Oracle MySQL Database Service with HeatWave & Autopilot

Sets the Bar Orders of Magnitude Higher Than AWS, Azure, Google, & Snowflake

by, Marc Staimer

October 2021

Research Premises

Less than a year after Oracle completely shook up the MySQL cloud database service market with its introduction of the MySQL HeatWave integrated in-memory analytics engine, they've done it again. With the introduction of Autopilot to the MySQL Database Service with HeatWave, Oracle has completely redefined what a MySQL database cloud service can be. And make no mistake, it's impressive.

This research is based on the following premises:

- MySQL databases are the most popular databases in the world garnering more than 20% market share according to Enlyft[1].
- The most precious non-renewable resource is time. Time only flows in a forward direction. There is no reality rewind button. Once time is spent it can never be recovered. Time invariably translates into money. Time-to-market. Time-to-revenue. Time-to-profit. Time-to-actionable insights that lead to greater revenues and profits.
- Faster time-to-market leads to earlier revenues that would never have been captured before. First-to-market leads to first mover advantage capturing more revenues and market share than competitors. Both lead to faster profits. Faster time-to-actionable insights means adapting and reacting faster to changing conditions than competitors. That in turn leads to more revenues and profits. In contrast, having a slower time-to-market/actionable insights lead to lower productivity, revenues, and profits. Time really does translate into money.
- Saving time means reducing the amount of time required to complete a task. The two types of saving time are for people and applications.

[1]Source: Enlyft statistics as of June 2021

Saving time for people translates into higher productivity. The IBM red book research report on "The Economic Value of Rapid Response Time" showed a very close relationship between application response time, productivity, and financial results called the Doherty Threshold. The Doherty Threshold is when a computer, its applications, and users interact at a pace where neither has to wait on the other, productivity soars and the quality of their work improves noticeably. For example: when application response times decreased from 3s to 300ms (.3s), productivity more than doubled. Additionally, application response times less than or equal to 400ms were addictive. Response times greater than 3s resulted in plunging productivity. Saving time for applications provides faster results and response times.

Saving application time equates into faster response times. This is hugely impactful for analytical applications that derive insight from large amounts of data. Faster insights translate into faster-time-to-action that in turn leads to increased revenues and/or decreased costs.

- Human manual labor consumes much more time than application or machine automation, no matter how skilled the laborer. Especially, repetitive tasks for which humans are not well suited. For MySQL tasks the DBA (database administrator) has historically needed to be highly skilled, knowledgeable, and experienced. MySQL generally lacks much in the way of automation. In addition, most data warehouse cloud services – ***excluding Oracle Autonomous Data Warehouse and Oracle MySQL Database Service with HeatWave and Autopilo****t* – lack much in the way of automation.
- Automation can eliminate the majority of the tedious, pedantic, and repetitive MySQL tasks that DBAs have to perform saving a lot of time. Database and data warehouse performance tuning alone is highly manual and labor-intensive. It can take all of a DBA's time to manage this single incredibly important process. Even with the advent of high-performance computing, networking, and storage, database and data warehouse performance tuning are more art than science. Automation can free the DBA's time from laborious non-strategic work.
- MySQL database cloud services consistently require that the MySQL data be extracted, transformed, and loaded (ETL) from the MySQL database to a data warehouse or equivalent cloud service, for analytics. ETLs take time, skills, experience, knowledge, and ongoing effort in documentation, maintenance, updating, patching, and tuning. They consume costly IT resources. It is common for ETL processes to be continuously tweaked, modified, and tuned to maintain performance as data sources are occasionally changed. DBAs generally do not like ETLs or ETL processes. They see them as a MySQL necessity. It takes a lot of manual labor (time) and frequently cause data integrity, security, and availability issues despite best efforts in error detection and correction. ETLs consume more time when troubleshooting the inevitable problems made worse when utilizing multiple cloud services.

  One more issue from the utilization of a separate data warehouse or analytics service is the duplication of data and its storage, creating data siloes. Cloud storage is not cheap, even cloud object storage. And collating data across siloes is another time-consuming effort.
- As data ages it becomes less valuable, and any insights derived from aged data are stale. Time and value go hand-in-hand.
- Transactional database, data warehouse, analytical database, and ETL cloud services are not free. As the old cliché goes: "There is no such thing as a free lunch."
- Oracle MySQL Database Service with HeatWave and Autopilot in Oracle Cloud Infrastructure (OCI) is the first to solve all of these MySQL database cloud service issues. The addition of Autopilot to the MySQL Database Service with HeatWave systematically automates performance tuning, system setup, data load, query execution, and failure handling. It does this through the extensive utilization of automated machine learning (AutoML).
- Oracle MySQL Database Service with HeatWave and Autopilot takes performance and scalability to much greater heights than the previous iteration of Oracle MySQL Database Service with HeatWave and orders of magnitude better than competitive offerings. And it still runs both transaction processing and analytics concurrently without ETLs.

