HeatWave Turbocharges MySQL Database Service

Issue 1

Gartner®

HeatWave Turbocharges MySQL Database Service

As per Gartner, data and analytics leaders must decide the extent to which they will use multiple specialized DBMSs. In the cloud, rich provider portfolios may complicate this choice. A single multimodel DBMS may be more manageable and cost-effective, and suitable for a large percentage of use cases.

Key Challenges

- Developers are increasingly challenged to choose between specialty engines and existing multimodel DBMSs to match their target use cases — especially for use as services for cloud deployment.
- Organizations often find that they need leading multimodel DBMSs, which have significant capabilities for specialty use cases, and offer advantages of their own, including the ability to use existing skills.
- Although specialty engines can add well-matched functional capabilities and tools, integration with other data and applications required for the use case may introduce complexity.

Recommendations

Data and analytics leaders looking to optimize their data management choices should:

- Identify the capabilities of the organization's predominant, strategic DBMSs (many are multimodel) and familiarize the organization with these features by creating a feature and capability reference chart. Use this as a baseline when considering new projects.
- Assess the perceived advantages of any specialty engine by comparing it to the capabilities and roadmap of existing strategic multimodel products that are in use.
- Develop or extend your process for classifying potential projects by mapping them to DBMS technologies, while assessing all the associated costs.
- Offer support to developers who are outside formal IT processes by engaging them and providing a place to test alternatives that promote a modern application architecture — using inexpensive cloud-based utilities where possible.

MySQL Database Service with HeatWave: Single database for OLTP and real-time analytics and accelerated performance

MySQL is the world's most popular open-source database used by enterprises large and small. MySQL is optimized for OLTP and many enterprises use a specialized analytic database for running their analytic workloads which requires them to ETL the data from the MySQL database into the analytic database. The ETL process and managing two databases introduces cost, complexity, security vulnerability and leads to data in the analytic database being out of sync with the MySQL database.

HeatWave is a new, tightly integrated, high-performance query accelerator for MySQL Database Service, providing a single database platform for transactional and analytic workloads. In fact, MySQL Database Service with HeatWave is the only service that enables database developers to run OLTP and OLAP workloads directly from their MySQL database, eliminating the need for complex, time-consuming, and expensive ETL processes to move data to a separate analytics database.

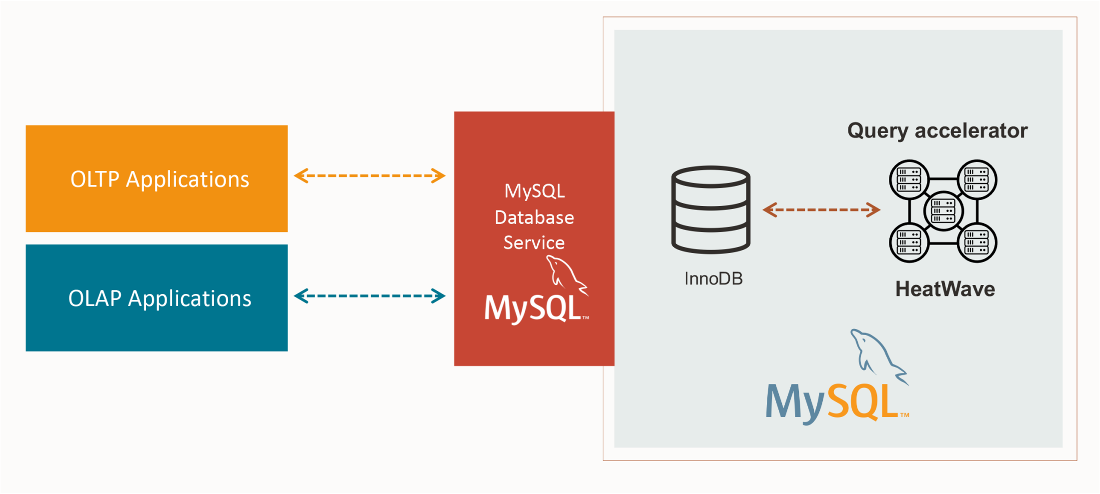
Analytic and complex OLTP queries are transparently accelerated by HeatWave without any intervention by the user or changes to the application. If there are any changes made to the MySQL database, those changes are transparently available in real-time. This ensures that queries are always accessing the latest data. To top it off, HeatWave accelerates MySQL performance by over two orders of magnitude for analytics and complex transactional queries and scales out to thousands of cores.

