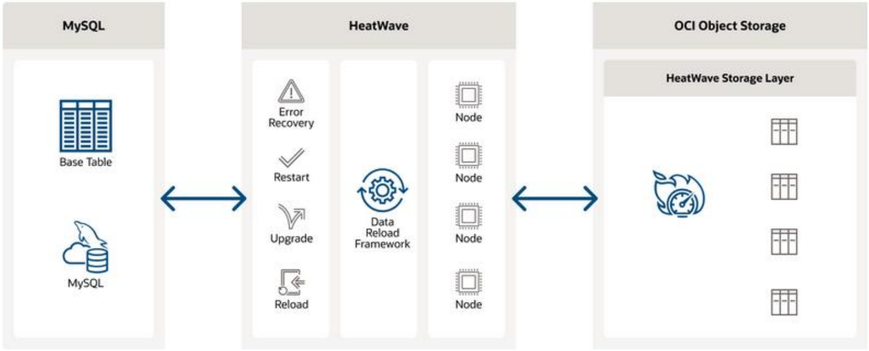


Figure 18. HeatWave scale-out storage layer

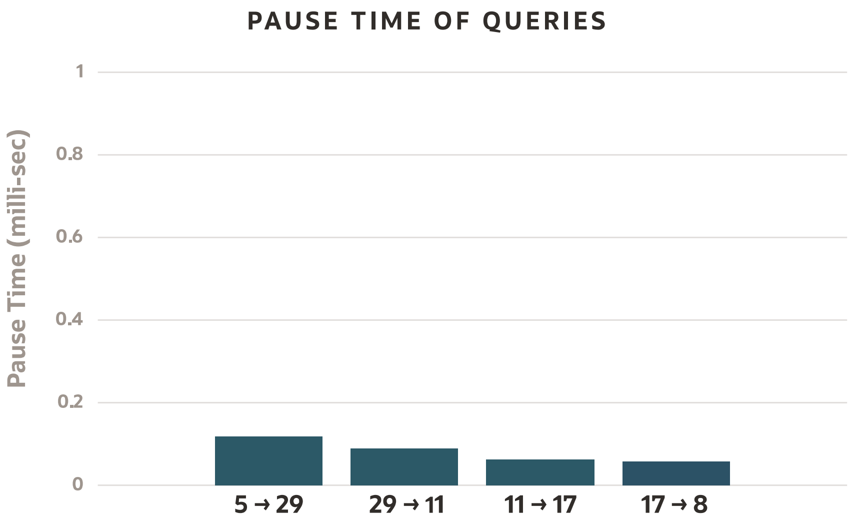
Figure 18 shows the new HeatWave architecture with the scale-out storage layer. In the HeatWave storage layer, persisted data is organized in the same way as that of in-memory data. Each HeatWave node can restore data independently and in parallel, allowing a very fast and near constant time data reload.

Real-time Elasticity

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

The HeatWave cluster can be scaled up or down to any number of nodes. During the scaling operation, the cluster continues to work, there is no interruption in client connections for read or write and there is minimal effect on query performance in HeatWave. For sizing up, data is loaded from the HeatWave storage layer to the newly created nodes. Once the data is loaded, HeatWave updates its metadata and queries. They will then be processed in all the nodes in the new cluster size. For sizing down, additional data is loaded from the HeatWave storage layer to the HeatWave nodes that will be kept for the new reduced cluster size. Once the additional data is loaded, HeatWave updates its metadata and queries, and they 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.

Figure 19 shows how fast and predictable the data loading time based on different cluster sizes, and the neglectable time (in microseconds) that the HeatWave cluster needs to update it metadata.



Always available:
Only paused for updating partition
maps in memory

Figure 19. HeatWave real-time elasticity performance

MySQL Autopilot

MySQL Autopilot automates many of the most important and often challenging aspects of achieving high query performance at scale - including provisioning, 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. These machine learning models are then used by MySQL Autopilot to execute its core capabilities. MySQL Autopilot makes the HeatWave query optimizer increasingly intelligent as more queries are executed, resulting in continually improving system performance over time. MySQL Autopilot also provides capabilities to improve the performance and price-performance of OLTP workloads.

MySQL Autopilot focuses on four aspects of the service lifecycle: system setup, data load, query execution and failure handling (Figure 20).

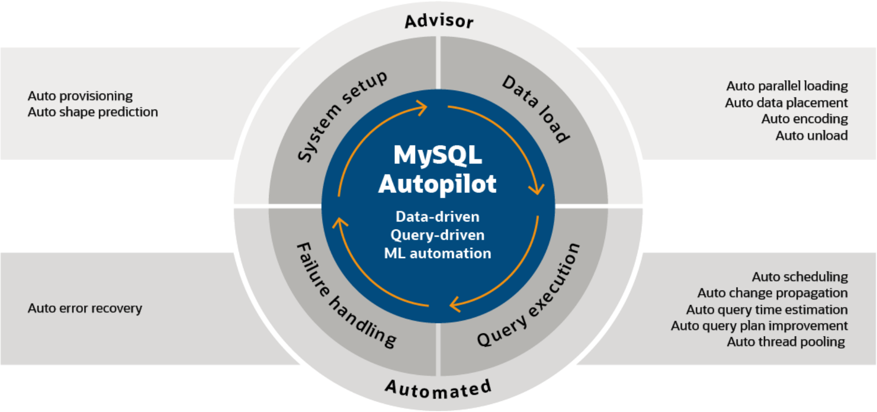


Figure 20. MySQL Autopilot automates different aspects of the service to improve performance, scalability and usability of the system

