Hewlett Packard Enterprise

HPE-CTY

Network Flow Database

A Report On

MariaDB

Documented By:
M N Nagaashrith

Introduction :

MariaDB is an open-source relational database management system (RDBMS) that is a fork of MySQL. It was created by the original developers of MySQL after concerns arose over the acquisition of MySQL by Oracle Corporation. MariaDB is designed to maintain compatibility with MySQL while also offering additional features, performance improvements, and security enhancements. It is widely used in various applications ranging from small personal databases to large-scale enterprise systems. MariaDB is named after Maria, the daughter of its main developer, Michael "Monty" Widenius.

Description of MariaDB :

MariaDB is a powerful open-source relational database management system (RDBMS) that offers a wide range of features suitable for various use cases.

Relational Database Management System (RDBMS): MariaDB follows the relational database model, storing data in tables with rows and columns, and supporting SQL (Structured Query Language) for data manipulation and retrieval.

Open Source: Being open source, MariaDB is freely available for use, modification, and distribution under the GNU General Public License (GPL).

Compatibility with MySQL: MariaDB maintains compatibility with MySQL, allowing for easy migration from MySQL to MariaDB with minimal or no code changes.

High Performance: MariaDB is optimized for high performance, supporting features such as multi-threading, parallel replication, and query optimization to handle large volumes of data efficiently.

Scalability: MariaDB is designed to scale horizontally and vertically, making it suitable for both small-scale deployments and large-scale enterprise environments.

High Availability: MariaDB offers features like master-slave replication, clustering, and automatic failover to ensure high availability and data redundancy.

Security: MariaDB provides various security features such as user authentication, role-based access control, encryption, and auditing to protect data integrity and confidentiality.

Main use cases of MariaDB :

Web Applications: MariaDB is commonly used as the backend database for web applications, powering content management systems (CMS), e-commerce platforms, social media sites, and more.

Enterprise Applications: MariaDB is suitable for enterprise-level applications such as customer relationship management (CRM), enterprise resource planning (ERP), and business intelligence (BI) systems.

Analytics and Reporting: MariaDB can be used for storing and analyzing large volumes of data, making it suitable for analytics, reporting, and data warehousing applications.

Mobile Applications: MariaDB can serve as the backend database for mobile applications, providing data storage, synchronization, and offline capabilities.

IoT (Internet of Things): MariaDB can handle the data generated by IoT devices, storing sensor data, telemetry data, and other types of IoT data for analysis and processing.

DevOps and Cloud-Native Environments: MariaDB is used in DevOps and cloud-native environments for database-as-a-service (DBaaS), containerized deployments, and microservices architectures.

Reasons why it fits for implementing the network flow database :

Implementing a network flow database involves storing and analyzing network traffic data, which requires a database system capable of handling large volumes of data efficiently, supporting complex queries, and providing high availability and scalability. MariaDB is well-suited for implementing a network flow database due to several reasons:

Performance: MariaDB is optimized for high performance, with features such as multi-threading and query optimization, making it capable of handling the high throughput of network traffic data.

Scalability: MariaDB can scale both horizontally and vertically, allowing for the addition of more resources or nodes as the volume of network traffic data increases.

Data Integrity and Security: MariaDB provides robust security features such as user authentication, role-based access control, and encryption to protect the integrity and confidentiality of network flow data.

Compatibility with Analytical Tools: MariaDB is compatible with various analytical and visualization tools, allowing for the analysis and visualization of network flow data to identify patterns, anomalies, and security threats.

Open Source and Cost-Effective: MariaDB is open source and freely available, making it a cost-effective solution for implementing a network flow database compared to proprietary database systems.

MariaDB's performance, scalability, high availability, security, compatibility, cost-effectiveness, and community support make it a suitable choice for implementing a network flow database to store, analyze, and manage network traffic data effectively.

Advantages of MariaDB :

Open Source: MariaDB is open source and freely available, which means there are no licensing costs associated with using it. This can lead to significant cost savings, especially for organizations with limited budgets.

Compatibility with MySQL: MariaDB maintains compatibility with MySQL, allowing for easy migration from MySQL to MariaDB with minimal or no code changes. This ensures smooth transition for existing MySQL users.