> "We have a mixed workload of OLTP and analytic queries. By migrating to HeatWave, our query times reduced from minutes to milliseconds. Migration was seamless, since there is almost no change to our application needed. And HeatWave reduced our cost by half from our previous provider."
>
> Pablo Lemos
> Co-Founder
> Tetris.co

Detailed Performance Comparison

Performance Comparison Details

FIGURE 1



Source: Oracle

Designed for the cloud

HeatWave has been designed from the ground up with the cloud architecture in mind to ensure the best performance, massive scalability and lowest cost. Data is stored in memory in a hybrid columnar format and is optimized for vector processing. HeatWave has a massively parallel architecture which is enabled by a massively partitioned architecture. Data is partitioned at near memory bandwidth and fits in cache size of the underlying shape. These partitions are processed very efficiently by new state of the art distributed query processing algorithms which have been developed at Oracle. Furthermore, HeatWave has an intelligent scheduler which overlaps computation time with inter-node network communication time to provide great scalability across servers.

Machine learning-based automation

HeatWave uses machine learning to intelligently automate various operations. Auto Provisioning is one example that uses machine learning-based automation to predict the number of HeatWave nodes needed to run a given workload. The number of nodes needed depends upon the size of the data and its characteristics. Other vendors require manual estimation of cluster size, which often results in an inaccurate provisioning of resources. Auto Provisioning provides a highly accurate prediction of the memory usage, which is then used to predict the cluster size.

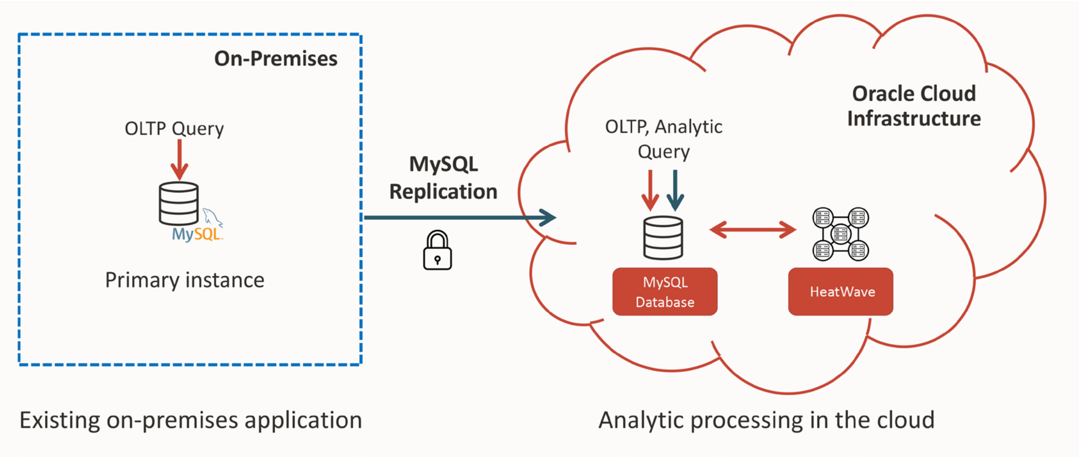
Hybrid deployment with no change to MySQL applications

No changes are required to existing applications to use HeatWave. Customers can use standard MySQL replication between their on-premises database and the MySQL Database Service with HeatWave in the cloud. This enables on-premises databases to leverage HeatWave for query acceleration.

"The performance of MySQL HeatWave is over 20x faster than the analytics service of one of the largest cloud vendors, and accelerates our analytics workload by 45,000x compared to MySQL."

