db insight

RESEARCH NOTE:
Oracle MySQL HeatWave embraces the Data Lakehouse

Autopilot turbocharges the data lake

Published November 2022

Executive Summary

Trigger

Among the flurry of announcements coming out of Oracle CloudWorld 2022 was the beta for the latest addition to Oracle's MySQL HeatWave cloud-based in-database transaction, analytics, and machine learning service. The new feature takes HeatWave into "Data Lakehouse" territory by extending analytics to cloud object storage. While Oracle is hardly the only cloud data platform to embrace the lakehouse, its implementation is notable in making data sitting in cloud object storage a first class citizen, not only where it comes to governance and security, but also in performance. Thanks to optimizations provided by HeatWave's MySQL Autopilot feature, query performance of data in object storage is virtually level with data sitting in HeatWave's in-memory column stores. What is the significance of Oracle's opening of the lakehouse, and where would we like to see Oracle go from here?

Our Take

As we've written previously, Oracle MySQL HeatWave is not a vanilla implementation of the popular open source platform that, through a twist of history, Oracle owns. It entered the stage as the only cloud MySQL service designed to support analytics, in addition to transaction processing. Since then, Oracle has doubled down with machine learning capabilities, both to run the database and for customers to run inference. Given growing vendor support for data lakehouse architectures, before this we were wondering, not whether, but when Oracle would enter the fray. The highlight of Oracle's MySQL HeatWave Lakehouse approach is an extension of the Autopilot optimizations to data stored in cloud storage, with Oracle's benchmarks, not only showing performance multiples in queries and data loading over Amazon Redshift and Snowflake in a 400TByte TPC-H workload using 512 nodes, but *equivalent* performance with data stored in HeatWave's in-memory column store.

This clearly differentiates HeatWave Lakehouse from the usual suspects, but we would add a couple related disclaimers. First, virtually all lakehouse implementations are still at the v1 stage, and therefore, performance levels will be moving targets. Secondly, while data lakehouses are gaining rapid uptake in vendor support, this is a case where they are currently ahead of their customers when it comes to demand and awareness. That's related to our first disclaimer. Nonetheless, from a price/performance standpoint, HeatWave currently stands out among its peers. On the horizon, we would like to see

the platform embrace open source standards on the data lake side, as we believe that could widen MySQL HeatWave Lakehouse's appeal.

What is a Data Lakehouse?

Functionally, a data lakehouse is literally supposed to (choose one) deliver the best of both worlds and/or be all things to all analytics users, from data scientists and engineers to business analysts.

This is not a new idea. Hadoop providers like Cloudera introduced Impala to bring more data warehouse-like performance to big data clusters with data sitting in HDFS, and later, cloud object storage. AWS introduced the Athena ad hoc query service to make data sitting in cloud object stores accessible to SQL queries, while virtually every major cloud data warehouse (and in some cases, on-premises) data warehousing platforms introduced federated queries to expose data sitting in object stores as virtual, external tables. While these capabilities started to bring the worlds of data warehousing and big data together, they did not provide satisfactory experiences for multiple reasons. They were complex to operate and use: lacked a table structure that could deliver better performance because they required bulk file scans; lacked the type of row or column-based security and access controls that are table stakes with data warehouses; and lacked ACID transaction capabilities for ensuring that data being analyzed was consistent, current, and not corrupted during file refreshes or updates.

Databricks introduced the term "Data Lakehouse" back in 2017, based on its emerging technology Delta Lake. Simply stated (and there's a lot more rocket science to all this), it superimposed a logical table structure atop data sitting in cloud object stores. Lakehouse tables are designed for structured data that can be represented as rows and columns in relational tables (this is not about unstructured data).

Databricks partially open sourced the Delta Lake project, keeping premium features close to its vest until this year when it open sourced the remainder of the project. Apache Iceberg is the main competing project. It was jointly created by Apple and Netflix, and open sourced as an Apache community project that has drawn backing from the likes of Snowflake and Cloudera. There is also Apache Hudi, which is supported by OneHouse, with a community that is far smaller than that of Iceberg. In each case, the commercial ecosystems for competing lakehouse table formats are still at early states of formation. Adoption of these lakehouse table formats are currently confined to classic early adopters. For instance, in the Iceberg world, the biggest adopters (in order) are

Internet, computer software, financial services, and information technology and services companies – who comprise roughly 60% of adopters, according to Enlyft.

What's different with Oracle's implementation?

As we've noted before, when Oracle got serious about coming to market with a MySQL cloud service, it was anything but me-too. From the get-go, HeatWave added analytics to MySQL, something that no other cloud MySQL service has done. Of course, given how crowded the existing MySQL database-as-a-service market is (with offerings, not only from hyperscalers but also third parties ranging from IBM to independents like Instclustr, Percona, and Aiven), Oracle had to do something very different given its belated entry. And it did so with a flourish of innovations. Beyond analytics, which none of the other services were designed to support, HeatWave introduced optimized storage followed by machine learning-based automation (MySQL Autopilot); support for in-database machine learning training; along with explanations and inference (HeatWave ML). Additionally, MySQL HeatWave has gone multicloud with the service available natively on AWS and via the OCI-Azure high-speed interconnect for Azure customers. We have covered all pertinent releases with these features previously.

