



NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

NOVA UNIVERSITY OF LISBON

MSC IN COMPUTER SCIENCE

PERFORMANCE COMPARISON BETWEEN DBMSs UNDER TPROC-C WORKLOADS

José Costa (62637)
Rodrigo Albuquerque (70294)
Rodrigo Silva (70567)

DATABASES SYSTEMS

JUNE 8, 2025

Contents

1	Introduction	1
2	Overview of HammerDB	1
2.1	Overview of TPROC-C	1
2.2	TPROC-C vs TPROC-H	1
3	Problem & DBMS Summary	1
4	Benchmark Description	2
5	Methodology	2
5.1	Hardware and Software Setup	3
5.2	Database Setup	3
6	Results	3
7	Discussion	3
8	Conclusions	3
A	TCL Scripts	5
A.1	MySQL TCL Script	5
A.2	MariaDB TCL Script	6
A.3	PostgreSQL TCL Script	7
B	Docker	8
B.1	Docker Compose File	8
B.2	Dockerfile for MySQL	9
B.3	Dockerfile for MariaDB	9
C	Configs	10
C.1	MySQL Configuration File	10
C.2	MariaDB Configuration File	11
C.3	PostgreSQL Configuration File	13

1 Introduction

This project aims to benchmark and compare the performance of different database systems using the TPC-C workload, a standard for evaluating OLTP (Online Transaction Processing) environments.

Automated scripts are used to run identical tests across multiple databases, each configured with similar settings to ensure a fair comparison.

Key metrics such as Transactions Per Minute (TPM) are measured under varying levels of concurrent users.

The results help identify the strengths and limitations of each database in handling transactional workloads.

In this project, the databases used were PostgreSQL, MySQL and MariaDB.

In total, we ran 54 tests:

- 4 tests scaling the number of virtual users (2 4 8 12) and warehouses (VU*5) on all PCs and databases (48 tests in total);
- 1 test with the number of virtual users set to the same as the number of threads in that PC, warehouses set to VU*5, *allwarehouse* = true on all databases on just one PC (3 test in total);
- 1 test with the number of virtual users set to the same as the number of threads in that PC, warehouses set to VU*5, with default config on all databases on just one PC (3 test in total).

2 Overview of HammerDB

HammerDB is a free, open-source tool for benchmarking the performance of relational databases [1].

It supports popular databases like Oracle, SQL Server, PostgreSQL, MySQL, and more. HammerDB uses industry-standard workloads such as TPROC-C and TPROC-H to simulate real-world database activity.

It offers both a graphical interface and command-line options, making it suitable for developers, DBAs, and system administrators to test, compare, and tune database performance.

In some cases we used HammerDB in docker containers to run the tests, which allows for easy setup and isolation of the testing environment.

In another case, we used the Windows version of HammerDB to run the tests on a Windows machine.

2.1 Overview of TPROC-C

TPROC-C is a benchmark designed to evaluate the performance of database management systems (DBMS) using a transactional workload. It simulates a typical online transaction processing (OLTP) environment, focusing on operations like inserts, updates, and deletes across multiple tables.

2.2 TPROC-C vs TPROC-H

TPROC-H is a benchmark designed for data warehousing and analytical workloads, while TPROC-C is focused on transactional processing. TPROC-H emphasizes complex queries and large data sets, whereas TPROC-C simulates real-time transactions with a focus on insert, update, and delete operations.

3 Problem & DBMS Summary

Modern applications progressively depend on robust, scalable database systems to effectively manage workloads in a transactional manner. Choosing the right Database Management System (DBMS) is critical to achieve the best possible performance, especially where concurrency is high and the workload is varied. With so many DBMSs to select from, with their own strengths, configurations, and community support, getting the right one can be problematic.

The main objective of this study is to provide a clear comparison of how three of the most popular open-source DBMSs—PostgreSQL, MySQL, and MariaDB—perform under TPROC-C workloads. This

is a close approximation of OLTP environments and thus is suitable to use in evaluating systems for high-throughput transaction processing.

