Practical MySQL

Session 11

Performance Optimization in MySql

Session Overview

- Explain optimizing queries with MySQL query optimization guidelines
- Outline the process of optimizing workload
- Define how to monitor fundamental resources
- Describe restructuring queries
- Describe optimizing subqueries
- Identify scalability problems
- Explain queue cache
- Describe and explain automating of configuration
- Explain specific types of queries

Optimizing Queries with MySQL Query Optimization Guidelines

Optimization refers to the performance increase of any given system. Currently, many applications are database-driven.

If the SQL Queries are not written properly or if they are improperly designed, then it degrades the database application performance.

In such cases, planning for optimizations or troubleshooting the queries and configurations after understanding the crux of the problem is essential.

Benefits of MySQL Database Queries Optimization

- Optimization improves the speed at which queries are executed and aids in managing the resources.
- Optimized queries run faster by taking less time and less computing power.
- Optimization reduces the cost or price of hardware.
- Optimization allows the server to efficiently run by consuming less power and less memory.

Optimizing Queries [1-2]

By using EXPLAIN statement: The `EXPLAIN` statement shows how MySQL executes a statement. It provides information on what the statement would do, how it would be processed, how the table is joined, and so on.

By normalizing the tables: Normalization keeps data non-redundant and avoids repetition of lengthy values such as name, categories, and contact details.

By using the most appropriate data types: Proper data types should be used in columns so that the columns can take the values that belong only to the specified datatype thereby minimizing the usage of space.

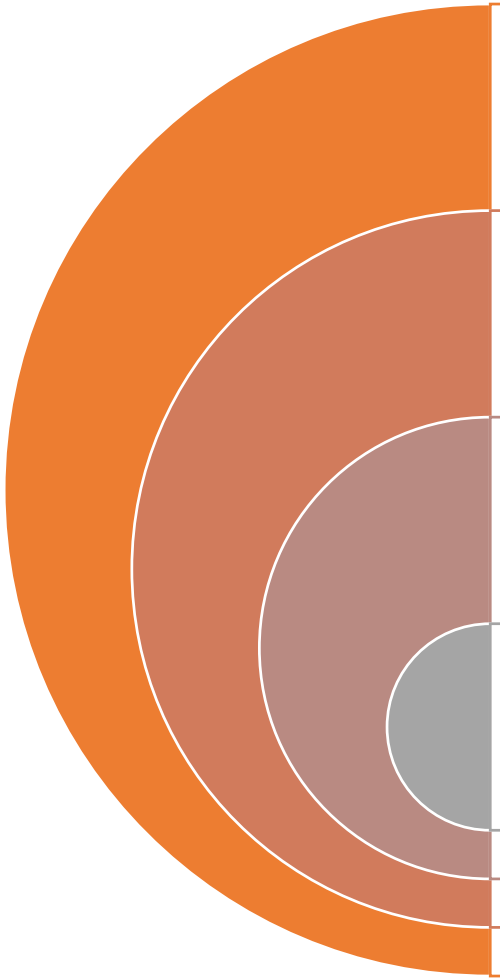
By avoiding Null values: Using the NOT NULL constraint for the columns ensures that the column can never have NULL values and ensures that the index mechanism works.

By using an index for required columns: Data retrieval time can be reduced by keeping the index on a column that is always in use. It will be easy to find the specific value faster by keeping indexes on the columns.

Optimizing Queries [2-2]

- **By using wildcard characters**: Wildcard characters are used in conjunction with `MySQL LIKE` Operator.
- **By using LIMIT**: Most often, only a specified number of rows is required from a result set.
- **By Using MySQL Query Caching**: Query caching provides database caching functionality that stores the result set.
- **By converting OUTER JOIN to INNER JOIN**: Convert the `OUTERJOIN` to `INNER JOIN` to enhance performance.

