

Performance Benchmarking and Scalability Analysis of MySQL and MariaDB Using YCSB Workloads

Samuel Aidoo
sa19104@georgiasouthern.edu
Georgia Southern University
Statesboro, GA 30458

Hayden Wimmer
hwimmer@georgiasouthern.edu
Georgia Southern University
Statesboro, GA 30458

Jie Du
dujie@gvsu.edu
Grand Valley State University
Allendale, MI 49401

Abstract

Database performance can vary substantially across workload configurations and deployment environments, making controlled comparisons important when evaluating relational database management systems. Although MySQL and MariaDB share a common lineage, fewer studies have examined their throughput behavior under identical YCSB workload, concurrency, and virtualized conditions. This study compares MySQL and MariaDB using YCSB Workload A, a standardized synthetic workload consisting of 50% read and 50% update operations. Experiments were conducted in a Linux-based virtual machine using thread counts of 1, 2, 4, and 8 and operation counts of 1,000, 5,000, and 10,000. Each configuration was repeated three times. Across the 36 matched experimental conditions, MariaDB achieved a mean throughput of 2,534.44 ops/sec compared with 643.39 ops/sec for MySQL. A paired-samples t-test indicated a statistically significant mean throughput difference of 1,891.06 ops/sec, $t(35) = 15.95$, $p < .001$, 95% CI [1,650.44, 2,131.68]. The results show that MariaDB achieved higher throughput than MySQL under the specific workload and virtualized configuration evaluated in this study. These findings should therefore be interpreted as environment-specific benchmark results rather than evidence of general performance superiority across all database workloads and deployment environments.

Keywords: Database Performance Benchmarking, YCSB, MariaDB, MySQL, Throughput Analysis, Scalability

Performance Benchmarking and Scalability Analysis of MySQL and MariaDB Using YCSB Workloads

Samuel Aidoo, Hayden Wimmer and Jie Du

1. INTRODUCTION

Relational database management systems (RDBMSs) dominate today's computing environments because of their ability to efficiently store, manage and process large amounts of structured data. As enterprise systems and web-based applications are becoming more prevalent due to the development of cloud computing, database systems are expected to deliver high performance, scalability and reliability against increasing workloads and concurrent user requests. Therefore, performance benchmarking has become an important process to evaluate the behavior of database systems under different workload conditions and levels of concurrency.

MySQL and MariaDB are the most widely used open-source relational database management systems (DB Engines, 2026). MySQL has been widely used in enterprise and web applications due to its stability, flexibility and large community support. Then came MariaDB, fork of MySQL, that intended to be compatible but also to address performance, scalability and optimization opportunities. The architecture and features of the two systems are similar, but the internal implementation and the way of handling workloads may be different, which may lead to differences in performance in the face of concurrent operations.

Benchmarking tools for databases provide a common way to assess and compare the performance of systems. One of the most popular benchmarking frameworks is the Yahoo! Cloud Serving Benchmark (YCSB), which aims to test database systems with various workload patterns and operational settings (Cooper et al., 2010). YCSB lets researchers measure important performance indicators such as throughput, latency, and scalability in controlled experiments. We chose YCSB Workload A for this study because it provides a balanced transactional workload with 50% reads and 50% updates, which is suitable for testing mixed database operations found in typical real-world applications.

Performance behavior of relational database systems under concurrent workloads is important for choosing suitable database technologies in a production environment. This research provides comparative performance benchmarking and scalability analysis of MySQL and MariaDB using YCSB workloads in Linux based virtualized environment. Experiments were conducted with thread counts of 1, 2, 4, 8 and operation counts of 1,000, 5,000, 10,000 operations. Each test configuration was run 3 times to improve the reliability of the measurements and to reduce the performance variability. The key performance measure used to assess workload handling, and scalability behavior was throughput. We examined the collected throughput measurements to detect any performance variations between the two DBMSs, and to analyze the impact of concurrency and the number of operations on the observed results. The results are presented in Section 4 and interpreted according to the limitations of the experimental environment. This paper contributes to the research on database benchmarking and provides practical insights into the performance characteristics of MySQL and MariaDB under concurrent transactional workloads.

Database benchmarking has been extensively studied, but the available performance results are highly sensitive to workload characteristics, hardware resources, concurrency levels, database configuration and deployment environment. Previous research that has compared MySQL and MariaDB has also been based on different benchmarking tools, configurations and workloads, making it hard to generalize their findings to different environments. There is value in measuring both systems on a common YCSB mixed read/update workload while systematically varying both concurrency and total operations and repeating each configuration. In the present study, we fill this gap by providing a controlled comparison between MySQL and MariaDB in the same virtualized environment and matched YCSB Workload A configurations. The study evaluates the behavior of the two systems under the specific workload and experimental conditions tested, rather than trying to figure out which DBMS is better in general.