- Autopilot supercharges MySQL Database Service with HeatWave to unprecedented performance, scalability, automation, and security. As a result, it recoups extraordinary amounts of time leading to much greater productivity, much faster time-actionable-insights, and much faster time-to-market.

Executive Summary

This research will help CIOs determine which MySQL database cloud service best meets their performance, scalability, time savings, and budgetary requirements. There are several MySQL database cloud services in the market today including Amazon Aurora-MySQL, Amazon RDS for MySQL, Azure MySQL, and GCP Cloud SQL for MySQL. There is little that distinguishes them from one another. All of them carry forward the many MySQL shortcomings into the cloud. As far as performance enhancements go, the Amazon Aurora MySQL database cloud service proves to be nominally faster than Amazon MySQL database cloud service. But that's about it. None of these MySQL database cloud service have a built-in analytics engine. There is minimal database automation based on policy and even less based on machine learning. In addition, these implementations have forked from MySQL at various points and, therefore, require application changes to move from one to another.

Oracle is delivering the first and currently only MySQL database cloud service that has a built-in in-memory analytics engine, machine learning (ML) based automation, end-to-end security, best-in-class performance based on TPC-H[2] benchmarks, and unmatched scalability. None of the other MySQL database cloud services can say the same. To provide equivalent analytics necessitates a separate analytics or data warehouse service. That service must be manually integrated with an ETL tool or service. These net additional services are appreciably slower than Oracle MySQL Database Service with HeatWave and Autopilot because they are not tightly integrated. ETL and integration add significant costs and delays in insights.

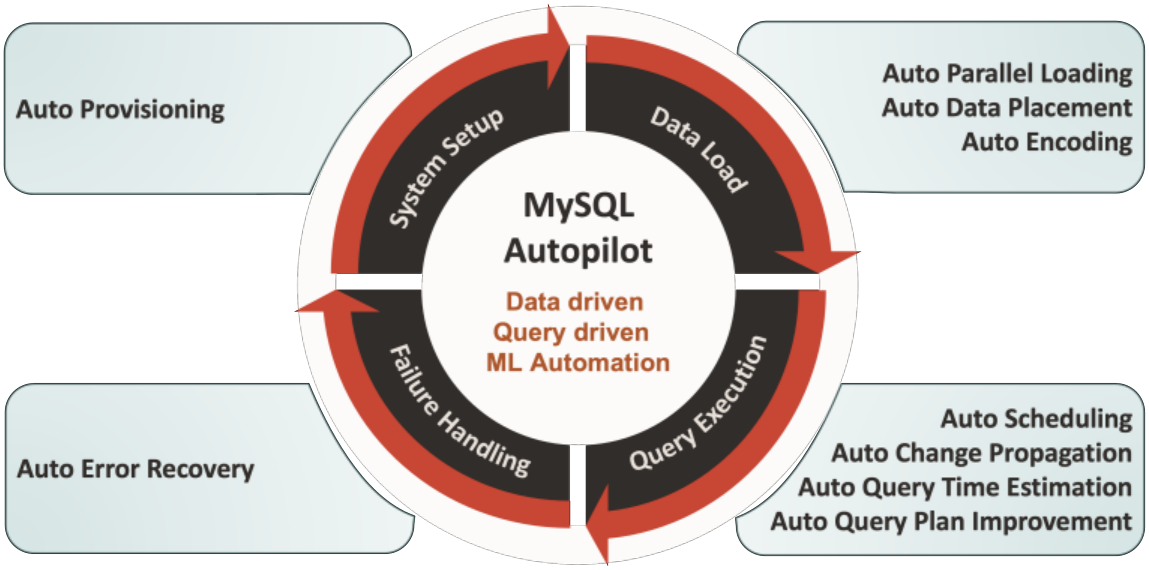
Public cloud services are the epitome of "time-is-money." The vast majority charge consumption on a vCPU (virtual CPU represents ½ core) per unit of time basis. The performance of MySQL Database Service with HeatWave and Autopilot is so much faster than the other MySQL database cloud services and data warehouse cloud services based on the TPC-H benchmarks[3], that it utilizes far less vCPU processing time. Less computing time translates into much lower costs. Documented users are seeing cost decreases ranging from 1/3 to as low as 1/6 that of AWS, Azure, Google MySQL, or Snowflake cloud services. This is a direct result of MySQL HeatWave and Autopilot's advanced automation.

HeatWave is a massively parallel, in-memory, scalable, native analytics engine tightly integrated with MySQL Database Service. It converges OLTP and analytics into one unified MySQL cloud database, improving the performance of both, while reducing total cost and complexity.

