Guide to Open Source Database Selection

WHITE PAPER

MySQL vs. MariaDB

MariaDB®

Table of Contents

Executive Summary

Often driven by technological needs and familiarity to developers and administrators, deciding on a database can be influenced by past experiences and decisions. The goal of this document is to evaluate MariaDB and MySQL side by side to better inform the decision-making process. Whether to embark on a greenfield project with no database in place yet, or running MySQL in production and considering an upgrade to MariaDB. The two will be discussed through the following topics, summarized below:

	MySQL	MariaDB
Compatibility – Protocols	MySQL	MySQL + MariaDB
Community – Source Code	Open Source + Proprietary	Open Source
Community – Development	Closed	Open
Community – Collaboration	Limited	Extensive
Core – Replication	MySQL Replication	MariaDB Replication
Core – Routing	MySQL Router	MariaDB MaxScale
Core – Partitioning	MySQL Partitioning	MariaDB Partitioning
Tool – Editing	MySQL Workbench	SQLyog
Tool – Monitoring	MySQL Enterprise Monitor	Monyog
Tool – Backup	MySQL Enterprise Backup	MariaDB Backup
Scalability – Client Connections	MySQL Enterprise Thread Pool	MariaDB Thread Pool
Scalability – Clustering	MySQL Group Replication	MariaDB Enterprise Cluster
Security – Encryption	MySQL Enterprise Encryption	MariaDB Encryption
Security – Data Masking	No	MariaDB MaxScale
Security – Firewall	MySQL Enterprise Firewall	MariaDB MaxScale
Security – Auditing	MySQL Enterprise Audit	MariaDB Audit
Analytics	No	MariaDB ColumnStore
SQL – Common Table Expressions	In-development (MySQL 8.0)	Yes
SQL – Window Functions	In-development (MySQL 8.0)	Yes
Temporal – Log-based Rollback	No	Yes
Temporal – System Versioned Tables	No	In-development (MariaDB 10.3)
JSON – Functions	21	26

Open Community

MariaDB embraces open source, encouraging community participation and innovation, difficult with MySQL's increasingly closed development.

From its inception, MariaDB has been committed to open development and transparency. Alternatively, Oracle has been cutting access to the development cycle and incorporating closed-source features. In addition, due to potential cannibalization of Oracle Database, feature development in MySQL has been limited. Seeing proprietary code and lack of access to basic test cases as a threat to a healthy ecosystem, the creators of MySQL took action. They created a fully open source database, MariaDB, and began fostering a community again. Full, public access to new features and test cases ensures users and administrators can confidently track issues and progress. In the open source model embraced by MariaDB, information flows both ways. MariaDB has incorporated contributions from the community, including encryption from Google and a storage engine from Facebook. The open source community welcomed this approach, which led MariaDB to become the default database in leading Linux distributions – including Red Hat Enterprise Linux, SUSE and Ubuntu.

A Shared Core Engine

MariaDB is a drop-in replacement for MySQL, so the risk of upgrading is minimal.

MariaDB and MySQL have a shared history, forming a common foundation between the two. Originally built on MySQL, MariaDB inherited many of the core behaviors and interfaces of MySQL. Developers familiar with interacting with MySQL won't have to change how they connect and use the database when swapping out MySQL for MariaDB.

For application developers, the connection to MariaDB is identical to MySQL, meaning MySQL clients can connect to MariaDB without knowing MySQL has been replaced. On the server, you simply uninstall MySQL then install and start the equivalent version of MariaDB. The procedure is almost identical to upgrading MySQL, which makes the swap feel like just another upgrade.

Extensibility

MariaDB is built on an extensible architecture, allowing each layer of architecture to be extended to fine-tune the database and better support a wide range of use cases.

In order to extend the vision of open source and leverage strong community innovation, MariaDB modernized the architecture by adding new components in every layer.

One of the most important elements of any database is the storage engine used to read and write data. Different use cases have different data access patterns. Thus, using the right storage engine for the right job can make all the difference in the world when it comes to performance. Each storage engine fits a particular use case. For example, use MyRocks, a SQL-based key-value store workload built on RocksDB, when high performance and compression is required.

Analytics

MariaDB introduced scalable, high-performance analytics with MariaDB ColumnStore, allowing developers to use SQL for both OLTP and OLAP.

MariaDB ColumnStore supports scalable, high-performance analytical use cases using distributed, columnar storage and ANSI SQL. MariaDB ColumnStore is designed as a pluggable storage engine purpose-built to handle analytic workloads while keeping the same ANSI SQL interface that is used across all the storage engines of the MariaDB portfolio. With massive parallel processing (MPP) architecture, MariaDB ColumnStore runs queries in parallel across all nodes. For complex analytics, MariaDB ColumnStore supports queries like complex joins, aggregation, window functions and user-defined functions. MySQL does not offer any analytics feature in its portfolio.

High Availability

MariaDB offers a clear route to multi-master clustering and high availability.

Serving production traffic to a database is critical to a company's line of business and any outage has a considerable cost every moment the database is down. Not only can the cost be measured in dollars lost from transactions that couldn't be completed, but also in customer trust and perception of reliability. Clustering and dependable failover is at the heart of high availability for any database. MySQL offers over half a dozen different overlapping approaches[1] and applications to resolve parts of this problem, most requiring separate installations and having slightly different purposes. By comparison, MariaDB provides three built-in solutions, one of which is a multi-master clustering solution, Galera Cluster, that can be easily configured to provide multiple, highly available MariaDB nodes capable of accepting both reads and writes.

