



What's New in MySQL 8.0

A MySQL White Paper

April 2018



Introduction	3
Performance Improvements	4
MySQL Document Store.....	6
Character Sets: Default to utf8mb4	7
Improved JSON Support	8
SQL Enhancements.....	9
Improvements to Query Consistency	11
Optimizer Histogram Statistics.....	12
GIS.....	13
Resource Groups	14
New Data Dictionary	15
Information Schema	16
Performance Schema	17
Improvements to Defaults.....	18
Cloud Friendly	19
Security Enhancements	19
Replication Improvements.....	21
Upgrade Checker Utility	25
MySQL Enterprise Edition.....	26
MySQL Cloud Service	28
Conclusion.....	28
Additional Resources.....	28



Introduction

MySQL is the most trusted and widely used open source database platform in use today. 10 out of the top 10 most popular and highly-trafficked websites in the world rely on MySQL due to its ubiquity across heterogeneous platforms and application stacks and for its proven performance, reliability, and ease of use. MySQL 8.0 builds on this momentum by delivering across the board improvements designed to enable innovative DBAs and Developers to create and deploy the next generation of web, embedded, mobile and Cloud/SaaS/PaaS/DBaaS applications on the latest generation of development frameworks and hardware platforms.

Some of the MySQL 8.0 highlights include:

- **MySQL Document Store** enabling developers to create NoSQL document database applications using schemaless JSON collections. Now developers have the flexibility of developing both relational and NoSQL document applications using a single database, without having to use a separate document database.
- **MySQL 8.0 is 2x faster** than MySQL 5.7 and sets new benchmarks records. Sysbench Performance Benchmarks demonstrate that MySQL is 2x faster in both Read Write and Read Only tests. Furthermore, MySQL 8.0 sets new benchmark records with up to 1.8 Million Queries/Second.
- **Improved Reliability** with a Transactional Data Dictionary. Previously database data dictionary meta data was stored in multiple file formats and different locations. The MySQL 8.0 Data Dictionary is implemented as a set of SQL tables stored in a single InnoDB tablespace. DDL statements are atomic and crash safe, resulting in improved reliability.
- **Improved Security** with multiple enhancements including SQL Roles. MySQL 8.0 implements standard SQL Roles. A role is a named collection of privileges. One can grant roles, grant to roles, create roles, drop roles, and decide what roles are applicable during a session. Also provided is an SQL function `ROLES_GRAPHML()` returning a graphml document representing role subgraphs.

This paper serves as a DBA and Developer guide to MySQL 8.0 as it highlights the key new features. To make the process of upgrading to MySQL 8.0 as quick and easy as possible, MySQL is introducing a new Upgrade Checker utility, that runs a series of checks. If any issues are discovered, it displays them along with any advice targeted at resolving those issues.

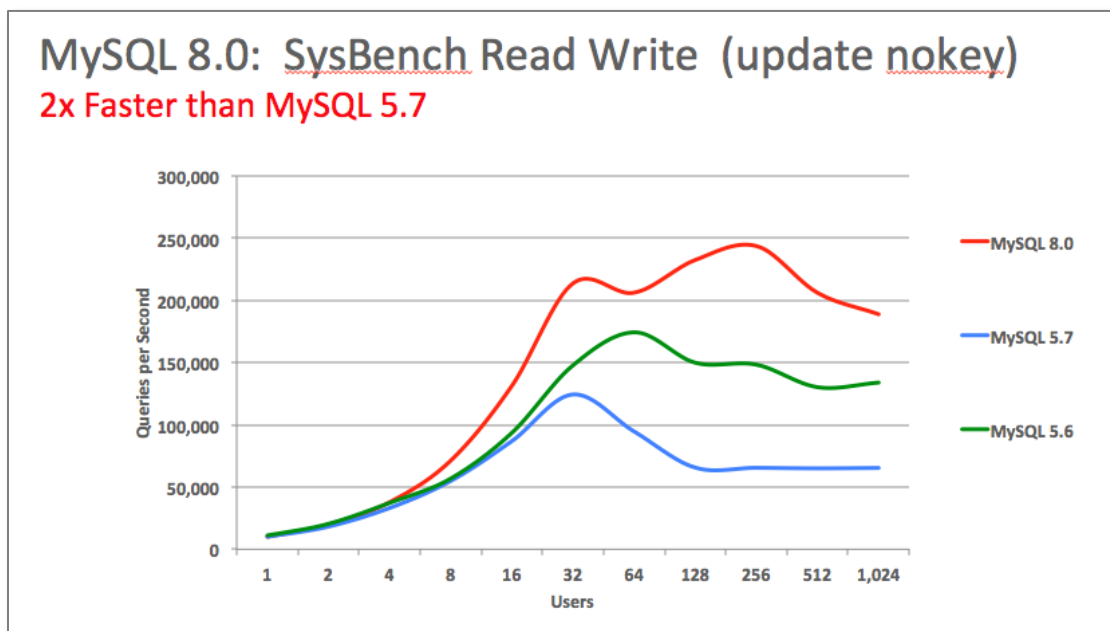


To conclude, there is a summary of how MySQL 8.0 can be implemented using best practices comprising production-ready support and services of the MySQL Enterprise Edition offering.

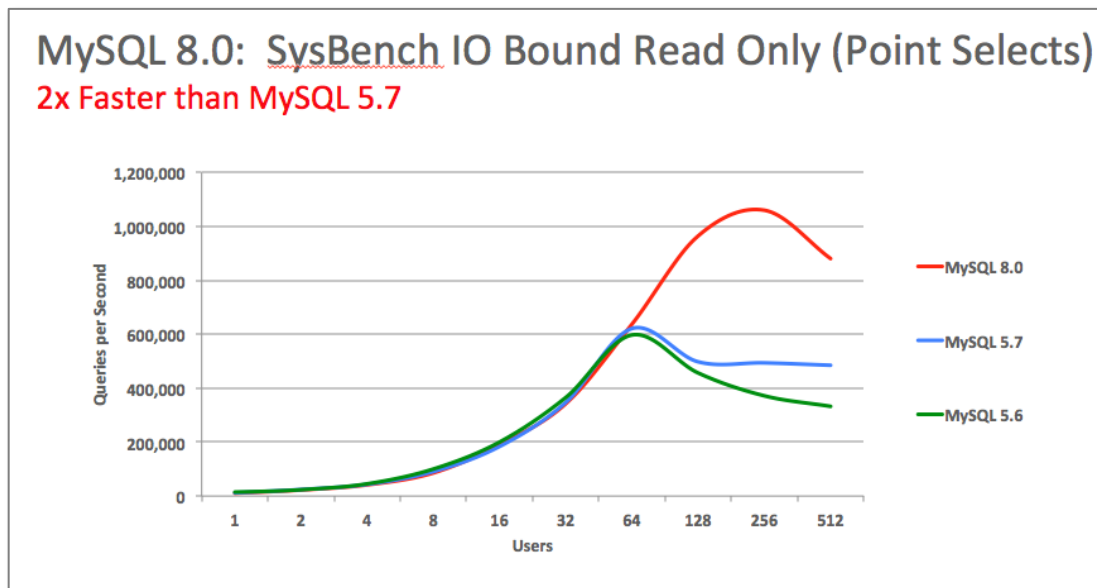
Performance Improvements

MySQL 8.0 delivers significant performance improvements and sets new benchmark records.

