

MySQL HeatWave: Teaching the Competition a Lesson in ML

The News: Oracle announced that Oracle MySQL HeatWave now supports in-database machine learning (ML) in addition to the previously available transaction processing and analytics, providing users a new MySQL cloud database service solution to do so. MySQL HeatWave ML is designed to fully automate the ML lifecycle and store all trained models inside the MySQL database, eliminating the need to move data or the model to a machine learning tool or service. Eliminating ETL reduces application complexity, lowers cost, and can improve security of both the data and the model. HeatWave ML is included with the MySQL HeatWave database cloud service in all 37 Oracle Cloud Infrastructure (OCI) regions. [Read the Oracle Press Release here.](#)

MySQL HeatWave: Teaching the Competition a Lesson in ML

Analyst Take: Oracle's MySQL HeatWave ML announcement represents the third major release of MySQL HeatWave in just over 15 months. The MySQL HeatWave launch in December 2020 united OLTP with a query accelerator, giving customers real-time analytics. The MySQL HeatWave launch in summer 2021 delivered MySQL Autopilot, 64 node scalability, introduced HeatWave scale out storage for reloading data and improved performance advances which the company says has resulted in customers migrating from AWS Aurora, RDS, Redshift, Teradata, SAP HANA, and Google BigQuery platforms. Throughout these migrations, Oracle prioritized MySQL customer feedback with a view toward further improving the overall MySQL experience. As a result, four major areas of prioritization were identified, with in-database machine learning as the top customer request.

While I will examine several of HeatWave ML's features, model explanation is critical for enterprise workloads. This is particularly the case when explanations are integrated with the training pipeline along with model-agnostic techniques, which can explain any HeatWave ML model. From my perspective, the ML explanation of any and all models in HeatWave ML can help organizations in fulfilling key organization-wide objectives such as:

- **Regulatory compliance:** May imply right to an explanation for algorithms-affecting users
- **Fairness:** By allowing validation that predictions are unbiased
- **Repeatability:** Ensures that small changes in input do not lead to large changes or disruptions in the explanation
- **Causality:** Allows verification that only causal correlation between features and predictions are selected
- **Trust:** Interpretable explanations encourage ML-based protection

I see these new ML explanations in HeatWave providing differentiation opportunities for Oracle across usability and interpretability, quality, along with scalable performance. For usability and interpretability, the model-agnostic techniques enable explanation of all HeatWave models that do not require a reference dataset when performing explanations. As such, intuitive explanations assist users in determining which factors contribute most to a prediction.

Quality is improved by leveraging dataset characteristics which provide repeatable and stable explanations and support novel improvements that more accurately explain the model's behavior. Improved repeatability means less variance between explanations and lower error nearby to the sample to explain. Performance and

scalability are augmented, since explanations scale linearly, and not exponentially, with the number of features. This results in real-time explanations that are possible due to distribution of work among workers and cores.

MySQL HeatWave ML: Competitive Advantages Against Rivals

As I explore qualitative and quantitative comparisons, note that I am basing my assessment on data presently made available to us. In this case, I am using all the ML Benchmarks and all the TPC-DS benchmarks, including all the scripts, configs, and details, that are posted on GitHub for full transparency, allowing anyone to replicate them for verification. The TPC-DS benchmarks were performed by an independent third-party benchmarking organization and the ML benchmarks were performed by MySQL HeatWave Engineering.

Exploring this data, in direct comparison to rival cloud DB services, I am impressed that HeatWave ML demonstrates clear competitive differentiation against Amazon Redshift ML and Snowflake ML across eight key features. Note that These features include AutoML, model and prediction explanations, user-facing API, user control of ML functionality, data/model locality, data available, supported number of nodes for training, and data sampling as indicated in the chart below:

Qualitative comparison with Cloud DB ML solutions

Feature	HeatWave ML	Redshift ML	Snowflake ML
AutoML - Integration - Completion Criteria	Native within HeatWave Convergence Based	Via SageMaker AutoPilot Time Budget Based	Not available. Enables the user to use third-party libraries
Model & Prediction Explanations	Both	Model Explanations for limited models only	Not available. Enables the user to use third-party libraries
User-facing API	Stored Procedures / Functions	SQL integration In-line SELECT statements	User writes code using Java, Scala, Python
User control of ML functionality	Simplified interface geared toward business users	Flexible interface with additional knobs for expert users	Not Applicable
Data/model locality	Stays in HeatWave	Exported to SageMaker	Data exported for training
Data Available	Both OLTP and OLAP	OLAP, Aurora ML for OLTP	OLAP
Supported #Nodes for training	Configurable	Not configurable by customers for training	Not Applicable
Data Sampling	Intelligent sampling	Random sampling	Enables the user to use third-party libraries