Security

MariaDB includes more comprehensive encryption and auditing than MySQL. Attack mitigation with MariaDB MaxScale is more flexible and configurable than MySQL Enterprise Firewall.

Digital security is paramount to success, providing a solid foundation of reliability and trust between a business and its customers. Three key elements of security are encryption, attack prevention and auditing.

Encryption

Encryption can be further broken down into two contexts: data in motion and data at rest. Data in motion consists of any active communication between two servers or between a client and a server, including all requests to the database from application servers. Both MariaDB and MySQL encrypt data in motion using Transport Layer Security (TLS), a secure and reliable cryptographic protocol.

Data at rest consists of information that is stored to the disk drive. Both MariaDB and MySQL offer encryption of files storing InnoDB tables. Beyond MySQL, MariaDB also offers full encryption using the XtraDB storage engine.

[1] Replication, Group Replication, ZFS Replication, MySQL Fabric, MySQL Cluster, Oracle MySQL Cloud Service, Oracle Clusterware Agent for MySQL, MySQL with Solaris Cluster

MariaDB also supports the encryption of the binlog, undo/redo log, relay log, and temp logs and using simple command line flags or configuration settings.

Attack Prevention

Attackers look to steal sensitive information from databases. MariaDB and MySQL both utilize an extra layer of security, MaxScale and MySQL Enterprise Firewall, respectively. MaxScale and MySQL Enterprise Firewall both seek to block certain types of queries or query patterns, but they take different approaches in how they define firewall rules.

MySQL Enterprise Firewall generates whitelist rules through manually training the database. This requires a training session where all database access must be simulated in order to be captured and whitelisted. Generated firewall rules require downtime to add additional behavior and leave virtually no room for tuning or customization.

Instead of "learning" firewall rules from actual database requests, MaxScale requires firewall rules to be explicitly documented and assigned to users. The explicit, declarative rule definitions used in MaxScale lead to more versatile and expressive possibilities than can be "learned" by MySQL Enterprise Firewall.

In addition to blocking behavior based on queries, MaxScale can also temporarily block requests based on the frequency of queries, as in the case of a DDoS attack. The architecture of MySQL Enterprise Firewall is incapable of filtering such traffic patterns.

Auditing

With new laws and industry standards being introduced that require or encourage user access and activity logging, auditing is becoming an important fixture of any database. Both MySQL and MariaDB offer a zero-downtime installation of an audit plugin that can be configured to record this kind of data. Written to a local file by default, both audit plugins can be tuned to capture certain types of events from certain users. MariaDB takes this a step further by adding the ability to stream the log messages to a different server. This approach provides additional convenience, safety and security. Shipping multiple servers' logs via rsyslog to a single location makes processing logs much more convenient since it's all in one place. It's safer because a critical disk failure or other disasters in a single server won't destroy the audit logs. By storing audit logs on a separate server, you can lock down access to them more effectively than on your database server. MySQL does not provide any remote logging capabilities in its audit plugin.

Supporting Ecosystem and Tools

MariaDB simplifies ecosystem interaction and requires fewer tools to do the same work as MySQL.

Upgrading from MySQL to MariaDB is a simple process. As mentioned earlier, the upgrade happens in-place on the server and clients can be agnostic to the change. Once the initial upgrade has been completed, several other resources will be useful to ease the transition into a new ecosystem. The following chart shows ecosystem tools to transition to in order to maximize the upgrade to MariaDB.

Purpose	MySQL	MariaDB
Monitoring	MySQL Enterprise Monitor	Monyog
Backup	MySQL Enterprise Backup	MariaDB Backup
SQL Management	MySQL Workbench	SQLyog
Load Balancing and Routing	MySQL Router	MaxScale
Database Firewall	MySQL Enterprise Firewall	MaxScale

Enterprise Database Compatibility

MariaDB is closing the gap with legacy enterprise databases.

MariaDB Server 10.2 and 10.3 introduces new features for enterprise database compatibility, such as Oracle Database. MariaDB Server 10.2 features include window functions and common table expressions (CTE) for analytical use cases. MariaDB Server 10.3 includes a PL/SQL parser, which will help users to migrate from Oracle Database to MariaDB. In addition, MariaDB is investing in temporal features. MariaDB's log-based rollback (10.2) allows instances, databases or tables to be rolled back using the binlog. Another feature, AS OF and system versioned tables will provide full temporal processing capabilities including data change history and comparing data within a certain timespan.

Conclusion

MariaDB surpassed MySQL Enterprise Edition in most categories above. In addition to the technical advantages MariaDB has over MySQL, it also holds advantages over MySQL by fully leveraging the open source model. As always, individual use cases vary, so the best way to evaluate MariaDB as a replacement for MySQL is by simulating production load in a test environment. Taking as little as two minutes per server to install or upgrade, the cost of testing is negligible when compared to the benefits outlined above. Whether a project is just starting or has been running in production for a decade, it's the perfect time to upgrade to MariaDB.

Website: mariadb.com
Americas: sales-AMER@mariadb.com
Europe, Middle East, Africa: sales-EMEA@mariadb.com
Asia Pacific: sales-APAC@mariadb.com

© Copyright 2017 MariaDB Corporation Ab, Tekniikantie 12, 02150 Espoo, Finland. MariaDB is a trademark or registered trademarks of MariaDB Corporation.