Why we're excited about MySQL 8

Practical Look for Devs and Ops

Peter Zaitsev, CEO

November 11, 2017

Percona University Kiev

PERCONA

In This Presentation

Few Words about Percona

Few Words about Percona University Program

Exiting things MySQL 8 has to bring

PERCONA

Thank you Creative Quarter

CREATIVE QUARTER

WARP YOUR BUSINESS

Creative Quarter is a Ukraine-based managed infrastructure with flexible office space, supplies, facilities management and a full range of operational services

PERCONA

Thank you CNA

Few Words about Percona

Percona's Purpose

To Champion Unbiased Open Source Database Solutions

We Do

Support, Managed Services for MySQL and MongoDB

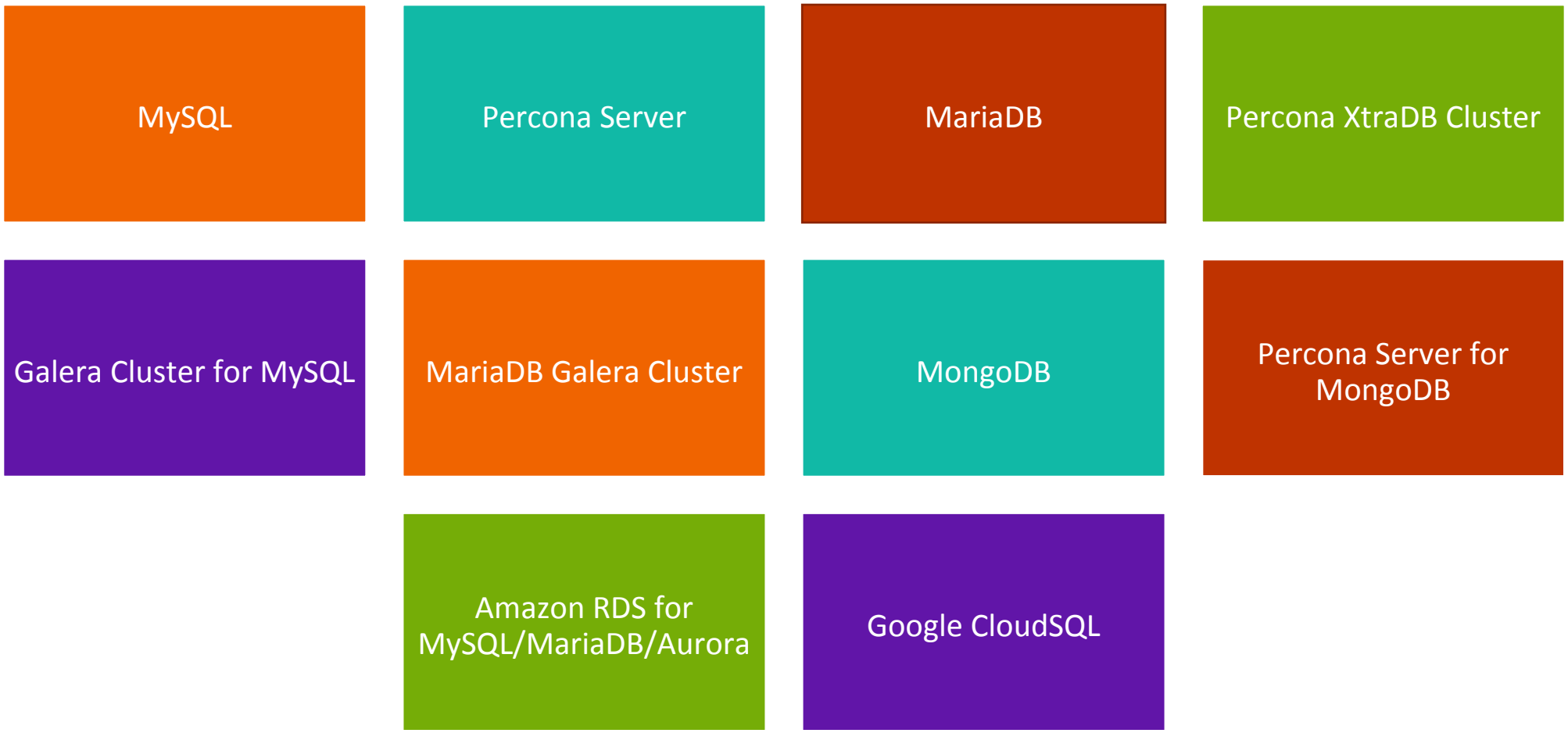
Also Consulting and Training

Helping companies to migrate to Open Source Database

Develop Open Source Software

Solutions to maximize your success

Broad Software Ecosystem Support



PERCONA

100% Free and Open Source Software

Percona Server for MySQL