System Setup

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

Data Load

3. ***Auto parallel loading*** optimizes the load time and memory usage by predicting the optimal degree of parallelism for each table being loaded into HeatWave.
4. ***Auto encoding*** determines the optimal representation of columns being loaded into HeatWave taking queries into consideration. This optimal representation provides the best query performance and minimizes the size of the cluster to minimize the cost.
5. ***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.
6. ***Auto unload*** determines unused or rarely used tables in HeatWave, and predicts the memory saved from unloading the tables.

Query Execution

7. ***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.
8. ***Auto query time estimation*** estimates the execution time of a query prior to executing the query, allowing quick tryout and test on different queries
9. ***Auto change propagation*** intelligently determines the optimal time when changes in MySQL Database should be propagated to the

HeatWave storage layer. This ensures that changes are being propagated at the right optimal cadence.

10. ***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.
11. ***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

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

Auto Provisioning

Auto Provisioning provides recommendation on how many 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 HeatWave cluster memory. The size of the cluster needed depends on tables and columns required to load, and the compression achieved in memory for this data. Figure 20 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.

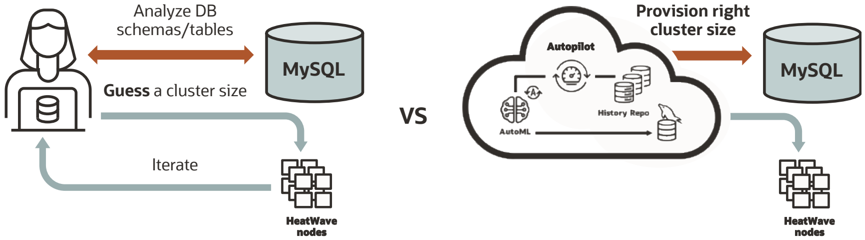


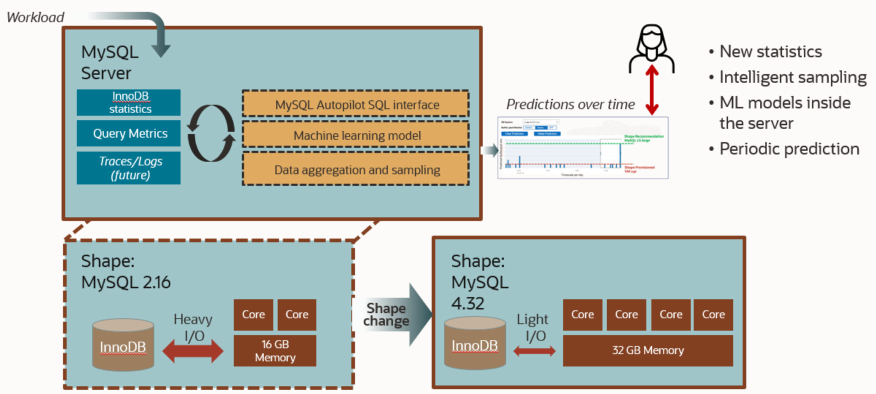
Figure 21. Comparison of a manual provisioning vs Auto provisioning

The right side of figure 21 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 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. SiInce 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 Parallel Load

Loading data into HeatWave involves several manual steps. The time required to perform these steps depends on the number of schemas, tables, and columns, and statistics. Auto parallel load automates these steps by predicting the degree of parallelism per table via machine-learning models to achieve optimal load speed and memory usage.

Auto Encoding

HeatWave supports two string column encoding types: variable-length and dictionary. The type of encoding affects the query performance as well as the supported query operations. It also affects the amount of memory required for HeatWave nodes. By default, HeatWave applies variable-length encoding to string columns when data is loaded, which may not be the optimal encoding choice for query performance and cluster memory usage for certain workloads.

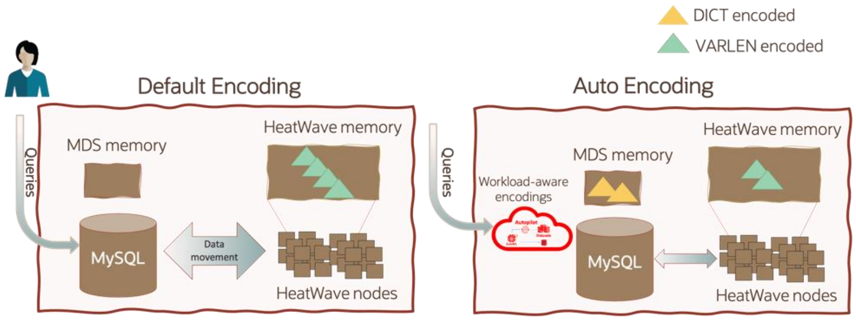


Figure 22. Comparison of a default encoding vs auto encoding

Auto encoding provides recommendations for string columns that reduce memory usage and help improve query performance. Figure 22 shows the difference between default encoding and auto encoding. In the default case, the variable-length encoding ensures best query offload capability. However, this can impact ideal query performance due to data movement between HeatWave nodes and MySQL nodes. Auto encoding uses machine learning to analyze column data, HeatWave query history, and available MySQL node memory to identify which string columns can be coded in dictionary encoding. When the suggestion is applied, the overall query performance is improved due to reduced data movement in system, and HeatWave memory usage is reduced due to efficient (i.e., smaller) dictionary codes and the corresponding dictionaries that maintain the mapping between the strings and the codes reside in the memory of the MDS node.

