MCKNIGHT CONSULTING GROUP

High-Concurrency TPC-C Benchmarking

Performance Evaluation: MariaDB Enterprise Server 12.3 vs MariaDB Enterprise Server 11.8 and MySQL Community Edition 9.7.2

McKnight Consulting Group

William McKnight and Jake Dolezal

Sponsored by MariaDB

October 2026

Table of Contents

Executive Summary 2
Benchmark Methods and Platforms 3
- HammerDB TPROC-C Workload 3
- Infrastructure 4
- Test Procedure 4
- Database Configurations 5
- Test Measurements 7

Benchmark Results 7
- MariaDB 12.3 vs MariaDB 11.8 9

Conclusion 11
About MariaDB 13
About McKnight Consulting Group 14
Disclaimer 15

Executive Summary

Database performance is crucial to an enterprise, directly impacting the user experience with applications and AI agents, as well as business operations and data integrity. Fast transaction processing, low latency, and high throughput enable businesses to operate efficiently, drive revenue, and maintain competitiveness. Conversely, poor performance leads to frustrated users, reduced productivity, and increased risk to the business.

Modern servers now pack 96 or more cores into a single socket. On hardware this large, the hardware is rarely the limit. The question is whether the database itself can turn those cores into transactions. A database that stalls on internal contention leaves expensive capacity idle, and the organization pays for compute it cannot use.

MariaDB Enterprise Server is an enterprise open-source relational database, built on the MariaDB community server and compatible with MySQL. It runs transactional applications that demand high performance, scalability, and reliability.

We tested MariaDB Enterprise Server 12.3.3-0 against its predecessor, MariaDB Enterprise Server 11.8.3-1 released in late 2025, and against MySQL 9.7.2 Community to determine comparative transactional performance on identical Google Cloud infrastructure. We used a workload derived from TPC-C, which is a widely recognized benchmark for measuring database performance. All tests ran with the database fully in memory, which makes this a test of how well each database scales across a high core count.

While MariaDB Enterprise Server 12.3.3-0 delivers a 21% higher overall peak throughput compared to MySQL 9.7.2 (2.65M vs. 2.20M NOPM), its performance advantage is even more pronounced under identical concurrency levels. **At equivalent active loads between 64 and 120 Virtual Users, MariaDB ES 12.3.3-0 runs 25% faster than MySQL 9.7.2.** Compared to its predecessor (MariaDB ES 11.8.3-1), the new release delivers 56% higher peak throughput overall and up to 60%+ higher throughput at high concurrency levels.

Because all three databases ran on the same instance at the same hourly cost, the throughput advantage is also a price-performance advantage: **MariaDB 12.3.3-0 processes 21% more transactions per dollar than MySQL and 56% more than MariaDB 11.8.3-1.**

These tests show that MariaDB 12.3 is a substantial step forward over the 11.8 release and that it now outperforms MySQL on large, many-core hardware, making it a strong choice for enterprises running demanding transactional workloads.

Benchmark Methods and Platforms

This section analyzes the methods we used in our database transactional performance testing. The benchmark was executed using the following setup, environment, standards, and configurations. We include the full configuration of each database so that the results can be reproduced.

HammerDB TPROC-C Workload

The workload and data used in the benchmark were a workload derived from the well-recognized industry standard benchmark TPC-C. This is not an official TPC-C. From tpc.org[1]:

> "TPC-C is an on-line transaction processing (OLTP) benchmark. TPC-C involves a mix of five concurrent transactions of different types and complexity either executed on-line or queued for deferred execution. The database is comprised of nine types of tables with a wide range of record and population sizes. … While the benchmark portrays the activity of a wholesale supplier, TPC-C is not limited to the activity of any particular business segment, but, rather represents any industry that must manage, sell, or distribute a product or service."

To perform the workload, we use a driver. There are several TPC-C-like drivers available for free. We used HammerDB TPROC-C v6.0[2].