Percona Server for MongoDB

Percona XtraDB Cluster

Percona Xtrabackup

Percona Toolkit

Percona Monitoring and Management

PERCONA

Percona University

Educational Technical Presentations

Multiple Locations in the World

Partnering with Local Companies

Affordable to Attend (Free)

PERCONA

Percona University – What to Expect

Several presentations on different topics

Feel free to only attend those you're interested in

Keep it Interactive! Ask Questions

Breaks

Prize Give away in the end

What’s Exciting in MySQL 8 ?

For Devs and Ops

Warning

This assessment is done for Pre-GA MySQL 8, based on documentation and limited testing. We're yet to see how they behave in production

Source Notes

Examples liberally borrowed from Oracle team presentations and Blog Posts

MySQL 8 for Ops

Ops care about

Native Data Dictionary

About 10 years overdue

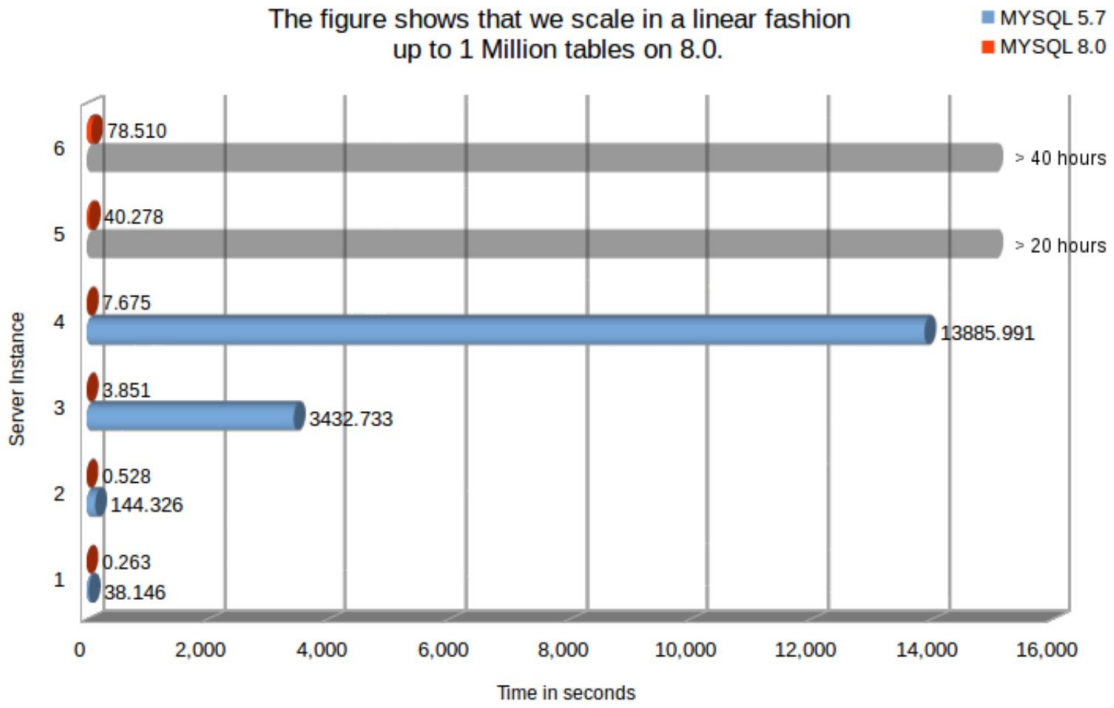
Atomic (Crash Save) DDLs

Much Faster Information Schema

No more MyISAM System Tables!