Auto Data Placement

Data placement keys are used to partition table data when loading tables into HeatWave. Partitioning table data by JOIN and GROUP BY key columns can improve query performance by avoiding costs associated with redistributing data among HeatWave nodes at query execution time.

Determining the best data placement key is a tedious task requiring understanding query access patterns and system behavior. Moreover, picking wrong partitioning keys can lead to sub-optimal performance due to increased data distribution costs during query execution time.

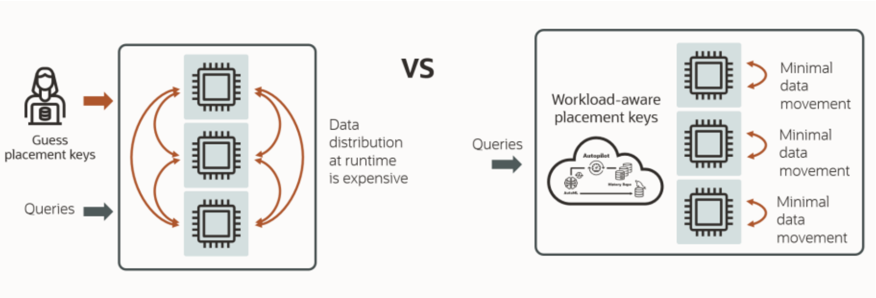


Figure 23. Comparison of manual data placement vs auto data placement

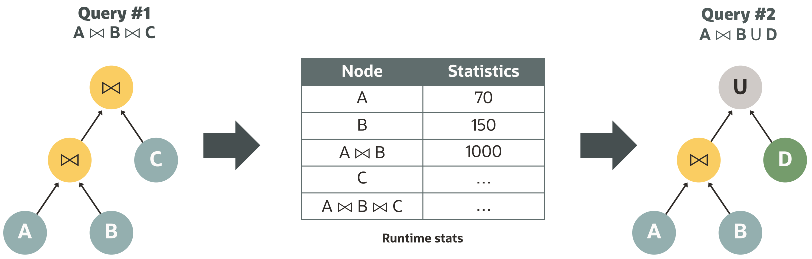
Figure 23 depicts a comparison between default query execution and execution with auto data placement. Based on machine learning models, auto data placement recommends appropriate data placement keys by analyzing table statistics and HeatWave query history and provides an estimation of query performance improvement. Once the suggestion is applied, query performance is improved by minimizing the data movement between nodes during execution.

Auto Unload

Users can save cost by unloading tables that are never or rarely queried in HeatWave to lower the total memory required and decrease the HeatWave cluster size. However, this requires understanding all queries and access patterns of the tables. Auto unload automates and tracks table usage and provides recommendations along with an explanation to support them.

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, HeatWave creates better query plan and makes better decisions on the underlying physical operators; consequently improves the overall query performance.



Node	Statistics
A	70
B	150
A ⋈ B	1000
C	…
A ⋈ B ⋈ C	…

Figure 24. Query 2 benefits from statistics of a previous similar query (query 1) with auto query plan improvement

Figure 24 shows how auto query plan improvement works without user intervention. After a query (Q1) executes on 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 Query Time Estimation

Users are often interested in accurate query time estimates before running the query. Such functionality allows users to estimate their application performance better and to understand the resource needed. Auto query time estimation not only provides user-visible estimations for query run times, but it also uses the same building blocks internally to improve query performance by optimizing query (sub-)plans.

Instead of using static, analytical models, auto query time estimation integrates a data-driven query time estimation module, which improves as queries run. To do so, HeatWave leverages load- and run-time statistics and dynamically tunes query cost models during execution. As a result, auto query time estimation improves with time as more queries are executed on the system.

Auto Change Propagation

Data updated in MySQL is propagated and persisted to HeatWave data layer as change logs. During data reload, HeatWave first restores data from the base data, then applies the data from the change logs. Over time, the persisted change log volume increases, which can result in an increased reload time as all the change logs need to be applied to the base data. So, the change logs are consolidated from time-to-time to alleviate increased reload latency as shown in Figure 25. However, determining when to consolidate is not an easy task, which depends on several factors such as transaction rate, system load, failure probability.

