

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ



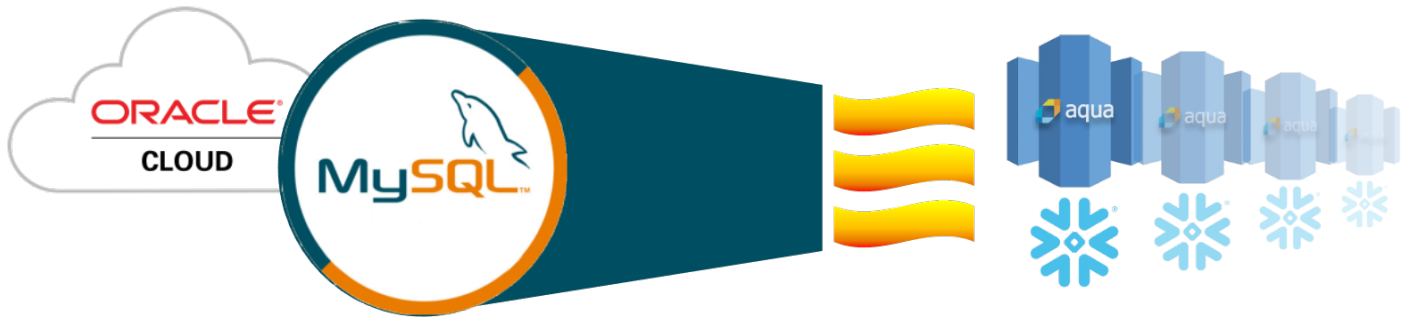
Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

by, Marc Staimer

March 7th, 2022

Research Premises This research is based on the following seven premises: Open-source MySQL relational database management system (RDBMS) is the most popular database in the world[1] per Enlyft1. It meets or exceeds the needs of organizations of all sizes. MySQL has approximately 20.62% of the transactional database market with more than 190,304 unique customers. The [...]

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ



Research Premises

This research is based on the following seven premises:

- Open-source MySQL relational database management system (RDBMS) is the most popular database in the world[1] per Enlyft¹. It meets or exceeds the needs of organizations of all sizes. MySQL has approximately 20.62% of the transactional database market with more than 190,304 unique customers. The next closest is Microsoft SQL Server with 19.59% and approximately 180,486 customers. The reported MySQL market share and number of customers does not include the MySQL forks—binary drop-in replacements—of MariaDB and PerconaDB. Nor does it include the customers of database cloud services utilizing MySQL, MariaDB, and PerconaDB such as those offered by AWS RDS and Azure. It's important to note that PostgreSQL, the other popular open-source relational database, is very distant fourth in popularity with less than 5% of the market.
- MySQL popularity ranges across large, medium, and small IT organizations including some very recognizable names. Impressive [names](#) such as NASA, Netflix, Tesla, Facebook, Twitter, Spotify, YouTube, GitHub, the United States Navy, Airbus/EADS, WeChat, Zendesk, JP Morgan Chase, CERN, Airbnb, Caterpillar, UN, Marketo, Verizon Wireless, Bayer, Booking.com, and many more.
- There are many illustrations of MySQL popularity among developers and applications. One of the more ubiquitous examples is the open-source application [Magento](#), an Adobe company. It is the world's leading eCommerce platform and specifically designed around MySQL. Another good example is the broadly utilized [Marketo](#) marketing automation software. There are thousands more.
- Contrary to its popularity, MySQL users have unresolved issues with this database. The most common complaint is its extremely limited analytics capabilities. There is no built-in analytics engine within MySQL, making performance atrocious. Complex queries or scaling queries are non-starters. This major shortcoming does not eliminate MySQL user requirements for reporting and analytics that generate insights into their data. It just forces MySQL users to use an additional database optimized for analytics such as a data warehouse. This always looks better on paper than empirical reality. The problem arises from having to move the MySQL transactional data to that data warehouse. This is a non-trivial task commonly referred to as an ETL or Extract, Transform, and Load process. ETLs take considerable time and effort. That means reporting and analytics do not occur in real-time. There is always some delay (a.k.a. latency), to the point of delivering insights that can be too late to take advantage. And anytime data is moved and manipulated there is an increased probability of corruption, deletion, unauthorized access, or errors.

ETLs require costly engineering resources to develop, document, maintain, and upgrade on an ongoing basis. The ETL processes are generally continuously tweaked and modified to improve performance or to change/increase data sources—because time is money and ETL delays cost both. Database administrators do not like ETL processes. They take a lot of work and frequently cause data integrity issues despite best efforts in error detection and correction.

Although there are reasonably effective third-party ETL tools, services, and cloud services, they add cost.

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

They also frequently consume time when troubleshooting the invariable problems that arise with multiple vendors, as documented in Murphy's Law.

- The emergence of MySQL, MariaDB, and PerconaDB database cloud services presented hope as a solution to these analytics problems. Those hopes have been dashed as it became clearly evident that transitioning to a MySQL, MariaDB, or PerconaDB database cloud service merely moved the problems from on-premises to the cloud. The MySQL analytic, performance, scalability, and data integrity issues don't change, they simply cost more.

Even AWS RDS (Relational Database Services) running Aurora—MySQL version—requires additional services such as their “GLUE” ETL as a service to move transactional data to the Redshift data warehouse. This does not address ETL issues, nor does it address MySQL scalability issues.

- Oracle MySQL Database Service with HeatWave in Oracle Cloud Infrastructure (OCI) is the first and currently only MySQL database cloud service that addresses all these issues. It scales both capacity and performance near linearly without compromising data integrity. It runs both transaction processing and analytics concurrently without ETLs. And it accelerates transactions **and** analytic queries to a level of performance previously unseen or imagined. In a nutshell, MySQL Database Service with HeatWave is the database cloud service that MySQL users have been waiting for.

Executive Summary

This research is designed to help CIOs utilizing MySQL in their decision on which MySQL database cloud service to adopt. It does so by taking a deep dive into the crucial MySQL database issues users are dealing with today, how current MySQL database cloud services deal with them, and what Oracle MySQL Database Service with Heatwave does uniquely better.

Oracle is and has been the primary development and support organization for open-source MySQL since acquiring Sun Microsystems in 2010. That support made it inherently obvious to them the severity of these persistent MySQL issues. Oracle determined the best way to solve them was to offer MySQL database as a cloud service, but with a unique innovation called “HeatWave” to address the analytics issue.

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

All existing MySQL or forked applications run without any changes. Oracle MySQL Database Service with HeatWave does this while delivering unprecedented scalability and performance. It scales to thousands of cores and 24TB of data. TPC-H[2] testing as well as production customers show a 400x increase or more in performance over MySQL databases on-premises or other MySQL database cloud services. Common wisdom is that massive increases in scalability and performance will cost more. MySQL Database Service with HeatWave defies common wisdom by costing less.

This DSC research examines in detail the previously mentioned MySQL problems Oracle set out to solve and how they solved them. It then compares and contrasts the unique Oracle MySQL Database Service with HeatWave versus AWS Aurora, Redshift, AQUA, and Glue in addition to Azure MySQL Cloud Service, Snowflake, and Google BigQuery. That analysis makes crystal clear why other MySQL and data warehouse cloud services are not competitive. In truth, there is no other MySQL database cloud service or licensed software that comes anywhere close to MySQL Database Service with HeatWave.

[1]Source: [Enlyft](#) statistics as of June 2021

[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

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

chosen to have broad industry-wide relevance. 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, it is technically out of compliance with the TPC-H rules. Non-optimized standard MySQL implementations will compare poorly against MySQL Cloud Service with HeatWave.

The Persistent Fundamental MySQL User Issues

MySQL is very easy to use and generally low cost, making it the most popular open-source database in the world. And yet user complaints persist. Those complaints can be summarized into three categories:

1. MySQL is optimized for transactions. It does them quite well and a key reason to its popularity. It does not do analytics very well at all nor was it designed to do so. To get effective reporting and analytics from a MySQL database has required doing those analytics in a separate database such as a data warehouse, or equivalent data warehouse in the cloud. That in turn requires a process known as Extraction, Transform, and Load, a.k.a. ETL. ETLs can be sped up through the use of third-party tools, or equivalent ETL cloud services. Regardless, ETLs take time, effort, and money, making the transactional data stale and the analytics out-of-date.
2. Query performance degrades precipitously in many data warehouses as the amount of analyzed data grows. It is not unusual for large data warehouses to have queries that take hours or never complete at all.
3. Utilizing the MySQL database cloud services in major cloud service providers such as AWS or Azure currently do not fix these issues. They are more likely to exacerbate them—while costing considerably more.