Understanding the Workload



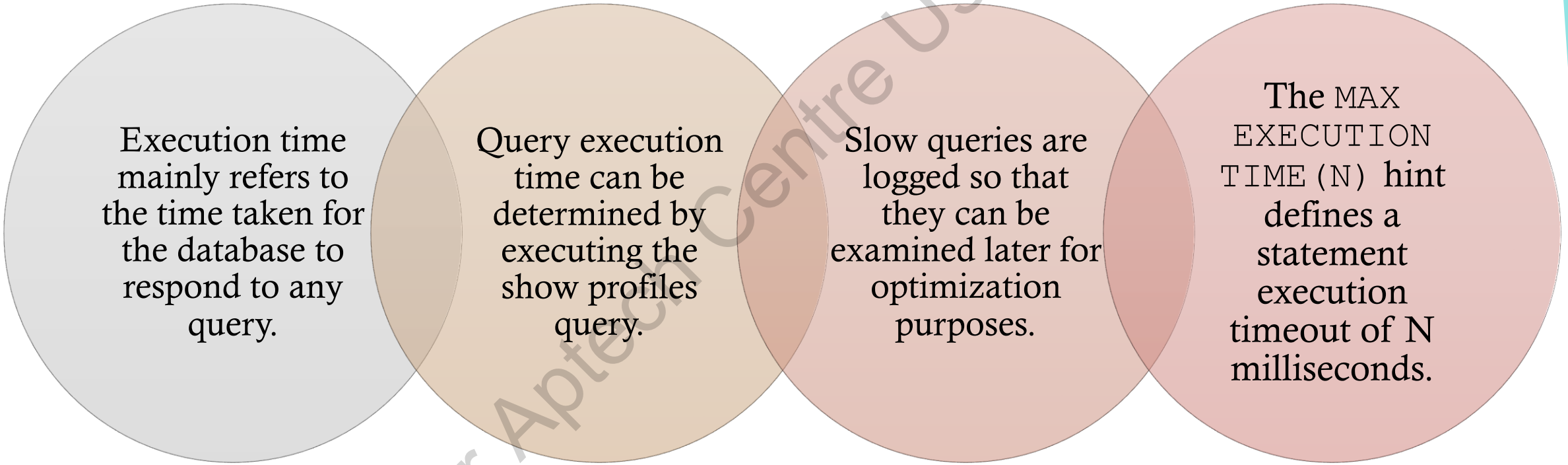
The workload of the MySQL database can be explained by understanding what is currently happening with the database.

In absence of accurate baseline and workload profiling, it is very difficult to identify problems and determine how to use potential optimization measures.

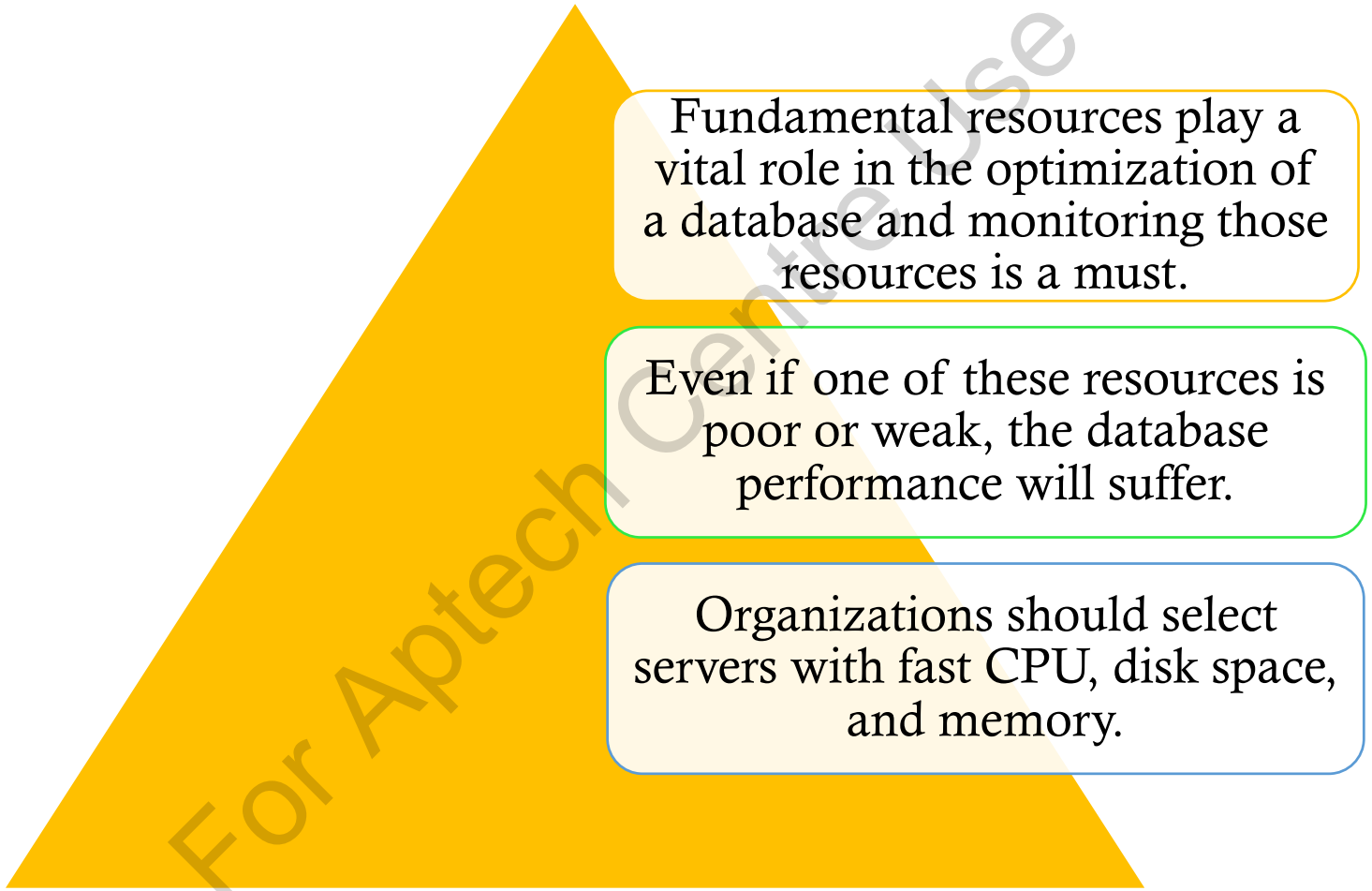
Looking at the workload, the user can find out which query is more expensive and is affecting the performance of the database.