2x Faster than MySQL 5.7: SysBench Read Write



2x Faster than MySQL 5.7: SysBench Read Only (Point Selects)



For context, the benchmark shown above was run on the following platform configuration:

- OS : Oracle Linux 7.4
- CPU : 48cores-HT Intel Skylake 2.7Ghz
- (2CPU sockets, Intel(R) Xeon(R) Platinum 8168 CPU)
- RAM: 256GB
- Storage : x2 Intel Optane flash devices
- (Intel (R) Optane (TM) SSD P4800X Series)

For more information on the MySQL 8.0 benchmarks, visit:

<http://www.mysql.com/why-mysql/benchmarks/>

The SysBench benchmark tool is freely available for application use-case specific benchmarks and can be downloaded here:

<http://dev.mysql.com/downloads/benchmarks.html>

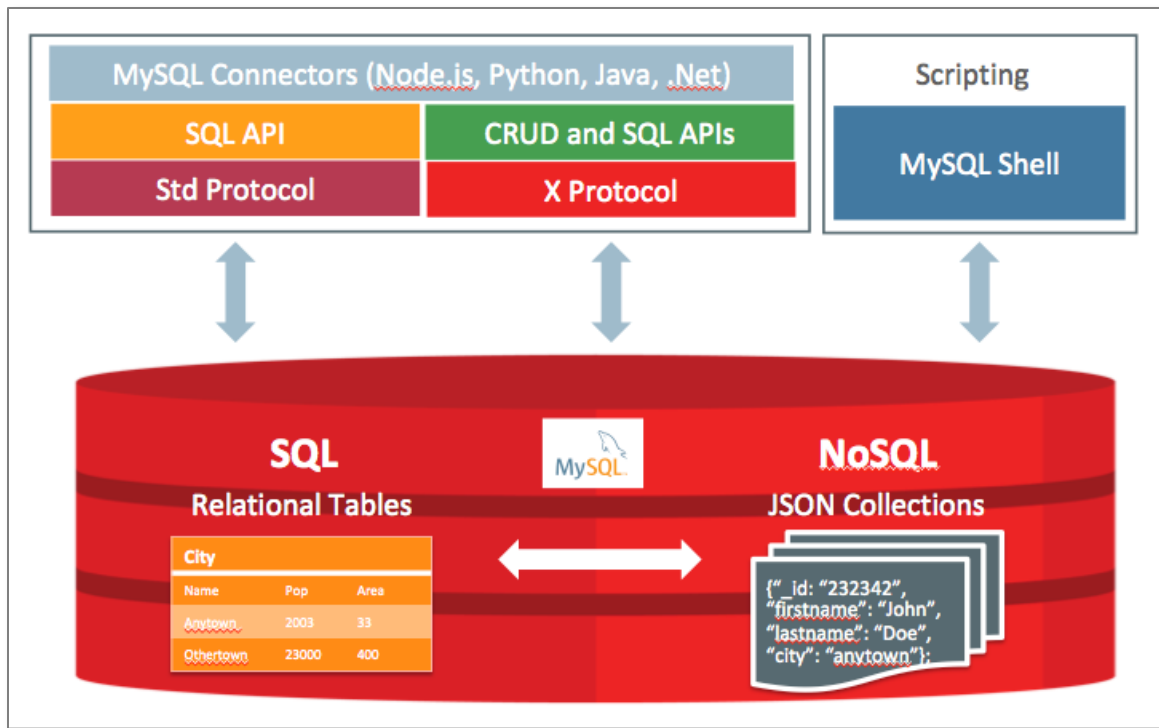


MySQL Document Store

MySQL Document Store allows developers to work with relational tables and schema-less JSON collections. To make that possible MySQL has created the X Dev API, which puts a strong focus on SQL and CRUD by providing a fluent API allowing you to work with JSON documents in a natural way. The X Protocol is a highly extensible and is optimized for both SQL and CRUD API operations. MySQL Shell is an advanced command-line client that offers scripting capabilities for JavaScript and Python.

The MySQL Document Store architecture consists of the following components:

- **MySQL X Plugin** - The X Plugin enables the MySQL Server to use the X Protocol and uses MySQL Shell to act as a client to the server. The X Plugin is supplied with MySQL 5.7.12 or higher.
- **X Protocol** - For the MySQL Document Store a new client protocol was developed based on top of the Protobuf industry standard. The X Protocol works for both, CRUD and SQL operations.
- **X DevAPI** – The X DevAPI is a new, modern, async developer API for CRUD and SQL operations on top of X Protocol. It introduces Collections as new Schema objects. Documents are stored in Collections and have their dedicated CRUD operation set.
- **MySQL Shell** - The MySQL Shell is an interactive Javascript, Python, or SQL interface supporting development and administration for the MySQL Server. You can use the MySQL Shell to perform data queries and updates as well as various administration operations.
- **MySQL Connectors** - The following MySQL Connectors support the X Protocol and enable you to use X DevAPI in your chosen language to develop applications that communicate with a MySQL Server functioning as a document store.
 - MySQL Connector/J
 - MySQL Connector/Node.js
 - MySQL Connector/Net
 - MySQL Connector/Python
 - MySQL Connector/C++



MySQL Document Store Architecture and Components

For More Information:

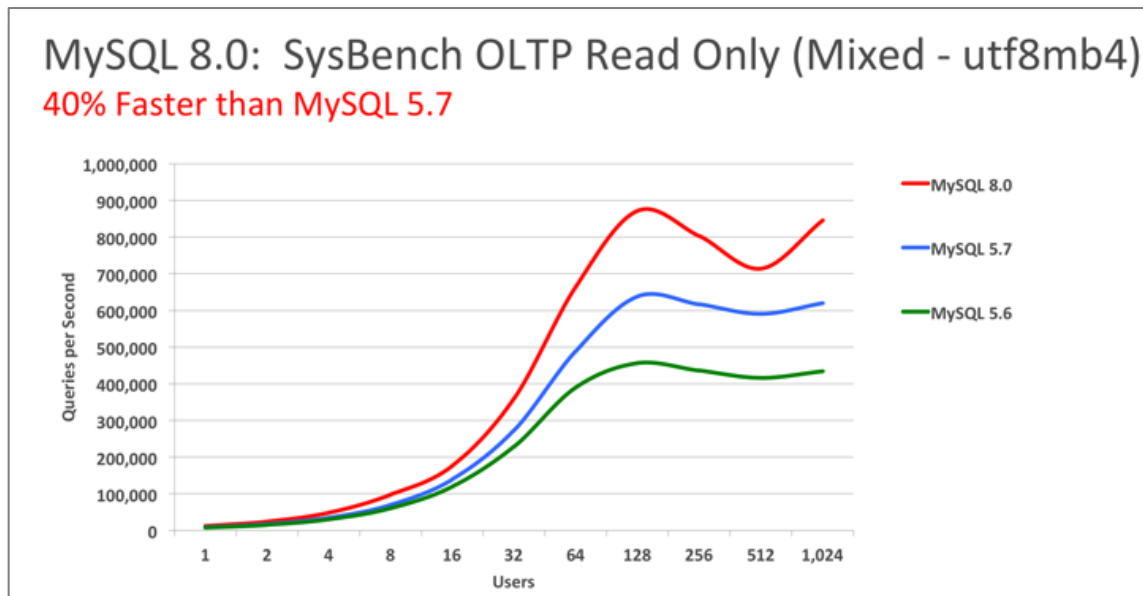
- [Documentation: Setting Up MySQL as a Document Store](#)

Character Sets: Default to utf8mb4

With MySQL 8.0, one of the priorities is mobile first for making MySQL better for delivering modern web applications. In the mobile world, emojis and a multitude of charsets and collations need to coexist. Unicode (or more specifically UTF-8 encoding) has become universal even in English speaking markets. This is why the default character set has been changed from latin-1 to utf8mb4. MySQL supports now the latest Unicode 9.0 with new collations based on DUCET, accent and case sensitive collations, support for Japanese including kana sensitivity.

The challenge with variable length character encodings has always been performance, which the MySQL development team has worked hard on improving during the development of MySQL 8.0. Current users of MySQL 5.7 will notice a 40% performance increase by upgrading to one of the new collations in MySQL 8.0. The earlier utf8mb3 character set has been deprecated.

40% Faster than MySQL 5.7: utf8mb4



Improved JSON Support

JSON support was perhaps the most well received feature of MySQL 5.7, and MySQL 8.0 includes a number of useful additions.