Performance: MariaDB is optimized for high performance, with features such as multi-threading, query optimization, and indexing. This enables efficient handling of large volumes of network flow data and complex queries.

Scalability: MariaDB can scale both horizontally and vertically, allowing for seamless expansion of resources as the volume of network flow data grows. This ensures that the database can handle increasing demands without sacrificing performance.

High Availability: MariaDB offers features such as master-slave replication, clustering, and automatic failover, ensuring high availability and continuous access to network flow data even in the event of hardware failures or network disruptions.

Security: MariaDB provides robust security features such as user authentication, role-based access control, encryption, and auditing to protect the integrity and confidentiality of network flow data.

Disadvantages of MariaDB :

Complexity of Setup and Configuration: Setting up and configuring MariaDB, especially for high availability and scalability, can be complex and require expertise in database administration. This may pose challenges for organizations with limited technical resources.

Resource Intensive: Running MariaDB in high-performance and high-availability configurations may require significant hardware resources, including CPU, memory, and storage. This can increase infrastructure costs, especially for large-scale deployments.

Learning Curve: While MariaDB maintains compatibility with MySQL, there may still be a learning curve for users transitioning from other database systems or those new to relational databases. Training and familiarization may be required for effective utilization of MariaDB's features and capabilities.

Community Support: While MariaDB has a large and active community of users and developers, the level of community support may vary compared to commercial database systems. Organizations relying heavily on community support may face challenges in obtaining timely assistance for complex issues.

Vendor Lock-In: While MariaDB is open source, organizations relying heavily on proprietary features or extensions specific to MariaDB may face vendor lock-in, limiting their ability to migrate to alternative database solutions in the future.

Few more MariaDB properties :

Relational/Non-Relat	In memory/Disk based (RAM v/s SSD)	Publisher-Subscriber notification support for row/column updates (Y/	Partial key search support (Y/N) (Update with performance data)	Opensource License used	Client Library language support (C, Python, Go, etc?)
Relational	Disk based	No	Yes	GNU General Public License (GPL) version 2	C/C++,Python,Go,Java,Node.js,PHP

1. Relational/Non-Relations :

MariaDB is a relational database management system (RDBMS). It is an open-source fork of MySQL and uses Structured Query Language (SQL) for defining, manipulating, and querying data. As an RDBMS, MariaDB organizes data into tables with predefined schemas, consisting of rows and columns, and supports key relational database features like:

1. ACID Compliance
2. SQL Queries
3. Data Integrity
4. Normalization

2. In memory/Disk based :

MariaDB is primarily a disk-based database management system. It stores data on disk, allowing it to handle large datasets that exceed the size of available memory. However, MariaDB also leverages in-memory operations to improve performance, such as using memory for caching data, query results, and indexes to speed up access times.

3. Publisher-Subscriber Notification Support for Row/Column Updates:

MariaDB does not natively support a built-in publisher-subscriber (pub/sub) notification mechanism specifically for row or column updates. Unlike some databases like PostgreSQL, which have the LISTEN and NOTIFY commands for pub/sub functionality, MariaDB lacks this feature out of the box.

4. Partial Key Search Support:

MariaDB supports partial key searches through various means, allowing you to efficiently query parts of your data within a column. Here are several methods to achieve partial key searches in MariaDB

5. Open Source License Used:

MariaDB is licensed under the GNU General Public License, version 2 (GPLv2). This open source license allows users to freely use, modify, and distribute the software, provided that any distributed versions of the software or derivative works are also licensed under the GPLv2. This ensures that the software and any modifications to it remain open and available to the community.

6. Client Library Language Support:

MariaDB provides client libraries for a variety of programming languages, allowing developers to connect to and interact with MariaDB databases from their applications. Here are the primary languages supported by official or community-supported client libraries:

- C and C++
- Java
- Python
- PHP
- Node.js
- Ruby
- Perl
- NET
- Go
- Rust.
- R

7. Multi-Threaded Publisher/Subscriber Support:

MariaDB does not have built-in support for a multi-threaded publisher/subscriber (pub/sub) model in the same way that some message brokers do (such as RabbitMQ, Kafka, or Redis Pub/Sub), you can implement a pub/sub mechanism using a combination of database features, external tools, and multi-threading in your application. Here's a comprehensive approach to achieving multi-threaded pub/sub functionality with MariaDB.