The Dismal Consequences of the MySQL Analytics Issue

Few if any database administrators (DBAs) truly like or enjoy doing ETLs. They do them only when they're necessary. They would prefer them not to be necessary. ETLs require time, skill, and expertise.

The MySQL DBA choices have generally been unsavory if they don't want to leave MySQL or its forks. They can do nothing and continue to accept painfully slow analytics within the MySQL database, ETL to a data warehouse on-premises or in a cloud, or they can move their applications and MySQL database to a public cloud while leveraging that cloud's data warehouse cloud service. There are several drawbacks to all three choices.

Do Nothing - i.e. Keep the Reporting and Analytics Within the MySQL Database

By doing nothing there is no additional out-of-pocket cost. But there is a substantial time cost that invariably leads to personnel costs and revenue losses. Consider that much slower analytics and reporting results in much slower time-to-actionable-insights, missed opportunities, inability to respond to changing market conditions, reduced competitiveness, lost customers, and lower revenues. Time is money.

Slow analytics and reporting have several additional highly negative impacts. It can and often will consume database resources, noticeably slowing transactions. Slower response times reduce internal productivity, frustrate customers, and lose revenue. This often demands that DBAs run those analytics and reports in off hours, late nights, weekends, and holidays. That adds significant DBA overtime costs while making the results that much colder.

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

Move the Data and Analytics to a Data Warehouse or Data Warehouse Cloud Service

Moving the data to be analyzed into a data warehouse or data warehouse cloud service will mitigate some of this problem by accelerating the analytics and reporting over that produced within MySQL or its forks. The reason it only mitigates the issue is the ETL caveats. Transactional data cannot be analyzed in real-time because ETLs take time. That time ages the data, making it stale. Any insights derived from stale data are out-of-date. Decisions and actions from those insights are questionable and error prone. Non-time sensitive reports are generally fine—it is the time sensitive ones that suffer.

ETLs are not free. ETLs can be manually performed or semi-automated with tools built into the data warehouse. Manual ETLs cannot be counted on for continuous analytics and reporting. Semi-automated ones require in-house scripting, testing, debugging, documenting, patching, and upgrading over time. Rarely is this done well and many of these steps are not done all.

Alternatively, DBAs can utilize third-party ETL tools or services. Many of these tools and services are quite effective. Several are highly automated. However, they are not free. Third-party tools have software licensing costs, hardware and software support costs, administrator training and administrator time costs. ETL services have service and processing per unit of time costs depending on the hardware. Both have additional storage costs for the data replicas required for ETLs and storage in the data warehouse or data warehouse cloud service.

In conclusion, this process merely mitigates the issue. It does not come close to eliminating it, and it adds cost.

Move Apps, MySQL DB, and Data to Cloud Utilizing Cloud Data Warehouse

It is perfectly understandable why DBAs might expect that moving everything to a managed public cloud service would solve this problem. Unfortunately, as they quickly discover, it doesn't. They commonly end up with noticeably slower queries that the cloud service makes up for it by costing more.

MySQL database cloud services from AWS, Azure, GCP and others have not added any software or hardware to the database to improve the inherently poor analytics performance. Effective reporting and analytics still require ETLs from the MySQL database cloud service to a data warehouse cloud service.

AWS Relational Database Service (RDS) Aurora-MySQL and Redshift data warehouse cloud service exemplifies this. AWS has three different methodologies to provide analytics and reporting to the Aurora (MySQL) transactional database from their data warehouse, Redshift. Although the data warehouse cloud service could just as easily be Snowflake running on AWS except when utilizing AWS "Glue." (NOTE: Azure has a similar MySQL database cloud service and Synapse data warehouse cloud service, as does Google with BigQuery. However, their ETL tools are primarily customer-driven and labor intensive today—more in line with AWS Lambda and Kinesis Firehose.)

AWS Lambda with Kinesis Firehose ETL Process

The easiest task is providing a simple operational report. Shouldn't be too difficult and yet there are many steps to this process. The first is to write a trigger in the OLTP database to invoke an AWS Lambda function. Each trigger is typically a mere 10 to 20 lines of code. However, the number of triggers can rapidly escalate and run into thousands of lines of code. This code makes the AWS cash register ring every time it's triggered. AWS may charge only a few cents each time a Lambda function is called. It is common to assume then it shouldn't cost very much. And yet in reality it adds up quickly and can result in a surprisingly high

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

bill each month. The Lambda function then calls Kinesis, another AWS chargeable service. It's the AWS equivalent of the Apache Kafka service. The AWS Kinesis service stores the data into Amazon's Object Storage S3 bucket. In other words, it duplicates the MySQL data requiring analysis, adding yet more cost. AWS Redshift can then finally query the data from that S3 bucket in parallel.

This AWS process isn't simple, fast, nor cost effective. The other two ETL processes between Aurora MySQL and Redshift are called data pipeline and Glue.

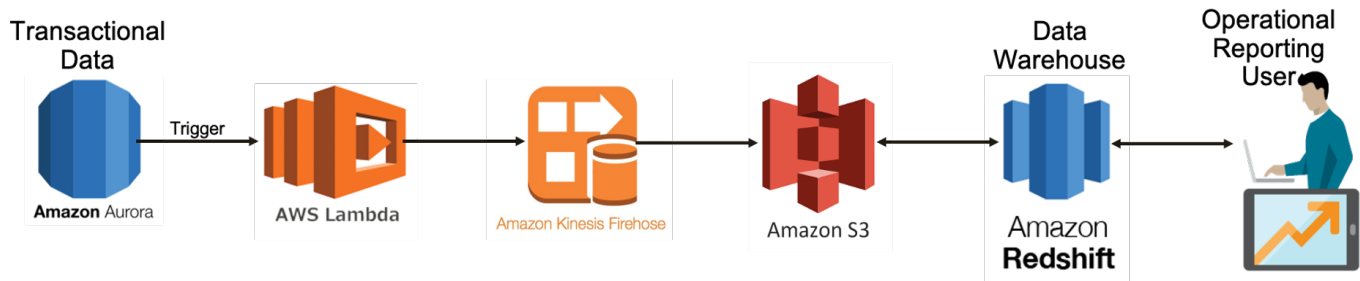


Figure 1: AWS Lambda and Kinesis Firehose ETL Process

AWS Data Pipeline

AWS Data Pipeline cloud service is a semi-automated ETL process. It's flexible with several built-in options for data handling. It can also control the instance and cluster types while managing the data pipeline. The data pipeline has built-in templates in the AWS console that theoretically simplify implementations. Depending on business logic, condition check, and job logic these templates can be relatively user-friendly vs user-coded custom scripting. This is a major improvement over the more manual AWS Lambda and Kinesis process.

However, there are serious shortcomings to the data pipeline process. Data pipeline is not serverless, meaning there is an ongoing instance cost even when it is not in use. In addition, AWS Data Pipeline internally triggers the EMR ETL cluster running behind the scenes with multiple S3 storage buckets and two copies of the data adding yet more cost. These fees add up quickly and can be more costly than the more manual Lambda with Kinesis Firehose process. Job handling for complex pipelines is problematic. It requires significant skill sets and expertise. Occasionally it can produce a non-meaningful exception error that significantly complicates troubleshooting. AWS Data Pipeline is not available in all regions—refer to the [AWS website](#).