For HammerDB TPROC-C, we used the following testing parameters:

Parameter	Value
Benchmark	TPROC-C
Warehouses	768
Virtual Users	Stepped ladder, one test per level (1, 16, 32, 64, 88, 96, 120, 152)
Ramp-up	2 minutes
Duration	5 minutes (timed)
Driver mode	Stored procedures, Standard (not Use All Warehouses)
Keying and thinking time	false
Time profile	true (response-time percentiles)
Connection	Local Unix socket; HammerDB ran on the same instance

Table 1: HammerDB TPROC-C parameters

These parameters are configured to measure maximum database engine throughput and tail latency under heavy stress while eliminating external system bottlenecks. Running HammerDB locally over a Unix socket removes network and protocol overhead, disabling keying and thinking time maximizes continuous transactional pressure, and using stored procedures keeps execution logic directly on the server. A scale of 768 warehouses creates a realistic 75–80 GB data footprint to prevent artificial

[1] More can be learned about the TPC-C benchmark at http://www.tpc.org/tpcc/.
[2] We used HammerDB 6.0 (cloud pre-release build). For more information about or to download HammerDB, visit https://www.hammerdb.com/.

cache shortcuts or premature lock bottlenecks, while the 2-minute ramp-up ensures caches and buffers reach a stable state before the 5-minute timed run. Finally, stepping virtual users from 1 to 152 tests hardware scaling up to and beyond core saturation, with response-time profiling enabled to reveal queueing and latency degradation as concurrency climbs.

These are typical settings for HammerDB engine stress tests.

This TPC-C workload simulation tests database performance under demanding conditions. The buffer pool of every database was sized to hold the entire database, and the databases ran without durable commits or binary logging. This removes disk I/O from the critical path and isolates the ability of each database engine to process transactions across 192 vCPUs. Each virtual user issues transactions back to back with no keying or thinking time, so the number of virtual users equals the number of concurrent transactions in flight.

Infrastructure

To perform our testing, we compared the performance of:

- **MariaDB Enterprise Server 12.3.3-0** (12.3.3-0-0)
- **MariaDB Enterprise Server 11.8.3-1**
- **MySQL 9.7.2** Community, built from source

All three databases ran, one at a time, on the same Google Cloud instance. The following table details the deployed system:

Item	Value
Cloud	Google Cloud
Region / zone	us-east1-b
Instance type	c4d-highmem-192-lssd
CPU	AMD EPYC 9B45 (Turin), 1 socket, 96 cores, SMT on (192 vCPU), 1 NUMA node, 384 MiB L3
Memory	1,487 GiB
Storage	16 × 375 GB Local SSD, mdadm RAID 0 (512 KiB chunk), 5.9 TB
File system	ext4 (stride=128, stripe_width=2048), mounted noatime, nodiratime
Operating system	Ubuntu 24.04.5 LTS, kernel 7.0.0-1011-gcp
Load driver	HammerDB 6.0, on the same instance

Table 2: Test infrastructure

This testbed is a top-tier cloud compute configuration designed to eliminate external I/O and memory bottlenecks.

Test Procedure

We followed the same procedure for every database:

- **Fresh build:** the schema was rebuilt for every test series, after an fstrim of the array and a drop of the operating system page cache.
- **Clean start for each test:** before each virtual-user level, we waited until the buffer pool had flushed dirty pages, so that every test started from the same state.
- **Every level runs:** each database ran its entire virtual-user ladder. No test was stopped early, and we report the best result from each ladder.
- **Verified settings:** the tuned flushing, purge, deadlock-detection, and thread-pool settings were read back from the running server before the first test.

Database Configurations

The three databases share a common, non-durable InnoDB configuration. Table 3 lists the settings common to all three. Table 4 lists every setting that differs between them.

Setting	Value (all three databases)
skip_log_bin	1 (binary log off)
innodb_flush_log_at_trx_commit	0
innodb_doublewrite	0
innodb_buffer_pool_size	256G
innodb_log_buffer_size	64M
innodb_log_file_size	64GB
innodb_adaptive_hash_index	0
innodb_adaptive_flushing	1
innodb_flush_neighbors	0
innodb_read_ahead_threshold	0
innodb_lru_scan_depth	128
innodb_read_io_threads / innodb_write_io_threads	8 / 8
innodb_use_native_aio	1
innodb_open_files	500005
innodb_stats_persistent	1
innodb_log_write_ahead_size	4096
innodb_undo_log_truncate	on
innodb_max_purge_lag / _delay	0 / 1
innodb_fast_shutdown	1
innodb_buffer_pool_dump_at_shutdown / load_at_startup	0 / 0
innodb_io_capacity / _max	5000 / 40000
innodb_max_dirty_pages_pct / _lwm	90 / 0
innodb_adaptive_flushing_lwm	10
innodb_purge_threads / innodb_purge_batch_size	32 / 5000
innodb_deadlock_detect	OFF
innodb_lock_wait_timeout	50 (default)
table_open_cache / table_open_cache_instances	2000 / 31
max_connections	100000
back_log	1500
max_prepared_stmt_count	128000