Source: Oracle

For example, HeatWave ML natively supports AutoML, including integration and completion criteria capabilities, and is convergence-based. In contrast, Redshift ML requires users to use Amazon SageMaker with the meter running as its use is based on time budgets. Snowflake ML does not support AutoML at all, requiring users to rely on third-party libraries or chargeable services. Both Redshift ML and Snowflake ML require users to move both data and models outside of the database, which increases security risks by exposing more surface area

attack possibilities and makes data stale by the time all the data movement and manipulation is completed. Consideration must also be given to costs incurred when using separate ML tools and the learning curves associated therewith, which can result in lost employee productivity when evaluating potential solutions.

For data sampling, HeatWave ML brings intelligent sampling, whereas Redshift ML uses random sampling and Snowflake again compels users to resort to third-party libraries. From my perspective, demonstrating such differentiators can broaden HeatWave’s addressable market, win more mindshare across the cloud DB ecosystem, and accelerate Oracle sales cycles.

MySQL HeatWave ML: Let the Competition Begin

I see HeatWave ML achieving solid differentiation across a wide range of top cloud DB services selection criteria that augment the competitive advantages across the eight key features. These selection criteria include self-tuning, training time, pause & resume, costs, and other enhancements.

In the area of tuning, for example, it’s beneficial that HeatWave ML is self-tuning, relieving users of the uncertainties of pre-selecting parameters. Conversely, Amazon Redshift ML requires the user to specify the desired run time and maximum number of cells, including the headache of using larger values which can trigger higher costs. Users might face the added frustration of not getting a better model by simply spending more.

Based on testing by Oracle, and available for anyone to replicate on [GitHub](#), HeatWave ML is both 25X faster and more accurate than Redshift ML at 1% of the cost, based on the default training time that the Amazon solution delivers as indicated in the chart below:

HeatWave ML is *faster* and *more accurate* than Redshift ML

Default training time of Redshift ML

Dataset	Accuracy		Training Time (minutes)		Speedup
	Redshift ML	HeatWave ML	Redshift ML	HeatWave ML (2 nodes)	
Airlines	0.5	0.6524	90.00	2.71	33.21
Bank	0.8378	0.7115	90.00	3.72	24.19
CNAE-9	FAILED	0.9167	FAILED	5.91	N/A
Connect-4	0.6752	0.6970	90.00	7.13	12.62
Fashion MNIST	FAILED	0.9073	FAILED	181.85	N/A
Nomao	0.9512	0.9602	90.00	3.30	27.27
Numerai	0.5	0.5184	90.00	0.34	264.71
Higgs	0.5	0.758	90.00	68.58	1.31
Census	0.7985	0.7946	90.00	1.22	73.77
Titanic	0.9571	0.7660	90.00	0.47	191.49
CC Fraud	0.9154	0.9256	90.00	29.06	3.10
KDD Cup	FAILED	0.5	FAILED	3.55	N/A
GEOMEAN	0.712	0.754	90.00	3.561	25.271

Source: Oracle

Based on the default training time required by Redshift, it appears as though HeatWave ML is also more affordable than Redshift ML, as shown below.