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

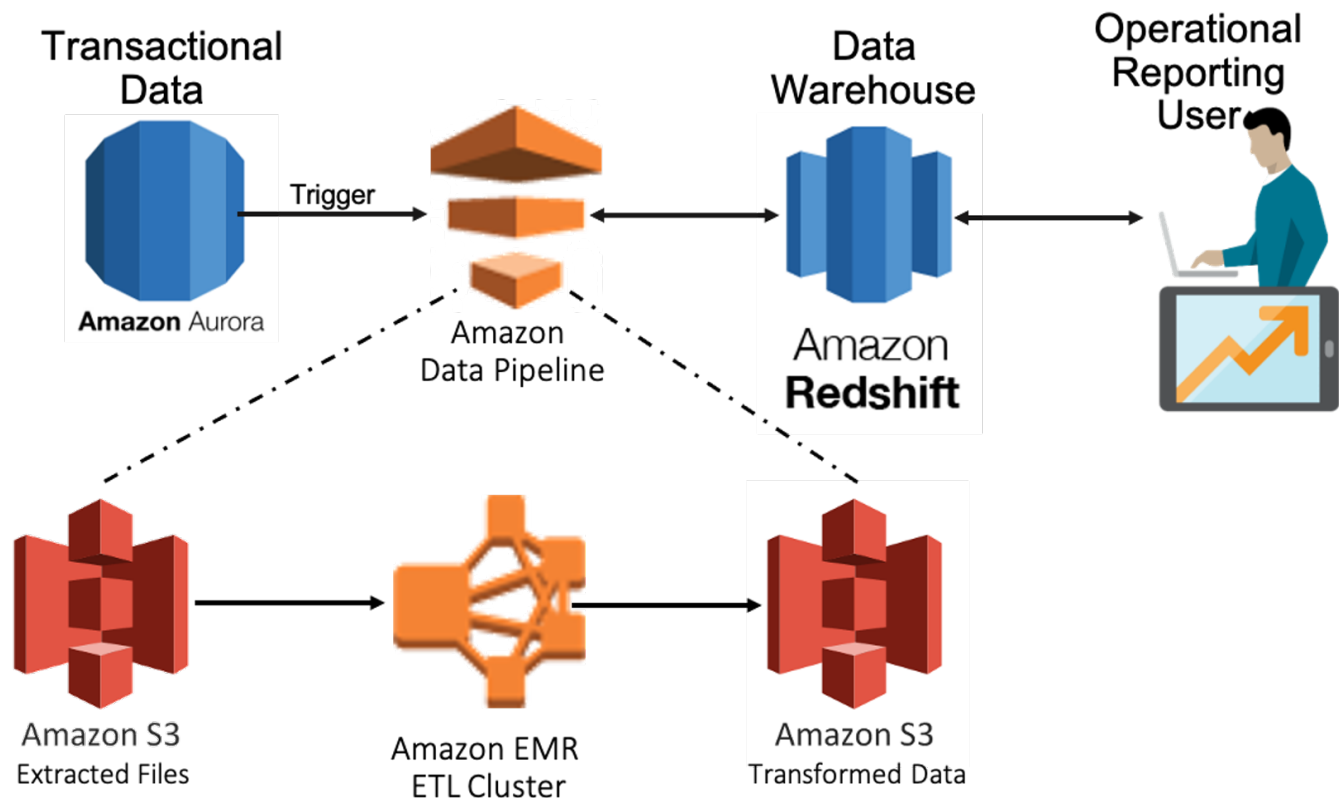


Figure 2: AWS Data Pipeline ETL Process

AWS Glue

Amazon's most sophisticated cloud ETL service is called Glue. It's an AWS managed service with the primary components of data catalog, ETL engine, and scheduler. The data catalog is a common location storing, accessing, and managing metadata such as databases, tables, schemas, partitions, etc. Glue uses data source crawlers to infer schemas and objects within the data sources. It utilizes this to create tables with metadata in the AWS Glue data catalog. From the data catalog the ETL engine enables the DBA to create an ETL job by selecting source and target data stores. Target schemas are also provided by the data catalog. The DBA defines each ETL job with Glue, generating PySpark code. That code commonly needs to be customized based on validation and transformation requirements. The Glue scheduler can then run the ETL jobs on-demand, at a designated time, or triggered by the completion of a separate job. The scheduler can be set to automatically retry failed jobs.

Unlike Data Pipeline, Glue is completely serverless and does not carry any instance charges. It requires no resource management. Glue is charged based on the query time and the data per unit, or DPU rate.

Glue has its shortcomings as well. It currently only works with Amazon Database Cloud Services. Glue does not currently work with third-party database cloud services such as Snowflake. It's not recommended for complex ETL logic. It's also region limited—refer to the [AWS website](#)—similar to Data Pipeline and, restricted internally to the Spark environment to process data.

Spark also has significant issues that affect Glue. Issues such as: no support for real-time processing; small files; no file management system; high memory consumption; inadequate number of algorithms; extensive labor-intensive manual optimization required; iterates in batches with each iteration scheduled and executed separately; relatively high latencies; no support for record-based window criteria - only time-

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

based window criteria; and labor-intensive manual back pressure handling. One more thing about Glue, when dependent jobs and success/error handling are invoked, Glue requires in-depth knowledge of the other AWS data cloud services being utilized such as Lambda, CloudWatch, and Data Pipeline. AWS Glue may simplify ETLs; however, it will add complexity, performance latency, and, most significantly, manual labor and considerable cost.

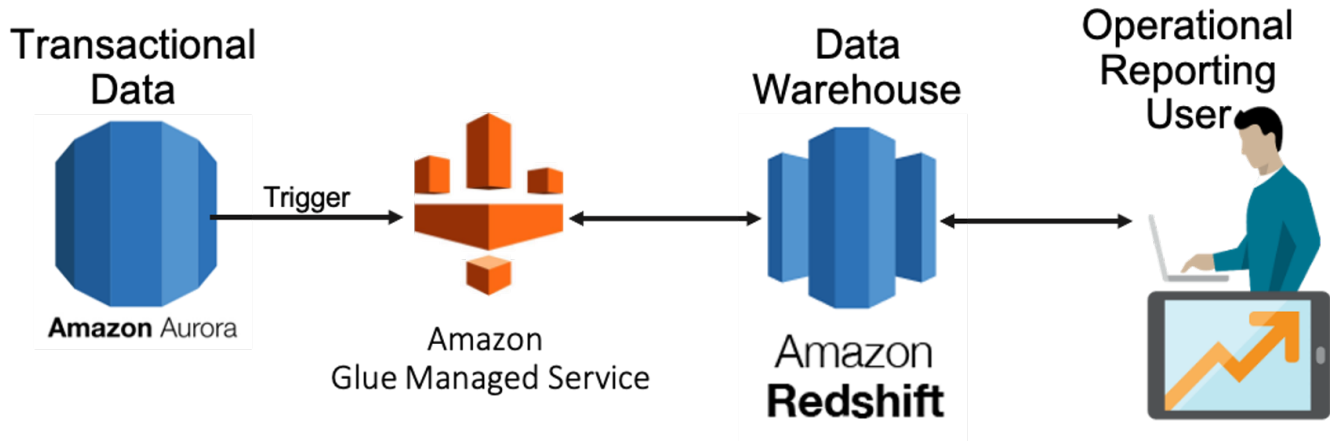


Figure 3: AWS Glue ETL Process

What about other MySQL cloud services combined with data warehouse cloud services such as Azure Synapse, Google BigQuery, and Snowflake? In all cases, an ETL from MySQL must occur. There are manual tools similar to AWS Lambda, available from each of these cloud services. Or there are third-party services and tools that simplify ETLs but at a significant additional cost. Tools and services such as Hevo, Talend, Data Virtuality, and even the industry leading ETL tool from Oracle called GoldenGate. None eliminate ETLs, their need to be managed, or their cost both in time (stale data, stale insights, lost opportunity, and revenue) and treasure.

These third-party tools and services are attempting to manage the symptoms of the MySQL reporting and analytics problem without solving it. It's analogous to taking drugs to manage the symptoms of a disease versus taking a drug that cures it. And, like the drug analogy, managing the symptoms has a much higher cost.

Oracle's Solution: MySQL Database Service with HeatWave – The First Complete Solution

Oracle set out to solve these difficult MySQL problems with clever innovation. It started with several **reporting** and **analytics** objectives **must haves**:

- MySQL data stays within MySQL.
- No ETLs required.
- No change to the applications using MySQL or MySQL compatible databases
- Perform considerably better than native MySQL or MySQL database cloud services.
- Perform significantly better than data warehouse cloud services.