Setting	Value (all three databases)
join_buffer_size / sort_buffer_size	32K / 32K
character_set_server / collation_server	latin1 / latin1_swedish_ci
transaction_isolation	REPEATABLE-READ
performance_schema	OFF
event_scheduler	OFF

Table 3: Configuration common to all three databases

Setting	MariaDB 12.3.3-0	MariaDB 11.8.3-1	MySQL 9.7.2
thread_pool_size	192 (default: one per vCPU)	192	n/a
thread_pool_max_threads	1024	1024	n/a
Buffer pool instances / page cleaners	1 / 1	1 / 1	48 / 48 (automatic)
metadata_locks_instances	61	61	n/a
innodb_index_shrink	OFF	n/a (not in 11.8.3)	n/a
innodb_undo_tablespaces	1	1	default
innodb_max_purge_lag_wait	0	0	n/a

Table 4: Settings that differ between the databases

The common settings are configured to maximize raw in-memory transactional compute by stripping out persistence, locking, and diagnostic bottlenecks across all three engines. By disabling binary logging, transaction-commit disk flushes (innodb_flush_log_at_trx_commit = 0), and doublewrite buffering, the workload bypasses write latency and per-commit fsync stalls. Allocating a 256 GB buffer pool ensures the entire 768-warehouse working set remains cached in RAM without disk reads, while a 64 GB redo log prevents sync-checkpoint interruptions during the test window. Finally, internal serialization points and runtime overhead are eliminated by turning off adaptive hash indexing, deadlock detection, and the performance schema, alongside tuning I/O capacity and thread buffers for high-concurrency SSD throughput.

The differing settings reflect engine-level architectural divergences, version deprecations, and feature parity gaps between the MariaDB and MySQL branches. MariaDB leverages its built-in Community thread pool sized to 192 vCPUs to prevent thread-thrashing at high concurrency, whereas Community MySQL lacks this open-source feature and must rely on standard connection threading. In memory management, MariaDB enforces a single unified buffer pool instance due to internal mutex redesigns, while MySQL divides its cache across 48 instances and page cleaners to partition lock contention. Additional differences stem from MariaDB-specific controls like innodb_index_shrink, partitioning metadata locks (metadata_locks_instances = 61), and explicit purge-wait throttling knobs, which are either obsolete, unexposed, or managed under internal automated heuristics in modern MySQL.

Test Measurements

Our benchmark collected three measurements:

- **NOPM (new orders per minute):** HammerDB's primary metric. It counts the New Order transactions committed per minute and is comparable across databases, because it is taken from the database's own tables rather than from database-specific counters.
- **TPM (transactions per minute):** measures the throughput of all database transactions that a system can handle per minute, reflecting its transaction processing capability.
- **CPU utilization:** the share of the instance's 192 vCPUs in use during each test, which shows how efficiently each database converts compute into transactions.

For each database, we report the best NOPM across its virtual-user ladder, along with the TPM, CPU utilization, and New Order 99th-percentile response time at that point. Because all three databases ran on the same instance, the infrastructure cost is identical, and relative throughput is also relative price-performance.

Benchmark Results

The benchmark test compares the transactional performance of MariaDB Enterprise Server 12.3.3-0, MariaDB Enterprise Server 11.8.3-1, and MySQL 9.7.2 on the same Google Cloud c4d-highmem-192-lssd instance.