Description of the table present in the Database :

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| scr_ip     | varchar(45) | NO   | PRI | NULL    |       |
| des_ip     | varchar(45) | NO   | PRI | NULL    |       |
| scr_port   | int(11)     | NO   | PRI | NULL    |       |
| des_port   | int(11)     | NO   | PRI | NULL    |       |
| ip_version | varchar(10) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

- A table is created with the fiends source_ip, destination_ip, source_port, destination_port and ip_version.
- All the five tuples are made primary key.
- source_ip, destination_ip and ip_version are of type varchar.
- source_port, destination_port are of type integer.

Default Database Setting :

Memory and Cache Settings

key_buffer_size:

16M Size of the buffer used for index blocks for MyISAM tables.

innodb_buffer_pool_size:

128M The size of the buffer pool where InnoDB caches table and index data.

Innodb_log_file_size:

48M Size of each log file in the log group for InnoDB. Innodb_log_buffer_size:

8M The size of the buffer that InnoDB uses to write to the log files on disk.

Server Settings

port: 3306

The default port on which the MariaDB server listens for incoming connections.

bind-address: 127.0.0.1

The IP address that the server binds to. By default, it binds to localhost, restricting access to local connections only.

Socket:

/var/run/mysqld/mysqld.sock on Linux or C:/Program Files/MariaDB X.Y/data/mysql.sock on Windows. The socket file for local connections.

Performance Settings

query_cache_size: 1M The amount of memory allocated for caching query results.

max_connections: 151 The maximum permitted number of simultaneous client connections.

thread_cache_size: 8 The number of threads to keep in the cache for reuse

Code Explanation :

Libraries Used

- csv: This is a built-in Python library used for reading and writing CSV (Comma Separated Values) files.
- mysql.connector: This is a MySQL driver written in Python. It allows Python code to interact with MySQL (or MariaDB) databases.
- Time: This measures the time.

Connecting to the Database

```
# Connect to MariaDB
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="ashrith789",
    database="ashrith"
)
```

- host: The hostname where the database server is running (e.g., `localhost` for local server).
- user: The username to authenticate with the database server.
- password: The password for the given username.
- database: The name of the specific database to connect to.

Example CSV File Format

CSV File format:

scr_ip ,des_ip, scr_port, des_port, ip_version
192.168.1.1,192.168.1.2,1234,80,IPv4
192.168.1.3,192.168.1.4,1235,443,IPv4

API Versions and Compatibility

mysql-connector-python: Compatible with Python 3.6 and above. Ensure your MySQL server is compatible with the MySQL Connector/Python version you are using. For MariaDB, this connector is also compatible since MariaDB is a drop-in replacement for MySQL.

Bulk Operations :

Bulk Operations					
	100	1,000	10,000	1,00,000	10,00,000
Insertion	0.013	0.119	0.323	2.18	50.72
Updation	0.012	0.023	0.041	0.376	-
Deletion	0.16	4.15	25.13	222.49	-

CPU and Memory utilization :

		CPU %	Memory %
Bulk Operations	Insertion	26.4	1.7
	Updation	35	1.5
	Deletion	18	2.5