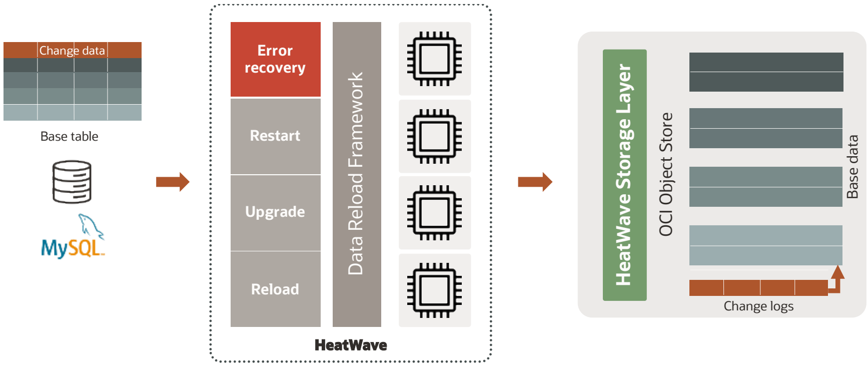


Figure 25. Auto change propagation

To minimize consolidation time during reloading from the storage layer, auto change propagation uses data driven mechanism to determine the best change propagation interval and choice. Auto change propagation analyzes rate of changes, incoming DMLs, object storage resources, and previously seen change activity. As a result, the changes are propagated at the best time interval, which results in optimized consolidation time for critical system operations.

Auto Scheduling

Traditional database systems process queries based on their arrival time, which can result in long-running queries starving short-running queries, as shown in Figure 26.

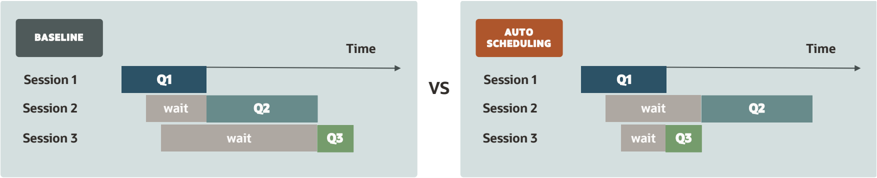


Figure 26. Traditional database system vs HeatWave auto scheduling

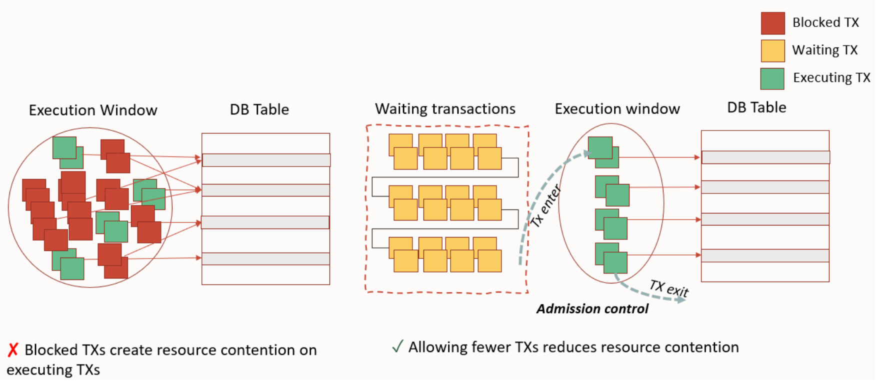
On the left, is a sub-optimal case where three queries (Q1, Q2, Q3) from three user sessions arrive one after the other and are scheduled in the FIFO order. After the execution completes, one can identify that waiting time for Q3 could be reduced significantly with minimal impact on Q2 latency.

On the right, it shows how auto scheduling improves user experience for short running queries in a multi-session application. Auto scheduling identifies and prioritizes short-running queries by automatically classifying queries into short or long queries using HeatWave data driven algorithms. Therefore, Q3 is prioritized before Q2 as Q3 is identified as a short-running query.

Auto Scheduling reduces 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.

Auto Thread Pooling

With Auto Thread Pooling, MySQL HeatWave prioritizes not only peak single-thread performance, but also high throughput in the presence of concurrent clients running concurrent queries on a MySQL server. With this feature, the MySQL server now can perform 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.



Auto Error Recovery

HeatWave automatically provisions new 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 Autopilot are available in MySQL HeatWave database on AWS, and will be available soon on OCI.

HeatWave AutoML

Current challenges of Machine Learning in databases

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

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.

The current approach to use machine learning in MySQL requires the user to perform ETL (Extract, Transform, Load) on the database table. The data must be extracted from the database, the user must learn and use third-party tools and libraries to train a model and then perform inference and explanations. In addition to being onerous and time consuming, this process also has the potential to proliferate data outside of the database, causing data security and governance issues.