Oracle has taken a similar approach with MySQL HeatWave Lakehouse. While the Data Lakehouse as a market is still in its infancy, others have stepped up to the plate ahead of Oracle, from born-in-the-cloud services like Databricks and Snowflake, but also the likes of IBM, Teradata, and Cloudera. But Oracle has done so with differentiated performance by extending Autopilot to address different optimizations for data in Parquet or CSV format sitting in cloud object storage as well as Amazon Aurora and Redshift exports. Specifically, Oracle has public benchmarks backing its claim that queries to data sitting in object storage will perform on par with data in HeatWave's original in-memory column store tables.

We'll provide a few examples of how Oracle pulls this off. They include *adaptive sampling*, where Autopilot literally samples a small subset of data for characteristics such as min/max, degree of cardinality, data types (e.g., string, integer, or floating point). From that, Autopilot *infers schema*, predicts the *optimal shape* (e.g., layout of storage), and then automatically determines compute and storage based on sampled stats and data profiling. It then loads, and processes queries in parallel, predicting change propagation and query time.

Taking what is by now its standard approach to benchmarking, Oracle has published the benchmarks and put the test suite on GitHub for customers to test drive on their own. For 10- and 30-TByte workloads, query times for the HeatWave "when the source data is in MySQL database and object stores, were identical. By comparison, in our conversations with Snowflake, their beta implementation of Iceberg delivers almost 90% of the performance of their proprietary tables, but keep in mind that their results are not directly comparable to Oracle's because both databases carry different designs and flavors of SQL.

Competitive benchmarks for Lakehouse vs. the usual suspects are just starting to trickle in; to date, Oracle claims 17x faster query performance and 30% lower cost than Snowflake as well as 2.7x faster load time; and 6x and 8x faster than Amazon Redshift for the corresponding tests. Both tests were run against a 400TByte TPC-H derived workload, which is new territory for MySQL. However, as noted above, all these results (Oracle and Snowflake) are of beta technology. While we always take a verify then trust attitude towards vendor benchmarks, Oracle has been highly transparent. Notably, none of Oracle's rivals have responded to the benchmarks that Oracle has published for HeatWave to date, and we don't expect such practice will change anytime soon. Expect this field to be a moving target.

Takeaways

Oracle's pre-production version of MySQL HeatWave Lakehouse will be competitive in price and performance vs. the usual suspects. It has cleverly applied optimizations that have turbocharged MySQL performance and has demonstrated massive scale-out capabilities for both query processing and loading data. Adding lakehouse capability is the logical next step forward after previously rolling out the analytics engine, AutoML for in-database machine learning, as well as machine learning-based automation in Autopilot. It stands to reason that more of the *right data*, with the *right quality*, and the *right currency* will make HeatWave AutoML even more useful.

One of the interesting phenomena that has made data lakehouses potentially compelling is that there is critical mass activity on the open source side with Delta Lake (led by Databricks), and community-based Apache Iceberg and Hudi projects. They have already started getting vendor commitments for support. For instance, next year Google Cloud's BigQuery data lakehouse will eventually include support for all three open

source projects. MySQL HeatWave Lakehouse currently runs on an Oracle-designed proprietary format; Teradata has adopted the same strategy.

Given that price/performance and a scale-out architecture is key to HeatWave's value proposition, we would like to see Oracle test these open source formats to determine whether they are industrial strength. The upsides for Oracle are twofold: (1) It reinforces Oracle's message of openness and promises that it will not lock-in customers; and (2) it might be able to make HeatWave accessible to non-Oracle customers (we're still assessing openness and portability – watch this space). Oracle should settle the issue once and for all whether their own data lakehouse table format is part of that difference.

From the get-go, Oracle has engineered MySQL HeatWave to deliver higher performance than vanilla MySQL. Oracle hasn't disappointed on performance with MySQL HeatWave Lakehouse, where it has turbocharged the data lake.

Author

Tony Baer, Principal, dbInsight

tony@dbinsight.io

LinkedIn https://www.linkedin.com/in/onstrategies/

About dbInsight

dbInsight LLC® provides an independent view on the database and analytics technology ecosystem. dbInsight publishes independent research, and from our research, distills insights to help data and analytics technology providers understand their competitive positioning and sharpen their message.

Tony Baer, the founder and principal of dbInsight, is a recognized industry expert on data-driven transformation. *Onalytica* named him as a Top Cloud Influencer for 2022 for the fourth straight year. *Analytics Insight* named him one of the 2019 Top 100 Artificial Intelligence and Big Data Influencers. His combined expertise in both legacy database technologies and emerging cloud and analytics technologies shapes how technology providers go to market in an industry undergoing significant transformation.

dbInsight® is a registered trademark of dbInsight LLC.

Published November 2022