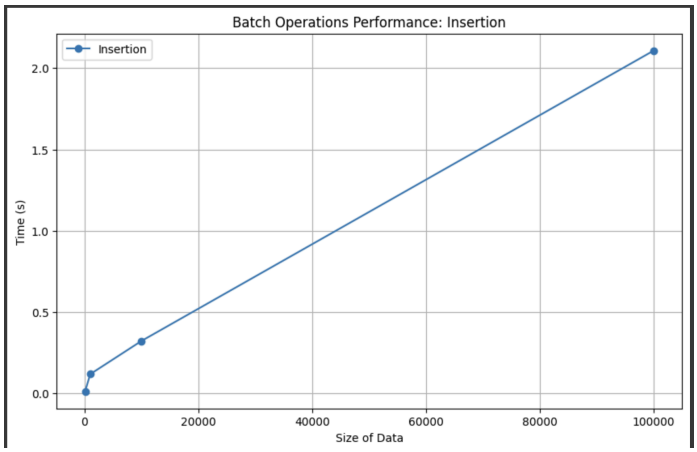
Code Links for Bulk Operations

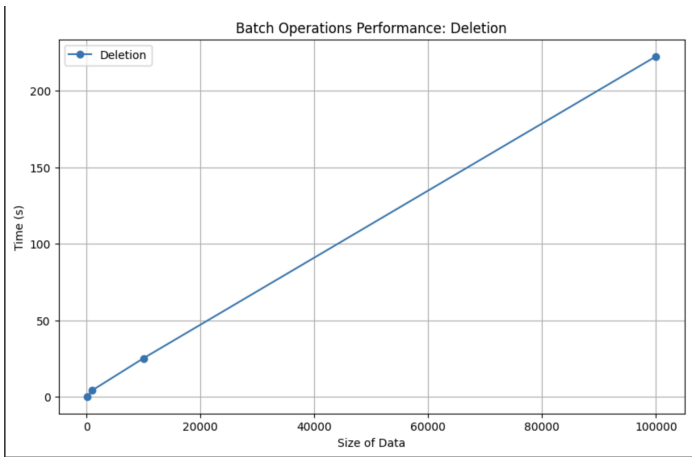
Insertion : https://github.com/ashrith5355/network_flow_db/blob/main/insert_bulk.py

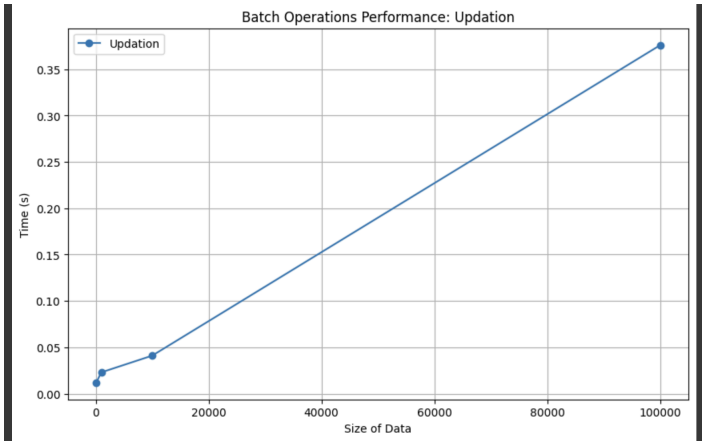
Updation : https://github.com/ashrith5355/network_flow_db/blob/main/update_bulk.py

Deletion : https://github.com/ashrith5355/network_flow_db/blob/main/delete_bulk.py

Graphs :


Batch Operations Performance: Insertion
Insertion
Time (s)
Size of Data
0.0
0.5
1.0
1.5
2.0
0
20000
40000
60000
80000
100000


Batch Operations Performance: Deletion
Deletion
Time (s)
Size of Data
0
50
100
150
200
0
20000
40000
60000
80000
100000


Batch Operations Performance: Updation
Updation
Time (s)
Size of Data
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
20000
40000
60000
80000
100000

Introduction to YCSB (Yahoo! Cloud Serving Benchmark)

Yahoo! Cloud Serving Benchmark (YCSB) is a widely-used benchmarking framework designed to evaluate the performance of various database systems, particularly NoSQL databases, but it can also be applied to SQL databases. YCSB was originally developed by Yahoo! as a means to standardize the performance measurement of different cloud-serving systems under a variety of workloads.

Key Features of YCSB:

- Workload Versatility: YCSB allows for the creation and execution of various workloads, simulating different types of database operations such as read, write, update, and scan operations.
- Extensibility: It supports multiple database systems through pluggable database adapters or clients, making it easy to extend and adapt YCSB to new database systems.
- Benchmarking Metrics: YCSB provides several important performance metrics, such as throughput (operations per second), latency (response time), and latency distribution, which are critical for evaluating database performance.
- Configurable Parameters: Users can configure many parameters, including the size of the data set, the distribution of the keys, the proportion of different types of operations, and more, allowing for customized benchmarking scenarios.

Workloads of YCSB:

YCSB comes with a set of predefined workloads (A-F), each representing different types of operations and access patterns:

- Workload A: Update heavy workload (50% reads, 50% updates).
- Workload B: Read heavy workload (95% reads, 5% updates).
- Workload C: Read only (100% reads).
- Workload D: Read latest workload (95% reads, 5% inserts).
- Workload E: Short ranges workload (95% scans, 5% inserts).
- Workload F: Read-modify-write workload (50% read-modify-write, 50% reads).