Fast Information Schema

```
1 SELECT t.table_schema, t.table_name, c.column_name
2   FROM information_schema.tables t,
3        information_schema.columns c
4   WHERE t.table_schema = c.table_schema
5         AND t.table_name = c.table_name
6         AND t.engine='InnoDB';
```

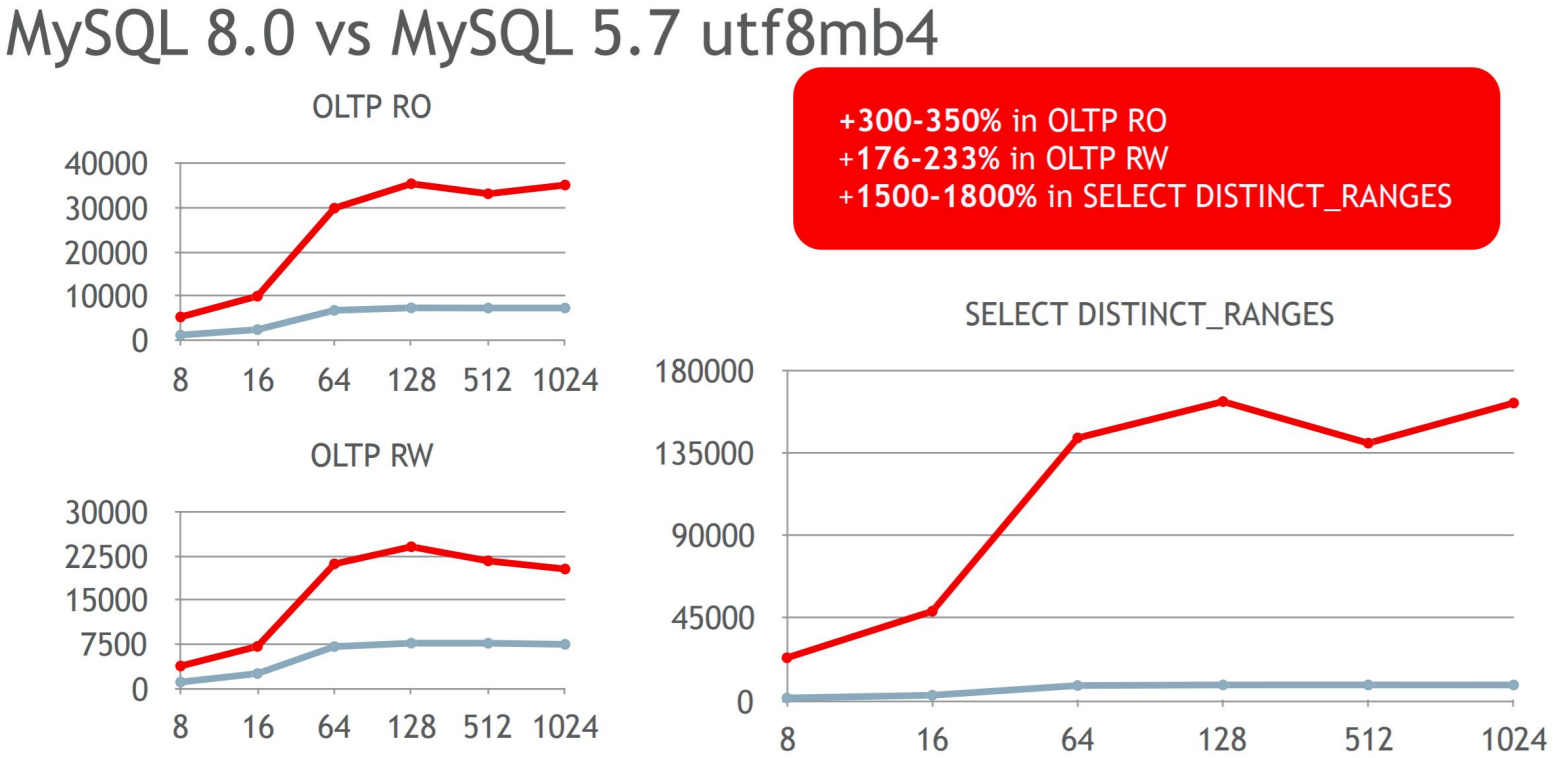


Scaling from 5.000 to 1.000.000 tables

PERCONA

Much Better and Faster UTF8

- **utf8mb4 as Default Charset**



Security

- ROLEs
- Breakdown of SUPER Privileges
- Password History
- Faster cached-SHA2 Authentication
- --skip-grants blocks remote connections
- Redo and Undo Logs are now encrypted if Table Encryption is enabled

Persistent Auto Increment

Another feature 10 years overdue

Do not reset AUTO INCREMENT to the max value in the table on restart

Auto-Managed Undo Tablespace

Do not use system table space for undo space any more

Automatically reclaim space on disk from large transactions

Self Tuning (limited to Innodb)

Set innodb_dedicated_server to auto-tune

- innodb_buffer_pool_size
- innodb_log_file_size
- innodb_flush_method

Partial In-Place Update for JSON

Can update field in JSON object without full re-write

Great for counters, statuses, timestamps etc