HeatWave ML is *cheaper* than Redshift ML

Default training time of Redshift ML

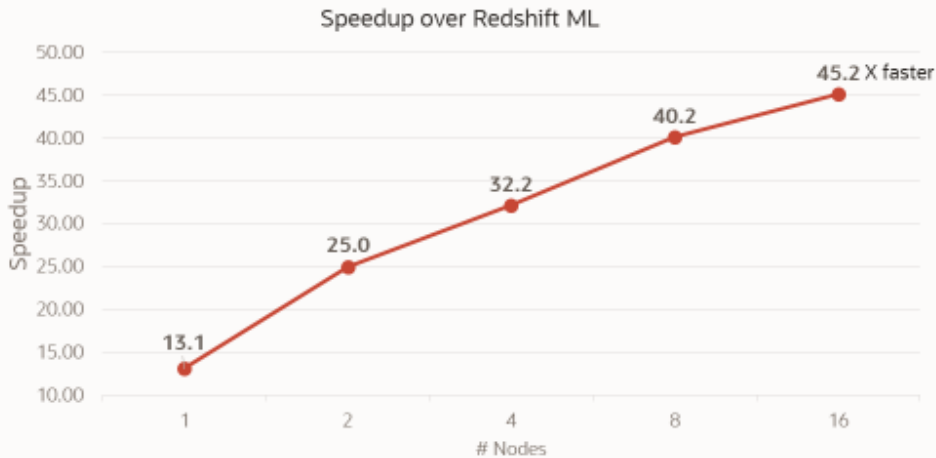
Dataset	Cost		Cheaper
	Redshift ML Cost with 1 year plan	HeatWave ML Cost \$ (2 nodes)	
Airlines	6.23	0.0479	130.03
Bank	5.68	0.0658	86.30
CNAE-9	FAILED	0.1045	N/A
Connect-4	6.18	0.1261	49.05
Fashion MNIST	X	3.2151	N/A
Nomao	5.96	0.0583	102.14
Numerai	5.49	0.0060	913.49
Higgs	7.27	1.2125	5.99
Census	6.12	0.0216	283.95
Titanic	5.60	0.0083	674.32
CC Fraud	6.70	0.5138	13.03
KDD Cup	FAILED	0.0628	N/A
GEOMEAN	6.115	0.063	97.129

Source: Oracle

The HeatWave ML differentiation cause is furthered by providing automated tuning and training of models, especially in the scaling of challenging AutoML applications. Each stage of a given pipeline has unique considerations for parallelism, which entails tradeoffs between runtime, accuracy, and scalability. For instance, iteration-free pipelines with fewer trials improve runtime but trades off parallelism, while reduced synchronization points in AutoML pipeline trades off accuracy for improved scalability. Likewise, convergence of the pipeline creates fewer trials at the end of the query, trading off scalability for faster runtime. In sum, I credit Heatwave ML's better performance across larger cluster sizes as validating its value in handling AutoML scalability as shown in the following chart provided by Oracle:

Better performance with HeatWave ML with larger cluster size

Default settings for Redshift ML



Source: Oracle

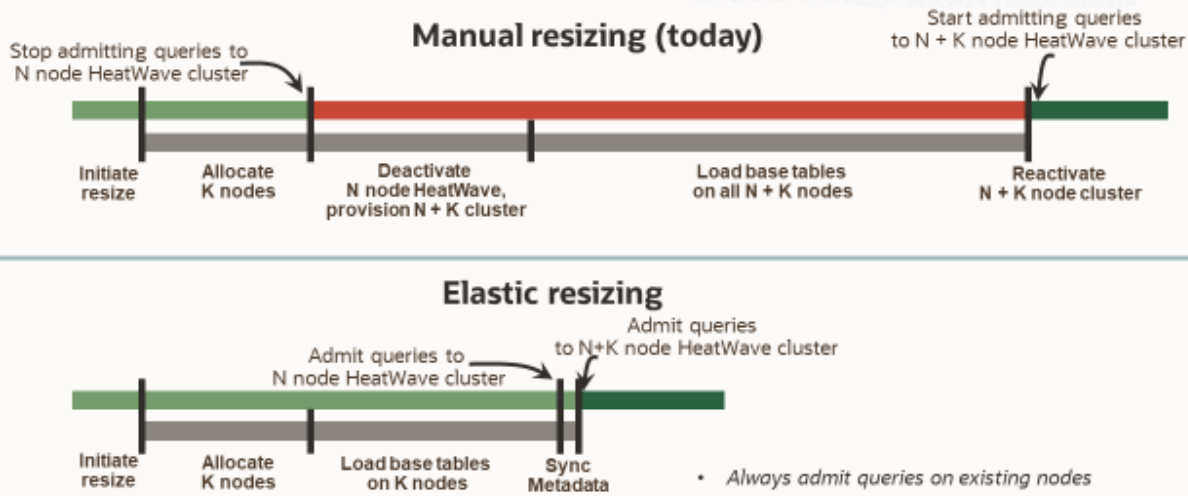
These cost and performance differentiators make HeatWave ML a very attractive offering for customers of all sizes.

