RESEARCH NOTE:
Oracle takes MySQL HeatWave to AWS

Where will Oracle's multicloud strategy go next?

Published September 2022

Executive Summary

Trigger

During its Q3 FY2022 investor call earlier this year, Oracle chairman and chief technology officer Larry Ellison promised that it would take MySQL HeatWave multicloud. Oracle just made good on the first step of that promise, releasing MySQL HeatWave on AWS. The latest update to Oracle's service, which extends MySQL to new ground with analytics, in-database machine learning (ML), and Autopilot now runs on AWS as a fully managed service in addition to OCI. MySQL HeatWave on AWS is the second announcement in almost as many months where Oracle is bringing some of its database services to non-Oracle clouds. To the customer, MySQL HeatWave on AWS will provide the same capabilities as the original OCI-based service. But beneath the hood, it is not a basic lift and shift of the OCI service; instead, it was engineered specifically for the AWS cloud infrastructure. Following on the recent announcement of Oracle Database Service for Microsoft Azure, what can customers expect from Oracle's new multicloud strategy?

Our Take

Since the unveiling of MySQL HeatWave two years ago, Oracle has come out swinging. It built features taking MySQL out of its pure OLTP roots, along with aggressive price/performance optimizations. With MySQL already a mature, and crowded market in the cloud, Oracle had to take MySQL to a different place. While a key part of differentiating MySQL HeatWave was in the Autopilot automation, taking it to AWS required reengineering those features and adding new ones for new cloud infrastructure outside of Oracle's control.

As an initial release, Oracle has aimed for middle ground, targeting AWS compute instances based on commodity Intel infrastructure akin to its AMD architecture currently running on OCI. That is a sensible choice for its first dip into the AWS pond. But, going forward, we hope that Oracle will look to Graviton, built on the ARM architecture, which could potentially deliver even better price/performance for some workloads. As to the future of Oracle's multicloud support, we expect that a subset of existing OCI SaaS services such as analytics and BI, machine learning, and Oracle Database Service (but *not* Autonomous Database) would make good candidates. Not surprisingly, we also expect to see MySQL HeatWave also end up on Azure soon, as it was also mentioned on the same earnings call.

Going where the customers are

A key pillar of Oracle's public cloud strategy has been optimizing its SaaS services specifically for OCI. In fact, that is a logical sequel to Oracle's engineered systems strategy that yielded Exadata. The going notion was optimizing software for the hardware and vice versa.

But as we noted when Oracle introduced its database service for Azure via its special high-speed cloud interconnect, multicloud is the reality for most organizations. Most already have a variety of platforms, applications, and databases across their on-premises technology portfolios, and when it comes to the public cloud, those habits didn't change. Multicloud may result from:

- Merger and acquisition, when newly acquired entities have distinct environments;
- Deliberate policy, for avoiding dependence on any single vendor or providing fault tolerance; or
- Inertia, where platform choice has been decentralized where local business units make their own choices.

Our take is that, increasingly, enterprises are seeking to keep their cloud options open, and Oracle has heard their message.

In part, that drove Oracle's decision to take its MySQL HeatWave service to AWS. As the largest cloud provider by market share, it shouldn't be a shock that many Oracle customers use – or also use – AWS for some of their applications and databases. The flip side of the coin is Oracle wanted to lower barriers to entry for MySQL customers on AWS seeking to take advantage of MySQL HeatWave. Those barriers included data egress costs; latency of accessing another cloud from AWS; and/or lack of tight integration with other AWS services that they might or plan to be using. Opening up a beachhead with MySQL HeatWave inside AWS itself addresses those issues.

How MySQL HeatWave translated to AWS

As noted above, one of Oracle key differentiators in public cloud has been about optimization of software for the OCI environment. That strategy carried through to the initial release of MySQL HeatWave. Oracle had little choice given that MySQL in the cloud has become fairly commoditized; as a me-too, what additional value could Oracle deliver?

Oracle responded with a flourish: it extended MySQL to uncharted territory with an analytics in-memory hybrid columnar store and aggressively priced the service, backing its claims with several benchmarks showing an order of magnitude advantages over the usual suspects: Amazon Redshift, Aurora, Snowflake, and Google BigQuery. Highlights of subsequent releases included embedding ML, both for automating the operation of the database (Autopilot), and for running training and inference in-database.

Because optimization is so critical to MySQL HeatWave, taking the service to AWS (or any other public cloud) is not a trivial undertaking. Most cloud Database-as-a-Service (DBaaS) providers approach multi-cloud by packaging their offering as containerized, Kubernetes implementations that can be implemented fairly straightforwardly to other clouds. But as noted, with the extensive optimizations, such an approach wouldn't cut it for MySQL HeatWave.

For instance, MySQL Autopilot, which optimizes tasks such as provisioning, parallel data loading, data placement, query scheduling, schema change propagation, and error recovery, was originally designed especially for OCI compute; networking (which tends to be flatter than most public clouds); and storage. Furthermore, Autopilot was trained on workloads running on OCI. All of those optimizations had to be adapted for AWS instances, requiring deep knowledge of foreign infrastructure. And to run like a native AWS service, Oracle had to run and optimize the data plane, control plane, and console on AWS.