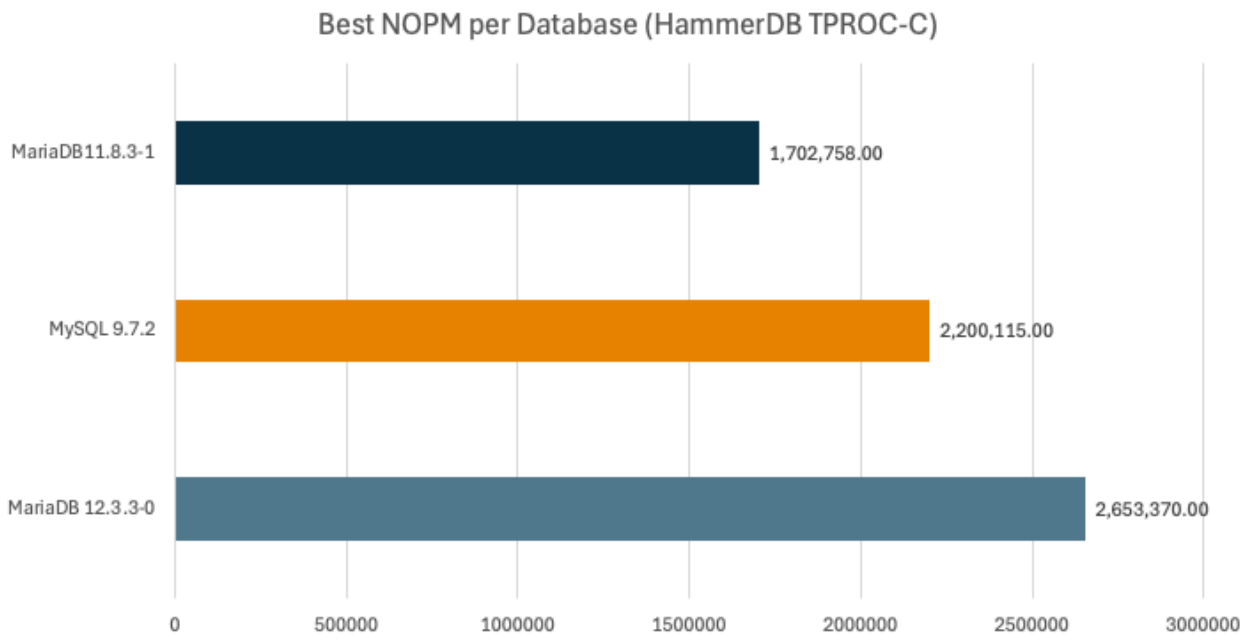


Figure 1: Best Throughput Achieved (NOPM)