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

- Cannot negatively impact MySQL transaction performance.
- Reporting and analytic tasks don't require different processes from MySQL.
- Scale much better than either MySQL or data warehouse cloud services.
- Leverage commodity cloud hardware and software.
- Cost less than MySQL database cloud services combined with ETLs and data warehouse cloud services.
- Available in ALL Oracle Cloud Infrastructure (OCI) data centers and Dedicated Region Cloud@Customer.

Oracle spent hundreds of person years and extensive R&D developing their MySQL Database Service with HeatWave. This cloud service doesn't simply meet those objectives, it exceeds them.

The Oracle MySQL Database Service with Heatwave Solution

The Oracle solution starts with the MySQL Enterprise Edition with Thread Pools. This is currently the highest performing MySQL version.

MySQL previously assigned a thread for every client connection. Performance drops as the number of concurrent users grows. A lot of active threads can reduce database performance considerably from the increased context switching, thread contention, and bad locality for CPU caches. Thread pools fixes this MySQL performance problem with a worker thread dynamic pool that enables active threads running on the server to be limited while minimizing thread churn. This helps prevent the server from running out of resources or crash with a memory error from connections burst.

Thread pools improve the performance of the short queries associated with OLTP. They do nothing for reporting and analytics or OLAP. That heavy lifting is performed by HeatWave, Oracle's unique massively scalable native query accelerator built into MySQL. HeatWave is a massively parallel system. It has a highly partitioned architecture that employs an in-memory engine specifically optimized for vector processing.

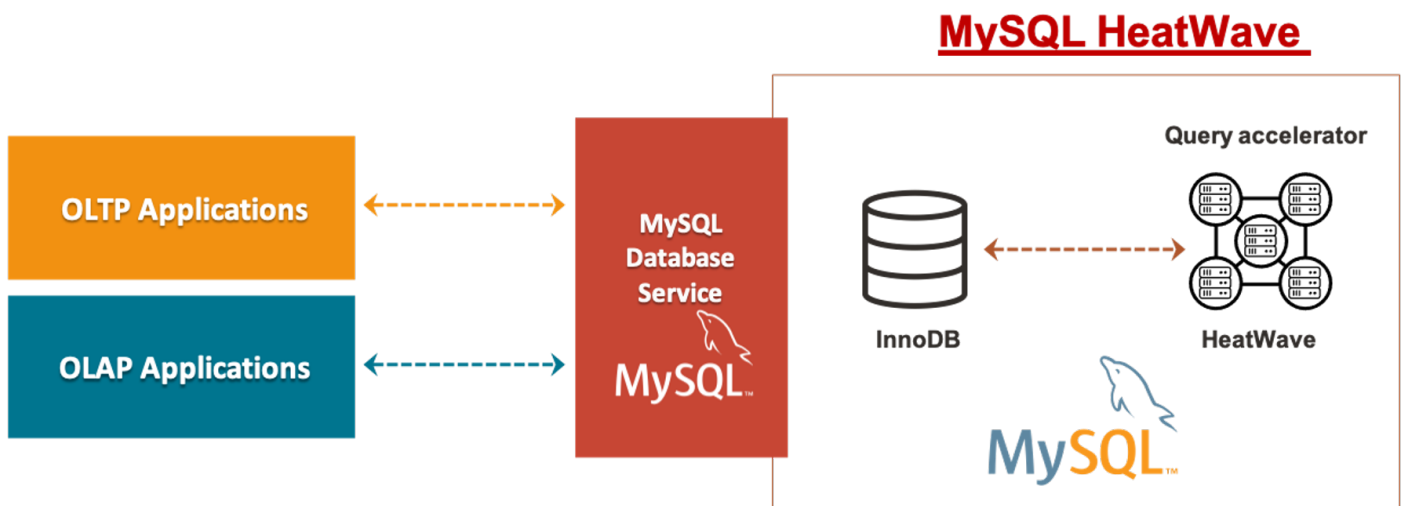


Figure 4: Oracle MySQL Database Service with HeatWave

When these are combined with columnar structure and Oracle's unique pioneering algorithms, they provide distributed joins processing with extensive overlap between compute and communications, creating

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

unmatched performance results, and massive cloud scalability at a much lower cost.

In-Memory Hybrid Columnar Processing

HeatWave's in-memory hybrid columnar processing delivers MySQL reporting and analytics in real-time.

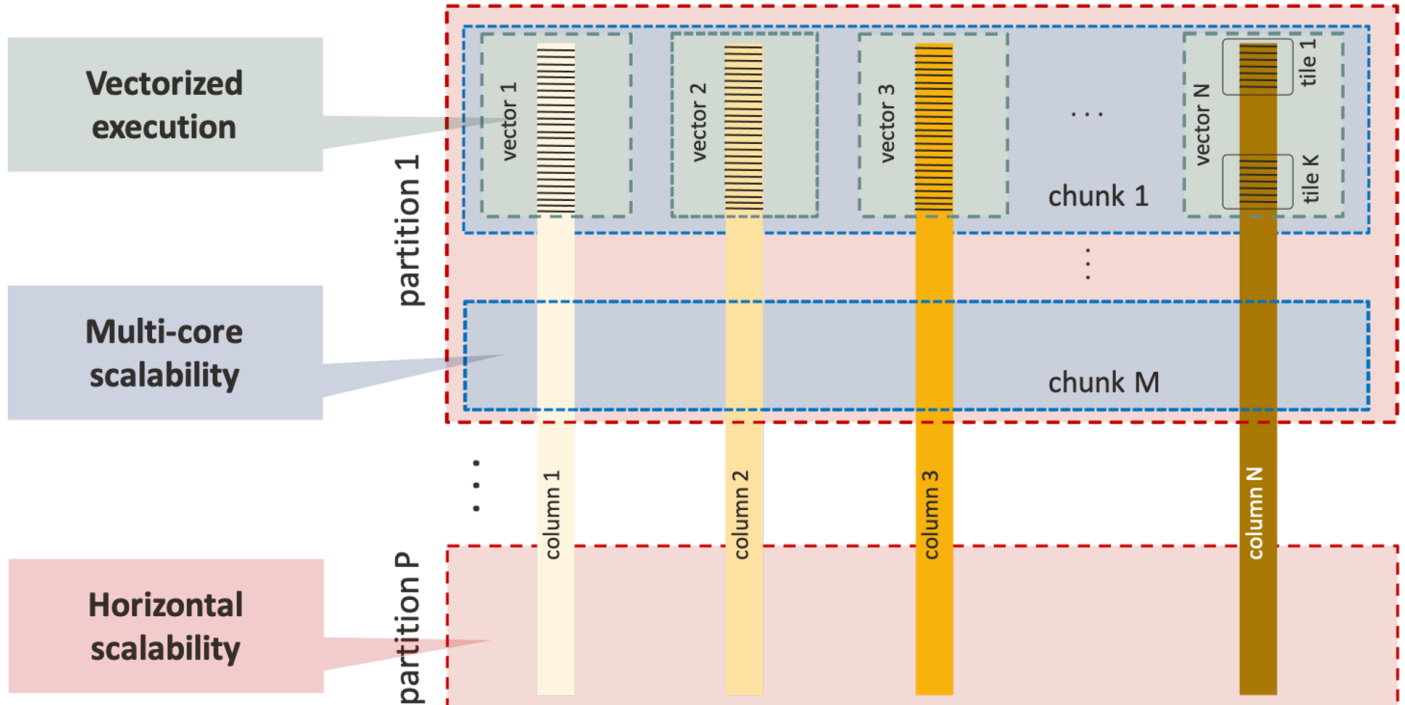
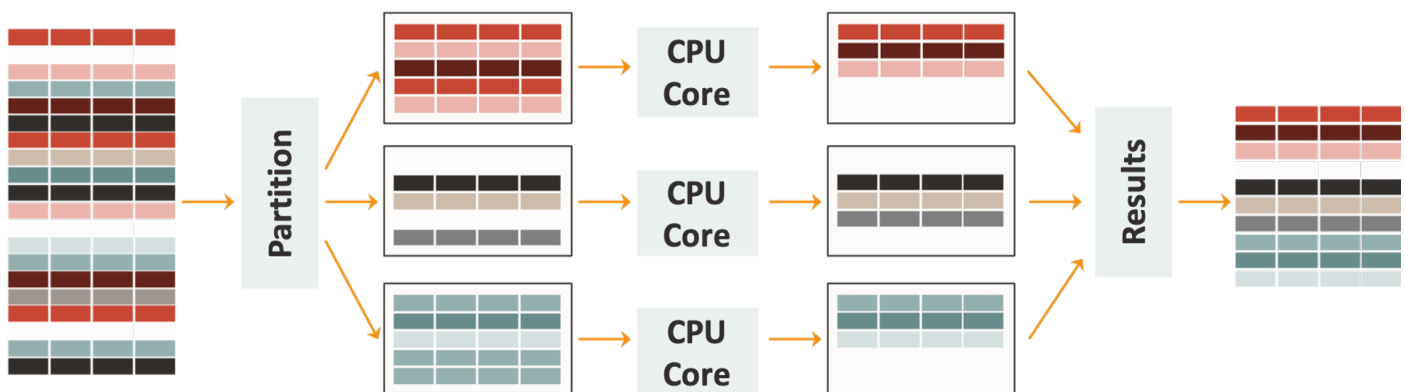