For AWS, Oracle chose to implement MySQL HeatWave on Intel instances that were similar to the AMD running on OCI. Let's break down the rationale in more detail:

- Why Intel? As a first release in a new cloud, it makes sense for Oracle to embrace familiar architecture. With Intel AWS instances, it is starting with the known commodity x86 architecture used by the AMD instances in OCI.
- Why not Graviton? That would require significant code rewrites as they are based on the ARM architecture.
- Why Not GPUs? They would incur significant additional cost, which goes against the price/performance targets that Oracle has set for this service. Furthermore, our take is that MySQL HeatWave customers are not likely to be running the types of highly complex, Convolutional Neural Network (CNN) deep learning models that would require them anyway.

A Moving Target

As a cloud service, the cadence of new features is a continual flow. Over the past few months, MySQL HeatWave has introduced enhancements and new features in areas ranging from business continuity to read scale-out, cluster sizing requirements, query performance, security, and other areas. These apply to both OCI, and now, the AWS version; and in one case, the new AWS version actually leapfrogged the original OCI edition. Hold that thought.

Among the highlights is adding a new transaction “admission control” capability for Autopilot to act as a traffic cop for transactions, reducing the likelihood of resource contention issues; in effect, lighter weight transactions are given a go ahead of complex ones to avoid tying up resources. This is crucial for high concurrency scenarios where resource contention is more likely.

Another major upgrade is a new real-time “Autoshape” feature that monitors database operation and predicts what is the best “shape” (or instance type) (e.g., configuration of compute and storage) to get the best price/performance for current workloads. Autoshape is a natural complement to Autopilot (which optimizes deployment and operations, encompassing provisioning, parallel loading, data placement, and so on. It factors in the instances that are unique for the specific target cloud and developing algorithms for generating suggestions to customers for staying the same, upgrading, or downgrading their own MySQL HeatWave infrastructure footprint.

Here’s the interesting part. Autoshape is debuting first for MySQL HeatWave on AWS *before* it comes to OCI (it will, soon). It shows, not only that such a feature cannot be carbon copied for each cloud implementation of MySQL HeatWave, but also that Oracle is not playing favorites. MySQL HeatWave on all clouds (OCI and AWS today, and Azure in the future) will get equal treatment when it comes to user features.

Takeaways

Oracle has taken the next logical step in its multicloud journey. Following on its announcement a few months back of Oracle Database Service for Azure, it is now bringing MySQL HeatWave to AWS. It provides a solution for AWS customers seeking to take advantage of MySQL HeatWave, minus the overhead of egress costs, cross-cloud latencies, and integration issues.

The common thread for Oracle’s multi-cloud announcements this year is that Oracle customers, like most enterprises, use multiple clouds, and these are Oracle’s first steps towards meeting its customers where they are. But this is not a cookie cutter strategy, and it may not necessarily apply to all of Oracle’s portfolio.

For instance, while the database service for Azure runs like an Azure service under an Azure-like console, it physically runs the Oracle Database on OCI but takes advantage of the fast Azure interconnect that both providers began opening several years ago. By contrast, MySQL

HeatWave has been engineered to run natively on AWS. That is significant because, with all the built-in MySQL HeatWave optimizations for OCI, Oracle had to reengineer them specifically for AWS; this was not simply a lift-and-shift Kubernetes implementation on a new platform.

Another interesting fact of the AWS support is that Oracle took a middle ground in adapting MySQL HeatWave. Specifically, it chose the AWS commodity Intel instances as the best price performance match rather than getting fancy with ports to ARM/Graviton for burstable workloads, or GPUs for complex AI models. To get MySQL HeatWave to AWS as soon as possible, it made sense for Oracle to stay on familiar ground. But as to Graviton, never say never. Graviton could present TCO advantages for select workloads owing to their low power consumption, which is certainly what Oracle is aiming for. It will take some time (as Graviton is a radically different chip architecture), but we'd like to see Oracle go for it.

An analysis of Oracle's release would not be complete without referencing Oracle's benchmarks, where Oracle claims at least an order of magnitude faster and better price/performance than the usual suspects (Redshift, Snowflake, and BigQuery) for analytics. Oracle also claims that on AWS, MySQL HeatWave's Auto Thread Pool enables 10x better sustained throughput than Aurora for OLTP workloads as concurrency demands grow. Like other recent Oracle benchmarks, Oracle has published them on GitHub for customers to run their own test suites. While we take benchmarks with grains of salt (verify before you trust), they demonstrate that Oracle is dead serious in making MySQL HeatWave performant. To date, no competitor has challenged Oracle's HeatWave benchmarks publicly.

As to the future of Oracle's multicloud support, our answer is, "it depends." It has already said that it will take MySQL HeatWave to Azure, so that's a matter of time. Beyond that, we believe that OCI analytics and BI, machine learning, and Oracle Database Service would make good candidates for release on other clouds. On the other hand, we don't see Autonomous Database, which requires total control of infrastructure, as well suited. Neither do we see enterprise apps such as the Fusion portfolio going multicloud. The footprint of the Fusion service is large, and when it comes to enterprise apps, we don't see customer demand forcing the issue. While Oracle continues to tout the differentiation of its own cloud infrastructure, excluding Autonomous Database and Fusion Applications, we expect to see more clouds in Oracle's future.

Author

Tony Baer, Principal, dbInsight

tony@dbinsight.io

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

Twitter @TonyBaer

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.