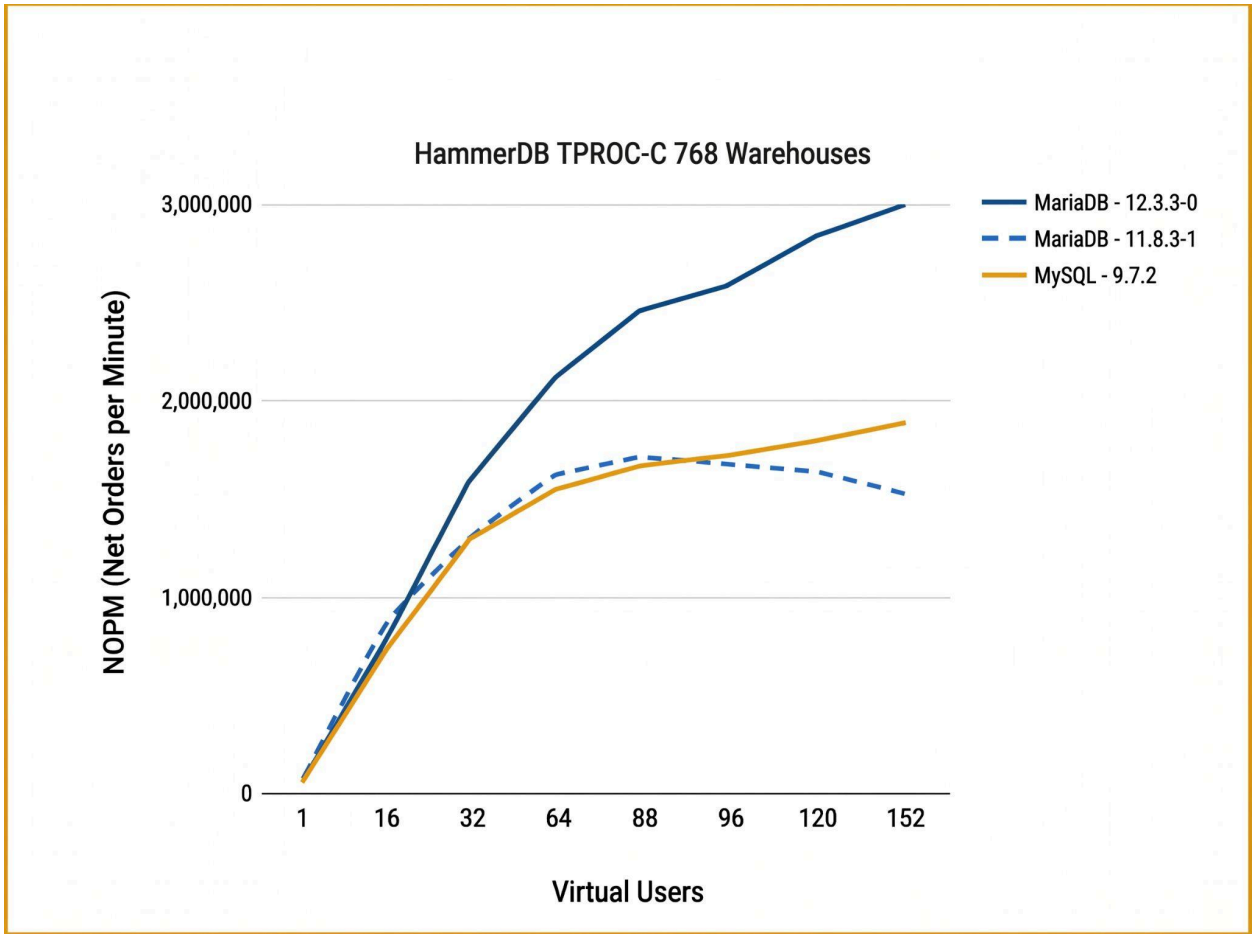


Figure 2: Throughput (NOPM) Scaling Across Virtual Users for MariaDB ES 12.3.3-0, MariaDB ES 11.8.3-1, and MySQL 9.7.2

Here are the key quantitative results:

Performance Metrics

- **MariaDB 12.3.3-0:** 2,653,370 best NOPM (6,162,972 TPM) @ 152 virtual users
- **MariaDB 11.8.3-1:** 1,702,758 best NOPM (3,955,632 TPM) @ 88 virtual users
- **MySQL 9.7.2:** 2,200,115 best NOPM (5,110,330 TPM) @ 152 virtual users

Performance Differences

- **Release over release:** MariaDB 12.3.3-0 is 56% faster than MariaDB 11.8.3-1 (calculated as (2,653,370 − 1,702,758) / 1,702,758)
- **Competitive:** MariaDB 12.3.3-0 is 21% faster than MySQL 9.7.2 (calculated as (2,653,370 − 2,200,115) / 2,200,115)

Measurement	MariaDB 12.3.3-0	MariaDB 11.8.3-1	MySQL 9.7.2
Best NOPM	2,653,370	1,702,758	2,200,115
TPM at best NOPM	6,162,972	3,955,632	5,110,330

Measurement	MariaDB 12.3.3-0	MariaDB 11.8.3-1	MySQL 9.7.2
Virtual users at best NOPM	152	88	152
CPU utilization at best NOPM	70%	40%	77%
New Order p99 response time	3.60 ms	6.54 ms	3.43 ms