Autopilot is what makes the latest release of MySQL Database Service with HeatWave even more impressive. It's a database and data warehouse optimization engine leveraging highly advanced machine learning-based algorithms. Autopilot automates many of the most difficult and manually labor-intensive DBA tasks. This first release includes nine new groundbreaking automation services illustrated in the image below.

[2] TPC-H is an open-source decision support benchmark. It consists of a suite of business-oriented ad hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance that simulates typical scenarios. This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity, and give answers to critical business questions. However, since Oracle optimized the baseline standard MySQL test machine to enable it to shine in the best possible light against MySQL Cloud Services with HeatWave and Autopilot, it is technically out of compliance with the TPC-H rules. Unoptimized standard MySQL implementations will compare less well against MySQL Cloud Service with HeatWave.

[3] AWS Redshift w/AQUA, Azure Synapse, Google Big Query, and Snowflake.



This level of automation is unheard of for database and data warehouse cloud services with the exception of Oracle Autonomous Database.

This updated research report examines in detail the previously discussed MySQL problems that Oracle set out to solve and how they solved them. It then compares and contrasts the unique Oracle MySQL Database Service with HeatWave and Autopilot versus Amazon Aurora, Azure MySQL cloud services, GCP Cloud SQL for MySQL, plus data warehouse cloud services Amazon Redshift with AQUA and Glue, Snowflake, Azure Synapse, and GCP BigQuery. That analysis reveals why all other MySQL and data warehouse cloud services are not very competitive. In truth, there is no other MySQL database cloud service, open-source distributions, or data warehouse cloud services that come close to matching Oracle MySQL Database Service with HeatWave and Autopilot.

Table of Contents

Common MySQL and MySQL Database Cloud Service Problems

The ubiquitous MySQL is generally pretty easy to use and very low cost as an open-source product. These factors have made it the most popular transactional database in the world according to Enlyft. These are some of the primary reasons why MySQL is also a very popular database cloud service, sometimes referred to as MySQL Database-as-a-Service or DBaaS. For the purposes of this research, it will continue to be referred to as a MySQL database cloud service.

There are three principal areas where MySQL database cloud services have serious issues. These issues are poor analytical performance, considerable amount of manual labor-intensive DBA tasks, and high costs in time and treasure.

Poor Analytical Performance

As fast as MySQL is at transaction processing, it's equally atrocious at analytics. Running reports and analytics on MySQL are dreadfully slow—as slow as watching paint dry. Transactional databases have historically not been very good at analytics. That's why there are so many data warehouse on-premises software products and cloud services. Unfortunately, that means the data that originates in the MySQL transactional database cloud service has to be copied, stored, massaged, reformatted, transferred, loaded, tested, and validated in the target data warehouse cloud service. The shorthand for this is 'Extract, Transfer, and Load' (ETL).

ETLs are not trivial processes and fraught with errors. The good news is there are several ETL tools and services that greatly facilitate ETL processes. These tools and services reduce the time, complexity, and errors. But they by no means eliminate them. And there is of course the additional cost. Just as the data warehouse cloud service is an additional cost.

Worse is the fact that the query performance rapidly degrades as the analyzed data grows or the queries become more complex with multiple joins. These situations are more common than many users realize—until they start utilizing the data warehouse service.

One of the most frustrating aspects of MySQL cloud services paired with data warehouse cloud services is the need for DBAs to find a way to tweak performance to meet requirements. This is especially daunting when the DBAs have far fewer levers to manipulate while keeping costs to a sane level.

Too Many Manual Labor-Intensive and Database-Disruptive DBA Tasks

To the uninitiated in the use of database cloud services, cloud database, and data warehouse services, there is a belief that they're supposed to be fully managed by the service provider. And to a certain degree, they are. The issue is what the service provider means by 'fully managed.' They typically mean they'll own the software and all of the supporting infrastructure required to operate the service. They'll also own all of the maintenance, patching, hot fixes, and upgrades. Those are important costly and time-consuming tasks. But even those tasks are rarely automated. Users quickly learn their elastic scalability does not necessarily mean automated. Life with MySQL database and data warehouse cloud services is not all wine and roses. There are many tasks that require user decisions and system disruptions, or in other words, downtime.

Take the underlying hardware infrastructure for example. The DBA has to determine, commonly through trial and error, what hardware shape they need to meet their MySQL database performance requirements. When there is a requirement for more memory and processing, the DBA will have to change that hardware shape. Changing the hardware shape is disruptive, requiring an outage as the system reconfigures. The same is true for the data warehouse with one exception. Snowflake allows automatic spin up of multiple hardware shapes identical to the original one, without requiring an outage. These additional shapes are conceptually similar to database shards in that they are separate from one another. They're treated as separate resources within the virtual data warehouse. Each spun up hardware shape adds an equal cost increase to the original hardware shape. This is not a cost-effective way to increase hardware infrastructure resources. If the data warehouse started out using a 32-vCPU hardware shape and needs just one or two more vCPUs, they will actually get and pay for 32 more, doubling the cost.