Using YCSB:

- Installation
- Configuration of MariaDB
- Build JDBC Driver (Connector)
- Load and Run workloads
- Tabulate the performance of DB

Conclusion:

YCSB is a powerful tool for evaluating the performance of different database systems under various conditions. Its flexibility, extensibility, and comprehensive metrics make it a valuable resource for developers, database administrators, and researchers looking to understand and improve database performance.

Configuration options used for YCSB :

```
# Copyright (c) 2010 Yahoo! Inc. All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License"); you
# may not use this file except in compliance with the License. You
# may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied. See the License for the specific language governing
# permissions and limitations under the License. See accompanying
# LICENSE file.


# Yahoo! Cloud System Benchmark
# Workload A: Update heavy workload
#   Application example: Session store recording recent actions
#
#   Read/update ratio: 50/50
#   Default data size: 1 KB records (10 fields, 100 bytes each, plus key)
#   Request distribution: zipfian

recordcount=100
operationcount=100
workload=site.ycsb.workloads.CoreWorkload

readallfields=true

readproportion=0
updateproportion=0
scanproportion=0
insertproportion=0
readmodifywriteproportion=1.0

requestdistribution=zipfian
```

Readcount and Operationcount are modified for every iteration for required batch size of data.

Readmodifywriteproportion is used for updating the data.

YCSB Bulk Operations :

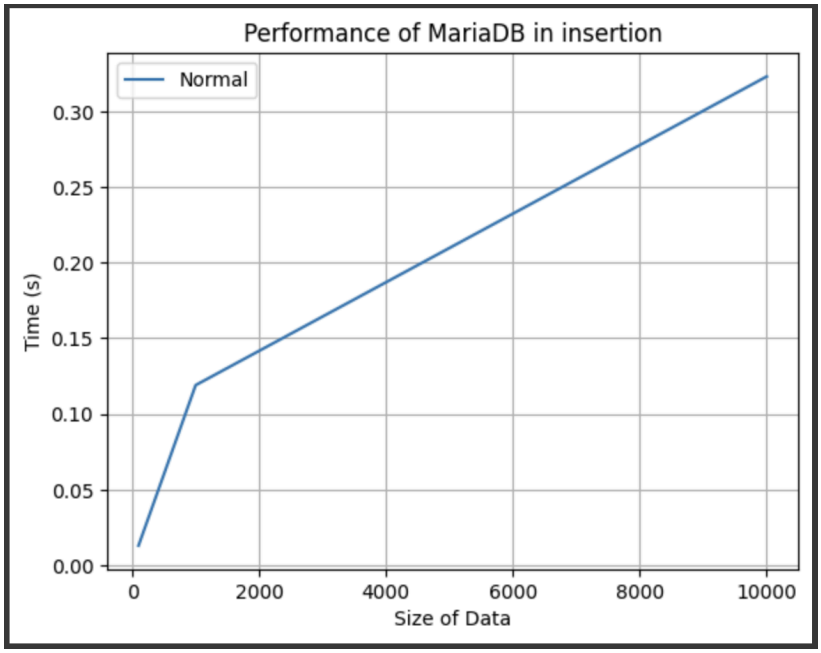
YCSB Bulk					
	100	1000	10,000	1,00,000	10,00,000
Insert	0.888	10.83	126.08	-	-
Update	0.862	10.58	118.19	-	-