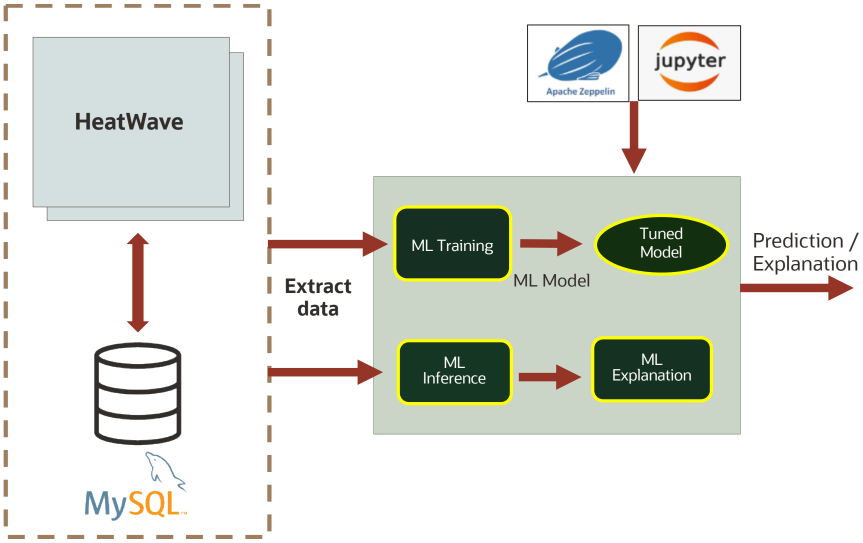


Figure 27. The current approach to machine learning requires the cost, complexity and risk of ETL from the database.

HeatWave AutoML Advantages

HeatWave AutoML enables MySQL users to train a model, generate inferences and explanations, without extracting data out of the MySQL database. It provides several advantages:

- **Fully Automated:** 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 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 HeatWave AutoML is much better at a lower cost than competing services such as Redshift ML. Furthermore, HeatWave AutoML scales with the size of the cluster.
- **Easy Upgrades**: HeatWave AutoML leverages state-of-the-art open-source Python ML packages that enable continual and swift uptake of newer (and improved) versions.

All these capabilities are available to MySQL HeatWave customers without any additional charge.

The ML functionality in HeatWave is incorporated within the database. By taking this approach, the data does not have to be extracted from the database. Training, inference, and explanation activities are performed in-database, without moving the data.

Analytics queries and ML queries share a common query queue, and analytic queries are prioritized over ML queries. Note that analytics data and ML data will share memory resources, with ML memory usage constrained to pre-set memory limits.

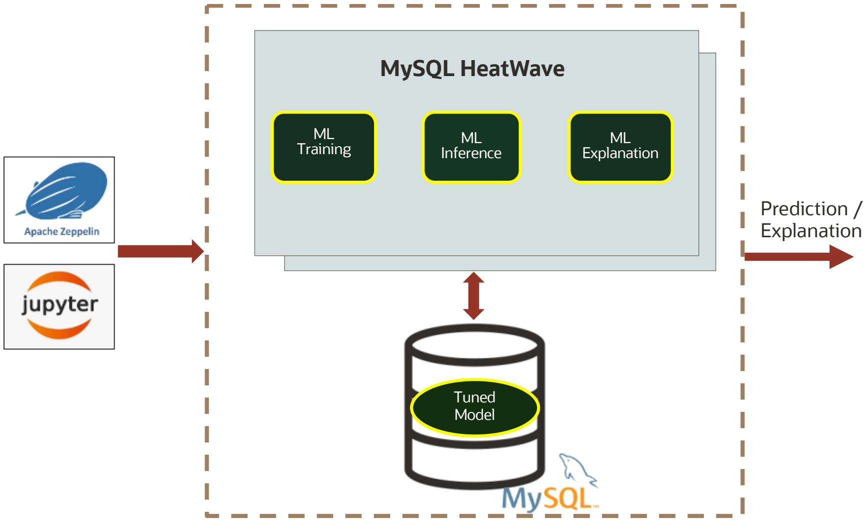


Figure 28. HeatWave AutoML is a native, in-database solution, eliminating the cost, complexity and risk of ETL.

Technology Background

HeatWave AutoML leverages Oracle AutoML, which automates the task of generating models. It replaces the laborious and time-consuming tasks that a data scientist performs as listed below:

1. Preprocess the data
2. Select an algorithm from a set of algorithms to create a model
3. Select a suitable representative sample of data
4. Select only the relevant features to speed up the pipeline and reduce over fitting
5. Tune the hyperparameters
6. Ensure the model performs well on unseen data (also called generalization)

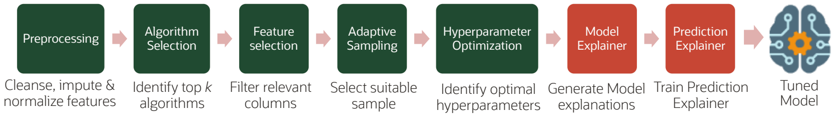

Preprocessing
Cleanse, impute & normalize features
Algorithm Selection
Identify top *k* algorithms
Feature selection
Filter relevant columns
Adaptive Sampling
Select suitable sample
Hyperparameter Optimization
Identify optimal hyperparameters
Model Explainer
Generate Model explanations
Prediction Explainer
Train Prediction Explainer
Tuned Model

Figure 29: Machine learning pipeline automated by HeatWave AutoML

Oracle AutoML has a scalable design, minimizes the number of trials by extensive use of meta-learning, and provides an optimal model given a time budget. This proven technology has been integrated and in use in various Oracle products, including the OCI Data Science Service and the Oracle Database.