This gap in the research results in the following research questions of the study:

RQ1: What is the difference in throughput between MySQL and MariaDB with identical YCSB Workload A settings and is it statistically significant?

RQ2: How does the number of threads and operations affect the throughput of MySQL and MariaDB in the tested virtualized environment?

This work provides a controlled, repeated comparison of MySQL and MariaDB under the same YCSB Workload A configuration. This experiment is not a general comparison of the two database systems, but rather a comparison of throughput in the same virtualized environment with the matched combinations of concurrency and operation count. The study also presents a repeatable benchmarking procedure that can be adapted to database performance experiments and instructional laboratory activities.

This paper is divided into 6 sections. Section 2 contains a review of the relevant literature. Section 3 presents the methodology of the study. The results are reported in Section 4. The results and their implications are presented in Section 5. The paper is concluded in section 6.

2. LITERATURE REVIEW

2.1 Database Benchmarking

Database benchmarking is the process of evaluating system performance under controlled and repeatable experimental conditions (Darmont, 2009). Benchmarking allows researchers and system administrators to measure how database systems respond to different workload sizes, user concurrency levels, and transaction patterns. Performance benchmarking is particularly important in cloud and virtualized environments where system resources are shared among multiple services and applications. Through benchmarking, researchers can identify scalability limitations, workload bottlenecks, and performance trends that affect overall system efficiency.

One of the most used benchmarking frameworks for database evaluation is Yahoo! Cloud Serving Benchmark (YCSB). YCSB was developed to provide a standardized benchmarking framework capable of evaluating database systems using multiple workload types and configurable testing parameters (Cooper et al., 2010). The framework supports various database technologies and allows researchers to measure performance metrics such as throughput, latency, and scalability. YCSB workloads simulate realistic application behaviors by combining read, update, insert, and scan operations under different operational patterns.

2.2 Benchmarking Methodology

While benchmark frameworks such as YCSB define standardized workloads for evaluating database systems, a well-designed benchmarking methodology is essential to ensure fair and reliable performance comparisons. Li et al. (2010) proposed CloudCmp, a benchmarking methodology designed to compare the performance of public cloud providers using measurable metrics such as computation, storage, and networking performance. Through extensive experiments conducted across different cloud providers, they demonstrated that benchmarking results can vary significantly depending on workload type and infrastructure configuration. Their findings emphasized the importance of consistent benchmarking procedures and repeatable measurements when comparing system performance.

To measure and compare system performance under high computational workloads across different cloud environments, Mohammadi and Bazhrov (2018) conducted a comparative benchmarking study of cloud computing vendors using the High Performance Linpack (HPL) benchmark to evaluate computational scalability and performance efficiency. They found that benchmarking results vary depending on hardware allocation, virtualization overhead, and workload intensity. Their work emphasized the importance of standardized benchmarking methodologies and performance metrics for conducting fair system evaluations.

Leitner and Cito (2016) investigated performance variability and predictability in Infrastructure-as-a-Service (IaaS) cloud environments with the aim to determine how cloud performance fluctuates due to multitenancy, hardware heterogeneity, and workload unpredictability. Using extensive benchmark experiments across cloud systems, they found that performance measurements can vary significantly

even under similar workload conditions. Their results demonstrated that repeated benchmark executions are necessary to obtain reliable and representative measurements.

2.3 Performance Evaluation of MySQL and MariaDB

Previous studies have directly compared the performance of MySQL and MariaDB, but their findings vary depending on the workload and experimental environment. (Tongkaw & Tongkaw, 2016) compared MySQL and MariaDB using OLTP-Simple and OLTP-Seats workloads in a Xen 4.4 virtualized environment. Under their experimental conditions, MySQL 5.6 achieved better performance than MariaDB 10.0.21, particularly at higher concurrency levels. Their results demonstrate that the relative performance of the two database systems can depend on the workload, concurrency level, database version, and deployment environment.

Witham et al. (2019) conducted another direct comparison of MySQL and MariaDB by examining insertion and query performance using a dataset containing more than 500,000 records and two experimental test benches. Their findings generally showed stronger MariaDB performance, although the magnitude of the difference varied according to the operation and test environment. The contrast between their findings and those of (Tongkaw & Tongkaw, 2016) suggests that neither database system can be assumed to provide universally higher performance across all workloads and environments.