Figure 5: HeatWave In-Memory Hybrid Columnar Processing

Massively Parallel Architecture

HeatWave's combination of massively parallel architecture, high-fanout workload-aware partitioning, VMs and CPU cores doing further processing of partitioned data in parallel, optimized cache size and memory hierarchy of underlying hardware, accelerates analytical processing.



Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

Figure 6: HeatWave Massively Parallel Architecture

Distributed Algorithms Optimized for OCI

HeatWave's unique distributed algorithms partition the data to fit into the cache, partitioning at near memory bandwidth with shape specific optimizations—hash computation. It ensures partitions reside in CPU cache.

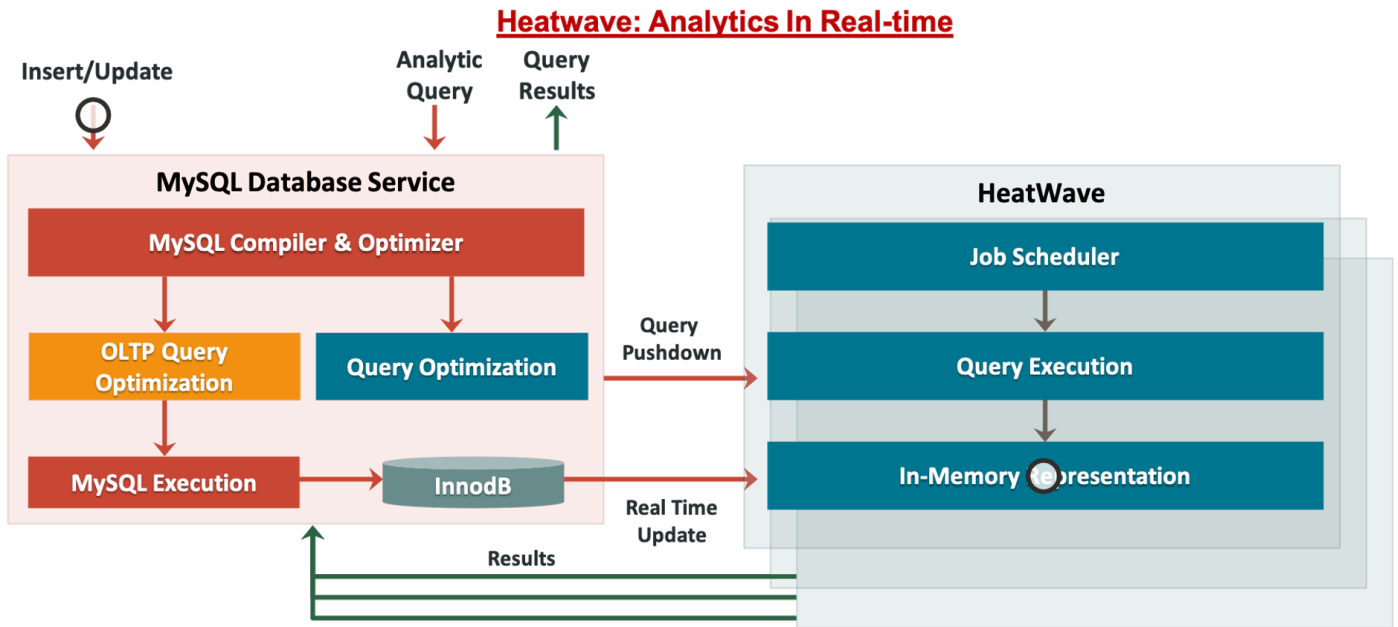


Figure 7: HeatWave Real-time Query Process

It processes partitions as fast as possible with highly vectorized build and probe join kernels, hardware-consciousness, and hand-tuned primitives for OCI—using AVX2. Another distinctive aspect of HeatWave's algorithms overlap compute with communication. HeatWave is network-optimized for the OCI interconnect while providing intelligent scheduling of execute and transfer. This combination of distributed algorithms optimized for OCI, in-memory hybrid columnar processing, and massively parallel architecture, uses real-time data and provides analytics in real-time. That in turn delivers incredibly fast time-to-actionable-insights.

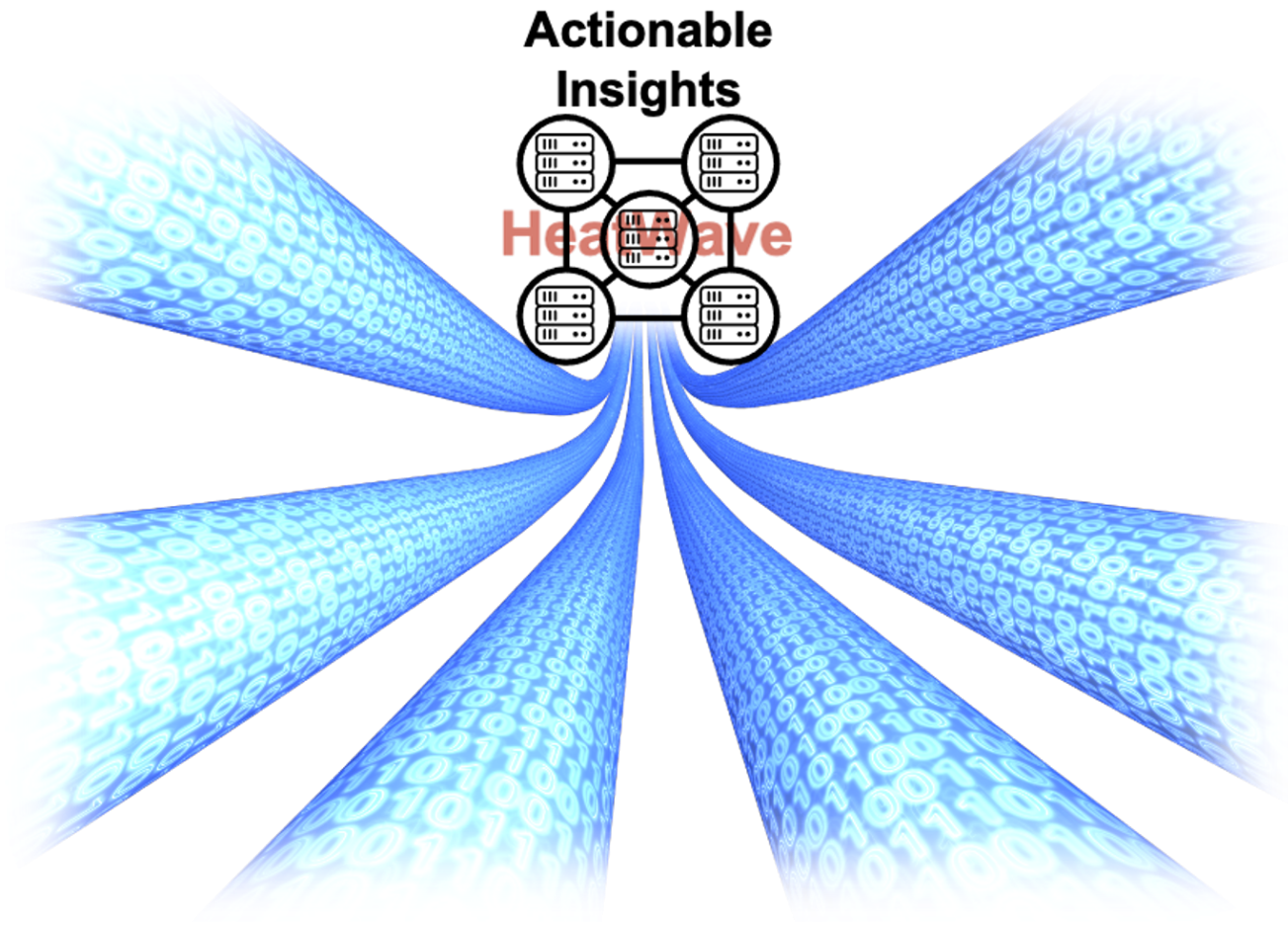


Figure 8: HeatWave Actionable Insights

Scalability

MySQL is not known for its scalability, especially when it comes to analytics. MySQL Database Service with HeatWave completely changes that paradigm. Solving the MySQL reporting and analytics problem requires being able to scale performance, cores, and capacity. MySQL Database Service with HeatWave has demonstrated linear TPC-H and TPC-DS performance from 80GB to 24TB. It's been tested up to 60 nodes, and thousands of cores. MySQL Database Service with HeatWave delivers scalability extremes.