That hardware infrastructure example is just one part of what the DBA has to do to tune the cloud-based MySQL performance AND the data warehouse performance. They also tweak storage and networking to squeeze out better performance. Performance tuning is a primary task DBAs perform for on-premises MySQL and data warehouses. A primary reason they move to cloud services is to offload that time consuming, error-prone, and difficult task. It's as much art as it is science. And yet they quickly find they still have to do it in the cloud, albeit with much less granularity and less control. So not only is the process not off-loaded, it's also harder, more time consuming, and more error prone. Yikes!

Patching is another big issue. In this era of escalating malware, ransomware, and data breach attacks, no one can afford to leave database, data warehouse, OS, virtualization, and hardware microcode vulnerabilities unpatched. That's just asking for a major disruption and unpalatable costs. The MySQL database and data warehouse cloud service providers are obligated to provide those patches. However, their patching is disruptive. That means they do not automatically apply them. They leave it to the customer's DBA to decide when to apply them. The DBA will schedule them for a time when the disruption has the least impact on the organization's business. This creates a window of vulnerability.

Exceedingly High Costs in Time and Treasure

This is not rocket science. MySQL database and data warehouse cloud services have many costs associated with them beyond the services themselves. These costs include multiple types of storage – block storage in the form of SSDs or HDDs for the MySQL database cloud service and object storage for the data warehouse cloud service – ETL tools or services, time managing each of these services and, of course, the disruptions for troubleshooting, patching, hot fixes, and upgrades. Remember, every cloud service must be managed, tuned, monitored, and maintained—each with its own costs.

Then there's the DBA's time for troubleshooting, disruptive patching, disruptive upgrades, and disruptive hardware shape changes. There are only 24 hours in a day. Time is precious. As highly trained professionals, a DBA's time is expensive. Time spent on many of these pedantic non-strategic tasks equates to less time available for more strategic pursuits such as SQL query construct optimization or application development. SQL query constructs are very important to application response times. And as previously stated, time is very much money.

Counter-intuitive is how cheaper, smaller hardware infrastructure shapes can end up costing more money than the more expensive hardware shapes. This is because charges are based on vCPU compute time. The MySQL database cloud service and the data warehouse cloud service charge per vCPU per second. Fewer vCPUs take much longer to process data, often resulting in higher compute costs. Balancing the two correctly takes more time.

Whereas the saying 'time is money' may be a cliché, it's definitely true for MySQL database and data warehouse cloud services.

The Oracle Answer: MySQL Database Service with HeatWave and now Autopilot

The previous research on Oracle MySQL Database Service with HeatWave clearly illustrated how this Oracle cloud service solved the MySQL analytics and performance problems. It was and still is the only MySQL database cloud service with a built-in in-memory data warehouse analytics engine that provides actionable insights in real-time.

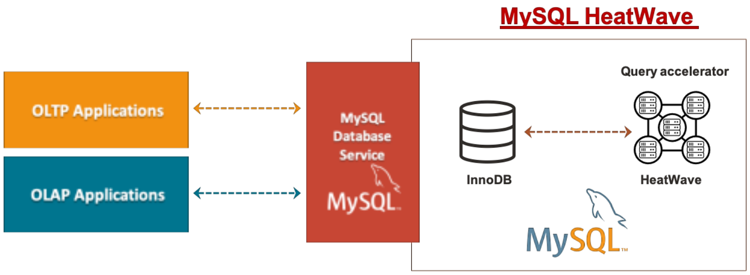


Figure 1: Oracle MySQL Database Service with HeatWave

MySQL Database Service with HeatWave allows data to stay within MySQL, requires no ETLs, runs any application using MySQL or MySQL databases with no changes, performs considerably faster than on-premises or cloud-based MySQL database services, scales significantly greater than other MySQL database cloud services or data warehouse cloud services, and costs materially less than other MySQL database cloud services combined with ETLs and data warehouse cloud services – as much as 5/6 or 83% less, and it's available in all 44 Oracle Cloud Infrastructure (OCI) data centers and as Dedicated Region Cloud@Customer deployments around the world.

Oracle established a new standard with the Oracle MySQL Service with HeatWave. This exclusive cloud service set an incredibly high bar that no other MySQL database or data warehouse cloud service comes close to meeting. Oracle has now raised that previously high bar even higher. With its latest release of MySQL Database Service with HeatWave, Oracle has increased performance, scalability, and now solved the manual, labor-intensive task problem with Autopilot.