A brief overview of the DBMSs under test is provided below:

- **PostgreSQL**: It is renowned for support of sophisticated query capabilities, extensibility, and strict support for data integrity features.
- **MySQL**: Extensively used in web development, valued as being easy to use and swift, with excellent ecosystem and support.
- **MariaDB**: A fork of MySQL with an emphasis on improved speed, open development, and additional storage engines.

Each DBMS was installed with identical hardware and software configurations to provide a level playing field for the tests. Experiments were designed to highlight differences in how each system handles transaction loads, concurrency, and configuration parameters. Through comparison of performance in a systematic manner across controlled tests, this study aims to guide database selection on the grounds of empirical evidence and not assumptions or anecdotal experience.

4 Benchmark Description

5 Methodology

5.1 Hardware and Software Setup

PC	1	2	3	4
OS	Windows 11	Windows 11	Linux (Unraid)	MacOS Sequoia
CPU	Intel i7-13700H	AMD Ryzen 5 3600	Intel i3-10100F	Apple M1
Cores	14 (6P 8E)	6	4	8
Threads	20	12	8	8
RAM	16GB	16GB	32GB	16GB
Disk	SSD M.2 NVMe	SSD M.2 NVMe	SSD M.2 NVMe	SSD M.2 NVMe
Read	3500 MB/s	2500 MB/s	3500 MB/s	3400 MB/s
Write	2700 MB/s	2100 MB/s	3300 MB/s	2800 MB/s
Test type	Docker	Bare metal	Docker	Docker

Table 1: Hardware used in the benchmarks

In PCs **1**, **3**, and **4**, we used HammerDB and all the databases in docker containers to run the tests, which allows for easy setup and isolation of the testing environment.

We used docker compose for this setup, which allowed us to easily run the same tests on different machines with the same configuration.

In PC **2**, we used everything installed on the host machine (Bare metal).

5.2 Database Setup

6 Results

7 Discussion

8 Conclusions

Bibliography

- [1] Wikipedia contributors. *HammerDB — Wikipedia, The Free Encyclopedia*. [Online; accessed 25-May-2025]. 2025. URL: <https://en.wikipedia.org/w/index.php?title=HammerDB&oldid=1275860580>.

A TCL Scripts

A.1 MySQL TCL Script

Listing 1: TCL script for MySQL

```
1  #!/usr/bin/tclsh
2  # Set Database & Benchmark
3  dbset db mysql
4  dbset bm TPC-C
5
6  # DB configs
7  diset connection mysql_host 172.22.0.3
8  diset connection mysql_port 3306
9  diset connection mysql_socket "/var/run/mysqld/mysqld.sock"
10 diset tpcc mysql_user root
11 diset tpcc mysql_pass 1234
12 diset tpcc mysql_dbase tpcc
13
14 # Default for WH and VU
15 diset tpcc mysql_count_ware 50
16 diset tpcc mysql_num_vu 10
17
18 # Driver script options
19 diset tpcc mysql_timeprofile true
20 diset tpcc mysql_async_scale false
21 diset tpcc mysql_driver timed
22 # Ensure test is limited by time
23 diset tpcc mysql_total_iterations 1000000
24 # Timed duration
25 diset tpcc mysql_rampup 2
26 diset tpcc mysql_duration 8
27 # Distribute load
28 diset tpcc mysql_allwarehouse false
29
30 # Transactions options
31 tcset refreshrate 10
32 tcset logtotemp 1
33 tcset unique 1
34 tcset timestamps 1
35 # Run
36 foreach z {2 4 8 12} {
37     set w [expr {$z * 5}]
38     diset tpcc mysql_count_ware $w
39     puts "Building Schema for $z TEST"
40     # Delete possible previous data
41     deleteschema
42     vudestroy
43     # Build and load
44     buildschema
45     vudestroy
46     loadscript
47     # Vuser options
48     vuset delay 500
49     vuset repeat 500
50     vuset iterations 1
51     vuset showoutput 0
52     vuset logtotemp 1
53     vuset unique 1
54     vuset nobuff 0
55     vuset timestamps 1
56     puts "Starting $z VU TEST"
57     tcstart
58     vuset vu $z
59     vucreate
60     vurun
61     puts "Waiting 1 for cleanup and collection"
62     after 60000
63     puts "Destroying VU"
64     vudestroy
65     tcstop
66 }
```