Only update and removal of element is supported

Only Optimizer and Replication support so far

Invisible Indexes

- Test impact of dropping indexes before actually dropping them
- Can use use_invisible_indexes to use invisible indexes in a session

```
1  CREATE TABLE t1 (
2    i INT,
3    j INT,
4    k INT,
5    INDEX i_idx (i) INVISIBLE
6  ) ENGINE = InnoDB;
7  CREATE INDEX j_idx ON t1 (j) INVISIBLE;
8  ALTER TABLE t1 ADD INDEX k_idx (k) INVISIBLE;
```

TmpTable Storage Engine

More efficient storage engine for Internal Temporary tables

Efficient storage for VARCHAR and VARBINARY columns

BLOB/TEXT Columns are not supported (yet?)

Backup Locks

Prevent operation which may result in inconsistent backups

LOCK INSTANCE FOR BACKUP

Optimizer Histograms

- **Detailed Statistics on Columns, not just Indexes**


```
{
  "buckets": [
    [
      1,
      0.3333333333333333
    ],
    [
      2,
      0.6666666666666666
    ],
    [
      3,
      1
    ]
  ],
  "null-values": 0,
  "last-updated": "2017-03-24 13:32:40.000000",
  "sampling-rate": 1,
  "histogram-type": "singleton",
  "number-of-buckets-specified": 128,
  "data-type": "int",
  "collation-id": 8
}
```