MySQL HeatWave ML: Real-Time Elasticity Further Fuels the Differentiation Engine

From my point of view, supporting real-time elasticity is a key feature that cloud DB users are increasingly demanding when aligning their key database analytics scalability requirements. I consider the most important real-time elasticity features as encompassing:

- **Availability:** All operations (queries, DMLs, load) are permitted on-cluster during resizing
- **Flexibility:** Upsize or downsize to any number
- **Balanced:** Data across nodes is balanced after resize
- **Performant:** Minimal data movement during resize. Data loaded at object store bandwidth
- **Predictable:** Resize time is constant and predictable

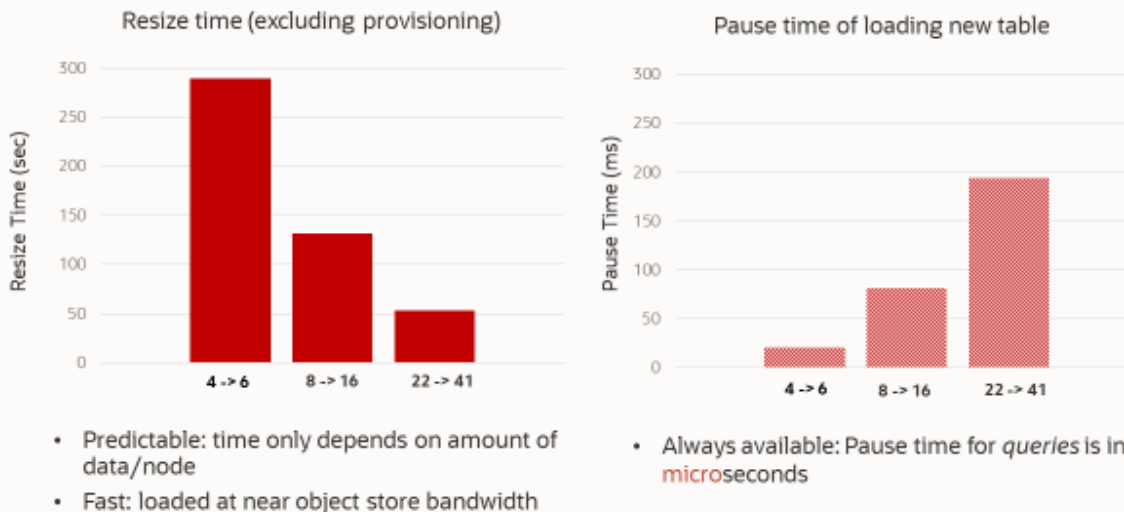
Elasticity versus manual resizing



Source: Oracle

Elasticity performance

2TB TPCH; fast, predictable and cluster is always available



Source: Oracle



Through delivering these features across the board, I regard MySQL HeatWave as providing the fully available up/down scaling to any size cluster, which helps Oracle stand out from the competition. Snowflake, for example, provides instant elasticity but with delayed loading of data on first query. Of equal concern, Snowflake only allows scale up or down in exponentially large cluster sizes such as threefold or fourfold cluster sizes (i.e., 64 to 128 CPUs). This limits the ability of simpler queries to fully utilize the large cluster as well as shoehorning customers into only up/down scaling the DB through a pre-determined, “shoe sizing” approach which doubles the cost. With Snowflake, if I need a pair of size 7 shoes but Foot Locker only has size 12, then I’m spending two to three times the money and my shoes resemble those of former employees at the Barnum & Bailey Circus.

Redshift ML resizing can take 10 to 15 minutes during which writes are not allowed, which can result in constraining user flexibility. In addition, during resizing, queries can be held back or timed out and if the user is not prepared with a snapshot, resizing can take even longer to implement. Adding to the potential for additional user frustration, any resulting data-skew must be identified manually and resolved with a classic resize, which can take hours or days. These are, of course, difficulties that any cloud DB user would prefer to avoid altogether.

AWS Redshift RA3 provides compute instances of 4, 12, and 48 vCPUs. When an organization needs 17 CPUs, they must buy 48—there is no other choice. This is yesteryear’s antiquated approach to cloud databases, and I believe not how customers want to be treated going forward. I can’t imagine AWS not modifying this at some point in the future, as it only makes sense.