This is an overview of some nice improvements:

- ->> has been added, this is the equivalent of `JSON_UNQUOTE(JSON_EXTRACT())`
- new aggregation functions `JSON_ARRAYAGG()` and `JSON_OBJECTAGG()`
- addition of `JSON_PRETTY()`
- new JSON utility functions like `JSON_STORAGE_SIZE()`, `JSON_STORAGE_FREE()`

One of the most impressive function that has been added in MySQL 8.0 for JSON documents is the `JSON_TABLE()` function. This function accepts JSON data and returns it as a relational table. Columns and casting can be specified. Users can query the result returned by the function as a regular relational table using SQL, e.g. join, project, and aggregate.



For more information:

- [MySQL 8.0: JSON Aggregation Functions](#)
- [MySQL JSON Documentation](#)

SQL Enhancements

Common Table Expressions (CTEs)

Common Table Expressions (also known as WITH queries) have been added to MySQL 8.0. A CTE is a named temporary result set that exists within the scope of a single statement and that can be referred to later within that statements, recursively or not. Non-recursive CTEs can be explained as “improved derived tables” as it allow the derived table to be referenced more than once. A recursive CTE is a set of rows which is built iteratively: from an initial set of rows, a process derives new rows, which grow the set, and those new rows are fed into the process again, producing more rows, and so on, until the process produces no more rows.

For more information:

- [MySQL 8.0: \[Recursive\] Common Table Expressions in MySQL \(CTEs\)](#)
- [MySQL 8.0: \[Recursive\] Common Table Expressions in MySQL \(CTEs\), Part 2](#)
- [MySQL 8.0: \[Recursive\] Common Table Expressions in MySQL \(CTEs\), Part 3](#)
- [MySQL CTEs Documentation](#)

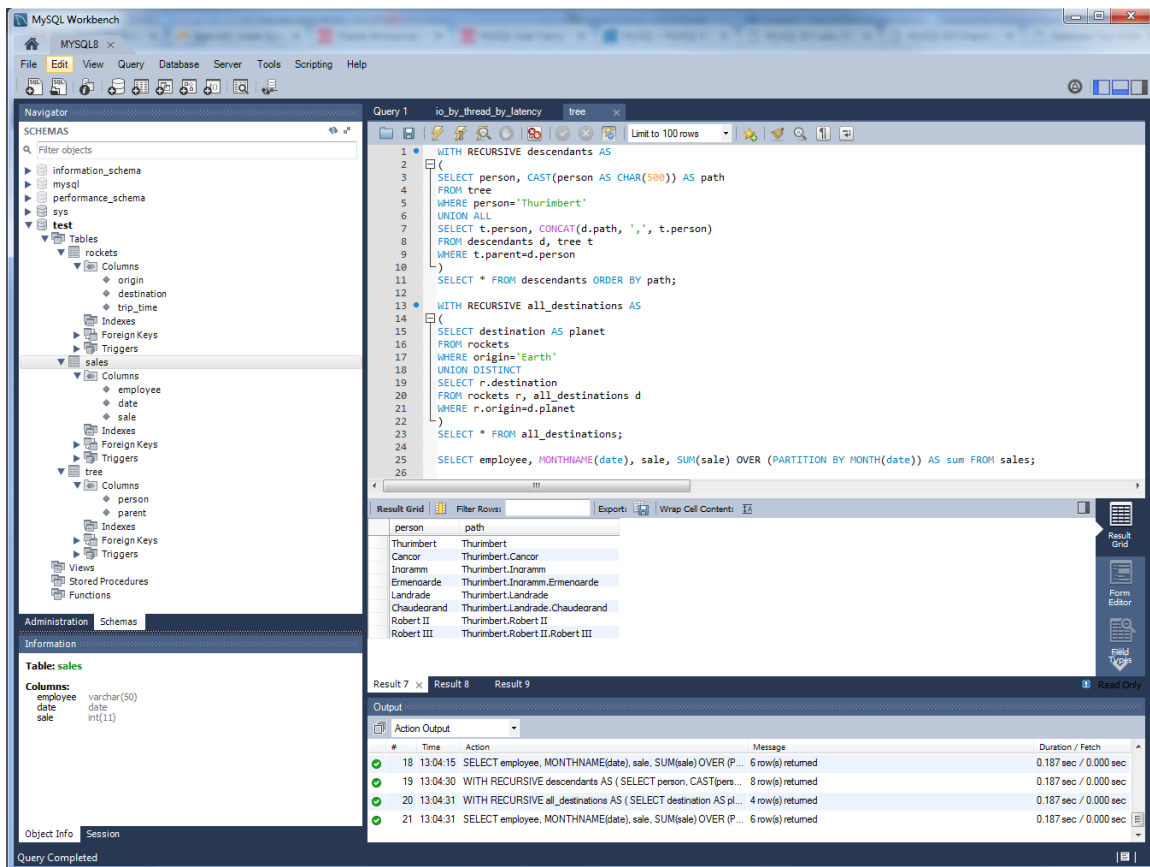
Window Functions

A window function, for each row from a query, performs a calculation using rows related to that row. It's like a GROUP BY but preserving rows rather than collapsing them. Window Functions are extremely powerful, and reduce code complexity significantly.

This is the list of the implemented Window Functions as in MySQL 8.0:

Name	Description
CUME_DIST()	Cumulative distribution value
DENSE_RANK()	Rank of current row within its partition, without gaps
FIRST_VALUE()	Value of argument from first row of window frame
LAG()	Value of argument from row lagging current row within partition
LAST_VALUE()	Value of argument from last row of window frame
LEAD()	Value of argument from row leading current row within partition
NTH_VALUE()	Value of argument from N-th row of window frame

Name	Description
NTILEQ	Bucket number of current row within its partition.
PERCENT RANKQ	Percentage rank value
RANKQ	Rank of current row within its partition, with gaps
ROW NUMBERQ	Number of current row within its partition



MySQL CTEs and Window Functions in MySQL Workbench

For more information:

- [MySQL 8.0: Introducing Window Functions](#)
- [MySQL 8.0 Window Functions Documentation](#)

Invisible Indexes

Many of today's applications run 24x7, leaving no clear maintenance windows for software upgrades or database changes. Customers have provided key feedback which is that *any change* to indexing can have unintended consequences.



Invisible indexes are a new feature in MySQL 8.0 that provide the ability to mark an index as unavailable for use by the Optimizer. This means that the index will still be maintained and kept up-to-date as data is modified, but no queries will be permitted to make use of the index (even if the query uses a `FORCE INDEX` hint).

One common use case is to find out which indexes are unused via the `SYS` schema. You can then set them to `INVISIBLE`, which is the database equivalent to a Recycle Bin. After some time has elapsed, you can then safely `DROP` the index. The same process can also be used in reverse for a *staged-rollout* of indexing changes.

For more information:

- [MySQL 8.0: Invisible Indexes](#)
- [MySQL 8.0: Invisible Indexes Documentation](#)

Descending Indexes

MySQL server now supports descending indexes, can be used to eliminate the need for sorting results, and lead to performance improvements in a number of queries. With the introduction of descending indexes, InnoDB can now store entries in descending order and the optimizer will take advantage of it when descending order is requested in the query.

For more information:

[MySQL 8.0: Descending Indexes in MySQL](#)

Improvements to Query Consistency

With SQL being a declarative language, a query is similar to what a street address is to GPS navigation. That is to say it conveys a final destination, and not a set of directions on how to get there. The optimizer is the part of the database system which “picks” the best execution plan (to extend the street analogy, is Google Maps). MySQL 8.0 contains several improvements designed to help picking the best plan more consistently.

Optimizer Histograms (not to be confused with Performance Schema histograms) have been introduced to help with cases where a high amount of data skew would have previously led to a poor execution plan choice.

The cost model has been refined to consider the effect of how much of the table or index is resident in memory. For example: On a given query, it may be preferred to range scan only when a large percentage of the index is in memory. The individual *constants* used in the cost model are user-configurable, and in



MySQL 8.0 the usability has been improved slightly by showing the default values.

Additional query hints have also been added for finer grain control of the optimizer if it makes incorrect choices. Database Administrators can also insert these hints by using the query rewrite feature on the MySQL server side.