Abubakar (2014) also included MySQL and MariaDB in a broader YCSB-based comparison of open-source relational database systems. Their study evaluated read, write, and update operations across different record sizes using repeated measurements. Together, these studies demonstrate that comparisons between MySQL and MariaDB should control workload characteristics, database configuration, concurrency, and experimental environment when interpreting performance differences.

2.4 Factors Affecting Benchmark Performance

Shah et al. (2021) evaluated the performance of virtual machines and containers under different cloud workload configurations to determine how virtualization technologies influence throughput and scalability. Their objective was to compare workload efficiency, latency, and resource utilization across multiple virtualized environments. Using benchmark-based experiments, they found that virtualization overhead and resource contention significantly affect system scalability and workload consistency.

De Sensi et al. (2022) examined the influence of system and network noise on application scalability in cloud computing environments. Their objective was to analyze how concurrency levels and infrastructure variability affect performance consistency during scalability experiments. Through extensive benchmarking and scalability testing, they demonstrated that increasing concurrency levels can introduce workload instability and variability in throughput measurements. Their findings support the present research because varying thread counts were used to evaluate the scalability and concurrency behavior of MySQL and MariaDB under mixed transactional workloads (De Sensi et al., 2022).

Mukherjee et al. (2014) investigated performance testing techniques for cloud-based web applications with the objective of evaluating how workload intensity and concurrent user activity influence system throughput and responsiveness. Using workload-based benchmark experiments, the authors demonstrated that performance measurements can fluctuate significantly depending on workload size and concurrency levels. Their results highlighted the importance of workload modeling and repeated testing when evaluating system performance in cloud and virtualized environments.

Kang et al., (2017) compared the performance of private OpenStack cloud systems with Amazon EC2 environments to evaluate how virtualization infrastructure affects workload efficiency and application performance. Their objective was to determine whether private cloud deployments could provide performance levels comparable to public cloud services. Using benchmark-driven experiments, they analyzed system throughput, scalability, and computational performance across different cloud infrastructures. Their findings showed that infrastructure configuration and virtualization layers significantly influence benchmark performance and workload consistency.

Zboril and Svatá (2025) investigated the performance characteristics of cloud virtual machines with the objective of analyzing how different virtualized environments influence scalability and throughput performance. They evaluated workload efficiency, system responsiveness, and resource utilization under

varying operational conditions. Their results demonstrated that virtual machine configuration and workload concurrency strongly affect system performance and scalability behavior.

The literature establishes that fair and reliable database performance comparisons depend on standardized benchmark frameworks, controlled experimental conditions, and appropriate performance metrics. Building upon these established best practices, this study employs YCSB to systematically compare the performance of MySQL and MariaDB under consistent experimental conditions across representative workloads.

2.4 Research Gap and Study Contribution

Previous research demonstrates that the relative performance of MySQL and MariaDB varies across experimental environments and workload conditions. Tongkaw & Tongkaw (2016) reported stronger MySQL performance under their OLTP and Xen-based experimental conditions, whereas Witham et al. (2019) generally observed stronger MariaDB performance in their insertion and query experiments. Abubakar (2014) further demonstrated the use of YCSB for comparing open-source relational database systems, including MySQL and MariaDB. These differing findings indicate that database performance is influenced by workload characteristics, concurrency, system configuration, and deployment environment.

However, differences in benchmarking tools, workload definitions, database versions, hardware configurations, and experimental procedures make it difficult to generalize previous MySQL and MariaDB performance findings across environments. There remains value in evaluating both systems under identical YCSB workload conditions while systematically varying concurrency and operation count and repeating each experimental configuration.

This study addresses this gap through a controlled comparison of MySQL and MariaDB under the same virtualized environment using identical YCSB Workload A configurations. Thread count and operation count are systematically varied, and each configuration is repeated three times. The study therefore provides matched experimental observations for evaluating throughput differences between the two database systems and examining how the observed performance changes across the tested workload configurations. The experimental workflow also provides a repeatable framework that can be adapted for instructional database-performance laboratories.

3. METHODOLOGY

3.1 Experimental Environment

The experiments conducted in this study were performed in a Linux-based virtualized environment to evaluate the performance of MySQL and MariaDB under controlled workload conditions. The experimental platform consisted of an Ubuntu virtual machine running on VMware virtualization software. A Java Development Kit (JDK) was installed to support execution of the Yahoo! Cloud Serving Benchmark (YCSB) framework and JDBC database connectivity. Both database systems were evaluated within the same virtualized environment to maintain consistent experimental conditions.