The user can profile or baseline the database performance by using tools such as To do list, Buffer, and so on.

Execution Time



Monitoring Fundamental Resources



Fundamental resources play a vital role in the optimization of a database and monitoring those resources is a must.

Even if one of these resources is poor or weak, the database performance will suffer.

Organizations should select servers with fast CPU, disk space, and memory.

Restructuring Queries

Restructuring queries is another prominent way to achieve query optimization.

Too many rows: A common mistake made while writing a query is that it retrieves data.

Too many columns: Obtaining unnecessary columns can also affect performance and is considered a common mistake.

Too much data analysis: Data analysis can maximize the time for execution.

Chopping Queries

Chopping the query means slicing up a big or complex query and running it in smaller chunks so that it affects only a smaller quantity of rows each time..

Chopping the queries is a concept that comes into the picture when the time spent fetching the result set keeps increasing.

There are two types of queries. The very first one is complex queries, and the second one is Simple queries.

Complex queries are the major ones that are mainly to be chopped to get good performance.

Identifying Scalability Problems

- MySQL was designed for a single node system and not for modern data centers.
- When considering scalability, scale or enhancement is not the only thing to take care of. What to do with the data and how to keep it while scaling should also be given a thought.
- Scalability can be upwards or downwards. In other words, it can mean an increase or decrease in hardware.
- In managerial terms, scalability refers to an application that can perform well even though it grows. This must be accounted for as well.

Query Cache and Automating Configuration

Query Caches is a technique that saves the text from the `SELECT` statement and its resultant output in cache. In the occasion of an identical statement being executed at a later point, the resultant is received from the query cache and displayed to the client or user. This technique works with the `SELECT` statement and is utilized when the table data is seldom changed.

Configuration mainly points to the arrangement or set-up of any system. Automating this process without regular human intervention is what automatic configuration refers to.

Optimizing Specific Types of Queries

Optimizing `COUNT()` Queries: `COUNT()` queries are one of the most frequently used queries in MySQL.

Simple Optimizations: Simple Optimization in MySQL is the procedure of selecting an efficient means of executing an SQL statement.

Optimizing `JOIN` Queries: `JOIN` is a concept where a result set can be retrieved from more than one table based on specific conditions. Imprudent use of the `JOIN` statement can result in affecting the data retrieval performance.

Optimizing `LIMIT` and `OFFSET`: `LIMIT` is mainly used to obtain the specified number of rows and it optimizes the SQL Queries depending on the count of rows to be fetched.

Summary

- An application executes multiple queries on the database concurrently and if the SQL Queries are improperly written or if they are badly designed, then it degrades the database application performance.
- Optimization improves the pace at which queries are executed and manages the resources.
- MySQL has a ceiling of 4,096 columns per table.
- In absence of accurate baseline and workload profiling, it is very difficult to discover problems and utilize potential optimization measures.
- Query Caches is a technique that stores the text of the SELECT statement and its result and uses it when a similar statement comes by.
- The procedure of selecting an efficient means of executing an SQL statement in MySQL is called Simple Optimization.