Kenichi Sasaki
SRE Team of Japan
Mercari Inc.

FIGURE 2



Source: Oracle

More Resources

[Visit the MySQL HeatWave product page]

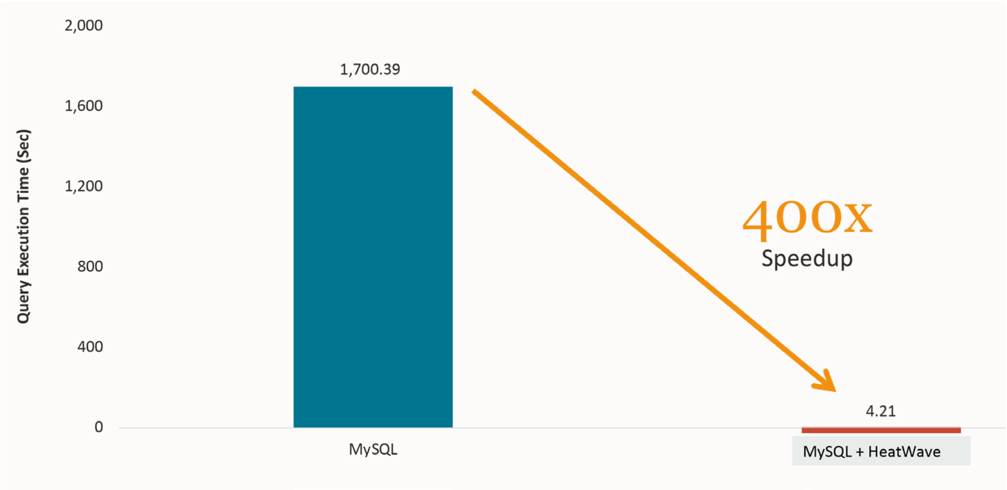
[Getting Started with MySQL HeatWave]

[HeatWave – Technical Brief]

Heatwave accelerates MySQL queries by orders of magnitude

HeatWave, a highly scalable, in-memory query accelerator for the MySQL Database Service, provides significant acceleration for both analytic and complex OLTP queries without the need to move the data out of the MySQL database or make changes to applications. In industry-standard benchmarks HeatWave accelerates the TPC-H workload with 400GB of data by 400x compared to MySQL 8.0 deployed on-premises. The performance improvement with HeatWave increases with the size of the database.

FIGURE 3



Source: Oracle

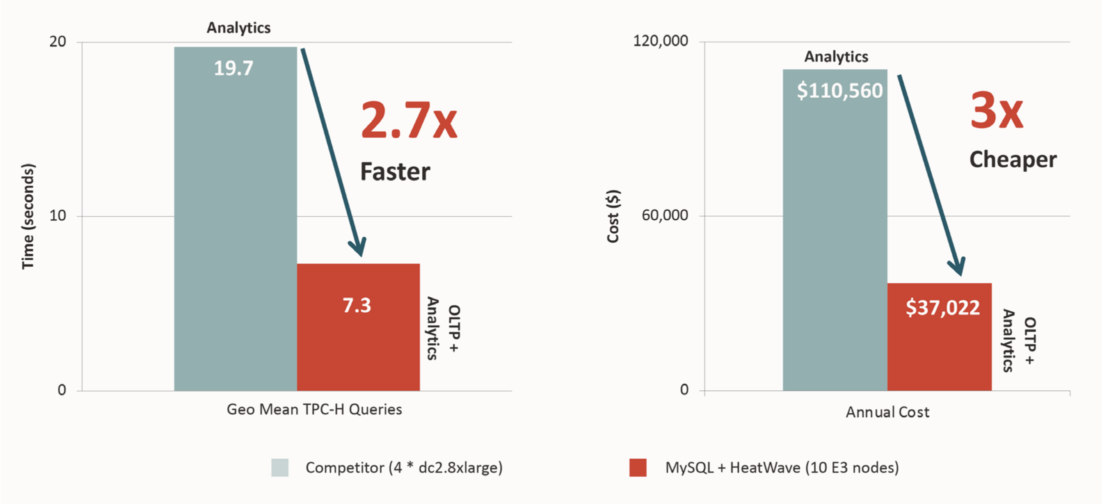
Much faster than any other analytics service