For more information:

- [MySQL 8.0: Query Optimizer Takes Data Buffering into Account](#)

Better Handling of Hot Rows: SKIP LOCKED and NOWAIT

Support for the lock modifiers SKIP LOCKED and NOWAIT. These two features allow you to better manage situations where you have tables with hot row contention – such as in the cases of a hot SKU in an e-commerce application or a set of worker threads that are all reading from the same table trying to find new rows to process.

When encountering locked rows, the default behavior is to queue waiting for up to 50 seconds for the lock holder to release. You now have the option to error immediately (NOWAIT) or non-deterministically skip past locked rows (SKIP LOCKED). You can even use both at the same time:

```
SELECT * FROM tickets  
WHERE id IN (1,2,3,4)  
AND order_id IS NULL  
FOR UPDATE  
NOWAIT;
```

Error immediately if
a row is already
locked

```
SELECT * FROM tickets  
WHERE id IN (1,2,3,4)  
AND order_id IS NULL  
FOR UPDATE  
SKIP LOCKED;
```

Non
deterministically
skip over locked
rows

MySQL 8.0 enables better management high volume ecommerce applications.

Optimizer Histogram Statistics

As of MySQL 8.0.3, you now have the ability to create histogram statistics in order to provide more statistics to the optimizer.



The query optimizer is the part of the database that is responsible for translating a SQL query into the most efficient execution plan possible. Sometimes, the query optimizer fails to find the most efficient plan and ends up spending a lot more time executing the query than necessary. The main reason for this is often that the optimizer doesn't have enough knowledge about the data it is about to query:

- How many rows are there in each table?
- How many distinct values are there in each column?
- How is the data distributed in each column?

A histogram is an approximation of the data distribution for a column. It can tell you with a reasonably accuracy whether your data is skewed or not, which in turn will help the database server understand the nature of data it contains.

Histograms comes in many different flavors, and MySQL supports two different types: The "singleton" histogram and the "equi-height" histogram. Common for all histogram types is that they split the data set into a set of "buckets", and MySQL automatically divides the values into buckets, and will also automatically decide what type of histogram to create.

For more information:

- [Histogram statistics in MySQL](#)

GIS

Modern applications are mobile, and (*besides using emoji!*), one of the characteristics of mobile is that it frequently make use of a user's location to provide context. MySQL 5.7 uses Boost.Geometry as a core library. The MySQL engineering team has since become regular contributors to Boost.

In MySQL 8.0 GIS is extended to support Geography and Spatial Reference Systems (SRS). With MySQL 8.0, the world is not only flat anymore, but it can be ellipsoidal too!

Some examples:

- axes have units
- geographic coordinate system wrap around
- geographic axes are not orthogonal
- geographic axes order matters
- relevance of axis direction

MySQL 5.7	MySQL 8.0
<ul style="list-style-type: none"> • The world is flat • The world is infinite • Axes are unitless • Axes are orthogonal • Axis order is irrelevant • Axis direction is irrelevant 	<ul style="list-style-type: none"> • The world can be flat or ellipsoidal • Geographic coordinate systems wrap around • Axes have units • Geographic axes are not orthogonal • Geographic axis order matters • Axis direction may be relevant

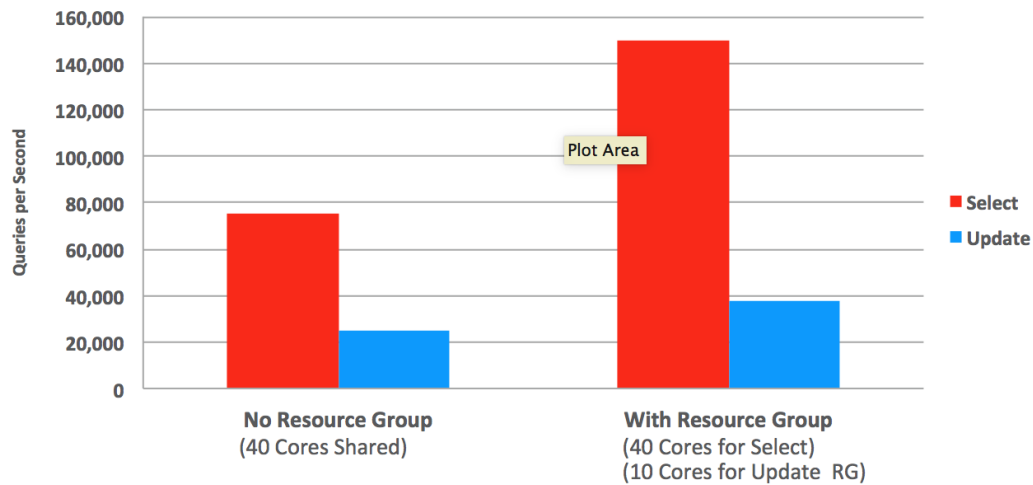
In MySQL 8.0, GIS is extended to support Geography and Spatial Reference Systems.

Resource Groups

The purpose of Resource Groups is to decide on the mapping between user/system threads and CPUs. This can be used to split workloads across CPUs to obtain better efficiency and/or performance in some use cases.

There are two default groups, one for user threads and one for system threads. Both default groups have 0 priority and no CPU affinity. DevOps/DBAs can create and manage additional Resource Groups with priority and CPU affinity using SQL CREATE/ALTER/DROP RESOURCE GROUP. Information about existing resource groups are found in INFORMATION_SCHEMA.RESOURCE_GROUPS. The user can execute a SQL query on a given resource group by adding the hint `/*+RESOURCE_GROUP(resource_group_name)*/` after the initial SELECT, UPDATE, INSERT, REPLACE or DELETE keyword.

MySQL 8.0: Resource Group Example



Performance benchmark improvements by using Resource Groups

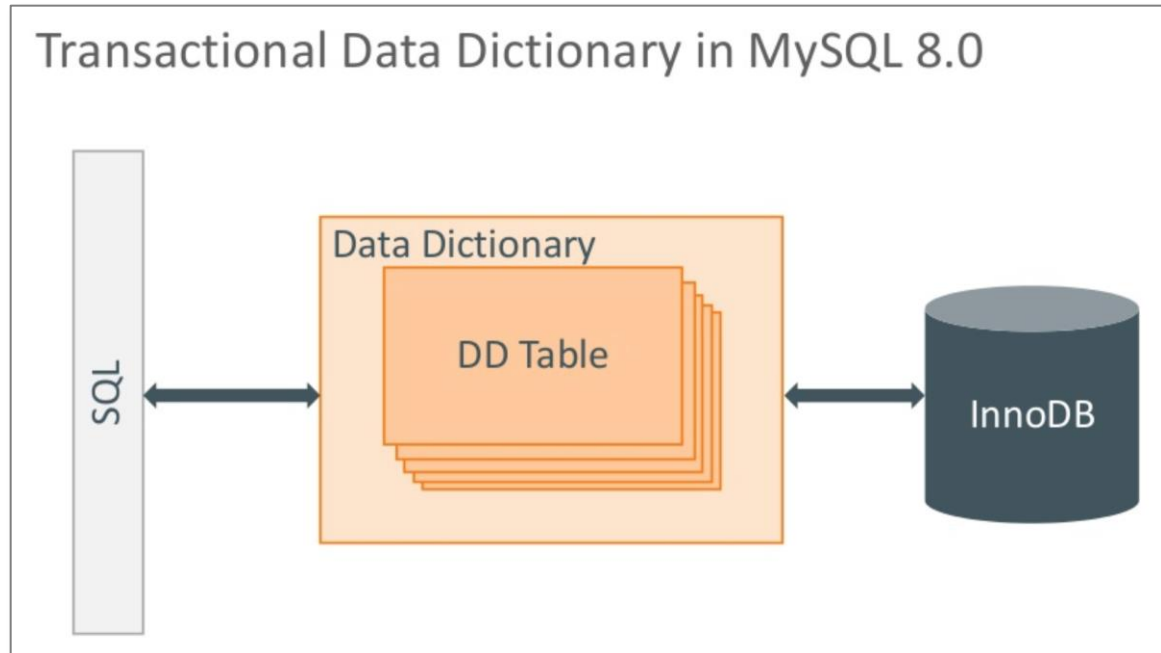
New Data Dictionary

Just as you use a database like MySQL to store your application data, MySQL must also store its meta data (schema names, table definitions etc) somewhere. Traditionally this meta data storage has been split between many different locations (.FRM, .PAR, .OPT, .TRN and .TRG files).