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

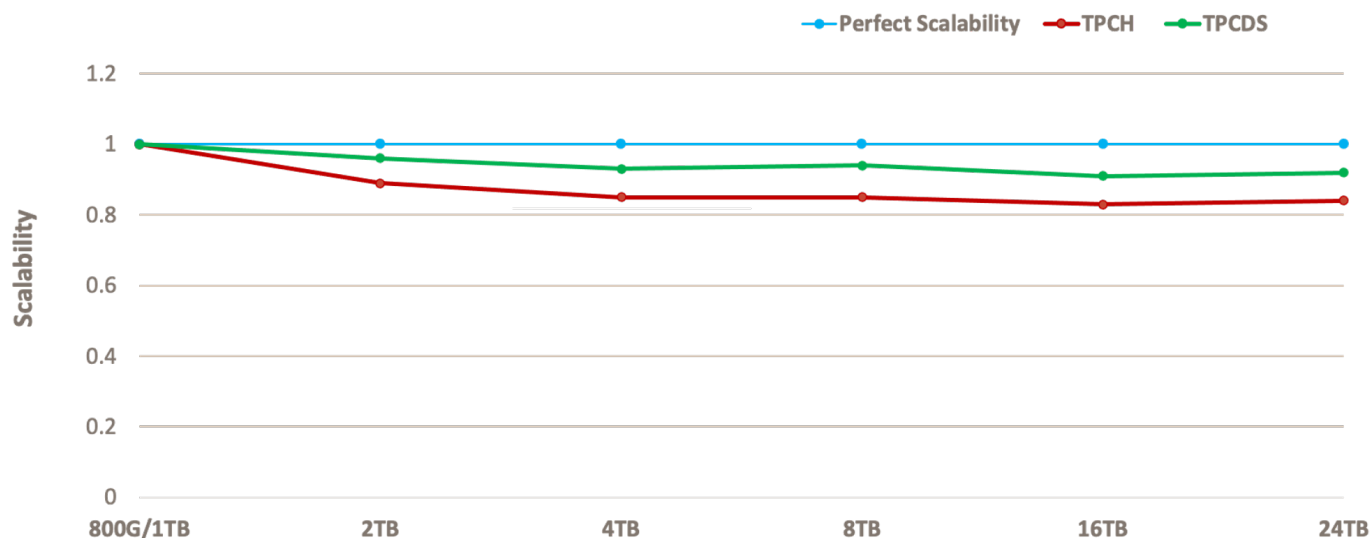


Figure 8: HeatWave TPC-H and TPCDS Linear Performance at Scale

The question then becomes how HeatWave performance compares to MySQL database cloud services and competitive dedicated stand-alone cloud data warehouses.

Comparative Cost/Performance Results

Oracle MySQL Database Service with HeatWave TPC-H[1] analytics testing, and actual customer results show a very impressive 400x[2] performance increase over MySQL databases on-premises or other MySQL database cloud services for 400GB data size. This 400x improvement in performance grows as the size of the database increases. How it compares to other database cloud services is more impressive.

Versus AWS Aurora—running in db.r5.24xlarge hardware shape—MySQL Database Service with HeatWave running in 10 E3 nodes testing with 4TB of data demonstrated that Oracle was 30x faster in preparation time, and delivers an 1100x faster GeoMean Query Run Time. Query performance was even more extraordinary. The MySQL Database Service with HeatWave advantage radically increases as the data size increases. At 256GB, MySQL Database Service with HeatWave queries were 151x faster. At 1TB, queries were 843x faster. At 4TB, queries were 1392x faster.

With that kind of performance advantage, MySQL Database Service with HeatWave should cost more. It doesn't. In fact, it costs ~ 71% less than AWS Aurora.

But for analytics the fair comparison is really versus AWS Redshift data warehouse cloud service. And once again the MySQL Cloud Database Service with HeatWave blows AWS Redshift away. Utilizing Redshift in 4*dc2.8xlarge shapes, MySQL Database Service with HeatWave was 3.7x faster and ~ 66% less expensive. When AWS instead uses a much less costly shape, such as the 2*ra3.4xlarge, MySQL Database Service with HeatWave was 17.7x faster and 3% less expensive.

What about AWS (Advance Query Accelerator for Redshift)? AQUA is the AWS high-speed scale-out cache on top of Redshift. It can process data in parallel across many AQUA nodes. AQUA uses AWS-designed analytics processors that are supposed to accelerate data compression, encryption, and data processing on queries that scan, filter, and aggregate large data sets. AWS claims customers can run queries up to 10x faster. Running the same tests, available in GitHub, showed no statistically significant difference in Redshift query performance versus Redshift with AQUA query performance. Redshift both with and without AQUA had many complex queries not complete even after 10 hours and were canceled. Furthermore, MySQL Database Service with HeatWave analyzes MySQL data in real-time. AWS Redshift cannot.

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

Results versus Snowflake are also impressive. By extrapolating the TPC-DS benchmark tests^[3] run by Fivetran –a data pipeline and ETL provider—on Snowflake and AWS Redshift, it becomes clear that MySQL Database Service with HeatWave is again noticeably faster. Snowflake ranged from approximately 2 to 3X faster than Redshift without AQUA, per Fivetran. Based on just the query processing within Snowflake, MySQL Database Service with HeatWave is at least 23 to 85% faster than Snowflake and more than 3x less expensive.

Remember, MySQL Database Service with HeatWave analyzes MySQL data in real-time. Changes made to the database are automatically available in HeatWave for analytics. Users are always working with the latest data. AWS Redshift, Redshift with AQUA, Snowflake, Azure Synapse, and Google BigQuery cannot do this. They must ETL the data from MySQL first. Users are always working with stale data.

For AWS Redshift, Redshift with AQUA, Snowflake, Azure Synapse, and Google BigQuery the real cost does not 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. The actual total cost savings for MySQL Database Service with HeatWave are considerably greater.

More Comparative Cost/Performance Results

GigaOM published TPC-H performance results for Snowflake, AWS Redshift, Azure Synapse, and Google BigQuery in October 2020. In February 2021 GigaOM published TPC-DS^[4] for these same data warehouse cloud services. In comparison, the HeatWave results advantage is staggering.