HeatWave is much faster than other cloud database services as demonstrated by multiple industry benchmarks: TPC-H and TPC-DS. In addition, customers using HeatWave have experienced much better performance compared to other services. For example, SCSK Corporation of Japan indicated that for their workload "HeatWave is 10x faster than the analytics service of another major cloud vendor and compared to MySQL, HeatWave is 4000x faster."

Much cheaper than any other analytics cloud service

HeatWave is 3x cheaper than any other cloud service. According to Red3i, a telecommunications company, Red3i's spend dropped to 40% by migrating from AWS Aurora to MySQL HeatWave.

FIGURE 4



Source: Oracle

Learn more about MySQL Database Service with HeatWave

Discover our featured events with the latest announcements, customer conversations, product-specific insights, and hands-on technical sessions.

Go to the MySQL page

Source: Oracle

Research from Gartner:

Choosing Between Multimodel DBMS and Multiple Specialized Engines

Data and analytics leaders must decide the extent to which they will use multiple specialized DBMSs. In the cloud, rich provider portfolios may complicate this choice. A single multimodel DBMS may be more manageable and cost-effective, and suitable for a large percentage of use cases.

Overview

Key Challenges

- Developers are increasingly challenged to choose between specialty engines and existing multimodel DBMSs to match their target use cases — especially for use as services for cloud deployment.

- Organizations often find that they need leading multimodel DBMSs, which have significant capabilities for specialty use cases, and offer advantages of their own, including the ability to use existing skills.

- Although specialty engines can add well-matched functional capabilities and tools, integration with other data and applications required for the use case may introduce complexity.

Recommendations

Data and analytics leaders looking to optimize their data management choices should:

- Identify the capabilities of the organization's predominant, strategic DBMSs (many are multimodel) and familiarize the organization with these features by creating a feature and capability reference chart. Use this as a baseline when considering new projects.

- Assess the perceived advantages of any specialty engine by comparing it to the capabilities and roadmap of existing strategic multimodel products that are in use.

- Develop or extend your process for classifying potential projects by mapping them to DBMS technologies, while assessing all the associated costs.

- Offer support to developers who are outside formal IT processes by engaging them and providing a place to test alternatives that promote a modern application architecture — using inexpensive cloud-based utilities where possible.

Introduction

The growing deployment of portfolios of specialty DBMS engines (for example, for document, graph, time-series, ledger/blockchain, key-value, log analysis and other contexts) has created opportunities and challenges for architects and developers. These new specialty engines are already gaining substantial acceptance. For example, of respondent organizations to Gartner's 2019 Data and Analytics Adoption Trends Survey, 28% indicated they are already using graph DBMS, and 23% are already using time series DBMS. In both instances, planned use of these technologies within the next 12 to 24 months is over 40%.[1]

The need for agility and rapid delivery should be balanced with sound cost consciousness, awareness of the ROI, and an assessment of the organization's readiness for the integration of new applications with existing infrastructure. Many traditional DBMSs have evolved to become multimodel, supporting a unified database for different types of data. They are designed to support multiple data models against a single, integrated back end. Some organizations choose to limit new expansions to their multimodel products that are already in use to avoid lock-in. Specialty engines may be well-designed for the specific requirements of one part of a complex business process. However, their ability to integrate with existing parallel, upstream and downstream systems may not yet be mature. The difficulty of operating and managing such a portfolio in a workflow with those systems — and ensuring that its policy compliance, security and resilience can be managed together with those in the products used — must be factored into the choice.