Also integral to the differentiation of MySQL HeatWave is the doubling of the data that can be processed per node, reducing costs by about 50%:

Data processed per node is doubled

Reduced memory for intermediate results (bloom filter) is very important



Base line (in August)

- 4TB TPCH, **10 Nodes**
- GEOMEAN: 6.35 sec

Bloom filter reduces intermediate results

- 4TB TPCH 10 Nodes
- IR memory reduction: 82GB → 39GB
- GEOMEAN: 5.05 sec

Bloom filter + Compression

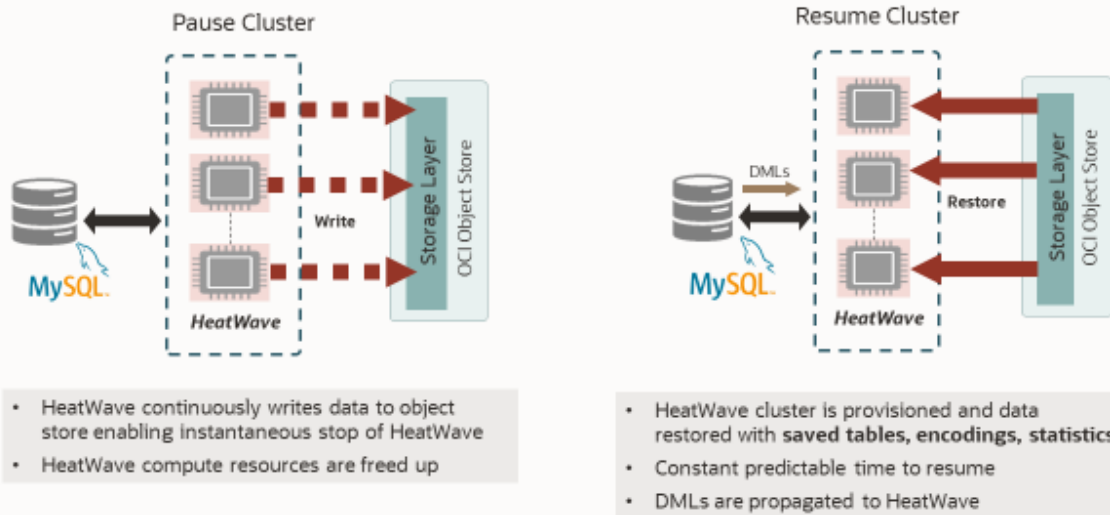
- LZ4HC algorithm
- 4TB TPCH **5 Nodes**
- GEOMEAN: 11.59 sec

Source: Oracle

MySQL HeatWave ensures that users experience no impact on load time when compression is in use, including reading into HeatWave memory and writing to object store processes. This is achieved by offloading compression to the HeatWave cluster. For example, when 4MB chunks are compressed and multiple compressed chunks are coalesced to form 4 MB chunks which are persisted on object store, the 4MB chunks result in maximized I/O performance.

Pause & Resume HeatWave

Instantaneous stop & constant time resume



Source: Oracle

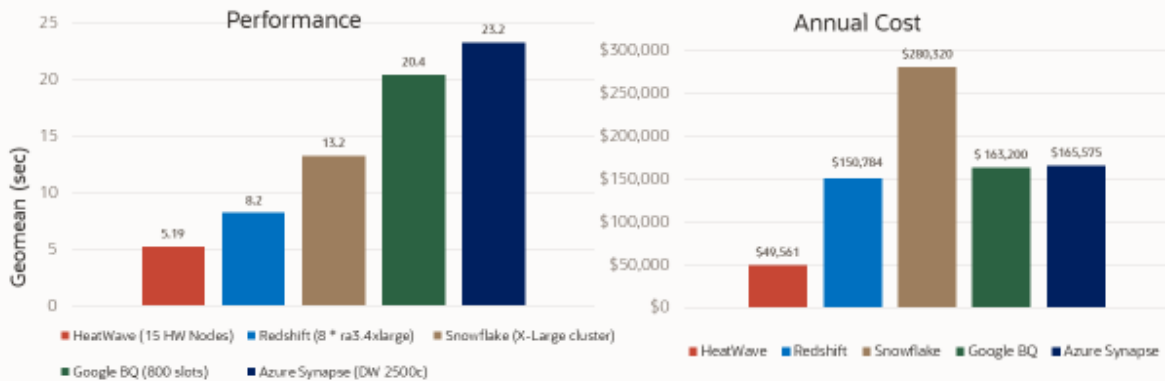
In addition, pause and resume functions now include instantaneous stop and constant time resume. To achieve this, HeatWave continuously writes data to object store, which enables instantaneous stop on HeatWave, also freeing up HeatWave compute resources. The HeatWave cluster also provisions and restores data through saved tables, encodings, and statistics that provide predictable time to resume capabilities. Customers can now pause HeatWave during the night or weekends when developers are not using the database, further reducing costs.