MySQL and MariaDB were installed locally within the Ubuntu virtual machine. JDBC drivers were used to establish communication between YCSB and the database systems. A common database schema and table structure were used for both systems to maintain consistency during benchmark execution. The virtual machine was configured with 20 GB of virtual storage.

3.2 Benchmark Framework

This study utilized the YCSB framework to evaluate database performance under configurable workload conditions. YCSB Workload A was selected because it provides a standardized synthetic workload consisting of 50% read operations and 50% update operations. The balanced read/update distribution provides a controlled basis for comparing the two database systems under identical mixed-operation conditions. Workload A is not intended to reproduce every characteristic of a production transactional application; rather, it provides a repeatable workload for evaluating database performance under controlled conditions.

3.3 Workflow Diagram

Figure 1 illustrates the benchmarking workflow used in this research. YCSB generated transactional workloads that were transmitted through JDBC drivers to both MySQL and MariaDB database systems. The databases processed the workloads, and performance metrics such as throughput and scalability were collected and analyzed for comparative evaluation.

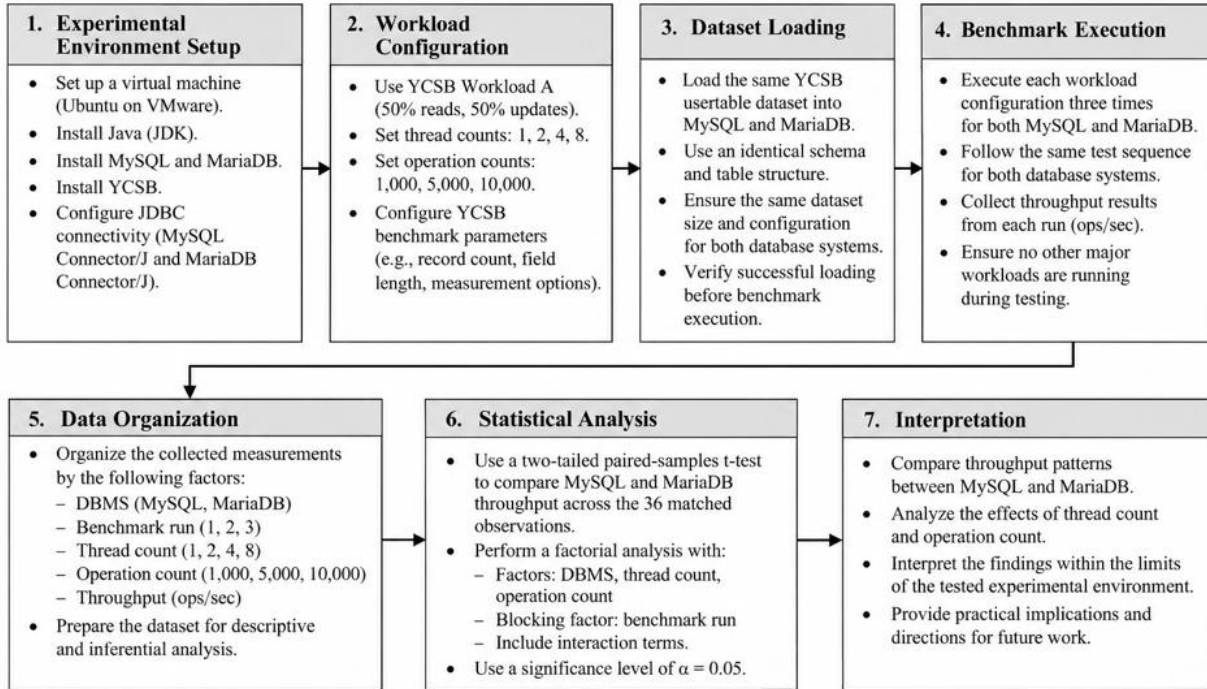


Figure 1. Benchmarking Workflow Diagram

3.4 Database and Workload Configuration

Separate databases were created for MySQL and MariaDB benchmarking experiments. The benchmark utilized the usertable dataset generated during the YCSB loading phase. Prior to executing benchmark tests, records were loaded into both database systems using the YCSB load operation. Both databases used identical dataset sizes and workload configurations to ensure fairness during performance evaluation.

The experiments were conducted using varying thread counts and operation counts to evaluate concurrency behavior and scalability performance (see Table 1). The three operation counts were selected to represent increasing workload sizes within the resource limits of the experimental environment. Using low, intermediate, and higher operation counts allowed the study to examine whether throughput patterns remained consistent as the number of benchmark operations increased. These values represent experimental workload levels and are not intended to correspond directly to specific production deployment sizes.