Security Benefits

HeatWave AutoML performs ML model training, inference and explanations on the data stored in MySQL database, without the data ever leaving the MySQL database. All data and operations are executed in memory within the HeatWave cluster and the trained model is automatically stored in MySQL database without any user data or model being transmitted to the client. The fully automated nature of the approach ensures that there is no human error that can cause security problems or errors in computation or data handling. This makes HeatWave AutoML far superior in terms of security to any existing ML solution for MySQL customers.

By taking the above approach, MySQL HeatWave enables the users to perform transactional, analytic as well as machine learning queries all within the same system.

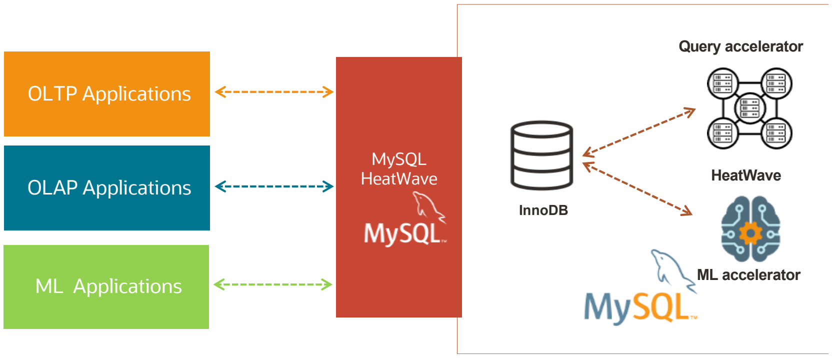


Figure 30: OLTP, OLAP and ML workloads in a single database

Performance and Scalability

HeatWave AutoML is designed for high performance and scalability. High performance is achieved by the automated machine learning pipeline that consists of a novel non-iterative architecture comprised of multiple sequential stages. This design speeds up the pipeline as every stage's decision is made in a feed-forward manner. Key to the design is the reliance on *proxy models*-fast-performing models that are indicative of the performance of the final tuned model on a subset of a dataset. Furthermore, the algorithm selection stage at the beginning of the pipeline ensures that the downstream algorithm-dependent stages perform well without the need to iterate over multiple algorithms. However, even with this non-iterative approach, each pipeline training can consist of hundreds of individual model training calls. This is a very compute intensive task, as each model fit can utilize a significant amount of time, compute, and memory resources. Therefore, it is crucial to take advantage of as many compute resources as are available to parallelize not only within an individual model fit call, but also across separate model fit calls.

The HeatWave AutoML pipeline is designed in a flexible and highly parallel fashion, enabling us to distribute individual model fits and multiple parallel fits to all available compute nodes on a given HeatWave cluster. HeatWave AutoML has been optimized for both intra- and inter-model parallelism to achieve optimal performance on HeatWave cluster nodes. HeatWave AutoML can scale to dozens of HeatWave nodes (hundreds of cores), significantly reducing the ML training runtime as the cluster scales up. Furthermore, as training data size grows, user can scale up the cluster size to minimize the increase in training time.

Explainability

HeatWave AutoML not only finds the most accurate ML model for the provided input table, with its integrated explainability module, it also helps users understand and interpret the model and its predictions.

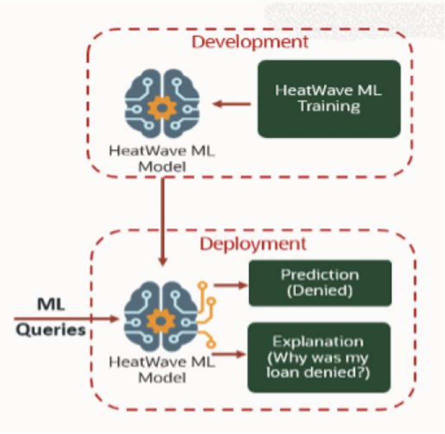


Figure 31: Model Development and Deployment

Deriving insights from the data and model helps the user answer questions around what factors matter most, why the model performs the way it does, and how it can be improved. Explainability can contribute to understanding the strengths and weaknesses in the user's data and within the predictions themselves.

Briefly, explainability performs knowledge discovery by explaining which factors matter most to the model (captured during model training), and which factors contribute the most to individual predictions (via explanations). Thus, it can help with the following:

- **Regulatory compliance,** may assist with the 'right to an explanation' for algorithms affecting users

- **Trust,** by involving the human analyst in the predictive process and providing an interpretable explanation. This encourages them to adopt machine learning systems.
- **Fairness,** by helping to ensure that predictions are unbiased and do not implicitly or explicitly discriminate against underrepresented groups.
- **Repeatability,** by conducting sensitivity analysis to help users ensure that small changes in the input do not lead to large changes in the prediction.
- **Causality,** by ensuring that users can check that only causal relationships are picked up.

The HeatWave AutoML explainability module has several key differentiators that set it apart from the competition:

Quality

- Model-agnostic techniques allow explanation of complex models
- Explanations are repeatable and reliable due to a random choice of evaluated combinations
- Does not require a reference dataset when performing explanations

Performance and Scalability

- Explanations scale linearly with the number of features (not exponentially)
- Real-time explanations are possible due to the distribution of work to exploit inter and intra node parallelism

Interpretability

- The intuitive interpretation of feature attributions assists users determine which factors contribute most to a prediction
- Additional explanation methods like what-if scenarios and counterfactual analysis are planned for future

Performance Comparison

Benchmarks are run on several datasets that are relevant to enterprise use cases and compared the performance of training time, quality of models and cost with Redshift ML. This was done for classification and regression datasets.