What is MySQL Autopilot and Why It's Important

Oracle MySQL Autopilot is the first AI machine learning (ML) powered database and data warehouse cloud service architected to automate the vast majority of the laborious time-intensive DBA tasks. Oracle Automated ML or AutoML is at the heart of Autopilot and is just as seismic as HeatWave. AutoML automates common machine learning tasks that eliminate many of the DBA's labor-intensive tasks that benefit both the novice and the highly skilled, knowledgeable, and experienced data scientist. It automates nine major aspects of Oracle MySQL Database Service with HeatWave. A comparative look at life without and with MySQL Autopilot for some of the more prominent automation capabilities makes this crystal clear.

Auto-provisioning

Take the typical MySQL database and data warehouse system setup. Traditional hardware infrastructure provisioning is commonly guesstimated based on performance requirements and the DBA's experience. It's rarely precise, often inaccurate, and most DBAs tend to be conservative with costly over-provisioning just to make sure those performance requirements are met. Provisioning is an extensive time-consuming task because there are several iterations based on trial and error as the DBA manipulates the hardware shapes to achieve the performance they're looking for.

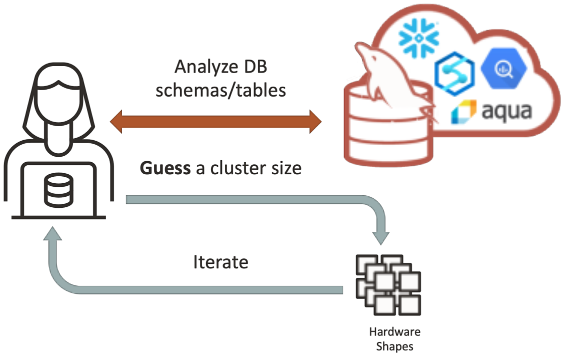


Figure 2: Manual Hardware Shape Sizing

Oracle MySQL Autopilot eliminates provisioning guesstimation with Auto-provisioning. Auto-provisioning utilizes AutoML to accurately predict memory usage and estimate cluster size.

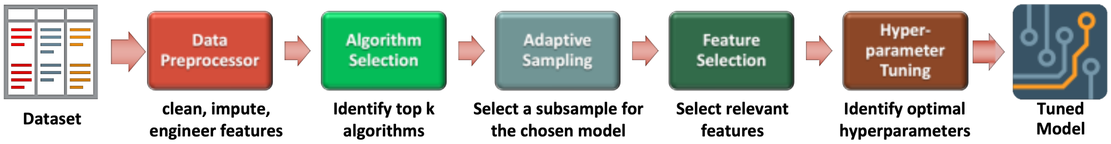
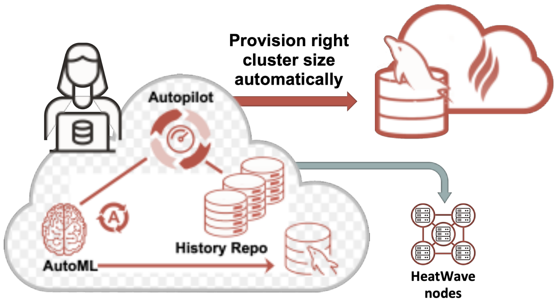
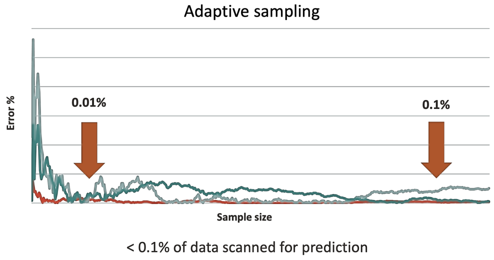
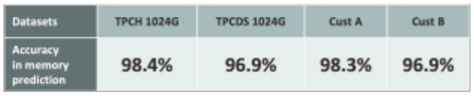


Figure 3: Auto-provisioning Process

It uses the automated pipeline to accelerate an accurate model. And it does it by utilizing less than 1% of scanned data. Auto provisioning results have proven to be very accurate both in the lab and with Oracle MySQL Autopilot customers, as illustrated below.







Datasets	TPCH 1024G	TPCDS 1024G	Cust A	Cust B
Accuracy in memory prediction	98.4%	96.9%	98.3%	96.9%

Figure 4: Auto-provisioning Results

Auto Data Placement

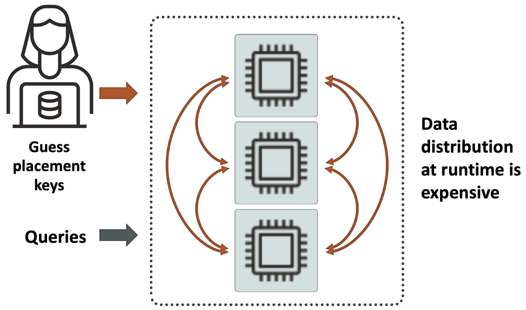


Figure 5: Manual Data Placement

Data placement is another of DBA's manual, labor-intensive guesstimation efforts without Autopilot. Performance is sub-optimal, requiring many trial-and-error iterations and even then, it's still likely to be suboptimal. Suboptimal performance in a cloud service adds runtime and considerable cost.