Table 5: Results at each database's best throughput

MariaDB 12.3 vs MariaDB 11.8

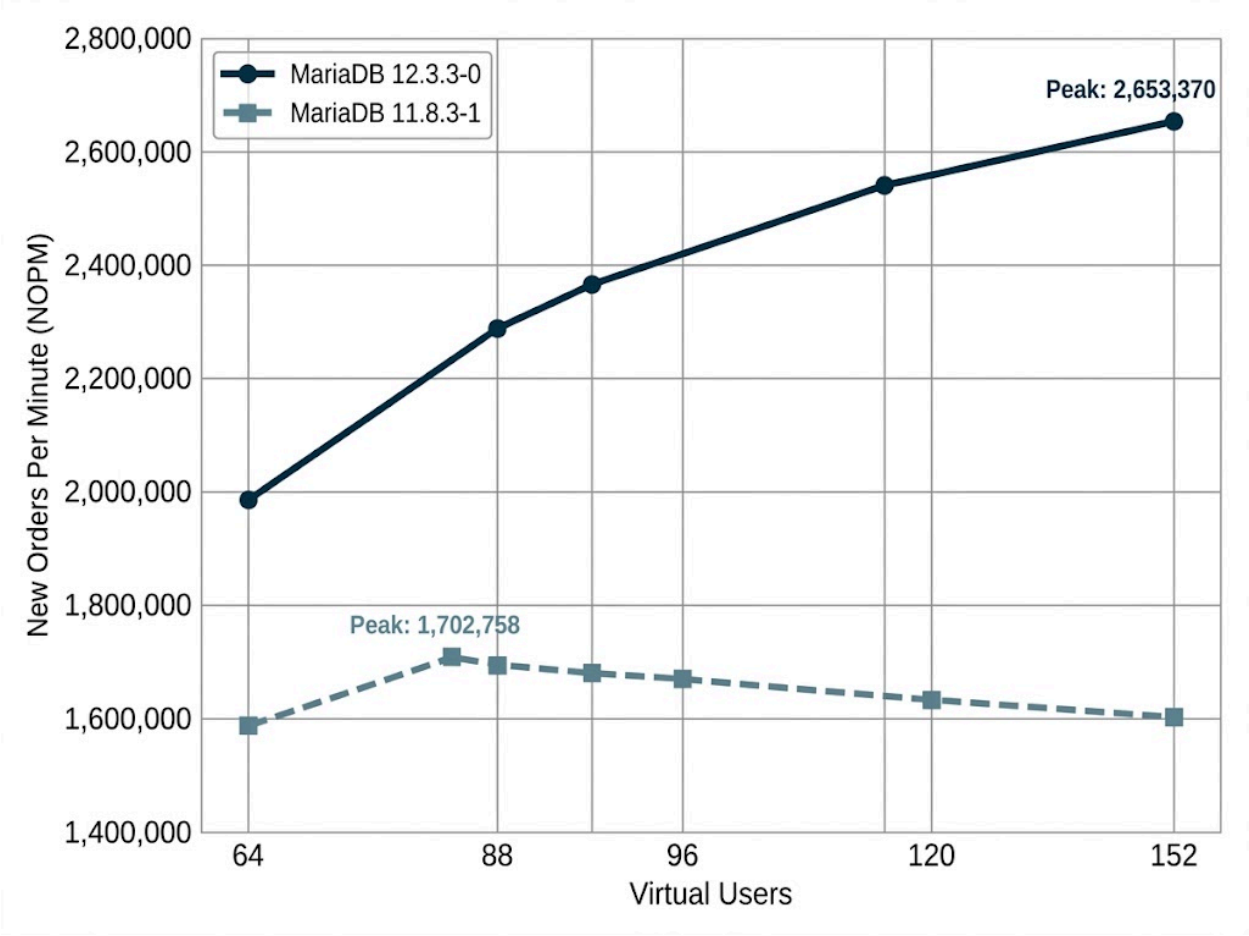


Figure 3: Throughput by Virtual Users, MariaDB 12.3.3-0 vs MariaDB 11.8.3-1

As seen in **Figure 3**, MariaDB 11.8.3-1 reaches its best result of 1,702,758 NOPM at only 88 virtual users. Adding more virtual users does not add throughput. Instead, throughput declines steadily to 1,629,449 NOPM at 152 virtual users, while 60% of the CPU sits idle at its peak. The database cannot put the additional cores to work.

In contrast, MariaDB ES 12.3.3-0 continues scaling to 152 VUs (2,653,370 NOPM), representing a **56% peak-to-peak gain**. At equivalent concurrency levels, ES 12.3.3-0 delivers **25% to 63% higher throughput** across the ladder, while reducing p99 latency from 6.54 ms to 3.60 ms

At every virtual-user level that both releases ran (88, 120, 152), MariaDB 12.3.3-0 delivers 25% to 63% more throughput than MariaDB 11.8.3-1. New Order response times improve as well: the 99th-percentile response time at the best result falls from 6.54 ms on 11.8.3-1 to 3.60 ms on 12.3.3-0, even though 12.3.3-0 is processing 56% more orders.