Parameter	Configuration
Benchmark Tool	YCSB 0.17.0
Workload	Workload A
Read/Update Ratio	50% Read / 50% Update
Thread Counts	1, 2, 4, 8
Operation Counts	1,000, 5,000, 10,000
Repetitions	3 per configuration
VM CPU	4 vCPUs
VM Memory	8 GB RAM
VM Storage	100 GB SSD
Operating System	Ubuntu 22.04 LTS
Virtualization Platform	VMware Workstation Pro 17
MySQL	MySQL 8.0
MariaDB	MariaDB 10.11
Storage Engine	InnoDB
Java	OpenJDK 11
YCSB	0.17.0
MySQL JDBC Driver	MySQL Connector/J 8.0
MariaDB JDBC Driver	MariaDB Connector/J 3.x
DBMS Tuning	Default configuration; no DBMS-specific performance tuning

Table 1. Experimental Workload Configuration

3.5 Performance Metrics

The primary performance metric evaluated in this study was throughput, measured in operations per second (ops/sec). Throughput was selected because it represents the ability of a database system to process concurrent transactional operations efficiently under varying workload conditions. Higher throughput values indicate better workload handling capability and improved scalability performance. Scalability behavior was also analyzed by examining how throughput changed as thread counts and operation sizes increased.

Scalability behavior was also examined by evaluating how throughput changed as thread count and operation count increased. The analysis considered whether throughput increased consistently across the tested concurrency and workload levels or whether the magnitude of improvement changed at higher levels. Because CPU utilization, memory usage, disk I/O, and virtualization scheduling were not measured, the experiment does not attempt to attribute observed throughput patterns to specific system-level mechanisms. YCSB also generated latency information during benchmark execution; however, latency measurements were not retained as part of the dataset used for statistical analysis. Consequently, the present study focuses on throughput. Future experiments should retain and analyze latency alongside throughput.

3.6 Experimental Procedure

The benchmarking process began by loading the initial dataset into both MySQL and MariaDB using the YCSB load operation. After the loading phase was completed, benchmark execution was performed using YCSB Workload A under varying thread counts and operation counts. For each workload configuration, benchmark results were collected and stored in output log files generated by YCSB. The experiments were repeated three times for each configuration to improve result reliability and reduce inconsistencies caused by virtualization noise and transient system behavior. We retained throughput measurements from all three repeated benchmark executions for statistical analysis rather than reducing each configuration to a single average observation. The complete dataset was organized by DBMS, benchmark run, thread count, and operation count for subsequent descriptive and inferential analysis.

3.7 Statistical Analysis

Statistical analysis was conducted using the throughput measurements collected from the three repeated benchmark runs. Because MySQL and MariaDB were evaluated under corresponding combinations of benchmark run, thread count, and operation count, the observations were treated as matched experimental conditions rather than independent samples.

To address RQ1, a two-tailed paired-samples t-test was used to compare MySQL and MariaDB throughput across the 36 matched experimental observations. Each MariaDB throughput measurement was paired with the MySQL measurement obtained during the same benchmark run and under the same thread-count and operation-count configuration.

The hypotheses used in this study are defined as follows:

$$H_0: \mu D = 0$$

$$H_1: \mu D \neq 0$$

where μD represents the population mean of the paired throughput differences between MariaDB and MySQL.

To address RQ2, throughput was additionally examined using a factorial model incorporating DBMS, thread count, and operation count as experimental factors, with benchmark run included as a blocking factor. Interaction terms were included to determine whether the effects of thread count, and operation count differed between the two database systems.

A significance level of $\alpha = .05$ was used for statistical inference. Statistical significance was interpreted within the tested experimental environment and was not considered evidence that either DBMS would necessarily exhibit the same performance relationship under different workloads or deployment configurations.

4. Results

4.1 Benchmark Run 1 Results

The first benchmark execution was conducted using YCSB Workload A under varying thread counts and operation sizes to evaluate the throughput performance of MySQL and MariaDB under concurrent transactional workloads. Throughput values were measured in operations per second (ops/sec), where higher values indicate better workload handling capability and scalability performance. Table 1 shows the throughput results.