No matter how experienced, knowledgeable, and skilled the DBA, determining effective data placement is difficult and laborious.

Autopilot's Auto Data Placement predicts and shows optimal columns to partition data in-memory based on queries. And it predicts the expected improvement in runtime. Measuring the expected improvement against actual results reveals a conservative bias on the part of Autopilot. It makes the correct prediction of columns to partition while under-predicting the amount of actual improvement. A nice bonus.

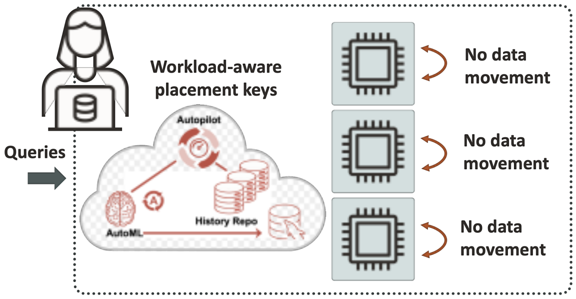


Figure 6: Auto-data Placement

Auto-Query Optimization

Then there are the DBA's migraine headaches with query acceleration, optimization, and management. Transactional and analytical queries are different and compete for the same resources when running in the same database. Analytical queries consume a much bigger chunk of memory and computer resources than transactional ones. They take longer. Both are using and contending for the same storage I/O. This results in serious performance deterioration for both, but more so for transactional queries.

The MySQL Database Service with HeatWave and Autopilot decouples transactional from analytical queries. Not only does the transactional query performance stay incredibly fast when analytical queries are run, but the analytical queries also execute noticeably faster. Best of all, it's completely automated. No DBA intervention is required to prioritize transactional or analytical queries over the other.

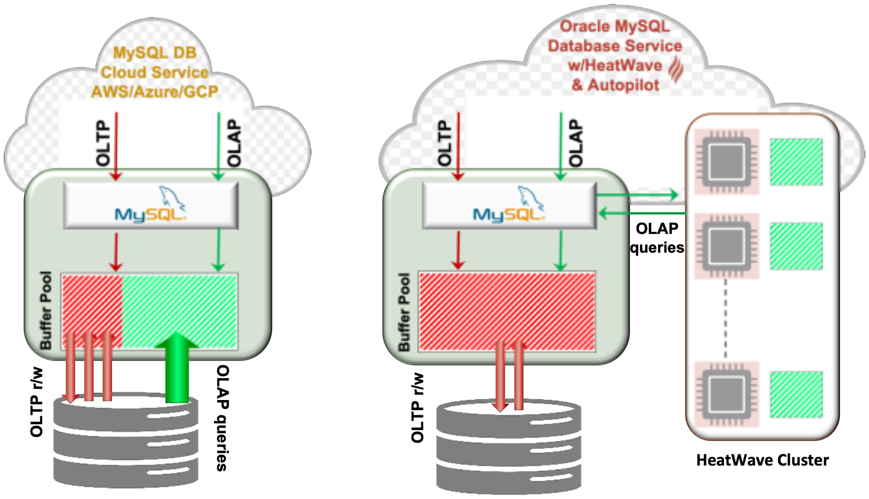


Figure 7: Comparison of Mixed Workloads

Autopilot accelerates and optimizes query performance with its Auto Query Plan Improvement and Auto Scheduler. The Auto Query Plan Improvement is an automated query optimizer that advances the query plan based on earlier executed queries. In other words, Oracle MySQL Database Service with Heatwave and

Autopilot uses ML to constantly improve the query plan. The proof is in the results. MySQL Autopilot improved TPCH, TPCDS 24TB performance by as much as 40%.

The Auto Scheduler fixes the lack of query prioritization problem. It automatically predicts execution time and prioritizes shorter queries based on ML to reduce overall wait times for mixed workloads. Testing results show short query times reduced by more than 45% while having a negligible impact on the longer more complex analytical queries. That's a noticeable and significant savings, keeping online customers happily purchasing products or services.

Auto-Scale Out Data Management

One thing DBAs abhor with a passion is having to recover their database due to an error or cluster outage. It's a hands-on stressful exercise and a considerable time sink. The root of that distaste is the lengthy amount of time and attention required to reload the entire database from an error recovery or an upgrade. The size of the database impacts the time to reload. Bigger takes longer. Time is also impacted by the speed of the storage. Cloud storage speed directly correlates to cost. Faster storage is more expensive, creating a bit of a conundrum. The major reason IT organizations choose a MySQL database cloud service is to keep costs down, which is difficult to do when the fastest storage is required.