Building an assessment framework that ensures that these issues are considered will prevent expensive redesign and remediation as new requirements emerge, or as existing connected systems change. The framework will also prepare the organization to adapt as new governance, compliance and security demands that affect all related systems emerge. Excessive bureaucracy is never a welcome addition, so building assessment capabilities without alienating designers and developers is essential. Approval processes that are perceived to be slow and inflexible will be circumvented, and the ease of deployment in the cloud makes it difficult to ensure compliance. Without an easily understood process that adds value and demonstrably eliminates wasted work, excessive cost and complexity, it will be difficult to enforce governance and standards while optimizing cost-effective deployment.

Analysis

Audit the Specialized Capabilities of Multimodel Engines Already in Use

Gartner inquiries about alternative DBMSs frequently reveal that the capabilities of existing DBMSs in wide use are not well-understood. This is particularly true for leading relational database management systems (RDBMSs), all of which have expanded their scope as multimodel offerings. Leading mature multimodel DBMSs contain significant functionality that may go untested and unused. It is likely that you are not certain of all the capabilities of your strategic DBMS platform(s). For example, document stores typically rely on JSON, and developers may demand a DBMS that supports it. Many RDBMSs — such as IBM Db2, Microsoft SQL Server, and Oracle DBMS — already have JSON support.

To help get the right fit, develop a more specific description of the requirements for your evaluation. For some application needs, the ability to add columns that contain documents formatted as JSON text, to parse and import JSON documents, and format relational data to JSON text may be sufficient. This may also preclude using an additional document engine.

Similarly, graph capabilities are present in Microsoft SQL Server, Oracle, Redis Enterprise and SAP HANA. Redis Enterprise and Teradata support time series in addition to their core engines, as do Amazon DynamoDB and Microsoft Azure CosmosDB for key-value capabilities.

Use the vendors (and Gartner's related research) to develop your understanding of what is possible. Vendors will see this as an opportunity to expand usage. They may offer an introduction, an assessment, design and conversion utilities, training, or help with a first application — especially if you let them know that alternatives are under consideration. Create a chart identifying what is already installed or available from vendors in use. Table 1 provides an illustrative example. Consult your vendors for a full list of offerings.

If resources are available, test the multimodel product even if a specialty engine seems to be the expected platform. Use the experience to determine how comparable the performance, scalability and other characteristics are. Note that other benefits of multimodel offerings may include reduced costs, less required training and increased skills acquisition.

Compare Specialty Engine Capabilities to Existing Multimodel Products

Choosing a specialized engine, especially from a portfolio of services in the cloud, can be effective where organizational capacity and interest in developing new skills are high. It may also be effective in instances where the use case pushes or even breaches the boundaries of the multimodel DBMS's functional capabilities. "Force-fitting" specialized uses into multimodel DBMSs can create complex development and design challenges. Thoroughly evaluate the capabilities of the specialty engine being proposed, especially if it offers features absent or not easily implemented in your multimodel alternative. Map any perceived advantages to the requirements driving the request for a specialized engine. Consider the future growth or enhancements that are likely once the engine is deployed.

Capability advantages will often be present for graph or time series use cases. Built-in capabilities for design and visualization in a graph DBMS, for example, may be far superior to "bolted on" added syntax or execution in a multimodel product.

Table 1: DBMS Engines Installed or Available From Vendors in Use

	AWS	IBM Cloud	Microsoft	Oracle
Row Store	RDS, Aurora	Db2 on Cloud, IBM Cloud Database for PostgreSQL, IBM Cloud Database for EnterpriseDB	SQL Server, Azure SQL	Oracle Database, MySQL, TimesTen
Column Store	RDS, EMR, Redshift	Db2 Warehouse on Cloud, SQL Query, Netezza Performance Server	SQL Server, Azure SQL, Azure Synapse Analytics	Oracle Database, MySQL, Essbase, TimesTen
Wide Column	DynamoDB, Managed Cassandra		Azure Cosmos DB, Azure Table Storage	Oracle NoSQL
Document Store	DynamoDB, DocumentDB	Cloudant, IBM Cloud Database for MongoDB	Azure Cosmos DB	Oracle Database, MySQL, NoSQL
Graph Store	Neptune		Azure Cosmos DB	Oracle Database, NoSQL
Time Series	Timestream Timestream	SQL Query, Informix	Azure Time Series Insights	Oracle Database, NoSQL, TimesTen
Key-Value	DynamoDB	IBM Cloud Database for Redis	Azure Cosmos DB, Azure Cache for Redis	Oracle Database, MySQL, NoSQL, Berkeley DB
Ledger	Quantum Ledger	Blockchain Platform		Oracle Database