With the new native Data Dictionary, MyISAM system tables are no longer needed. Those tables and the data dictionary tables are now created in a single InnoDB tablespace file named mysql.idb in the data directory. This means that if you don't explicitly use MyISAM tables you can have a MySQL instance without any MyISAM table.

With the new Data Dictionary, MySQL 8.0 now supports Atomic Data Definition Statements (Atomic DDLs). This means that when a DDL is performed, the data dictionary updates, the storage engine operation and the writes in the binary log are combined into a single atomic transaction that is either fully executed or not at all. This provides a better reliability where unfinished DDLs don't leave any incomplete data.

While there are a number of reasons why this was an important core-change, the new transactional data dictionary provides a lower barrier to entry for new features, and improved reliability as meta data management is unified in transactional storage.



The MySQL 8.0 native InnoDB-based Data Dictionary provides a simplified and uniform architecture to reliably handle dictionary data.

For more information:

- [MySQL 8.0 Data Dictionary: Background and Motivation](#)
- [MySQL 8.0: Data Dictionary Architecture and Design](#)
- [Atomic DDL in MySQL 8.0](#)
- [MySQL 8.0 Data Dictionary Documentation](#)

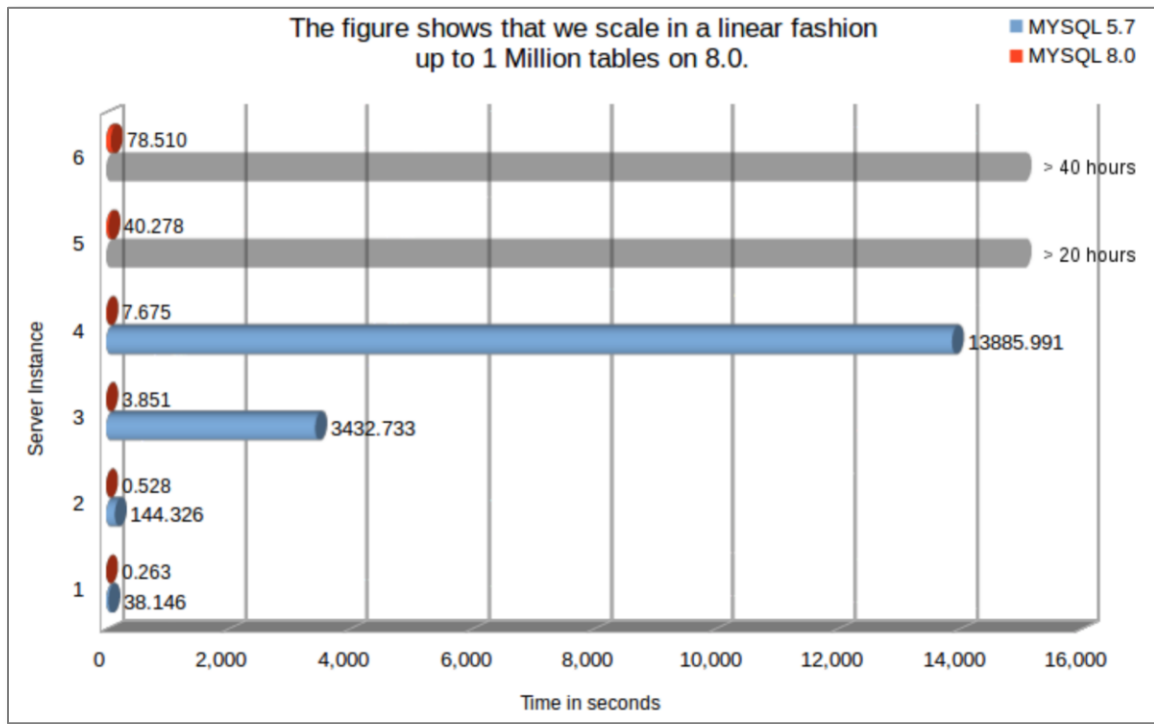
Information Schema

Coinciding with the new native data dictionary in MySQL 8.0, a number of useful enhancements have been made to the INFORMATION_SCHEMA subsystem design. One of the major changes in 8.0 is the introduction of a native data dictionary based on InnoDB. This change has enabled us to get rid of file-based metadata store (FRM files) and also help MySQL to move towards supporting transactional DDL.

Now that the metadata of all database tables is stored in transactional data dictionary tables, it enables us to design an INFORMATION_SCHEMA table as a database VIEW over the data dictionary tables. This is by far more efficient than the old implementation with up to 100 times performance improvement. This eliminates costs such as the creation of temporary tables for each INFORMATION_SCHEMA query during execution on-the-fly, and also scanning file-system directories to find FRM files. It is also now possible to utilize



the full power of the MySQL optimizer to prepare better query execution plans using indexes on data dictionary tables.



INFORMATION_SCHEMA queries execute several hundred times faster

More Information

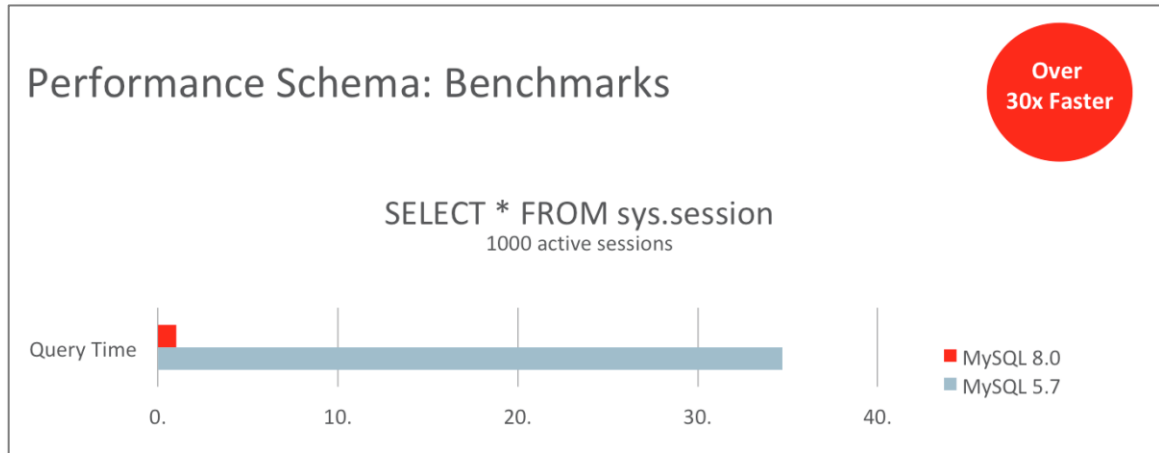
- [MySQL 8.0: Improvements to Information schema](#)
- [MySQL 8.0: Scaling and Performance of INFORMATION_SCHEMA](#)
- [Further improvements on INFORMATION_SCHEMA in MySQL 8.0](#)

Performance Schema

In MySQL 8.0 Performance Schema has been improved to support indexes. This increases the flexibility and ease of use of using performance_schema, as a number of monitoring queries will see dramatic performance improvements. A total of 115 indexes have been added in the performance schema in MySQL 8.0, to support better data access patterns in general.

This work was possible because the performance schema design, from its creation way back in MySQL 5.5, choose to expose the data as a storage engine instead of temporary tables. Additionally, as data access to the performance schema uses the same interface as data access to regular tables, it is able to benefit from future improvements to the query optimizer (better query plans and access methods), and reduces our maintenance burden.

The performance improvements from indexes can be very easily seen in many of the sys schema queries. With 1000 idle threads, the query `SELECT * FROM sys.session` drops from 34.70 seconds down to 1.01 seconds, a 30x improvement!



In MySQL 8.0, performance_schema queries are up to 30x faster.