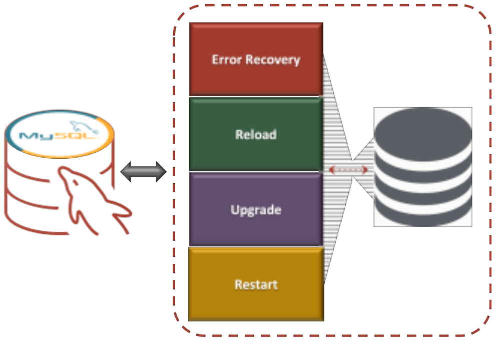


Figure 8: General MySQL DB Cloud Services Reload

MySQL Autopilot removes the DBA's reload effort and stress through automation and clever use of Oracle Cloud Infrastructure (OCI) object storage. This provides cost-effective and highly scalable data access, but most importantly, data is reloaded in a consistent timeframe for recoveries, upgrades, and restarts, regardless of the size of the database. It does this by:

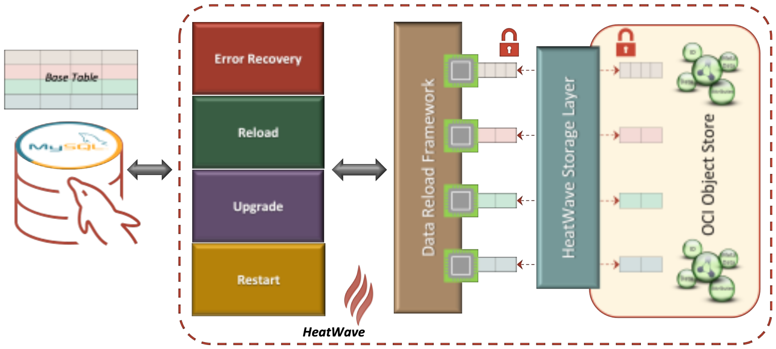


Figure 9: Comparison of Mixed Workloads

- Partitioning the data and storing it encrypted in OCI object store's in-memory format.
- Reloading data in parallel at object store bandwidth.
- Loading data at a fine level of granularity.
- Propagating MySQL Database changes to the OCI object store.

Upon a failure, analytics nodes are transparently re-provisioned, and the data is reloaded at very high speed. Recovery time is constant regardless of how many nodes need to be recovered. No DBA intervention is required. The reload performance is automated and with high performance. And not only is the data stored encrypted in the OCI Object Store, but so are the data transmissions between the analytic nodes and the OCI Object Store. This makes Oracle MySQL Database Service with HeatWave and Autopilot the most secure of the MySQL database cloud services currently on the market.

HeatWave Increases Scalability

Oracle Autopilot's unmatched automation was not the only massive improvement to this service. HeatWave's analytics capacity scalability was dramatically improved by 50% from 24TB to 32TB in 64 nodes. More importantly, the increased capacity does not negatively impact performance as seen in the chart below.

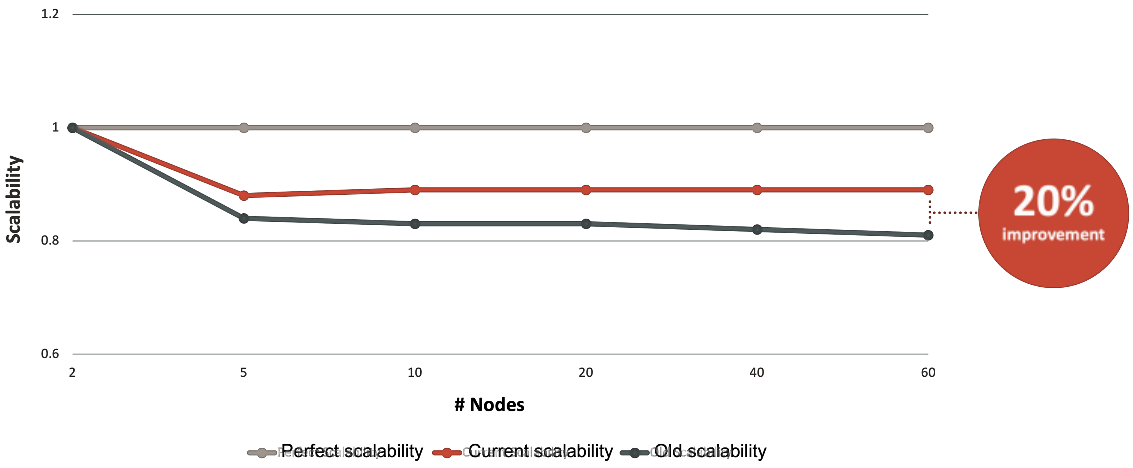
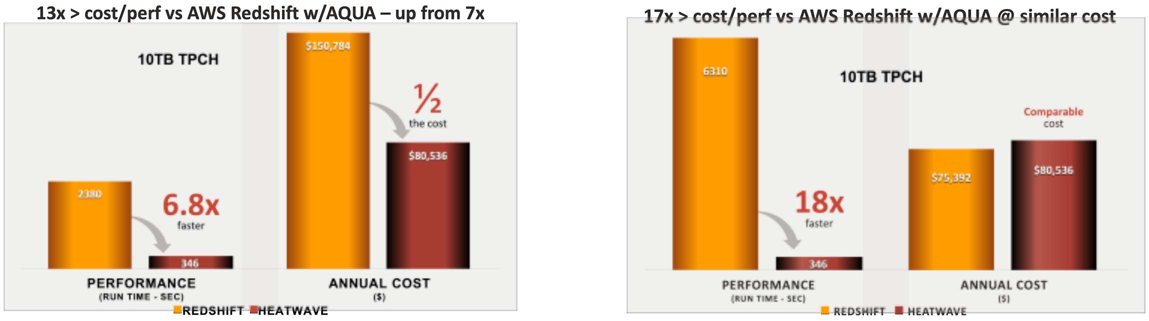