Improved Optimizer Cost Model

- Keep in account how much of data is cached vs on disk

More on MySQL 8 Optimizer

- **http://www.unofficialmysqlguide.com/**

Performance Schema

(Fake) Indexes for Faster Access

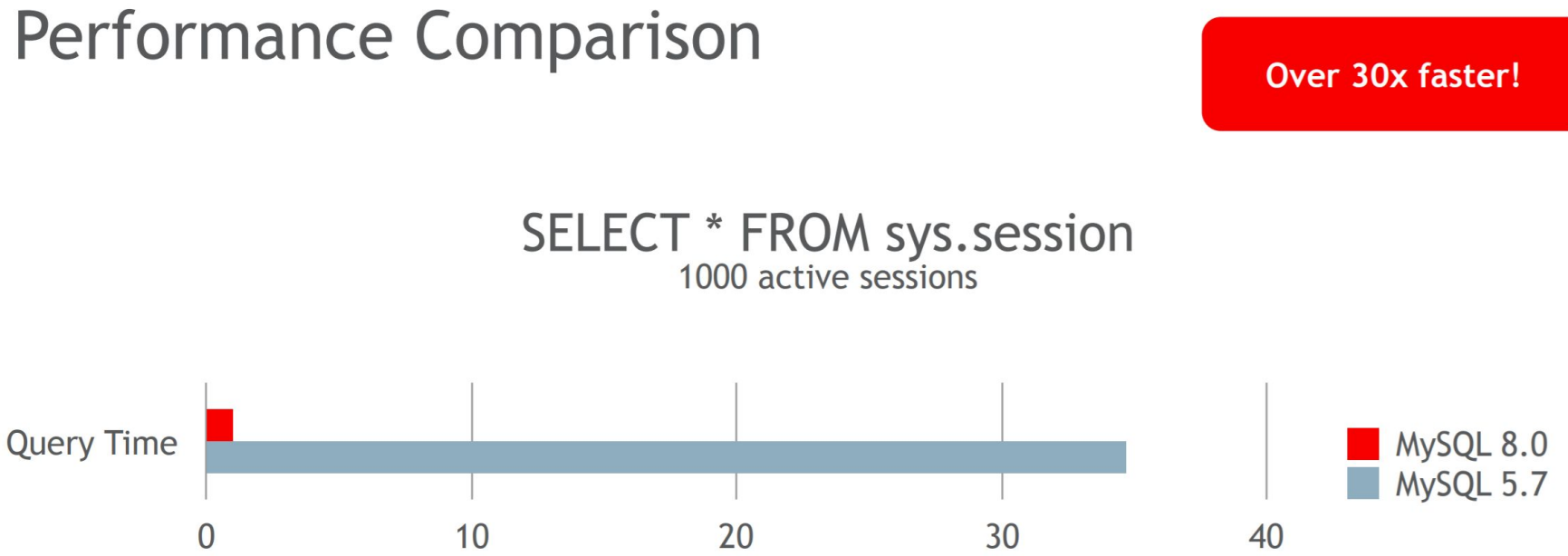
Error Instrumentation

Response Time Histograms (Global and Per Query Digest)

Query Examples for Summary by Digest

Performance Schema Performance

- **Now is Interactivelv Usable at Scale**



Persistent Global Variables

```
SET PERSIST innodb_buffer_pool_size = 1024 * 1024 * 1024;
```