For more information:

- [MySQL 8.0: Performance Schema, now with indexes!](#)
- [Performance Schema Reference Manual](#)

Improvements to Defaults

Generally speaking, a good default is “the best choice for most users, most of the time”. Out of respect to the existing user base changes to a default and not done just for the sake of it, so any change needs to have a good reason. The changes below have been motivated by several different objectives, such as :

- **Mainstream hardware is changing.** Machines have more CPUs and more memory than before, machines have SSDs rather than spinning disks etc.
- **User installations tend to be “bigger” than before**, thus on average need more resources such as bigger memory buffers.
- **MySQL should work “out of the box”**, thus features in general use should be ON by default. For example utf8 character set, binlog enabled, event scheduler enabled, more instrumentation, etc.
- **Better trade-offs** (based on experience), such as adjusting the InnoDB flushing behavior and bigger buffer for IP connection requests.
- **More standard SQL**, such as explicit defaults for timestamps.
- **More security**, such as enabling the password validation plugin.



The overall goal for defaults will always be to be “good enough” so that users will not have to worry about changing them.

In total over 20 defaults have been changed in MySQL 8.0 to improve the out of the box experience. Notable changes include several changes to replication, additional performance schema instruments enabled by default (memory, transactions, mdl), the default character set is now utf8mb4, and InnoDB now assumes SSDs by default.

The default configuration remains targeted for a system with 512M of memory, making it friendly for development environments and small cloud VMs. But in MySQL 8.0 it is easier to switch to a *dedicated* mode, where it will scale to use all system resources.

For more information:

- [New Defaults in MySQL 8.0](#)
- [Heads Up: The List of Replication Defaults That Have Changed in 8.0.2](#)
- [Plan to improve the out of the Box Experience in MySQL 8.0](#)

Cloud Friendly

It is becoming more common for MySQL to be deployed in a virtual machine, where the amount of system resources can change as the system is scaled up or down. With the new `innodb_dedicated_server` option, it's possible to auto-detect the system memory, and have MySQL adjust appropriately without having to edit configuration files.

It is also more common to have restricted or zero local access (shell or filesystem) to running database servers. This restriction makes it hard for administrators to change MySQL configuration. This was one of the use-cases for the new SET PERSIST feature, which allows you to persist global variables between MySQL restarts.

For more information:

- [MySQL 8.0: Persisting configuration variables](#)
- [MySQL 8.0: changing configuration easily and cloud friendly!](#)

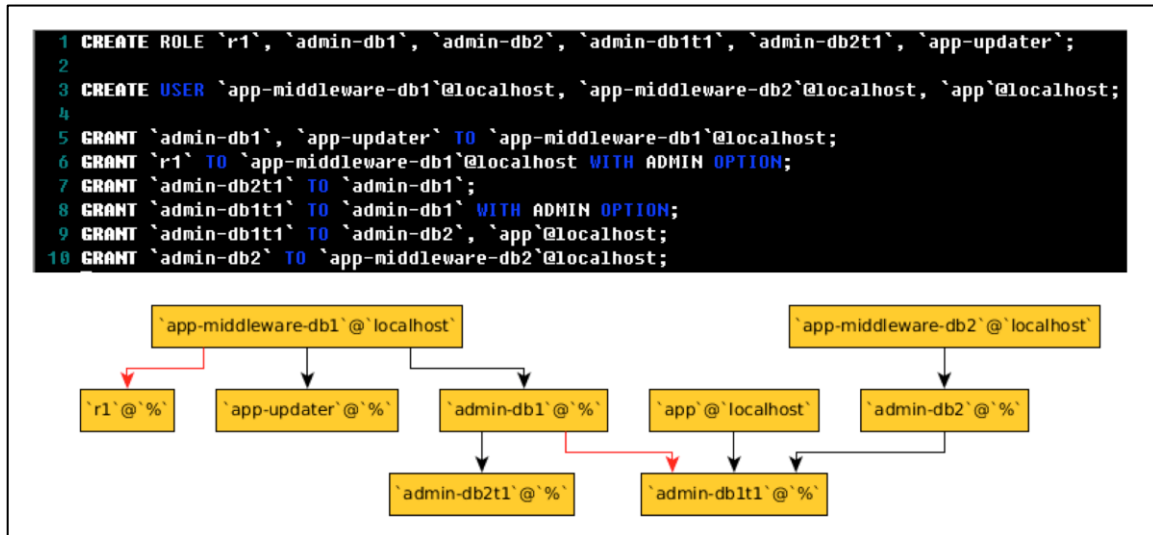
Security Enhancements

SQL Roles

MySQL 8.0 implements standard SQL Roles. A role is a named collection of privileges. One can grant roles, grant to roles, create roles, drop roles, and decide what roles are applicable during a session. Also provided is an SQL

function `ROLES_GRAPHML()` returning a graphml document representing role subgraphs.

Pluggable dynamic privileges provides a service to easily register new, granular global privileges with the server at runtime. It provides an easy way to GRANT and REVOKE these privileges through SQL and programmatically check if they're granted or not. This work also defines a set of new granular privileges for various aspects of what SUPER is used for today.



Atomic ACL Statements

Prior to MySQL 8.0 many ACLs or Access Controls, which appeared in a single statement, were actually executed in multiple parts. With the new transactional, ACID-based, InnoDB data dictionary in MySQL 8.0 each statement is atomic; it succeeds or it fails. So statements creating many users, or granting many users are atomic now. New locks within 8.0 were created to preserve this atomic feature for the access control statements.

Dynamic Privileges

Pluggable dynamic privileges provides a service to easily register new, finer grained administrative level access controls with the server at runtime. It provides an easy way to GRANT and REVOKE these privileges through SQL and programmatically check if they're granted or not.

Too often super is required for tasks when less privilege is really needed. This allows DBAs to follow the principle of least privilege, which is the idea that at any user should have only the bare minimum privileges necessary to perform a function.



Protection Against Brute Force Attacks

MySQL 8.0 introduces a delay in authentication process based on consecutive failed login. This will slow down brute force attacks on user passwords. It is possible to configure the number of consecutive unsuccessful attempts before the delay is introduced and the maximum amount of delay introduced. This is achieved via a plugin.

REDO & UNDO Logs Encryption

In MySQL 5.7 Tablespace Encryption for InnoDB tables stored in file-per-table tablespaces was introduced. This feature provides at-rest encryption for physical tablespace data files. In 8.0 this work was extended to include at-rest encryption of UNDO and REDO logs.

A new global, dynamic variable `innodb_undo_log_encrypt=ON/OFF` is used to enable and disable undo log encryption. Pages written to disk after setting `innodb_undo_log_encrypt=ON` are encrypted.

A new global, dynamic variable `innodb_redo_log_encrypt=ON/OFF` is used to enable and disable redo log encryption. Pages written to disk after setting `innodb_redo_log_encrypt=ON` are encrypted.

Caching sha2 authentication plugin

MySQL 8.0 introduces a new authentication plugin, [caching_sha2_password](#), which uses a caching mechanism to speed up authentication. `libmysqlclient` will now use `caching_sha2_password` as the default authentication mechanism, too.

For more information:

- [MySQL 8.0: New Default Authentication Plugin : caching_sha2_password](#)

Password Rotation Policy

MySQL 8.0 introduces restrictions on password reuse. Restrictions can be configured at global level as well as individual user level. Password history is kept secure because it may give clues about habits or patterns used by individual users when they change their password. As previously, MySQL offers a password expiration policy which enforces password change based on time. MySQL also has the ability to control what can and can not be used as password. This work restricts password reuse and thus forces users to supply new strong passwords with each password change.