A.2 MariaDB TCL Script

Listing 2: TCL script for MariaDB

```
1  #!/usr/bin/tclsh
2  # Set Database & Benchmark
3  dbset db maria
4  dbset bm TPC-C
5
6  # DB configs
7  diset connection maria_host 172.22.0.4
8  diset connection maria_port 3306
9  diset connection mysql_socket "/var/run/mysqld/mysqld.sock"
10 diset tpcc maria_user root
11 diset tpcc maria_pass 1234
12 diset tpcc maria_dbase tpcc
13
14 # Default for WH and VU
15 diset tpcc maria_count_ware 50
16 diset tpcc maria_num_vu 10
17
18 # Driver script options
19 diset tpcc maria_timeprofile true
20 diset tpcc maria_async_scale false
21 diset tpcc maria_driver timed
22
23 # Ensure test is limited by time
24 diset tpcc maria_total_iterations 1000000
25
26 # Timed duration
27 diset tpcc maria_rampup 2
28 diset tpcc maria_duration 8
29 # Distribute load
30 diset tpcc maria_allwarehouse false
31
32 # Transactions options
33 tcset refreshrate 10
34 tcset logtotemp 1
35 tcset unique 1
36 tcset timestamps 1
37
38 # Run
39 foreach z {2 4 8 12} {
40     set w [expr {$z * 5}]
41     diset tpcc maria_count_ware $w
42     puts "Building Schema for $z TEST"
43     # Delete possible previous data
44     deleteschema
45     vudestroy
46     # Build and load
47     buildschema
48     vudestroy
49     loadscript
50     # Vuser options
51     vuset delay 500
52     vuset repeat 500
53     vuset iterations 1
54     vuset showoutput 0
55     vuset logtotemp 1
56     vuset unique 1
57     vuset nobuff 0
58     vuset timestamps 1
59     puts "Starting $z VU TEST"
60     tcstart
61     vuset vu $z
62     vucreate
63     vurun
64     puts "Waiting 1 for cleanup and collection"
65     after 60000
66     puts "Destroying VU"
67     vudestroy
68     tcstop
69 }
```


A.3 PostgreSQL TCL Script

Listing 3: TCL script for PostgreSQL

```
1  #!/usr/bin/tclsh
2  # Set Database & Benchmark
3  dbset db pg
4  dbset bm TPC-C
5
6  # DB configs
7  diset connection pg_host 172.22.0.2
8  diset connection pg_port 5432
9  diset connection pg_sslmode prefer
10 diset tpcc pg_superuser postgres
11 diset tpcc pg_superuserpass 1234
12 diset tpcc pg_defaultdb postgres
13 diset tpcc pg_user tpcc
14 diset tpcc pg_pass tpcc
15 diset tpcc pg_dbase tpcc
16 diset tpcc pg_tspspace pg_default
17
18 # Default for WH and VU
19 diset tpcc pg_count_ware 50
20 diset tpcc pg_num_vu 10
21
22 # Driver script options
23 diset tpcc pg_timeprofile true
24 diset tpcc pg_async_scale false
25 diset tpcc pg_driver timed
26 # Ensure test is limited by time
27 diset tpcc pg_total_iterations 1000000
28 # Timed duration
29 diset tpcc pg_rampup 2
30 diset tpcc pg_duration 8
31 # Distribute load
32 diset tpcc pg_allwarehouse false
33 # Transactions options
34 tcset refreshrate 10
35 tcset logtotemp 1
36 tcset unique 1
37 tcset timestamps 1
38 # Run
39 foreach z {2 4 8 12} {
40     set w [expr {$z * 5}]
41     diset tpcc pg_count_ware $w
42     puts "Building Schema for $z TEST"
43     # Delete possible previous data
44     deleteschema
45     vudestroy
46     # Build and load
47     buildschema
48     vudestroy
49     loadscript
50     # Vuser options
51     vuset delay 500
52     vuset repeat 500
53     vuset iterations 1
54     vuset showoutput 0
55     vuset logtotemp 1
56     vuset unique 1
57     vuset nobuff 0
58     vuset timestamps 1
59     puts "Starting $z VU TEST"
60     tcstart
61     vuset vu $z
62     vucreate
63     vurun
64     puts "Waiting 1 for cleanup and collection"
65     after 60000
66     puts "Destroying VU"
67     vudestroy
68     tcstop
69 }
```