MySQL HeatWave: Let the Price/Performance Competition Begin

As a capstone, I believe HeatWave ML offers some competitive differentiation benefits including price/performance advantages over Amazon Redshift, Snowflake, Google Big Query, and Azure Synapse that make the Oracle offering consideration worthy. Based on the data made available to me, HeatWave provides better price-performance of 4.8x in relation to Redshift, 14.4x better than Snowflake, 12.9x better than BigQuery, and 14.9x better than Synapse:

Price-Performance comparison with Redshift, Snowflake, Google Big Query

10TB TPCDS: 4.8x better than Redshift, 14.4x better than Snowflake, 12.9x better than BigQuery, 14.9x better than Synapse



- Pricing for Redshift is based on 1 year reserved instance, paid upfront. For Snowflake is based on **standard edition**
- Pricing for Google Big Query is based on monthly flat rate commitment. For Azure Synapse is based on 1 year reserved pricing

Source: Oracle

As such, I see the very real possibility of HeatWave ML’s value proposition gaining broader consideration and making further inroads across the cloud DB services ecosystem due to HeatWave’s demonstrable price-performance competitive edge.

Key Takeaways on Oracle MySQL HeatWave ML Debut

The HeatWave ML news represents a three-pronged attack on the competition — with transaction processing, analytics, and now machine learning all engineered inside a single MySQL DB service. Why would you use two different databases, two to three different ETL tools and risk the movement of both data and models when you can do it with one database? The AWS and Snowflake approach to ML is analogous to going to the local auto parts store to buy a steering wheel, brakes, tires, and a transmission and then trying to put a car together yourself when you could just order one from Tesla online and be done with it. And therein lies the advantage for Oracle MySQL HeatWave ML.

From my perspective, the question for developers is: “Are you looking to be more productive, or spend time haggling with ETL tools and shuffling data back and forth?” The question for those in organizations that manage cloud IT expenditures consists of: “Are you in the business of making up for deficiencies in public cloud DB services or focusing on your core business competencies? And consider what will you do when approached by senior management when they ask—and eventually, they will ask—why are you spending 5-10x the cost on multiple services instead of one DB?” Based on what I’ve seen thus far, MySQL HeatWave represents the fiscally responsible approach to cloud databases while AWS Redshift and Snowflake represent the fiscally reckless approach.



I believe the new MySQL HeatWave ML offering demonstrates that Oracle has delivered more cloud database innovations over the last 15 months than most cloud database vendors have delivered in the last decade. From my point of view, the in-database HeatWave ML puts Redshift ML and Snowflake on notice. Are these solutions more like yesterday's tech in terms of engineering, performance, and cost? If they're slower and more expensive, chances are the answer is yes. It's clear that Oracle is innovating in cloud databases and gaining distinct competitive advantages over rivals. Will the competition respond or stagnate? Amazon has 15 overlapping specialized cloud DB services, and it's no doubt lucrative when customers use multiple services, for example Redshift ML and SageMaker for machine learning, forcing chargeable ETLs and data movement. But it's safe to say that Amazon is not known for sitting still for long, and I would guess that cloud database innovation is definitely on the table for the AWS team.

Snowflake continues its single-minded pursuit of analytics and likewise forcing customers to ETL, but to a chargeable third-party service.

Oracle's MySQL HeatWave is now positioned to drive increased innovation in the cloud DB market and force the competition to examine their offerings both in terms of technology and in terms of pricing — all of which will benefit customers in the end.