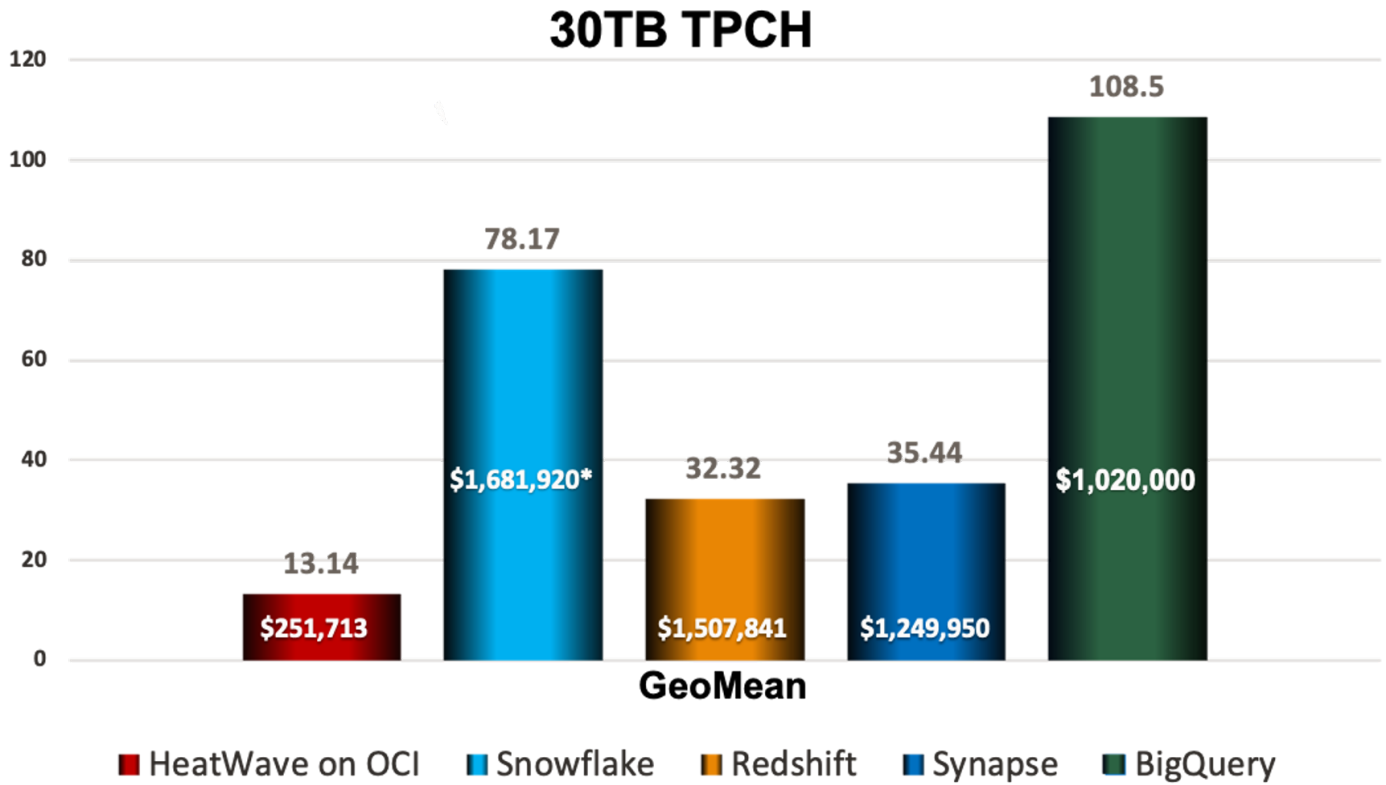
^[1] Benchmark queries are derived from TPC-H benchmark. But as previously mentioned the non-Oracle MySQL test results are technically out of compliance since they are tuned to put the competitors' best foot forward and so are not considered standardized. Oracle has put the entire test and configurations online at GitHub.

^[2] Two orders of magnitude times 4.

^[3] TPC-DS is an industry-standard benchmarking meant for data warehouses.

^[4] Comparing the performance of the 67 TPCDS queries executed on HeatWave those other services.

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

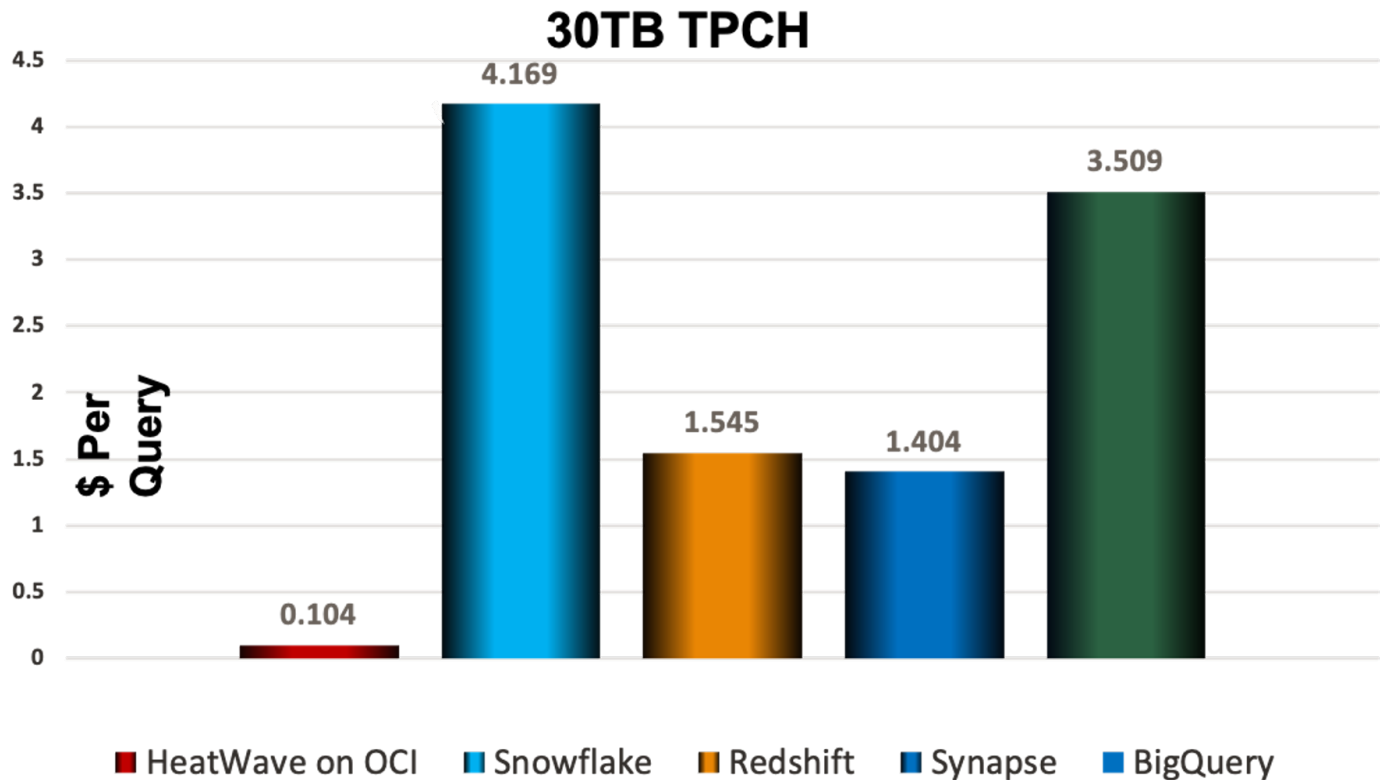


*Snowflake pricing is based on published PAYG. The rest are based on 1 year pricing

Figure 9: 30TB TPCH and TPCDS Cost/Performance Comparisons

HeatWave is obviously significantly faster at much lower cost than every one of its competitors. This becomes glaringly evident when looking at cost per query.

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ



*Snowflake pricing is based on published PAYG. The rest are based on 1 year pricing

Figure 9: 30TB TPCH and TPCDS Cost/Query Comparisons

Simply put, Snowflake, AWS Redshift, Azure Synapse, and Google BigQuery are simply not in the same league with MySQL Database Service with HeatWave in performance or cost.

How Oracle Keeps HeatWave Costs so Low

HeatWave is engineered specifically to leverage OCI commodity infrastructure. It's optimized for low-cost AMD VMs and standard Ethernet. The software has been tuned to achieve an optimal balance of compute, memory, and network bandwidth to achieve the levels of scalable performance required and that customers' production results experienced.

Scalable performance is achieved through the use of standard off-the-shelf common Ethernet. It doesn't require the more expensive RDMA networks that provide performance and delivery guarantees to increase performance. This keeps costs much lower.

Computer and network characteristics are modeled in cost functions to achieve best performance. HeatWave's exceptional performance means queries complete faster and, therefore, use much less OCPU[1] compute time. Less compute time equals lower cost.

Another key to the lower cost of MySQL Database Service with HeatWave is that it leverages the low cost OCI Object Storage without impacting performance. It does not require higher cost SSDs, block, or file storage to achieve optimum performance. OCI Object Storage improves availability and provides

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

elasticity—achieving the best reliability, scalability, bandwidth, and storage cost/performance.