B Docker

B.1 Docker Compose File

Listing 4: Docker Compose file for the databases and HammerDB

```
1 version: "3.9"
2
3 services:
4   mysql:
5     build:
6       context: .
7       dockerfile: Dockerfiles/MySQL/Dockerfile
8     restart: always
9     container_name: MySQL_SBD
10    environment:
11      MYSQL_ROOT_PASSWORD: 1234
12    volumes:
13      - mysql-volume:/var/lib/mysql
14    networks:
15      sbd_network:
16        ipv4_address: 172.22.0.3
17
18   postgres:
19     image: postgres
20     container_name: Postgres_SBD
21     command: postgres -c config_file=/etc/postgresql/postgresql.conf
22     environment:
23       POSTGRES_PASSWORD: "1234"
24     volumes:
25       - postgres-volume:/var/lib/postgresql/data
26       - ./Configs/Hard_Test/postgresql.conf:/etc/postgresql/postgresql.conf:ro
27     networks:
28       sbd_network:
29         ipv4_address: 172.22.0.2
30
31   mariadb:
32     build:
33       context: .
34       dockerfile: Dockerfiles/MariaDB/Dockerfile
35     restart: always
36     container_name: MariaDB_SBD
37     environment:
38       MYSQL_ROOT_PASSWORD: 1234
39       MYSQL_DATABASE: sbdDatabase
40     volumes:
41       - mariadb-volume:/var/lib/mysql:rw
42     networks:
43       sbd_network:
44         ipv4_address: 172.22.0.4
45
46   hammer-gui:
47     image: tpcorg/hammerdb:latest-cloudtk
48     container_name: HammerDB_SBD
49     restart: always
50     ports:
51       - "8081:8081"
52       - "8082:8082"
53       - "8080:8080"
54     depends_on:
55       - mysql
56       - postgres
57       - mariadb
58     volumes:
59       - ./Scripts:/home/HammerDB-4.10/scripts/tcl-scripts
60       - ./stats-logs:/exp-logs
61     networks:
62       sbd_network:
63         ipv4_address: 172.22.0.5
64
65 volumes:
66   mysql-volume:
```

```
67     driver: local
68   postgres-volume:
69     driver: local
70   mariadb-volume:
71     driver: local
72
73   networks:
74     sbd_network:
75       driver: bridge
76       ipam:
77         config:
78           - subnet: 172.22.0.0/16
```

B.2 Dockerfile for MySQL

Listing 5: Dockerfile for MySQL

```
1 FROM mysql:8.4
2
3 RUN mkdir -p /var/log/mysql && \
4     chown -R mysql:mysql /var/log/mysql
5
6 COPY ./Configs/Hard_Test/mysql.cnf /etc/mysql/conf.d/mysql.cnf
7
8 RUN chmod 644 /etc/mysql/conf.d/mysql.cnf
```