Threads	Operations	MySQL	MariaDB
1	1000	430	1634
1	5000	440	2032
1	10000	450	2409
2	1000	600	1700
2	5000	650	2300
2	10000	680	2600
4	1000	681	1700
4	5000	720	3180
4	10000	760	3600
8	1000	700	1750
8	5000	750	3400
8	10000	780	4008

Table 2. Benchmark Run 1 Throughput Results

Run 1 showed higher MariaDB throughput than MySQL across each of the 12 tested thread-count and operation-count configurations. At the highest tested configuration of 8 threads and 10,000 operations, MariaDB recorded 4,008 ops/sec compared with 780 ops/sec for MySQL. These run-level measurements

are presented descriptively; inferential comparisons across the complete set of repeated measurements are reported in Section 4.4.

4.2 Benchmark Run 2 Results

The second benchmark execution was performed using the same workload configurations and testing conditions to validate the consistency and reliability of the throughput measurements obtained during the first benchmark run. Table 2 shows the throughput results in the second run.

Threads	Operations	MySQL	MariaDB
1	1000	420	1504
1	5000	445	2225
1	10000	455	2406
2	1000	620	1838
2	5000	660	2400
2	10000	690	2700
4	1000	700	1700
4	5000	740	3100
4	10000	780	3500
8	1000	720	1800
8	5000	770	3500
8	10000	800	3586

Table 3. Benchmark Run 2 Throughput Results

Run 2 produced a pattern broadly similar to Run 1, with MariaDB recording higher throughput than MySQL across all 12 tested configurations. Minor differences were observed between repeated measurements, supporting the decision to repeat each experimental configuration rather than rely on a single benchmark execution.

4.3 Benchmark Run 3 Results

The third benchmark execution was conducted to validate the consistency of the benchmark measurements further and to reduce variability introduced by virtualization overhead and workload fluctuations.

Threads	Operations	MySQL	MariaDB
1	1000	435	1750
1	5000	438	2109
1	10000	448	2405
2	1000	610	1650
2	5000	655	2350
2	10000	685	2650
4	1000	690	1700
4	5000	730	3300
4	10000	770	3763
8	1000	710	1700
8	5000	760	3491
8	10000	790	3800

Table 4. Benchmark Run 3 Throughput Results

Run 3 again showed higher MariaDB throughput across the tested configurations. The overall pattern remained similar across the three benchmark executions, although individual throughput values varied between repetitions. Inferential analysis was therefore performed using the complete set of matched observations rather than conducting a separate hypothesis test for each benchmark run.

4.4 Overall Paired Throughput Comparison

Across the 36 matched experimental observations, MySQL achieved a mean throughput of 643.39 ops/sec, while MariaDB achieved a mean throughput of 2,534.44 ops/sec. The mean paired throughput difference was 1,891.06 ops/sec in favor of MariaDB.

A two-tailed paired-samples t-test showed that the mean throughput difference was statistically significant, $t(35) = 15.95$, $p < .001$, with a 95% confidence interval of approximately [1,650.44, 2,131.68] ops/sec. Thus, the null hypothesis for RQ1 was rejected. Within the configurations evaluated in this experiment, MariaDB produced higher throughput than MySQL.

4.4 Factorial Analysis

To address RQ2, a factorial linear model was used to examine the effects of DBMS, thread count, and operation count on throughput, with benchmark run included as a blocking factor. The analysis showed statistically significant effects for DBMS, $F(1, 46) = 12,808.39$, $p < .001$; thread count, $F(3, 46) = 309.95$, $p < .001$; and operation count, $F(2, 46) = 711.11$, $p < .001$. The benchmark-run blocking factor was not statistically significant, $F(2, 46) = 0.43$, $p = .655$.

Significant interaction effects were also observed between DBMS and thread count, $F(3, 46) = 102.27$, $p < .001$; DBMS and operation count, $F(2, 46) = 599.53$, $p < .001$; and thread count and operation count, $F(6, 46) = 42.68$, $p < .001$. The three-way interaction among DBMS, thread count, and operation count was also statistically significant, $F(6, 46) = 38.64$, $p < .001$.

These results indicate that throughput varied across both thread-count and operation-count levels and that these relationships were not identical for MySQL and MariaDB. Therefore, the effects of concurrency and operation count should be interpreted together with the DBMS being evaluated rather than as independent and uniform effects across both systems.

5. DISCUSSION

5.1 Discussion of Findings

The study compared the performance throughput of MySQL and MariaDB using the same configuration of YCSB Workload A in a virtualized Ubuntu environment. Benchmark results showed a consistent difference between the two database systems for the tested configurations. For the 36 matched observations, MariaDB had a mean throughput of 2534.44 operations per second compared to MySQL with 643.39 operations per second. The average paired difference was 1,891.06 operations per second. This difference was statistically significant by a paired-samples t-test, $t(35) = 15.95$, $p < .001$, 95% confidence interval [1,650.44, 2,131.68] operations per second. The results show that MariaDB outperformed (higher throughput) MySQL with the particular experimental setup and workload configurations used in this study.