Source: Gartner (August 2020)

Adding syntax or execution in a multimodel product may require internally developed, complex code that must be maintained by staff (who may not have the requisite skills), without tools that the specialty product provides. Similarly, performance can be a significant challenge in time series cases — where multiple windows need to be compared across time streams, for example. Creating extracts, constructing temporary datasets and building complex logic may be daunting — but specialty products already have functions for those purposes.

In many cases (though not all), the decision to use a specialty engine or a multimodel DBMS will align to selecting a native cloud service provider (CSP) offering or an independent software vendor (ISV) offering that runs on CSP infrastructure. Specialty engine-based portfolios are fundamental to the way many CSPs go to market. Amazon Web Services (AWS) and Google Cloud Platform (GCP) both predominantly use specialty engine approaches for their DBMS offerings. Multimodel options are more prevalent in Microsoft Azure, with offerings like Cosmos DB, Azure SQL Database and Synapse Analytics — though Synapse can be viewed as an amalgamation of speciality engines. DBMS offerings with robust multimodel capabilities often come from ISVs. This requires data and analytics leaders to weigh the trade-offs between consolidated and potentially simplified DBMS landscapes, and the potential increased overhead involved in integrating an ISV offering with the broader CSP environment on which it runs. Some leading vendors are hedging their bets by offering both specialized engines and multimodel offerings with overlapping functionality. When discussing options with vendors, data and analytics leaders should press them on which option is preferred for the concerned use case. Bear in mind that sales incentives may be in place for new products the vendor is launching, and assess the vendor's advice accordingly.

Even where specialty engines demonstrate some advantages, recognize that they also create additional requirements for effective integrations. Often, data must be moved to or duplicated in the new platform, creating synchronization challenges and additional processes that must run correctly. This creates dependencies and costs that must be considered, along with the skills and support that these dependencies demand.

Classify Projects and Map Them to DBMS Technologies

The relationship between data types, use cases and preferred engines can be subtle. Many engines can handle types of information for which specialized engines exist — the issue for developers is not whether the DBMS in use is capable of working with that data, but whether an additional engine is required to handle application requirements.

A common example is the use of "JSON data" — a phrase which can mean many things, but does not refer to a language or even a fixed format of any kind. JavaScript Object Notation is a way to mark up textual content for self-description — no more, no less. Document DBMSs add JSON-oriented operators for typical tasks. Storing, retrieving and computing with JSON data is well within the capabilities of many types of DBMSs. Its support for text, arrays and objects may meet the specific requirements of an application but not demand a specialized engine. Some use cases depend on combining JSON data with a very large amount of row-and-column data in an RDBMS — or with the nodes in a complex graph stored in a graph DBMS. In such instances, those other engines may be capable of delivering integration, performance and scale better than a document DBMS engine.

Similar comments can be applied to the use of time series information. For example, there are highly specialized functions available with time series DBMS engines that dramatically simplify the execution of comparisons across moving windows within multiple time streams. Such activities are challenging to program in other DBMSs and the resulting code is likely to be difficult to maintain — and even incomprehensible to those not experienced with the constructs.

Thus, the classification of projects should not be based solely on "what data type," but also on "what functions need to be executed." Statements such as "combining geospatial data with time-based sensor readings to compare the productivity of land management experiments over multiyear tests" are more meaningful than "key-value sensor readings and map data." Ensure that the business value of the specified requirements is sufficiently clear to support decisions that are required to "break a tie."