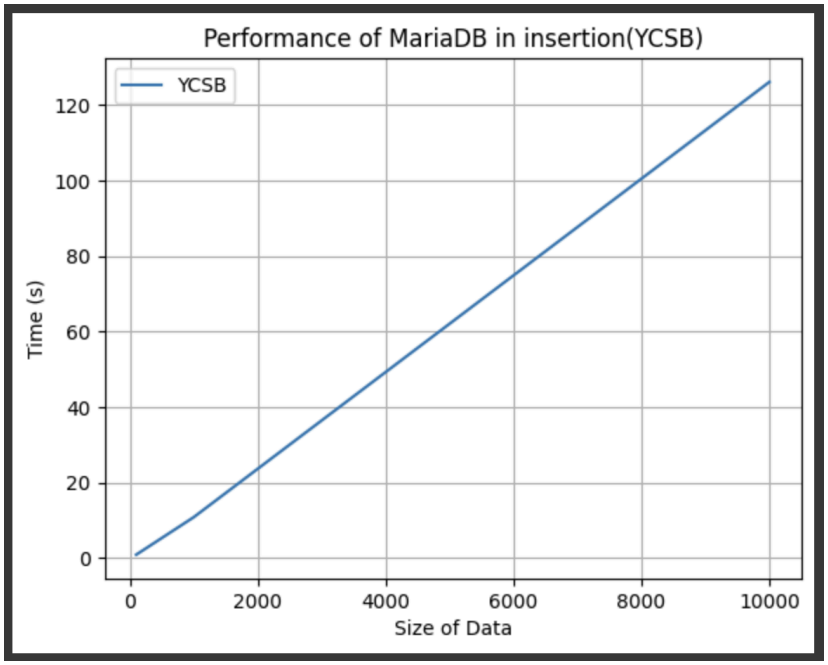
LOAD Command : bin/ycsb load jdbc -P workloads/workloada -P jdbc-binding/conf/db.properties -cp mysql-connector- java-5.1.49

RUN Command : bin/ycsb run jdbc -P workloads/workloada -P jdbc-binding/conf/db.properties -cp mysql-connector- java-5.1.49

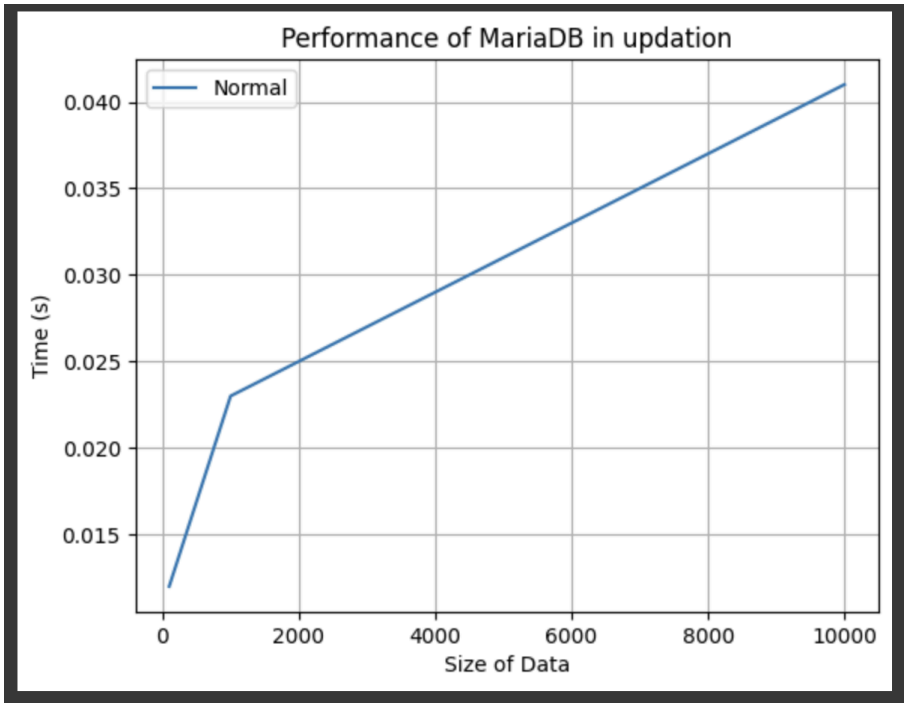
Graphs :