Classification Results

Here are the datasets which were chosen for classification.

Dataset	Explanation	# rows (Training set)	# features
Airlines	Predict Flight Delays	377,568	8
Bank Marketing	Direct marketing – Banking Products	31,647	17
CNAE-9	Documents with free text business descriptions of Brazilian companies	756	857
Connect-4	8-ply positions in the game of connect-4 in which neither player has won yet – predict win/loss	47,289	43
Fashion MNIST	Clothing classification problem	60,000	785
Nomao	Active learning is used to efficiently detect data that refer to a same place based on Nomao browser	24,125	120
Numerai	Data is cleaned, regularized and encrypted global equity data	67,424	22
Higgs	Monte Carlo Simulations	10,500,000	29
Census	Determine if a person makes > 50k	32,561	15
Titanic	Survival Status of individuals	916	14
Credit Card Fraud	Identify fraudulent transactions	199,364	31
KDD Cup (appetency)	Predict the propensity of customers to buy new products	35,000	231

Figure 32: Classification datasets

The table below compares the balanced accuracy, training times and cost of HeatWave AutoML with Redshift ML. HeatWave AutoML is 25x faster in training time with better accuracy and 97x cheaper than Redshift ML.

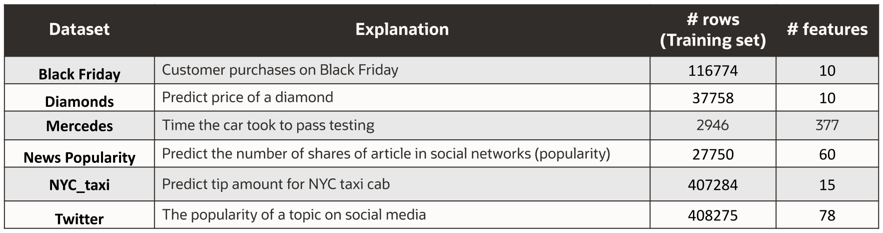
This enables users to retrain model more frequently, thereby keep models current with the data, which result in better prediction accuracy.

Dataset	Accuracy		Training Time (minutes)		Speedup	Cost ($)		Cheaper
	Redshift ML	HeatWave ML	Redshift ML (MAX_CELLS: 1M, MAX_RUNTIME: 5400s)	HeatWave ML (2 nodes)		Redshift ML 1 year plan	HeatWave ML	
Airlines	0.5	0.6524	90.00	2.71	**33.21**	6.23	0.0479	**130.03**
Bank	0.8378	0.7115	90.00	3.72	**24.19**	5.68	0.0658	**86.30**
CNAE-9	X	0.9167	X	5.91	**X**	X	0.1045	**X**
Connect-4	0.6752	0.6970	90.00	7.13	**12.62**	6.18	0.1261	**49.05**
Fashion MNIST	X	0.9073	X	181.85	**X**	X	3.2151	**X**
Nomao	0.9512	0.9602	90.00	3.30	**27.27**	5.96	0.0583	**102.14**
Numerai	0.5	0.5184	90.00	0.34	**264.71**	5.49	0.0060	**913.49**
Higgs	0.5	0.758	90.00	68.58	**1.31**	7.27	1.2125	**5.99**
Census	0.7985	0.7946	90.00	1.22	**73.77**	6.12	0.0216	**283.95**
Titanic	0.9571	0.7660	90.00	0.47	**191.49**	5.60	0.0083	**674.32**
CC Fraud	0.9154	0.9256	90.00	29.06	**3.10**	6.70	0.5138	**13.03**
KDD Cup	X	0.5	X	3.55	**X**	X	0.0628	**X**
GEOMEAN	**0.712**	**0.754**	**90.00**	**3.561**	**25.271**	**6.115**	**0.063**	**97.129**

Figure 33: Accuracy, training time and cost comparisons

Regression Results

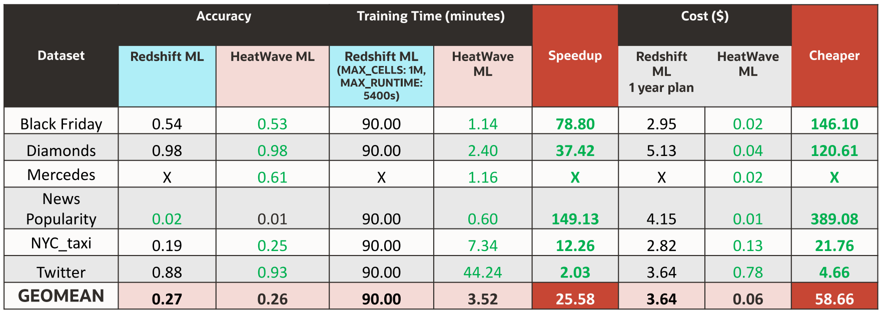
Following are the data sets used for Regression



Dataset	Explanation	# rows (Training set)	# features
Black Friday	Customer purchases on Black Friday	116774	10
Diamonds	Predict price of a diamond	37758	10
Mercedes	Time the car took to pass testing	2946	377
News Popularity	Predict the number of shares of article in social networks (popularity)	27750	60
NYC_taxi	Predict tip amount for NYC taxi cab	407284	15
Twitter	The popularity of a topic on social media	408275	78

Figure 34: Regression Datasets

The table below compares the r2 values, training times and cost with Redshift ML. HeatWave AutoML is 25x faster in training time with comparable accuracy and 59x cheaper than Redshift ML.