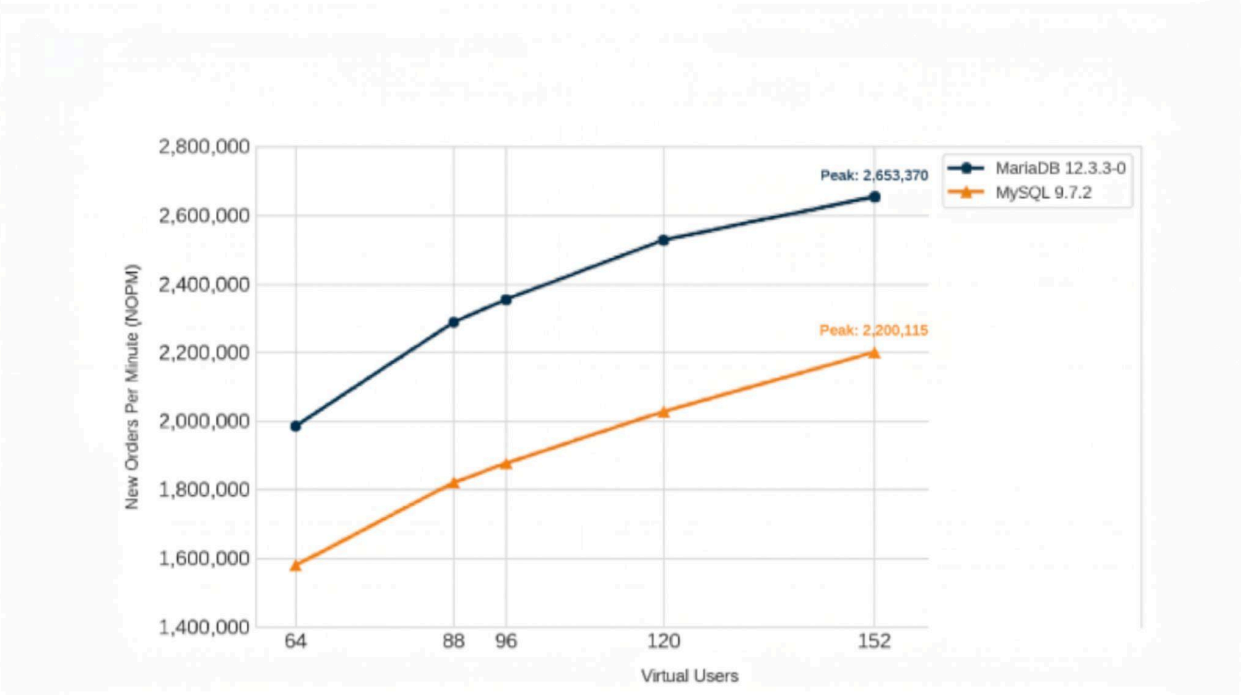


Figure 4: Throughput by Virtual Users, MariaDB 12.3.3-0 vs MySQL 9.7.2

As seen in **Figure 4**, MariaDB 12.3.3-0 outperforms MySQL 9.7.2 by 21% to 26% at every level from 64 to 152 virtual users, where MySQL reaches its own best result. MariaDB 12.3.3-0's best result of 2,653,370 NOPM is 21% higher than MySQL's best of 2,200,115 NOPM.

Comparing maximum peak results across different concurrency levels shows MariaDB 12.3.3-0 leading MySQL 9.7.2 by 21%. However, analyzing throughput at identical Virtual User (VU) levels reveals an even stronger competitive advantage in standard operational ranges:

- **64 to 120 VUs**: MariaDB ES 12.3.3-0 maintains a **24.7% to 25.7% performance lead** over MySQL 9.7.2
- **152 VUs**: At MariaDB's peak concurrency, it processes **20.6% more orders per minute** than MySQL at the same concurrency level

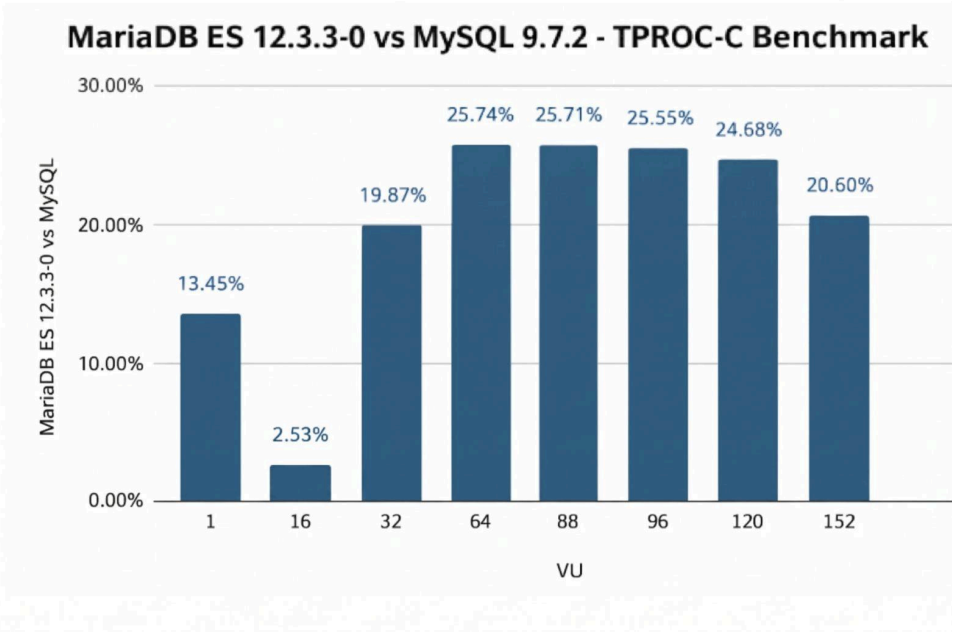


Figure 5: Percentage Throughput Advantage of MariaDB ES 12.3.3-0 over MySQL 9.7.2 Across Virtual User (VU) Levels

Beyond 152 virtual users, both databases are past their peaks.