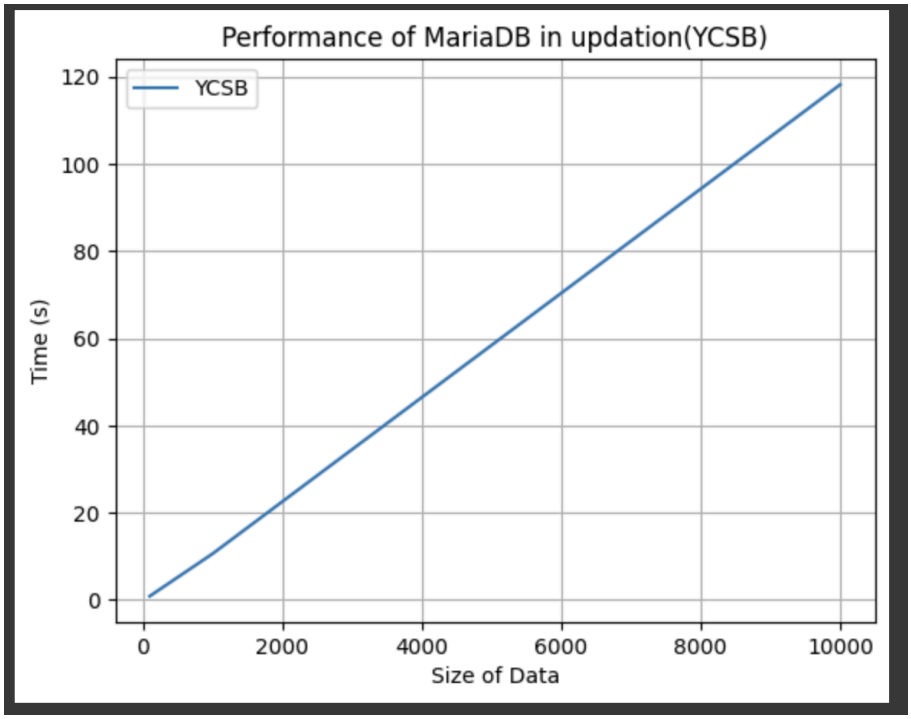
Insertion:





Updation:


Performance of MariaDB in updation
Normal
0.040
0.035
0.030
0.025
0.020
0.015
Time (s)
0
2000
4000
6000
8000
10000
Size of Data


Performance of MariaDB in updation(YCSB)
YCSB
120
100
80
60
40
20
0
Time (s)
0
2000
4000
6000
8000
10000
Size of Data

Batch Operations :

Batch Operations				
		Batch Size		
		100	1,000	10,000
Insertion	1,00,000	11.14	1.87	0.85
	10,000	0.86	0.16	-
	1,000	0.09	-	-
Updation	1,00,000	19.48	2.87	2.2
	10,000	0.88	0.24	0.08
	1,000	0.09	0.02	-
Deletion	1,00,000	10.41	4.85	161.49
	10,000	1.06	1	-
	1,000	1	-	-

- What we can observe is that the batch size must not be too small or too big, we have to choose an appropriate batch size for the best operation time.
- For insertion and updation it is okay to choose a large batch size.
- But, for deletion an appropriate batch size must be chosen.

CPU and Memory utilization :

		CPU %	Memory %
Batch Operations	Insertion	23.7	2.3
	Updation	48	2.2
	Deletion	15	2.2