Repeated benchmark runs also showed a consistent performance pattern. The individual throughput numbers varied slightly from run to run, but in every combination of thread and number of operations, MariaDB was able to deliver higher throughput than MySQL. This consistency across multiple measurements further supports the observed difference in performance during the experiment. However, this does not mean MariaDB is always faster than MySQL because performance of any database can depend on the configuration of the software, the resources of the hardware, the characteristics of the workload, the behavior of the storage, or the environment in which it is deployed.

Throughput changes also correlated with the number of threads and the number of operations. In general, the throughput increased with the number of concurrent threads, however, the size of the increase was not uniform for all configurations. MySQL and MariaDB also had different impacts when the thread counts increased. Throughput also varied for the different levels of operation count, which means that operation count cannot be considered to have no impact on the benchmark results.

The interplay of the experimental factors is especially crucial for the interpretation of the data. This difference in performance between MySQL and MariaDB was not consistent across all the thread and

operation-count configurations. Rather, the measured throughput depended on the database system, concurrency level and the number of operations. This highlights the need to evaluate database systems under different workload configurations, rather than draw conclusions based on a single benchmark configuration.

Some configurations exhibited less incremental throughput benefit with increasing concurrency. However, the present experiment did not collect CPU utilization, memory usage, disk I/O, cache behavior, or other system-level measurements that would be needed to determine why these patterns occurred. Such behavior may be due to resource contention, synchronization overhead, virtualization scheduling or other system-level effects, but the present results do not identify these mechanisms as causes. These explanations would need additional resource monitoring data for investigation.

The results, taken as a whole, provide answers to the two research questions posed by the experiment. To RQ1, we found a statistically significant throughput difference between MySQL and MariaDB using matched YCSB Workload A configurations, with MariaDB achieving a higher mean throughput. For RQ2, both concurrency and operation count were related to throughput changes, and their effects were database system dependent. The results show that database performance should be viewed as a function of the workload and the experimental configuration and not as an intrinsic property of DBMS itself.

5.2 Practical Implications and Insights

The study provides several important lessons for practitioners, system administrators and educators involved in database system deployment and performance evaluation.

The results demonstrate the need for benchmarking the candidate database systems in an environment like the environment where the system will be deployed. MariaDB achieved a higher throughput than MySQL for all tested YCSB Workload A configurations in the environment evaluated for this study. However, this should not be considered a blanket recommendation for all high-concurrency applications to choose MariaDB.

Besides, database selection should consider latency requirements, workload characteristics, storage configuration, compatibility, reliability, administration requirements, and deployment infrastructure. Indeed, the results emphasize the necessity of empirical testing for specific workloads rather than a one-size-fits-all best DBMS.

The results also show that increasing the thread count did not result in proportional increases in throughput for all tested configurations. This finding emphasizes the importance of empirically evaluating concurrency, rather than if adding additional concurrent threads will yield the same performance benefits. Because CPU utilization, memory consumption, disk I/O, and latency were not preserved for the analysis, the present study cannot identify the system-level mechanisms responsible for these throughput patterns.

From an educational point of view, the experimental workflow provides a practical framework for teaching performance evaluation of databases. Students can configure YCSB, set up JDBC connectivity, run controlled workloads, vary the level of concurrency, repeat experimental trials, and analyze performance using descriptive and inferential statistics. The laboratory can be extended to collect latency, CPU utilization, memory consumption and disk IO. This makes the experiment useful as a reproducible database performance laboratory activity for courses on database systems, information technology, cloud computing and performance engineering.

Finally, the study highlights the need for empirical testing of performance in virtualized environments, where workload variability and system overhead can have a significant impact on performance results. Practitioners and researchers should take these factors into account when designing experiments and interpreting benchmark results, especially in cloud-based or virtualized deployments.

5.3 Limitations

There are several limitations to bear in mind when interpreting the findings of this study. First, experiments were performed in a single virtualized Ubuntu environment. Performance characteristics

can vary between processors, memory sizes, storage systems, operating systems, virtualization platforms, and physical or cloud deployments.

Second, the evaluation was only done for YCSB Workload A, a synthetic workload with a 50% read and 50% update operation. The performance relationships may vary for read-heavy, write-heavy, scan-oriented and application-specific workloads.

Third, the experiment was limited to a small set of thread counts and operation counts. The results thus reflect the performance in the tested range rather than the full bounds of the scalability of either DBMS.

