MySQL Monitoring

Introduction

MySQL provides several tools and features for monitoring and optimizing database performance, and the Slow Query Log, Performance Schema, and SYS Schema are essential components for database administrators and developers to gain insights into MySQL's performance. By using the Slow Query Log, Performance Schema, and SYS Schema together, you can effectively monitor and optimize MySQL database performance. These tools provide valuable information about slow queries, resource usage, and overall database health, helping you to identify and address performance bottlenecks.

In this lab you will work with all three tools

Estimated Lab Time: 40 minutes

Objectives

In this lab, you will:

- Install airportdb
- Query Slow Query Log
- Query Performance Schema
- Query SYS Schema
- View Oracle Enterprise Manager for Monitoring MySQL Live Demo

Task 1: Install Airportdb

1. If not already connected to app-srv and mysql1 then do the following Connect with your SSH client using the public IP and the provided ssh Example of connections from Linux, MAC, Windows Powershell

```
<copy> ssh -i id_rsa_app-srv opc@<public_ip></copy>
```

2. Connect to shell-mysql1

```
<copy> ssh -i $HOME/sshkeys/id_rsa_mysql1 opc@mysql1 </copy>
```

3. Download the airportdb database file

```
<copy>wget https://downloads.mysql.com/docs/airport-db.zip</copy>
```

4. Unzip airportdb database file

```
<copy>unzip airport-db.zip</copy>
```

5. Login to MySQL

```
<copy>mysqlsh -uadmin -p -h 127.0.0.1 -P3307</copy>
```

6. Load the airportdb data files into the MySQL using the MySQL Shell Dump Loading utility.

```
<copy>\js</copy>
```

```
<copy>util.loadDump("airport-db", {threads: 16, deferTableIndexes: "all",
ignoreVersion: true, loadIndexes:false})</copy>
```

7. List databases

```
<copy>\sql</copy>
```

```
<copy>SHOW DATABASES;</copy>
```

8. Exit MySQL

```
<copy>\q</copy>
```

Task 2: Using Slow Query Log to monitor and analyze slow queries in MySQL

The Slow Query Log is a feature in MySQL that allows you to log queries that take longer than a specified threshold to execute. This is particularly useful for identifying and optimizing slow-running queries that may impact the overall performance of your database. The following steps help identify performance bottlenecks in your database by recording queries that take longer than a specified threshold to execute.

1. Login to MySQL

```
<copy>mysqlsh -uadmin -p -h 127.0.0.1 -P3307</copy>
```

2. Slow Query Log Variables

```
<copy>show variables like '%slow%';</copy>
```

3. Slow Query Log Variables

```
<copy>show variables like '%query%';</copy>
```

4. Slow Query Log Variables

```
<copy>show variables like '%examined%row%';</copy>
```

5. Set slow_query_log

```
<copy>SET GLOBAL slow_query_log='ON';</copy>
```

6. Set long_query_time

```
<copy>SET GLOBAL long_query_time = 1;</copy>
```

7. Change database to airportdb

```
<copy>use airportdb;</copy>
```

8. Show Tables

```
<copy>show tables;</copy>
```

9. Execute Query

```
<copy>SELECT * FROM booking ORDER BY RAND() limit 100;</copy>
```

10. Exit MySQL

```
<copy>\q</copy>
```

11. Read Slow Query Log File

```
<copy>sudo nano /mysql/log/sq_log.log;</copy>
```

12. Use mysqldumpslow to get summarized contents

```
<copy>sudo $(which mysqldumpslow) /mysql/log/sq_log.log</copy>
```

Task 3: Using MySQL Performance Schema for database monitoring and query analysis

The Performance Schema in MySQL is a powerful tool for monitoring and analyzing the internal execution of SQL statements and their impact on the database system. It provides detailed information about various aspects of MySQL's performance. It captures information about executed statements, table IO, memory usage, locking, and more. The following steps help identify performance bottlenecks, monitor query execution times, and analyze database operations.

1. Login to your mysql

```
<copy>mysqlsh -uadmin -p -h 127.0.0.1 -P3307</copy>
```

2. Verify if Performance Schema is enabled

```
<copy>SHOW VARIABLES LIKE 'performance_schema';</copy>
```

3. Change to performance_schema

```
<copy>use performance_schema;</copy>
```

4. Show Tables

```
<copy>show tables;</copy>
```

5. Change Database

```
<copy>use airportdb;</copy>
```

6. Run Query

```
<copy>SELECT a.airlinename, f.flightno, f.departure, f.arrival, b.seat,
b.price FROM airline a JOIN flight f ON a.airline_id = f.airline_id JOIN
booking b ON f.flight_id = b.flight_id WHERE a.airlinename like '%Air%'
ORDER BY f.departure, b.seat limit 10;</copy>
```

7. Get event_id for the query

```
<copy>SELECT EVENT_ID, TRUNCATE(TIMER_WAIT/1000000000000,6) as Duration,
SQL_TEXT FROM  performance_schema.events_statements_history_long WHERE
SQL_TEXT like '%airlinename%limit%';</copy>
```

8. Get query stage information

```
<copy>SELECT event_name AS Stage, TRUNCATE(TIMER_WAIT/1000000000000,6) AS
Duration FROM performance_schema.events_stages_history_long WHERE
NESTING_EVENT_ID=Entery event_id from previous query;</copy>
```

9. Get information about query from events_statements_history

```
<copy>select * FROM  performance_schema.events_statements_history Where
sql_text like '%flight%airlinename%limit%'\G;</copy>
```

Task 4: Using MySQL's Sys Schema to provide simplified views for performance analysis

The SYS Schema is a collection of views, procedures, and functions designed to simplify the process of monitoring and understanding MySQL performance. SYS Schema provides an easy-to-use interface for querying performance data, making it more accessible to DBAs and developers. The following steps help identify performance bottlenecks, track database usage patterns, and prioritize optimization efforts.

1. Switch to the sys schema database

```
<copy>use sys;</copy>
```

2. Queries with full table scans

```
<copy>select query, db, exec_count, total_latency, rows_sent, rows_examined,
rows_sent_avg, rows_examined_avg from sys.statements_with_full_table_scans;
</copy>
```

3. Query to get table statistics

```
<copy>select
table_schema,table_name,total_latency,rows_fetched,fetch_latency,rows_insert
ed, insert_latency,rows_updated,update_latency,rows_deleted
delete_latency,io_read_requests,io_read,io_read_latency from
sys.schema_table_statistics;</copy>
```

Task 5: Oracle Enterprise Manager for Monitoring MySQL Live Demo

Acknowledgements

- **Author** - Marco Carlessi, Principal Sales Consultant
- **Contributors** - Satish Senapathy, Senior MySQL Solution Engineer, Selena Sánchez, MySQL Solutions Engineer
- **Last Updated By/Date** - Perside Foster, MySQL Solution Engineering, March 2025