Assumes storage is SSD by Default

- Start of the long journey

Binary Log On by Default

bin_log is enabled by default

log_slave_updates is enabled by default

Expire logs after 30 days by default

Query Cache Removed

It’s design caused more problems than it fixed

Use ProxySQL (or other) external query cache instead

Native Partitioning Only

Only “Native” Partitioning supported, not Generic One

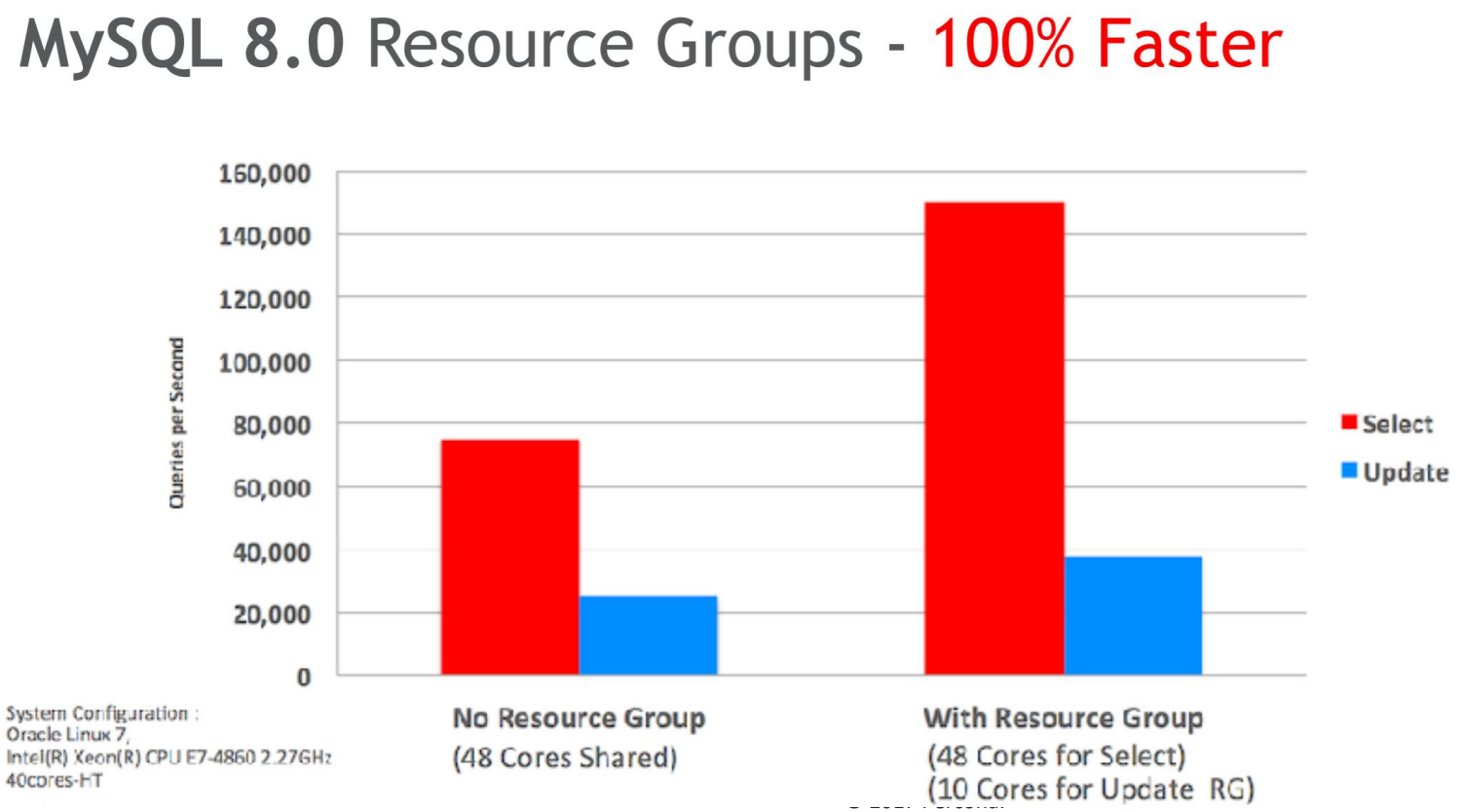
Remove partitions from MyISAM partitioned tables or convert them