Fourth, only throughput was retained as a metric for statistical analysis. Information on latency, CPU utilization, memory consumption, disk I/O and cache behavior would have given more insight into the observed performance differences.

Moreover, we restricted the number of repetitions to three per configuration; more repeated trials may provide a more robust estimate of benchmark variability.

Finally, the results are specific to the database versions and configuration used in this experiment. Performance can be influenced by alterations of storage engines, caching, buffer sizes, index management, connection management, and database tuning. The results should therefore not be interpreted as an indication that one DBMS is superior in general, but as a controlled experimental comparison.

6. CONCLUSIONS

This study used YCSB Workload A to analyze the throughput performance of MySQL and MariaDB in a controlled virtualized Ubuntu environment. The experiment tested 12 combinations of thread count and operation count. For each configuration 3 repetitions were made for both database systems.

For the 36 matched observations, MariaDB achieved a mean throughput of 2,534.44 ops/sec vs. 643.39 ops/sec for MySQL. The paired-samples analysis revealed a statistically significant difference in throughput between the two systems for the tested conditions. In addition, the factorial analysis showed significant effects of DBMS, thread count, and operation count as well as significant interaction effects. Thus, the throughput response to concurrency and operation count was different for the two database systems.

Database benchmark results should be interpreted according to a database system and its workload configuration, our results show. MariaDB achieved a higher throughput than MySQL in this experiment, but this result is not to be generalized beyond the studied virtualized environment and the YCSB Workload A configuration.

Future work should evaluate other YCSB workloads, larger datasets, higher concurrency, different database configurations, and different physical and cloud deployment environments. Future experiments should include latency, CPU utilization, memory consumption, disk I/O, and other resource-level measurements to give a stronger basis for explaining the mechanisms behind observed performance differences.

7. REFERENCES

- Abubakar, Y. (2014). Benchmarking popular open source RDBMS: a performance evaluation for it professionals. *International Journal of Advanced Computer Technology (IJACT)*, 3, 39-44.
- Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., & Sears, R. (2010). Benchmarking cloud serving systems with YCSB. *Proceedings of the 1st ACM symposium on Cloud computing*,
- Darmont, J. (2009). Database benchmarks. In *Encyclopedia of Information Science and Technology, Second Edition* (pp. 950-954). IGI Global Scientific Publishing.

- DB Engines. (2026). *DB-Engines Ranking of Relational DBMS*. Retrieved July 6 from <https://db-engines.com/en/ranking/relational+dbms>
- De Sensi, D., De Matteis, T., Taranov, K., Di Girolamo, S., Rahn, T., & Hoefler, T. (2022). Noise in the clouds: Influence of network performance variability on application scalability. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3), 1-27.
- Kang, M., Kang, D.-I., Walters, J. P., & Crago, S. P. (2017). A comparison of system performance on a private openstack cloud and amazon ec2. 2017 IEEE 10th International Conference on Cloud Computing (CLOUD),
- Leitner, P., & Cito, J. (2016). Patterns in the chaos—a study of performance variation and predictability in public iaas clouds. *ACM Transactions on Internet Technology (TOIT)*, 16(3), 1-23.
- Li, A., Yang, X., Kandula, S., & Zhang, M. (2010). CloudCmp: comparing public cloud providers. Proceedings of the 10th ACM SIGCOMM conference on Internet measurement,
- Mohammadi, M., & Bazhirov, T. (2018). Comparative benchmarking of cloud computing vendors with high performance linpack. Proceedings of the 2nd International Conference on High Performance Compilation, Computing and Communications,
- Mukherjee, J., Wang, M., & Krishnamurthy, D. (2014). Performance testing web applications on the cloud. 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops,
- Shah, S. A. R., Waqas, A., Kim, M.-H., Kim, T.-H., Yoon, H., & Noh, S.-Y. (2021). Benchmarking and performance evaluations on various configurations of virtual machine and containers for cloud-based scientific workloads. *Applied Sciences*, 11(3), 993.
- Tongkaw, S., & Tongkaw, A. (2016). A comparison of database performance of MariaDB and MySQL with OLTP workload. 2016 IEEE conference on open systems (ICOS),
- Witham, M., Bender, I., & Gomes, R. (2019). Comparative Analysis of MariaDB's Performance Efficiency as a Suitable Replacement for MySQL. *Proceedings of the 2019 Midwest Instruction and Computing Symposiu, Fargo, ND, USA*, 5-6.
- Zboril, M., & Svatá, V. (2025). Performance comparison of cloud virtual machines. *Journal of Systems and Information Technology*, 27(2), 197-213.