Figure 10: HeatWave TPCH and TCPDS Linear Performance at Scale

In fact, both HeatWave analytics and MySQL transactional performance have expressly increased. The competitive comparative cost/performance results are considerably more impressive.

Comparative Cost/Performance Results

Oracle MySQL Database Service with Heatwave and Autopilot's TPC-H[4] analytics testing literally blows away Amazon Redshift with AQUA (Advance Query Accelerator for Redshift) in both performance and cost/performance. Putting Amazon Redshift with AQUA in the best possible light utilizing the highest performing hardware shapes, MySQL Database Service with HeatWave and Autopilot is 6.8x faster and ~47% less expensive. That equates to ~13x better cost/performance for MySQL Database Service with HeatWave and Autopilot. By using AWS hardware shapes that are comparable in cost, Amazon Redshift with AQUA is ~18x slower resulting in Oracle MySQL Database Service with HeatWave and Autopilot coming in at ~17x better cost/performance.


13x > cost/perf vs AWS Redshift w/AQUA – up from 7x

17x > cost/perf vs AWS Redshift w/AQUA @ similar cost

Figure 11: HeatWave Cost/Performance vs. AWS Redshift with AQUA

What about Oracle MySQL Database Service with HeatWave and Autopilot versus Snowflake. By extrapolating the TPC-DS benchmark tests[5] run by Fivetran – a data pipeline and ETL provider – on Snowflake and Amazon Redshift, Oracle MySQL Database Service with HeatWave and Autopilot is once again faster and lower cost. Snowflake ranged from approximately 2 to 3X faster than Redshift without AQUA, according to Fivetran. Based on just the query processing within Snowflake, MySQL Database Service with HeatWave and Autopilot is ~7x faster than Snowflake at 1/5 the cost—that's 35x better cost/performance.

It is also imperative to keep in mind that there are no ETL tools, services, or duplicated storage required for Oracle MySQL Database Service with Heatwave and Autopilot. Those are required for all other MySQL database cloud services working with a data warehouse cloud service. That makes the cost comparisons

[4] Benchmark queries are derived from TPC-H benchmark. But as previously mentioned the non-Oracle MySQL test results are technically out of compliance since they are tuned to put the competitors' best foot forward and so are not considered standardized. Oracle has put the entire test and configurations online at GitHub.
[5] TPC-DS is an industry-standard benchmarking meant for data warehouses.

above excessively conservative. The real costs for AWS, Azure Synapse, Google BigQuery, and Snowflake should include the cost of the transactional MySQL database cloud service, duplicated data storage, ETL services, ETL tools, lost time, lost opportunities, and significant above the line lost revenue potential. If these costs were considered, the actual total savings for MySQL Database Service with HeatWave and Autopilot would be much greater. In fact, several of Oracle's customers claim Oracle MySQL Service with Heatwave and Autopilot costs come in 1/6th of what they were paying AWS.

Conclusion

Oracle MySQL Database Service with HeatWave was a revolutionary high performance MySQL database cloud service when it was first introduced late last year. It was, and still is, the only MySQL database cloud service with a highly integrated high-performance, in-memory analytics engine. Its performance and cost performance have no equal. Just as the market was getting used to this new high-performance low-cost paradigm, Oracle raised the bar even higher with the addition of MySQL Autopilot automation based on Oracle AutoML.

Autopilot eliminates or mitigates most of the most onerous manual, labor-intensive DBA tasks while further improving performance, saving yet more time and, of course, money. Whereas HeatWave fundamentally changed the MySQL database cloud service game, Autopilot has made it a tectonic change. It is now crystal clear that Amazon Aurora, Redshift with and without AQUA, Snowflake, Azure Synapse, and Google BigQuery are not in the same class as Oracle MySQL Database Service with Heatwave and Autopilot. They're not even in the same league.

For More Information on the Oracle MySQL Database Service with HeatWave and Autopilot

Go to: HeatWave