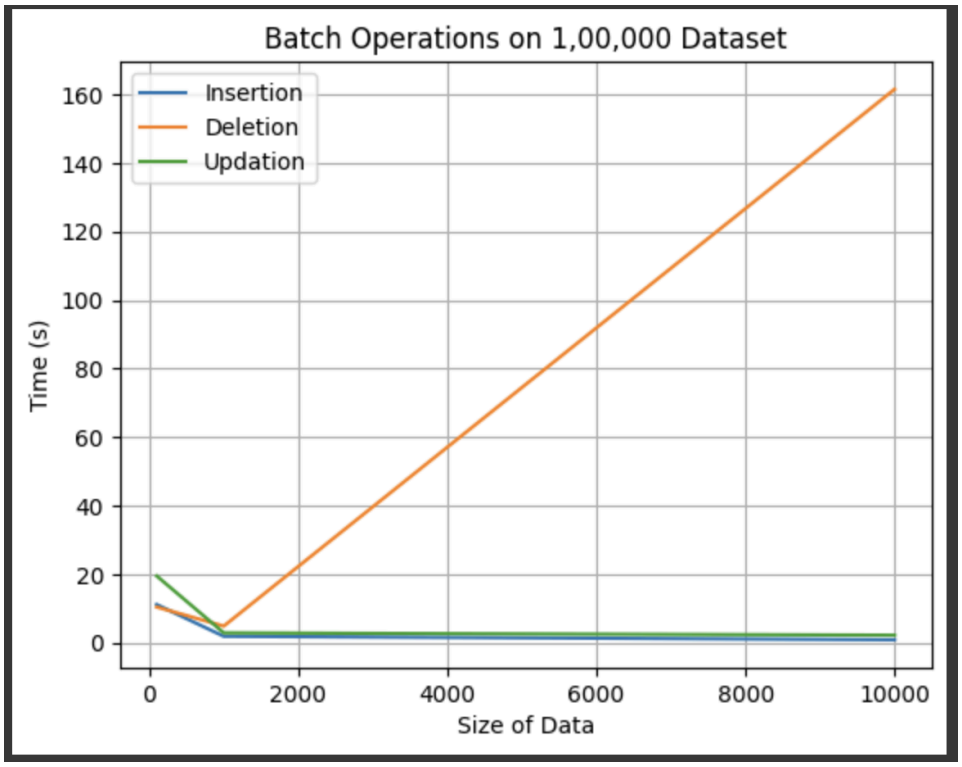
Code links for Batch Operations

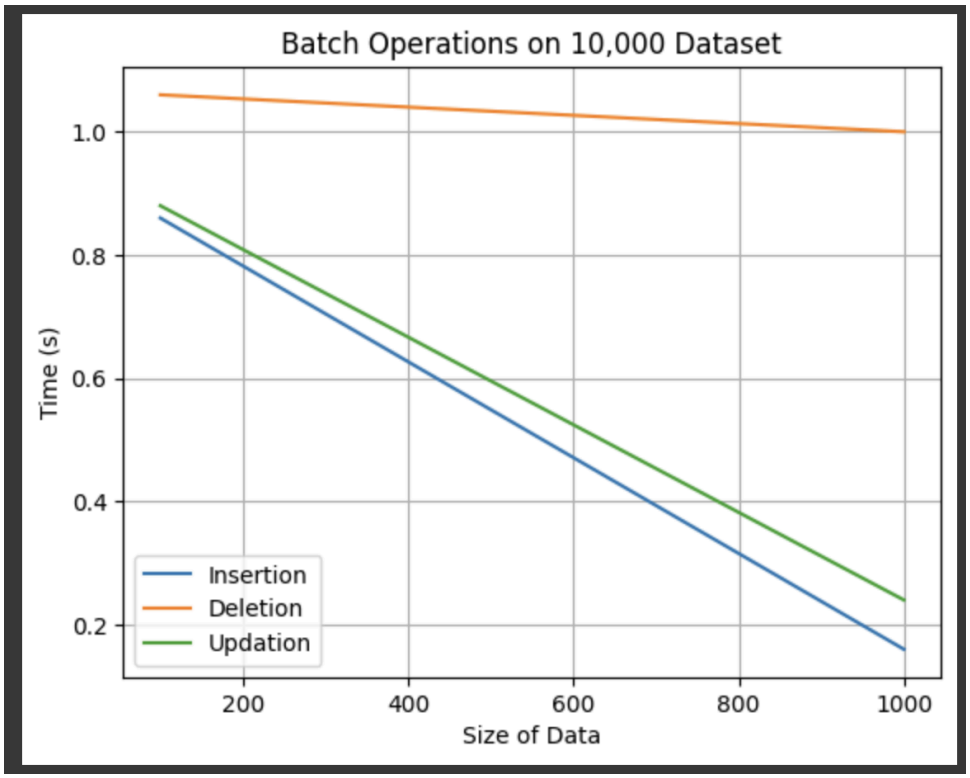
Insertion : https://github.com/ashrith5355/network_flow_db/blob/main/insert_batch_new.py

Updation: https://github.com/ashrith5355/network_flow_db/blob/main/update_batch_new.py

Deletion: https://github.com/ashrith5355/network_flow_db/blob/main/delete_batch_new.py

Graphs for Batch Operations :





Conclusion :

MariaDB is a robust and flexible relational database management system that excels in managing network flow data and general database needs. Its ability to scale horizontally and vertically, coupled with high performance and low latency, makes it suitable for real-time data analysis and large-scale data management. As an open-source solution, MariaDB offers cost efficiency and extensive customization options, ensuring seamless integration with existing systems. Enhanced security features and compatibility with MySQL further solidify its reliability. Overall, MariaDB's scalability, performance, flexibility, and security make it an ideal choice for efficiently handling network flow data and other complex database requirements.