One important and Oracle unique innovation in HeatWave is machine learning based AutoDB with auto provisioning. This machine learning-based advisor analyzes workload characteristics by sampling less than 0.1% of the data. AutoDB then recommends the number of servers and memory that are needed at table-level granularity, with 95% accuracy. There is currently nothing like it from any other database cloud service.

[1] OCPU is the Oracle virtual CPU. Each OCPU is a full core or twice the AWS vCPUs, which are half a core. Azure and GCP vCPUs are AWS equivalent.

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

Generate Analytics Cluster Node Count Estimate

Generates Analytics Cluster node count estimate.

[Regenerate Estimate](#)

This operation can take several minutes to complete.

Last estimate was generated on Wed, Feb 19, 2020, 17:20:12 UTC.

<input type="checkbox"/>	Name	Memory Size Estimate	Information	▼
<input checked="" type="checkbox"/>	dmm	37.23 GB	Number of Tables: 620 Number of Tables with Error comment: 82	▼
<input type="checkbox"/>	fullerton	22 MB	Number of Tables: 22	▼
<input checked="" type="checkbox"/>	sck	134 MB	Number of Tables: 134	▼

Total memory size selected: 37.36 GB

ANALYTICS NODE SHAPE

BM.Standard.E2.64

Summary

Shape: BM.Standard.E2.64
CPU Core Count: 64
Memory Size: 512 GB
Node Count: 1
Total Cluster Memory Required: 37.36 GB
Total Cluster Memory Size: 512 GB

[Apply Node Count Estimate](#) [Cancel](#)

Figure 10: Example of How HeatWave Keeps Costs Low

Availability

Oracle MySQL Database Service with HeatWave is available in every OCI region right now. It's also available from Oracle in Dedicated Region Cloud@Customer. But, what about MySQL users that want to keep their MySQL on-premises, take advantage of HeatWave, and don't need Oracle's Dedicated Region Cloud@Customer? HeatWave can also be utilized by MySQL databases on the customer's premises in a

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

hybrid configuration. The on-prem MySQL database replicates to the MySQL Database Service with HeatWave.

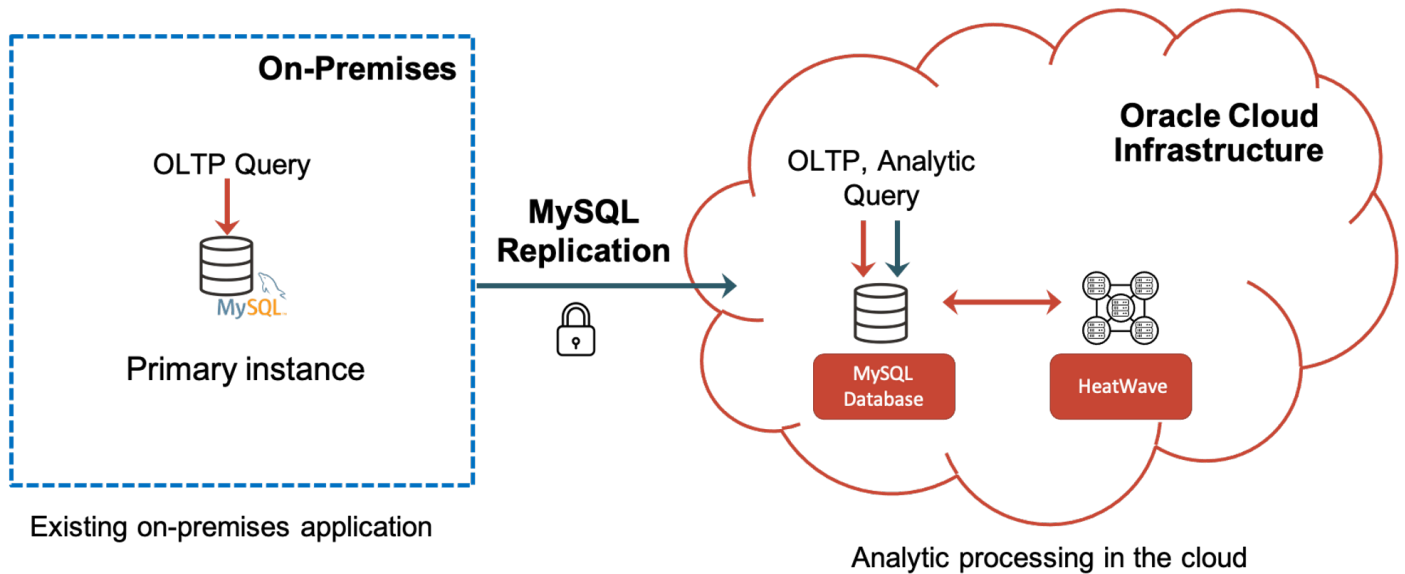


Figure 11: On-premises MySQL Utilizing OCI HeatWave

Production Proofpoints of Oracle Claims

How a product or service works in the lab is one thing. Production results in the real world are where the rubber meets the road.

Siemens Healthineers found Oracle MySQL Database Service with HeatWave reduced their query times from hours to seconds in the case of non-indexed queries. This enabled them to expand their services and opened the door to many more analytic-based capabilities while maintaining a single seamless database platform.

Square ENIX CO., LTD—the company behind the hit Final Fantasy video game franchise—ran the analytic workload of the Social Game Infrastructure group on HeatWave without requiring ETLs or changing their application. HeatWave ran up to 500x faster than their on-premises MySQL 5.7 instance.

SCSK Corporation found Oracle MySQL Cloud Service with HeatWave to be 10 times faster than the analytics service of another major cloud vendor. They no longer needed ETL. Compared to MySQL on-premises, HeatWave runs 4000 times faster.

Many customers have already gone into production with the HeatWave service. Several (like Tetris and Red3i), migrated their databases from AWS Aurora. These customers have reported significant acceleration of their workloads after moving MySQL Database Service with HeatWave while seeing their costs drop in half or more.

Conclusion

Oracle MySQL Database Service with HeatWave solves the MySQL complex queries and analytics problem. AWS Redshift with and without AQUA, Aurora, Snowflake, Azure Synapse, Google BigQuery, and everyone else, currently do not or cannot. They don't come close to HeatWave's performance and cannot match its real-time analytics capabilities. Every organization utilizing MySQL on-premises or in a public cloud, requiring reporting and analytics, should take a long hard look at the Oracle MySQL Database Service with HeatWave. It's much faster, more scalable, and less expensive than any other combination of MySQL

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

database and data warehouse.

HeatWave has fundamentally changed the MySQL game. That change completely vaporizes AWS Aurora, Redshift with and without AQUA, Azure Synapse, and Google BigQuery.

For More Information on the Oracle MySQL Database Service with HeatWave

Go to: [HeatWave](#)

Oracle MySQL Database Service with HeatWave Vaporizes AWS Redshift AQUA, Snowflake, Azure Synapse, & GCP BQ

President and CDS of Dragon Slayer Consulting in Beaverton OR since 1998, is renown for his in-depth and keen understanding of user problems, especially with storage, networking, applications, cloud services, data protection, and virtualization. Marc has published thousands of technology articles and tips from the user perspective for internationally renowned online trades including many of TechTarget's Searchxxx.com websites, Wikibon, Network Computing, and GigaOM. Marc has additionally delivered hundreds of white papers, webinars, and seminars to many well-known industry giants such as: Brocade, Cisco, DELL, EMC, Emulex (Avago), HDS, HPE, LSI (Avago), Mellanox, NEC, NetApp, Oracle, QLogic, SanDisk, Toshiba, and Western Digital. He has additionally provided similar services to smaller, less well-known vendors/startups including: Asigra, BrainChip, Cloudtenna, Clustrix, ConduSiv, DH2i, Diablo, FalconStor, Gridstore, ioFABRIC, Nexenta, Neuxpower, NetEx, NoviFlow, Pavilion Data, Permabit, Qumulo, SBDS, StorONE, Tegile, and many more. His speaking engagements are always well attended, often standing room only, because of the pragmatic, immediately useful information provided. Marc can be reached at marcstaimer@me.com, (503)-312-2167, in Beaverton OR, 97007.

Marc Staimer

marcstaimer@me.com