Virtual users	MariaDB 12.3.3-0	MariaDB 11.8.3-1	MySQL 9.7.2
64	1,985,075	1,588,057	1,578,719
88	2,288,094	1,702,758	1,820,166
96	2,355,413	1,681,738	1,876,118
120	2,528,641	1,667,529	2,028,137
152	2,653,370	1,629,449	2,200,115

Table 6: NOPM by virtual users (best result for each database: 152 and 88 virtual users)

Conclusion

MariaDB Enterprise Server 12.3.3-0 delivers superior performance in this TPC-C-derived benchmark, with a best result of 2,653,370 NOPM, outpacing both its predecessor, MariaDB 11.8.3-1, at 1,702,758 NOPM, and MySQL 9.7.2 at 2,200,115 NOPM.

The improvement over MariaDB 11.8 is dramatic: 56% more throughput, a peak reached at 152 virtual users instead of 88, and a 45% lower New Order 99th-percentile response time. Organizations on MariaDB 11.8 can expect to get considerably more

work from the same hardware by moving to MariaDB 12.3. This is amazing improvement from a direct successor release. Whatever you thought of MariaDB 11.8 needs to be revisited.

Against MySQL, MariaDB 12.3.3-0 delivers 21% more throughput at the peak and 21% to 26% more across the low-to-moderate concurrency levels typical of many production systems. It does so while using less CPU, leaving headroom for growth. Because cost per hour was identical across all tests, these performance advantages translate directly into price-performance: more transactions for every dollar spent on infrastructure, or a smaller instance for the same workload.

The results of this benchmark can inform database strategy for organizations running large transactional workloads. When evaluating database options, consider not only the raw performance of the database but how well it uses the hardware it runs on. MariaDB 12.3's strong showing on a 192-vCPU instance suggests that it is a compelling option for high-performance transactional database deployments.

As database technology continues to evolve, it is essential to stay informed about the latest developments and advancements. The results of this benchmark highlight the importance of ongoing evaluation, as a new release can change the performance picture significantly.

About MariaDB

MariaDB is the Unified Open Database Platform for the Agentic Enterprise. Its mission is to give every enterprise one real-time data platform they can trust, for everything they run today and everything they build next. With one reliable, real-time data foundation for every workload, organizations can stop moving data around and run their apps, analytics and agents on the same live data. Built outward from a proven transactional core, MariaDB brings operational, analytical and agentic workloads together with in-memory speed and real-time access. Its enterprise-grade open platform, built on curated open source, runs across any environment, giving organizations full control over performance, scale, cost and data sovereignty. MariaDB is trusted by organizations including Deutsche Bank, DBS, Nokia, Red Hat, Samsung and Virgin Media O2. For more information, visit mariadb.com.

About McKnight Consulting Group

McKnight Consulting Group exists to systematically advance enterprise data maturity for the AI era, institutionalize financial and empirical rigor in data architecture, deliver high-velocity time-to-value across strategy and execution, and foster continuous data leadership.

Operating with a focus on business justification, knowledge transfer, and early business results, the experienced team of strategists and architects delivers efficient, economical, and sustainable solutions recognized by the Inc. 5000 twice. Connect with our team today at info@mcknightcg.com or visit www.mcknightcg.com to accelerate your data strategy and turn information into an enduring competitive asset.



Disclaimer

McKnight Consulting Group (MCG) runs all its tests to strict ethical standards. The results of the report are the objective and unbiased results of the application of tests to the simulations described in the report. The report clearly defines the selected criteria and process used to establish the field test. The report also clearly states the data set sizes, the platforms, the methods, etc. that were used. The reader is left to determine for themselves how to qualify the information for their individual needs. The report does not make any claim regarding third-party certification and presents the objective results received from the application of the process to the criteria as described in the report. The report strictly measures the metrics listed and does not purport to evaluate other factors that potential customers may find relevant when making a purchase decision. This is a sponsored report. The client chose its configurations, while MCG chose the tests, configured the databases and testing applications, and ran the tests. MCG also chose the most compatible configurations for the other tested platforms. Choosing compatible configurations is subject to judgment. The information necessary to replicate this test is included. Readers are encouraged to compile their own representative configuration and test for themselves.