Where possible, be clear from the outset on your requirements for price/performance, especially where high concurrency is expected. Define the following:

- Security constraints and requirements.
- Expectations for development and management tools and utilities.
- The expected deployment model.
- The desired level of support for existing tools in use for BI, data science and machine learning.

Integration points with existing systems, and any migration or exchange of data with such systems, can be a significant challenge and should be noted.

As with any classification scheme, the following attributes should be included:

- Data volume
- Growth rate
- Sensitivity
- Security requirements

Data and analytics leaders should also consider:

- Concurrent usage demands
- Likely scalability requirements with attention to peaks that can and cannot be predicted
- Programming language support
- Integration needs for relating to the existing environment

See "Toolkit: RFP Template for Operational DBMS" for an extensible framework that you can adapt for this exercise.

Do not neglect the skills required for the use case. A multimodel DBMS may address the technology requirements, but offer very little in the way of design tools and training — which speciality DBMSs tend to build early in their market development.

Provide a Safe Space for Developers to Build and Test Alternatives

Nothing stifles innovation as much as the difficulties of getting projects into production. In a recent Gartner survey, data management teams reported spending 56% of their time on production initiatives, but only 22% of their time on innovation (see "Survey Analysis: Data Management Struggles to Balance Innovation and Control").[2] Supporting new functional requirements depends on establishing alignment between data management projects and business value. As the number of siloed projects increases, it becomes challenging for teams to continue delivering, while also maintaining existing projects. Ensure that those projects that make a leap to extended uses for existing products — or trialing entirely new ones — have support to experiment and fail.

In this context, a failed experiment is a result that helps define the boundaries of future experiments and the solution space they will operate in. Ensure that the reasons for the failure are clear — they may point to a different use for which the product that fell short is ideally suited. Scalability challenges are not the same as functional inadequacy — the latter may be permanently disqualifying for this type of project, while the former may simply suggest limits. The point is that even failures provide valuable insight and should be documented.

Make use of the cloud — and in particular, managed services — a priority if the corporate culture permits. This can eliminate much of the mechanical production work noted above. At the same time, it can impose new challenges if integration must span a hybrid environment, so keep the ultimate deployment target firmly in mind. An exciting new cloud-only offering may provide an excellent sandbox that ultimately has no practical relevance if deployment must be on-premises.

Where possible, test more than one alternative DBMS at the same time and compare the results. Assess:

- Time to delivery
- Support (if needed) from the vendor to overcome hurdles
- Quality of training if used
- Cost

Measure all-in costs — additional training may be required to get to full-scale production. New licenses from a new vendor create additional costs and if the tests show a high degree of difficulty, that increases the likely future costs as well. Engage the developers in assessing these issues — nothing reduces the risks of such projects for participants as much as being part of the evaluation team. It means they are entrusted with a key part of a decision.

Finally, be clear that the comparison is intended to be an honest one. Do not load the dice in favor of staying with the multimodel products that are already in place simply because that seems less challenging and less risky. Clearly articulate the perceived value of finding new, more cost-effective alternatives for the future of the enterprise, while guarding against change for its own sake.

Evidence

[1] Gartner's Data and Analytics Adoption Survey 2019 was conducted to learn how organizations use data and analytics.

The research was conducted online during November and December 2019 among 272 respondents from North America, Western Europe, and APAC. Companies from different industries were screened for having annual revenue less than $100 million.

Respondents were required to be at a managerial level or above and have a primary involvement in or be responsible for the organization's data and analytics solutions, including purchase and investments.

The study was developed collaboratively by Gartner analysts and the Primary Research Team.

[2] Gartner's Data Management Strategy Survey was conducted via an online survey from 19 August through 4 September 2019 with 129 Gartner Research Circle Members — a Gartner-managed panel. The survey was developed collaboratively by a team of Gartner analysts and was reviewed, tested, and administered by Gartner's Research Data and Analytics team.

This research is also informed by research data from Magic Quadrant and Critical Capabilities surveys; Research Circle data; vendor briefings on products; and inquiries from clients exploring new offerings.