Dataset	Accuracy		Training Time (minutes)		Speedup	Cost ($)		Cheaper
	Redshift ML	HeatWave ML	Redshift ML (MAX_CELLS: 1M, MAX_RUNTIME: 5400s)	HeatWave ML		Redshift ML 1 year plan	HeatWave ML	
Black Friday	0.54	0.53	90.00	1.14	**78.80**	2.95	0.02	**146.10**
Diamonds	0.98	0.98	90.00	2.40	**37.42**	5.13	0.04	**120.61**
Mercedes	X	0.61	X	1.16	**X**	X	0.02	**X**
News Popularity	0.02	0.01	90.00	0.60	**149.13**	4.15	0.01	**389.08**
NYC_taxi	0.19	0.25	90.00	7.34	**12.26**	2.82	0.13	**21.76**
Twitter	0.88	0.93	90.00	44.24	**2.03**	3.64	0.78	**4.66**
GEOMEAN	**0.27**	**0.26**	**90.00**	**3.52**	**25.58**	**3.64**	**0.06**	**58.66**

Figure 35: Accuracy, training time and cost comparisons

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 HeatWave (Figure 36).

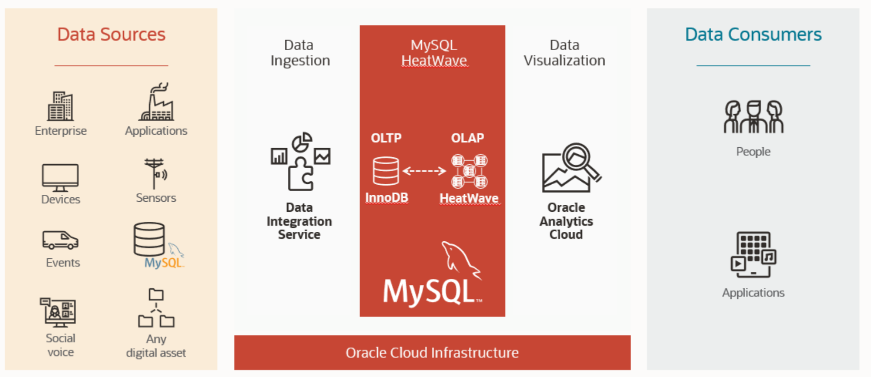


Figure 36. End to end integration with other Oracle Cloud Services from Data Ingestion to Data Visualization

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 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 HeatWave, expanding the scope of data that can take advantage of HeatWave.

1/2 the cost of AWS Redshift and Aurora one-year prepaid

The cost of using MySQL HeatWave depends on the number of HeatWave nodes provisioned. The size of a Heatwave cluster depends on the size of the dataset, and the characteristics of the workload. A single HeatWave node can hold approximately 1TB of data. Customers can expect to see a significant reduction in their costs when they migrate to HeatWave. Compared to Amazon Aurora and Redshift, HeatWave is 1/2 the cost of Amazon reserved instance one-year pre-paid cost (Figure 1 & 6).

Customers are likely to find that the cost benefit with HeatWave is higher because of the following:

- Amazon Aurora charges additional fees for Storage IO cost which can be substantial. There is no such cost with HeatWave
- When using HeatWave, the cost includes OLTP, OLAP, and machine learning capabilities. With Amazon Redshift the cost is only for OLAP, and additional costs are needed for OLTP and ML.
- Customers need to pay as they move their data around from one database to another with Amazon Aurora or Amazon Redshift.

MySQL HeatWave offers pay as you go and Universal Credit Annual Flex pricing.

MySQL HeatWave Lakehouse

MySQL HeatWave Lakehouse, currently available in beta, lets 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. With MySQL HeatWave Lakehouse, MySQL HeatWave provides one service for transaction processing, analytics across data warehouses and data lakes, and machine learning—without ETL across cloud services.

To learn more: read the MySQL HeatWave Lakehouse technical brief

Conclusion

MySQL HeatWave provides a single database for OLTP, OLAP, and machine learning applications, with compelling performance and cost advantages. Organizations using MySQL Database for managing their enterprise data can now run analytic queries with HeatWave with significantly better performance, lower cost, without ETL. HeatWave AutoML provides a native, in-database machine learning capability that allows users to build, train, predict, and explain machine learning models inside MySQL, without machine learning expertise. MySQL Autopilot provides machine learning automation that improves the performance, scalability, and ease of use of HeatWave. The service can be deployed in a cloud only or in a hybrid environment, and it simplifies management for transactional, analytic, and machine learning applications.

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

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.