ALTER TABLE ... REMOVE PARTITIONING

ALTER TABLE ... ENGINE=INNODB

Resource Groups

- **Isolation and Better Performance**



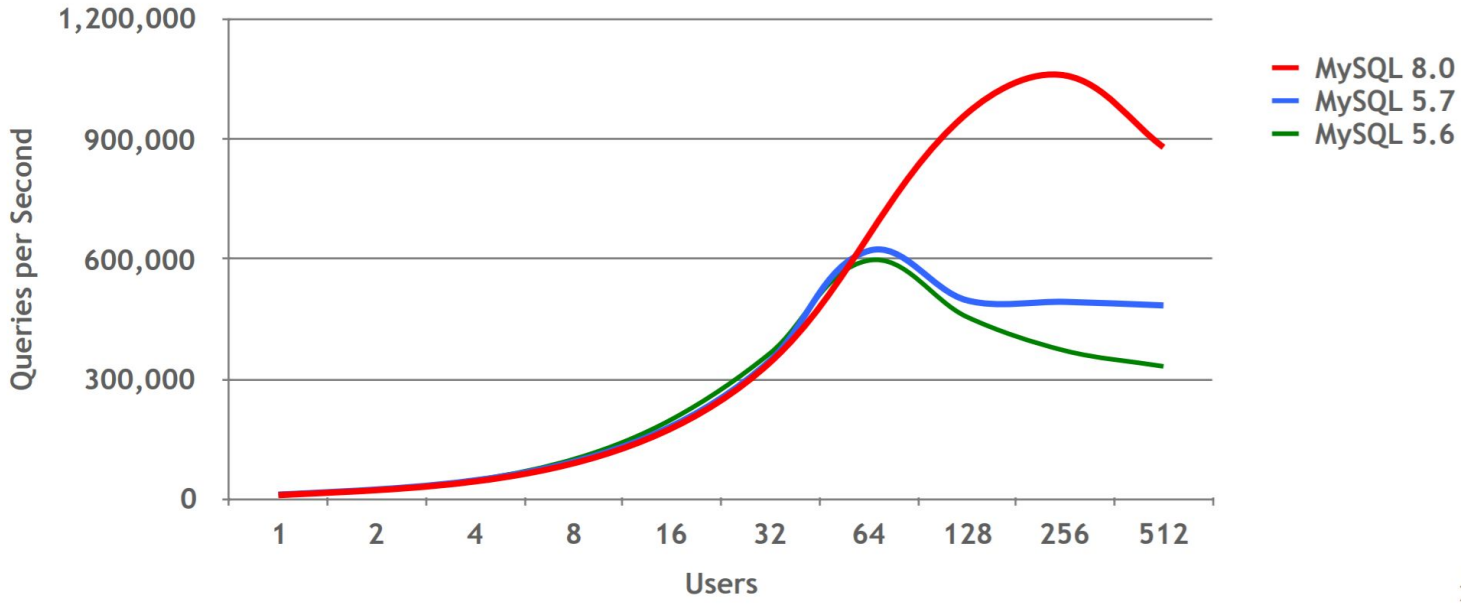
PERCONA

Plain Better Performance at Scale

Sysbench: OLTP_RO Point-Selects

2.1x Faster than MySQL 5.7

2.8x Faster than MySQL 5.6



PERCONA

Feature Requests

Better Single Thread Performance

Parallel Single Query Processing Please

MySQL 8 for Devs

Innodb NO WAIT and SKIP LOCKED

New! Better Handling of Hot Row Contention

```
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

Descending Indexes

Descending flag in index definition is no more ignored

Allows efficient handling of ORDER BY A ASC, B DESC queries

Aggregation of Relational into JSON

- JSON_ARRAYAGG() and JSON_OBJECTAGG()

```
mysql> SELECT id, col FROM t1;
+------+--------------------------------------+
| id   | col                                  |
+------+--------------------------------------+
|    1 | {"key1": "value1", "key2": "value2"} |
|    2 | {"keyA": "valueA", "keyB": "valueB"} |
+------+--------------------------------------+
2 rows in set (0.00 sec)

mysql> SELECT JSON_OBJECTAGG(id, col) FROM t1;
+-----------------------------------------------------------------------------------------+
| JSON_OBJECTAGG(id, col)                                                                 |
+-----------------------------------------------------------------------------------------+
| {"1": {"key1": "value1", "key2": "value2"}, "2": {"keyA": "valueA", "keyB": "valueB"}}  |
+-----------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```


PERCONA

JSON to Table Conversion (Labs)

Labs

```
SET @doc=(SELECT doc->"$.properties.amenities"
FROM seats WHERE id = 28100);

SELECT * FROM json_table(@doc, "$[*]" columns (
  id for ordinality, amenity_type varchar(100) path "$.type",
  distance float path '$.distance_in_meters')
) AS amenities
WHERE amenity_type IN ('snacks', 'bar')
ORDER BY distance;
+------+--------------+----------+
| id   | amenity_type | distance |
+------+--------------+----------+
|    2 | bar          |  100.538 |
|    3 | snacks       |  136.647 |
+------+--------------+----------+
2 rows in set (0.00 sec)
```