Replication Improvements



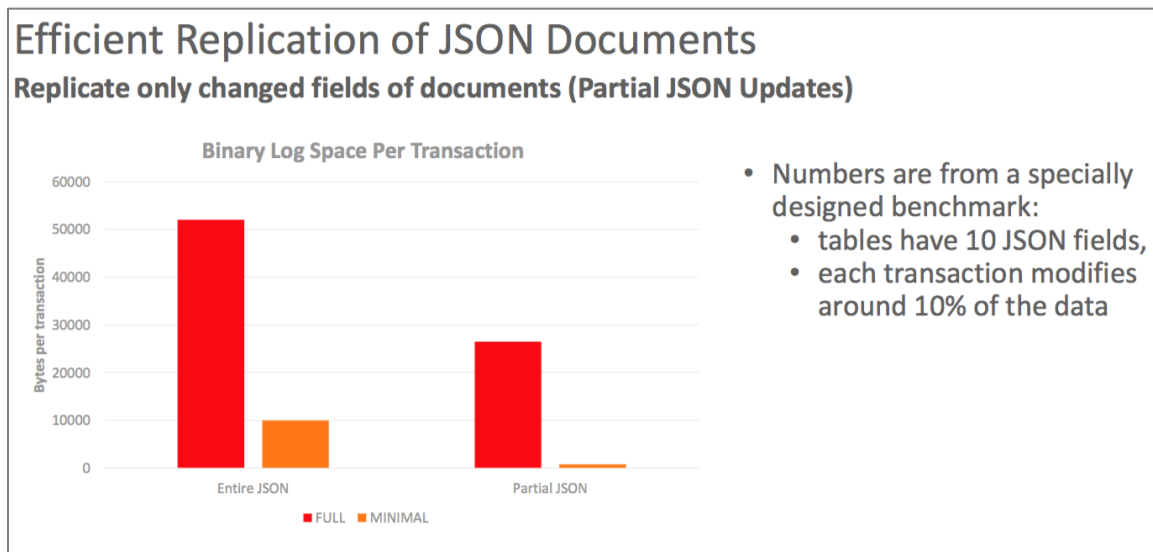
MySQL 8.0 introduces a wide range of new replication features. These range from more instrumentation and monitoring extensions, to additional metadata and information in the binary log, to better performance and efficiency and to improved configurability and overall management of the replication framework.

Additional Metadata in the binary log

The binary log from a MySQL 8.0.1 server shall contain additional table metadata information into the ROW-based changes. For instance, whether a numeric column is SIGNED or UNSIGNED, additional information about column character set, or even the name of the columns. The user is also able to control the amount of metadata logged.

Efficient JSON Replication

In MySQL 8.0 replication of JSON was made much more efficient. The new feature, called Partial JSON Replication, ensures that an UPDATE statement that only modifies parts of a JSON document will not write the full document to the binary log, but rather only those parts that were actually changed. Thus, the bigger your JSON document is, and the smaller part you modify, the more space will you save by using this feature. This reduces the disk usage for binary logs and relay logs, as well as the amount of data that needs to be sent over the network. Therefore, the throughput of the entire system can improve so that you get more transactions per second for JSON-heavy workloads.



Monitor Replication Lag with Microsecond Precision

Infrastructure for GTID based delayed replication and replication lag monitoring implements the infrastructure needed to properly monitor the replication lag. This work introduces two new timestamps that are associated with each transaction (not each event or statement) in the binary log:

- The `original_commit_timestamp` is in microseconds since the epoch when the transaction was committed on the original master,
- The `immediate_commit_timestamp` is in microseconds since the epoch when the transaction was committed on the immediate master.

MySQL 8.0 also improves existing performance schema replication status tables with new fields. These new fields help detecting and diagnosing lag at several points in the replication stream, more specifically lag that may be due to:

1. the connection thread, that retrieves transactions from the master and queues them in the relay log;
2. when multi-threaded slave mode (MTS) is enabled, the coordinator thread that reads the transactions from the relay log and schedules them to a worker queue;
3. the worker threads applying the transactions.

These threads and their activities can be directly mapped to three tables:

1. `replication_connection_status`: current status of the connection to the master;
2. `replication_applier_status_by_coordinator`: current status of the coordinator thread that only displays information when MTS is enabled;
3. and `replication_applier_status_by_worker`: current status of the thread(s) applying transactions received from the master.

More P_S Instrumentation for Group Replication

In order to improve observability and manageability of MySQL Group Replication (GR), instrumentation in MySQL 8.0 was enhanced. With these goals in mind, the usage of Performance Schema (P_S) was extended from this plugin to the included Group Communication System (GCS) module, as well as its submodule eXtended Communications (XCom) which are responsible for all communication involved in the dissemination of transactions to other MySQL instances.

Enable binary log by default

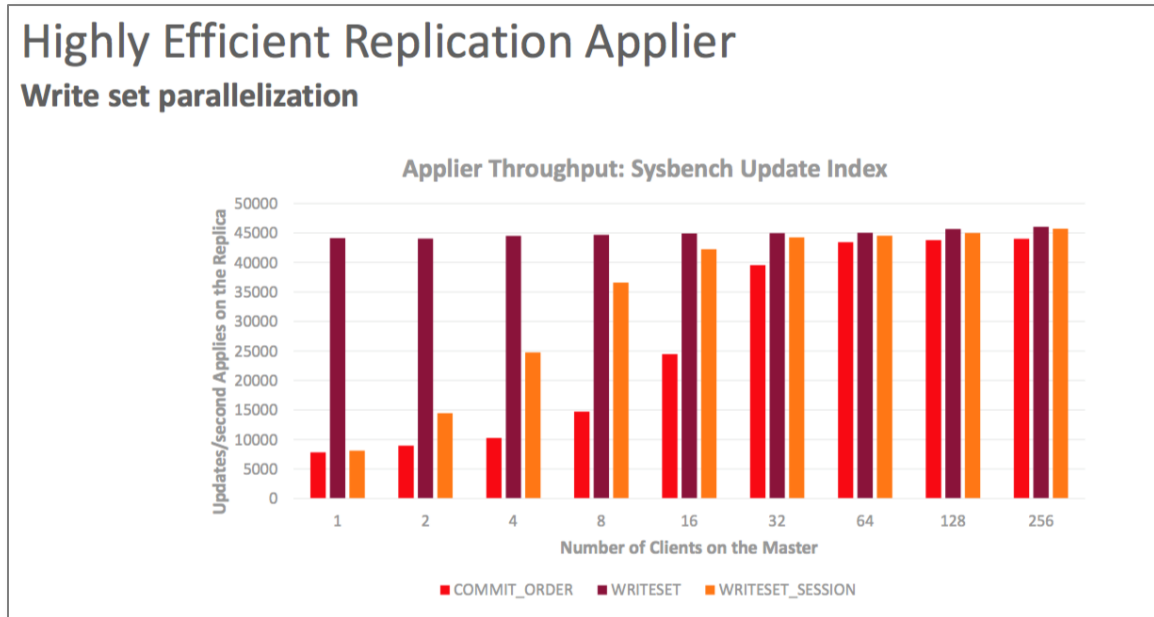
This changes the default of `–log-bin` from OFF to ON. Nearly all production installations have the binary log enabled as it is used for replication and point-in-time recovery. Thus, by enabling binary log by default one configuration step is eliminated (enabling it later requires a `mysqld` restart). By enabling it by default test coverage improves and it becomes easier to spot performance regressions.

Improving the Parallel Applier with Writeset-based Dependency Tracking

MySQL 8.0.1 introduces a new mechanism to track the dependencies between transactions. This feature is used to optimize the binary log applier on asynchronous replication slaves, improving the throughput significantly on workloads where there is low-concurrency on the master and/or on systems with very fast storage.



Testing shows that users can apply binlog on the slave about 4 times faster, even for a single-threaded workload, which is the most difficult workload for Multi Threaded Slaves (MTS).



Hostname support in Group Replication Whitelist

Since the release of Group Replication, one of its main security features is IP Address Whitelisting. This allows us to control whoever can send requests to enter in a group. In MySQL 8.0, the ability to configure IP Address Whitelisting with hostnames was added. All other communication specific parameters like Local Address and Seeds could be configured with hostnames, except Whitelisting. This opens a range of possibilities if your system is built around IP Pools that make a machine address vary in every restart since. This is a somewhat common scenario in Virtual Machines deployments. With this new mechanism, you just need to allow a machine with a certain hostname to enter the group!