B.3 Dockerfile for MariaDB

Listing 6: Dockerfile for MariaDB

```
1 FROM mariadb:11.4
2
3 COPY ./Configs/Hard_Test/my.cnf /etc/mysql/my.cnf
4
5 RUN chmod 644 /etc/mysql/my.cnf
```

C Configs

C.1 MySQL Configuration File

Listing 7: MySQL configuration file

```
1 [mysqld]
2
3 #-----
4 # BASIC SETTINGS
5 #-----
6 port = 3306
7 bind-address = 0.0.0.0
8 max_connections = 1000
9 max_connect_errors = 1000000
10 wait_timeout = 28800
11 interactive_timeout = 28800
12 connect_timeout = 10
13 back_log = 1500
14
15 #-----
16 # CHARACTER SET & COLLATION
17 #-----
18 character-set-server = utf8mb4
19 collation-server = utf8mb4_unicode_ci
20
21 #-----
22 # STORAGE AND INNODB ENGINE
23 #-----
24 innodb_buffer_pool_size = 4G
25 innodb_buffer_pool_instances = 4
26 innodb_log_file_size = 512M
27 innodb_log_buffer_size = 64M
28 innodb_file_per_table = 1
29 innodb_flush_log_at_trx_commit = 2
30 innodb_flush_method = O_DIRECT
31 innodb_io_capacity = 2000
32 innodb_io_capacity_max = 4000
33 innodb_read_io_threads = 8
34 innodb_write_io_threads = 8
35 innodb_purge_threads = 4
36 innodb_doublewrite = 1
37 innodb_autoinc_lock_mode = 2
38 innodb_stats_persistent = 1
39 innodb_lru_scan_depth = 2048
40 innodb_adaptive_flushing = 1
41 innodb_adaptive_hash_index = 0
42 innodb_change_buffering = none
43
44 #-----
45 # TEMPORARY TABLE & BUFFERS
46 #-----
47 tmp_table_size = 256M
48 max_heap_table_size = 256M
49 sort_buffer_size = 1M
50 join_buffer_size = 1M
51 read_buffer_size = 512K
52 read_rnd_buffer_size = 2M
53
54 #-----
55 # LOGGING
56 #-----
57 slow_query_log = 1
58 long_query_time = 2
59 slow_query_log_file = /var/lib/mysql/mysql-slow.log
60 general_log = 0
61
62 #-----
63 # BINARY LOGGING
64 #-----
65 skip-log-bin
66 sync_binlog = 0
```

```

67 | #-----
68 | # SECURITY & COMPATIBILITY
69 | #-----
70 | local_infile = 0
71 | sql_mode = STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION
72 | transaction_isolation = REPEATABLE-READ
73 |
74 | #-----
75 | # PERFORMANCE
76 | #-----
77 | performance_schema = OFF
78 | max_prepared_stmt_count = 12800
79 | table_open_cache = 2000
80 | table_open_cache_instances = 4
81 | open_files_limit = 65535
82 | thread_cache_size = 50
83 | thread_stack = 256K
84 |
85 | #-----
86 | # MONITORING
87 | #-----
88 | innodb_monitor_enable = '%'
89 |
90 | #-----
91 | # PLUGIN & AUTHENTICATION
92 | #-----
93 | require_secure_transport = OFF
94 | caching_sha2_password_auto_generate_rsa_keys = ON
95 |