PERCONA

Better JSON Document Data Extraction

```
mysql> CREATE TABLE t1 (doc JSON);
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO t1 VALUES ('[1, 2, 3, 4, 5]');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT doc->"$[1 to 3]" FROM t1;
+------------------+
| doc->"$[1 to 3]" |
+------------------+
| [2, 3, 4]        |
+------------------+
1 row in set (0.00 sec)

mysql> SELECT doc->"$[last-2]" FROM t1;
+------------------+
| doc->"$[last-2]" |
+------------------+
| 3                |
+------------------+
1 row in set (0.00 sec)
```


PERCONA

Common Table Expression

- **Recursive and Non-Recursive**

```
1  WITH RECURSIVE cte AS
2  (
3    SELECT category_id, name, 0 AS depth FROM category WHERE parent IS NULL
4    UNION ALL
5    SELECT c.category_id, c.name, cte.depth+1 FROM category c JOIN cte ON
6      cte.category_id=c.parent
7  )
8  SELECT * FROM cte ORDER BY depth;
9  +-------------+----------------------+-------+
10 | category_id | name                 | depth |
11 +-------------+----------------------+-------+
12 |           1 | ELECTRONICS          |     0 |
13 |           2 | TELEVISIONS          |     1 |
14 |           6 | PORTABLE ELECTRONICS |     1 |
15 |           7 | MP3 PLAYERS          |     2 |
16 |           9 | CD PLAYERS           |     2 |
17 |          10 | 2 WAY RADIOS         |     2 |
18 |           3 | TUBE                 |     2 |
19 |           4 | LCD                  |     2 |
20 |           5 | PLASMA               |     2 |
21 |           8 | FLASH                |     3 |
22 +-------------+----------------------+-------+
```


PERCONA

Window Functions

- **Like GROUP BY, But Preserving Rows rather than collapsing them**

```
mysql> CREATE TABLE t(i INT);
mysql> INSERT INTO t VALUES (1),(2),(3),(4);

mysql> SELECT SUM(i) AS sum FROM t;
+------+
| sum  |
+------+
|   10 |
+------+

mysql> SELECT i, SUM(i) OVER () AS sum FROM t;
+------+------+
| i    | sum  |
+------+------+
|    1 |   10 |
|    2 |   10 |
|    3 |   10 |
|    4 |   10 |
+------+------+
```


PERCONA

Much Better GIS

- **“Matching or Exceeding PostgreSQL GIS Feature Set”**

5.7	8.0
• The world is flat	• The world can be flat or ellipsoidal
• The world is infinite	• Geographic coordinate systems wrap around
• Axes are unitless	• Axes have units
• Axes are orthogonal	• Geographic axes are not orthogonal
• Axis order is irrelevant	• Geographic axis order matters
• Axis direction is irrelevant	• Axis direction may be relevant

MySQL Document Store

Full Text Indexing

GeoJSON Support

Anyone Using Document Store ?

Summary

MySQL 8 looks like release to be excited about

Has a lot of new features both for Devs and Ops

Before we take a Break…

SAVE THE DATE!

April 23-25, 2018
Santa Clara Convention Center

PERCONA
LIVE

CALL FOR PAPERS OPENING SOON!

www.perconalive.com


BECOME A
PERCONA SUPERHERO
WE'RE HIRING
CONTACT
careers@percona.com
PERCONA

Have a Friend ?

Refer a friend and get $1000 if one is hired by Percona

… and eternal gratitude for helping to get the most amazing job

Some of our the people we're looking for

- MySQL and MongoDB Operations Experts
- C/C++ Database Kernel Engineers
- Backend Engineers (Go)
- Technical Writers
- Front End Web Developers
- Front End Focused QA Engineer
- PMM Dashboard Intern

Thank You!