For more information:

- [Efficient JSON Replication in MySQL 8.0](#)
- [Hostname support in Group Replication whitelist](#)
- [MySQL 8.0 Binary logging and Replication chains are now enabled by default](#)
- [Efficient JSON Replication in MySQL 8.0](#)
- [More P S Instrumentation for Group Replication](#)
- [Improving the Parallel Applier with Writese-based Dependency Tracking](#)
- [New monitoring replication features and more!](#)



Upgrade Checker Utility

To make sure that your 5.7 system is ready for an upgrade there are certain steps you should take, described in our documentation.

Upgrade Checker is available as a function in Shell's scripting modes: "util.checkForServerUpgrade" in JavaScript and "util.check_for_server_upgrade" in Python, respectively. Its usage is straightforward, the only thing you need to do is to point the function to the server you want to check.

What Upgrade Checker does is connect to specified server and, if the server version is supported (at the moment only 5.7) and the user has enough privileges (ALL privileges with GRANT option), runs a series of checks. If any issues are discovered, it displays them along with any advice targeted at resolving those issues. Not all the issues discovered have the same gravity (they are all tagged either "notice", "warning", or "error") and results in failed upgrade process, but you should consider them all to be able to take full advantage of MySQL 8.0. At the end, UC prints a summary and returns an integer value describing the severity of the issues found:

- 0 – no issues or only ones categorized as notice,
- 1 – No fatal errors were found, but some potential issues were detected,
- 2 – UC found errors that must be fixed before upgrading to 8.0.

Typical UC run will look similar to this:

```
mysql-js> util.checkForServerUpgrade("root@localhost:3306")
The MySQL server at localhost:3306 will now be checked for compatibility issues
for upgrade to MySQL 8.0...
MySQL version: 5.7.19 - MySQL Community Server (GPL)

1) Usage of db objects with names conflicting with reserved keywords in 8.0
No issues found

2) Usage of utf8mb3 charset
Warning: The following objects use the utf8mb3 character set. It is recommended
to convert them to use utf8mb4 instead, for improved Unicode support.

simple_schema.city.name - column's default character set: utf8
simple_schema.city.country_code - column's default character set: utf8

3) Usage of use ZEROFILL/display length type attributes
Notice: The following table columns specify a ZEROFILL/display length attributes
. Please be aware that they will be ignored in MySQL 8.0

test.big_table.ORDINAL_POSITION - bigint(21) unsigned

4) Issues reported by 'check table x for upgrade' command
No issues found

5) Table names in the mysql schema conflicting with new tables in 8.0
No issues found

6) Usage of old temporal type
No issues found

7) Foreign key constraint names longer than 64 characters
No issues found

No fatal errors were found that would prevent a MySQL 8 upgrade, but some poten-
tial issues were detected. Please ensure that the reported issues are not sig-
nificant before upgrading.
1
```

For more information:



[Verifying Upgrade Prerequisites for Your MySQL 5.7 Installation](#)
[MySQL Shell 8.0.4: Introducing “Upgrade checker” utility](#)

MySQL Enterprise Edition

- **MySQL Database** – The most secure and up to date version of MySQL is used to power the most demanding online, web, cloud and OLTP applications and services. The MySQL commercial server is a fully integrated transaction-safe, ACID compliant database with full commit, rollback, crash- recovery and row level locking capabilities. MySQL is known for its performance, reliability and ease-of-use across all Linux, UNIX, Mac OS X, and Windows platforms.
- **MySQL Enterprise Backup** – MySQL Enterprise Backup performs online, non-blocking backups of your MySQL databases. Perform full, incremental, and partial backups for all InnoDB data while MySQL is fully available for transactional operations. All backup operations are executed in parallel for quick results and also support compression options that reduce the size of backup images by up to 90%. Recovery options include backward compatible full recovery, precise one-click point-in-time recovery, and partial recovery for restoring a specific set of objects
- **MySQL Enterprise Firewall** - Blocks SQL Injection attacks that can result in loss of valuable personal and financial data. Whitelist creation, real-time threat monitoring, SQL statement blocking and alerting all enable DBAs to protect vital data assets. Acting as an intrusion detection system, MySQL Enterprise Firewall can notify administrators to SQL statement activity that does not match an approved whitelist.
- **MySQL Enterprise Audit** - MySQL Enterprise Edition provides an easy to use, policy-based auditing solution that helps developers implement stronger security controls and satisfy regulatory compliance without changes to existing applications.
- **MySQL Enterprise Scalability** - To meet the sustained performance and scalability of ever increasing users, query, and data loads--MySQL Enterprise Edition provides the MySQL Thread Pool. The Enterprise Thread Pool provides a highly scalable, queue-based thread-handling model designed to reduce overhead in managing client connections and statement execution threads.
- **MySQL Enterprise Authentication** - MySQL Enterprise Edition provides ready to use external authentication modules so DBAs and developers can



easily integrate MySQL with existing security infrastructures, including Linux Pluggable Authentication Modules (PAM) and Windows Active Directory.

- **MySQL Enterprise Encryption** - To protect sensitive data throughout its lifecycle, MySQL Enterprise Encryption provides industry standard functionality for asymmetric encryption (Public Key Cryptography). MySQL Enterprise Encryption also provides encryption, key generation, digital signatures and other cryptographic features to help organizations protect confidential data and comply with regulatory requirements such as HIPAA, Sarbanes-Oxley, and the PCI Data Security Standard.
- **MySQL Enterprise High Availability** - MySQL Enterprise Edition offers a wide range of solutions for database high availability, to automatically detect and recover from failures--whether these occur at the network, host, OS or database layer--as well as minimize downtime resulting from scheduled maintenance activities.
- **MySQL Enterprise Monitor and Query Analyzer** – Continuously monitors MySQL databases and proactively alerts DBA to potential problems, queries, and tuning opportunities before they impact key systems or applications. The Monitor provides a set of MySQL expert advisors that provide insight and detailed guidance on fixing and tuning MySQL configurations and variables for optimal security, performance and availability. The built-in, advanced Query Analyzer allows developers to visually find and tune expensive query code without the need for Slow Query Log, SHOW PROCESS LIST or other labor intensive methods.
- **Oracle Enterprise Manager for MySQL**
The Oracle Enterprise Manager for MySQL provides Oracle developers and DBAs with real-time monitoring and delivers comprehensive performance, availability and configuration information for your MySQL databases. In addition to the default MySQL compliance rules, you can also create custom critical and warning thresholds can for each of the collected metrics.
- **MySQL Workbench** – Provides GUI-based data modeling, SQL development, deployment, database migration and comprehensive administrative tools (server configuration, user administration, object management) for database architects, developers, and DBAs.
- **Oracle Premier Lifetime Support for MySQL** – Provide direct access to expert MySQL Support engineers who are ready to assist users in the development, deployment, and management of MySQL applications. The MySQL Support team is composed of seasoned MySQL developers and database experts who understand the issues and challenges users face because they've overcome these same challenges themselves.



MySQL Cloud Service

Oracle MySQL Cloud Service is built on MySQL Enterprise Edition and powered by Oracle Cloud, providing an enterprise-grade MySQL database service. It delivers the best in class management tools, self service provisioning, elastic scalability and multi-layer security.

Conclusion

MySQL is the most popular and widely used open source database in the world because of its performance, reliability, and ease of use. MySQL 5.7 builds on this momentum by providing DBAs and developers with the added performance, scalability, and across the board improvements needed when building the next generation of web-based, cloud-based, and embedded applications and services. These improvements, further demonstrating that Oracle drives MySQL innovation, make MySQL 5.7 by far the most comprehensive and feature rich release in our history.

Additional Resources

MySQL 8.0 Downloads

<http://www.mysql.com/downloads/>

MySQL 8.0 Documentation

<http://dev.mysql.com/doc/refman/5.7/en/>

MySQL Enterprise Edition

<http://mysql.com/products/enterprise/>

MySQL Customers and Case Studies

<http://www.mysql.com/customers>

MySQL Training

<https://www.mysql.com/training/>

For more information on Oracle's MySQL products and services, please visit:

<http://www.mysql.com/products/>