```

C.2 MariaDB Configuration File

Listing 8: MariaDB configuration file

```

1 | [mysqld]
2 |
3 | #-----
4 | # BASIC SETTINGS
5 | #-----
6 | port = 3306
7 | bind-address = 0.0.0.0
8 | max_connections = 1000
9 | max_connect_errors = 1000000
10 | wait_timeout = 28800
11 | interactive_timeout = 28800
12 | connect_timeout = 10
13 | back_log = 1500
14 |
15 | #-----
16 | # CHARACTER SET & COLLATION
17 | #-----
18 | character-set-server = utf8mb4
19 | collation-server = utf8mb4_unicode_ci
20 |
21 | #-----
22 | # STORAGE AND INNODB ENGINE
23 | #-----
24 | innodb_buffer_pool_size = 4G
25 | innodb_buffer_pool_instances = 4
26 | innodb_log_file_size = 512M
27 | innodb_log_buffer_size = 64M
28 | innodb_file_per_table = 1
29 | innodb_flush_log_at_trx_commit = 2
30 | innodb_flush_method = O_DIRECT
31 | innodb_io_capacity = 2000
32 | innodb_io_capacity_max = 4000
33 | innodb_read_io_threads = 8
34 | innodb_write_io_threads = 8
35 | innodb_purge_threads = 4
36 | innodb_doublewrite = 1
37 | innodb_autoinc_lock_mode = 2

```

```

38 innodb_stats_persistent = 1
39 innodb_lru_scan_depth = 2048
40 innodb_adaptive_flushing = 1
41 innodb_adaptive_hash_index = 0
42 innodb_change_buffering = none
43
44 #-----
45 # TEMPORARY TABLE & BUFFERS
46 #-----
47 tmp_table_size = 256M
48 max_heap_table_size = 256M
49 sort_buffer_size = 1M
50 join_buffer_size = 1M
51 read_buffer_size = 512K
52 read_rnd_buffer_size = 2M
53
54 #-----
55 # LOGGING
56 #-----
57 slow_query_log = 1
58 long_query_time = 2
59 slow_query_log_file = /var/lib/mysql/mysql-slow.log
60 general_log = 0
61
62 #-----
63 # BYNARY LOGGING
64 #-----
65 skip-log-bin
66 sync_binlog = 0
67
68 #-----
69 # SECURITY & COMPATIBILITY
70 #-----
71 local_infile = 0
72 sql_mode = STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION
73 transaction_isolation = REPEATABLE-READ
74
75 #-----
76 # PERFORMANCE
77 #-----
78 performance_schema = OFF
79 max_prepared_stmt_count = 12800
80 table_open_cache = 2000
81 table_open_cache_instances = 4
82 open_files_limit = 65535
83 thread_cache_size = 50
84 thread_stack = 256K
85
86 #-----
87 # MONITORING
88 #-----
89 innodb_monitor_enable = '%'

```

C.3 PostgreSQL Configuration File

Listing 9: PostgreSQL configuration file

```
1 #-----
2 # FILE LOCATIONS
3 #-----
4 data_directory = '/var/lib/postgresql/data' # Important: match Docker volume
5 hba_file = '/var/lib/postgresql/data/pg_hba.conf'
6 ident_file = '/var/lib/postgresql/data/pg_ident.conf'
7
8 #-----
9 # CONNECTIONS AND AUTHENTICATION
10 #-----
11 listen_addresses = '*' # Allow external connections (Docker host network)
12 port = 5432 # Default PostgreSQL port
13 max_connections = 100
14
15 #-----
16 # RESOURCE USAGE
17 #-----
18 shared_buffers = 512MB # Adjust depending on host memory (e.g., 25% of RAM)
19 work_mem = 64MB # Suitable for OLTP like TPC-C
20 maintenance_work_mem = 256MB
21 effective_cache_size = 2GB # Depends on total system RAM
22
23 #-----
24 # WRITE-AHEAD LOG
25 #-----
26 wal_level = replica
27 synchronous_commit = off # Can improve write performance (acceptable for benchmarks)
28 checkpoint_timeout = 15min
29 checkpoint_completion_target = 0.9
30 max_wal_size = 2GB
31 min_wal_size = 512MB
32
33 #-----
34 # LOGGING (optional, but useful for debugging)
35 #-----
36 logging_collector = on
37 log_directory = 'log'
38 log_filename = 'postgresql.log'
39 log_statement = 'none'
40 log_min_duration_statement = 1000 # Log queries slower than 1s
```