

Benchmarking Suites

- A benchmark suite is a collection of benchmarks, being a benchmark a test or set of tests used to compare the performance of alternative tools or techniques.
- For Databases , the benchmarking suite is used to compare the performance of how different databases perform in terms of scalability , read/write performance , CPU and memory usage etc.
- Choosing a good benchmarking suite is important as the suite should be accessible and scalable. Also ,for this project , the benchmarking suite should be open-source and easily available.

TS-Benchmark

- The first benchmarking suite that I was going to use and implement for the project was TS-Benchmark .

GitHub : <https://github.com/dbiir/TS-Benchmark>

- This suite supported 4 time series databases.
- The steps to complete the test was generating the data , importing the data in the databases and lastly configuring the parameters and running the tests.

- The building of the benchmarking suite and downloading the databases in local was successful
- But generating the data and loading part got errors as a load.data file is missing from the original code.
- There is an issue raised on GitHub regarding that , But no update or solution of the bug is out yet.

```
[INFO] -----< cn.edu.ruc:TS-BM >-----
[INFO] Building ts-bench 1.0
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] --- clean:3.2.0:clean (default-clean) @ TS-BM ---
[INFO] --- resources:3.3.0:resources (default-resources) @ TS-BM ---
[INFO] Copying 1 resource
[INFO] --- compiler:3.3:compile (default-compile) @ TS-BM ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 5 source files to /Users/spartan/Downloads/TS-Benchmark/ts-benchmark/target/classes
[INFO] /Users/spartan/Downloads/TS-Benchmark/ts-benchmark/src/main/java/cn/edu/ruc/start/TSBM.java: /Users/spartan/Downloads/TS-Benchmark/ts-benchmark/src/main/java/cn/edu/ruc/start/TSBM.java uses or over
rides a deprecated API.
[INFO] /Users/spartan/Downloads/TS-Benchmark/ts-benchmark/src/main/java/cn/edu/ruc/start/TSBM.java: Recompile with -Xlint:deprecation for details.
[INFO] --- resources:3.3.0:testResources (default-testResources) @ TS-BM ---
[INFO] Not copying test resources
[INFO] --- compiler:3.3:testCompile (default-testCompile) @ TS-BM ---
[INFO] Not compiling test sources
[INFO] --- surefire:3.0.0-M8:test (default-test) @ TS-BM ---
[INFO] Tests are skipped.
[INFO] --- jar:2.4:jar (default-jar) @ TS-BM ---
[INFO] Building jar: /Users/spartan/Downloads/TS-Benchmark/ts-benchmark/target/benchmark.jar
[INFO] --- dependency:2.10:copy-dependencies (copy-dependencies) @ TS-BM ---
[INFO] --- resources:3.3.0:resources (default-resources) @ TS-BM ---
[INFO] Copying 1 resource
[INFO] --- compiler:3.3:compile (default-compile) @ TS-BM ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- resources:3.3.0:testResources (default-testResources) @ TS-BM ---
[INFO] Not copying test resources
[INFO] --- compiler:3.3:testCompile (default-testCompile) @ TS-BM ---
[INFO] Not compiling test sources
[INFO] --- surefire:3.0.0-M8:test (default-test) @ TS-BM ---
[INFO] Tests are skipped.
[INFO] --- jar:2.4:jar (default-jar) @ TS-BM ---
[INFO] --- dependency:2.10:copy-dependencies (copy-dependencies) @ TS-BM ---
[INFO] --- install:3.1.0:install (default-install) @ TS-BM ---
[INFO] Installing /Users/spartan/Downloads/TS-Benchmark/ts-benchmark/pom.xml to /Users/spartan/.m2/repository/cn/edu/ruc/TS-BM/1.0/TS-BM-1.0.pom
[INFO] Installing /Users/spartan/Downloads/TS-Benchmark/ts-benchmark/target/benchmark.jar to /Users/spartan/.m2/repository/cn/edu/ruc/TS-BM/1.0/TS-BM-1.0.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.237 s
```



基础依赖编译完成

```
spartan@IMS-089MBA ts-benchmark % cd ..
```

```
spartan@IMS-089MBA TS-Benchmark % cd tsdb-test
```

```
spartan@IMS-089MBA tsdb-test % vim pom.xml
```

```
[spartan@IMS-089MBA tsdb-test % sh run.sh
```

```
/Users/spartan/Downloads/TS-Benchmark/tsdb-test
```

```
Error: Could not find or load main class TSDBTest
```

```
Caused by: java.lang.ClassNotFoundException: TSDBTest
```

```
[spartan@IMS-089MBA tsdb-test % vim run.sh
```

```
[spartan@IMS-089MBA tsdb-test % cd ..
```

```
[spartan@IMS-089MBA TS-Benchmark % cd tsdb-test/data/load
```

```
[spartan@IMS-089MBA load % python3 generate_timescale.py
```

```
Traceback (most recent call last):
```

```
File "/Users/spartan/Downloads/TS-Benchmark/tsdb-test/data/lo
```

```
data_src = open('./load.data')
```

```
FileNotFoundError: [Errno 2] No such file or directory: './load
```

```
spartan@IMS-089MBA load % █
```

TSBS (Time Series Benchmarking Suite)

- Another benchmarking suite was searched called TSBS. This suite supports many time series databases along with NoSQL databases that can store time series data.

GitHub : <https://github.com/timescale/tsbs>

- This suite is relatively new, and more updates are given periodically as the GitHub community of this suite is quite active.
- The goal is to benchmark 4 databases (2 time series and 2 NoSQL) and to compare how NoSQL databases perform against native time series databases with 2 databases and a bunch of queries provided by the suite.

Selection of the Databases

- I am using Db-engine to get the rankings of the time series databases and then choosing the best databases that is supported by the suite.

<https://db-engines.com/en/ranking/time+series+dbms/all>

- As we can see from the figure in the next slide , Influx DB and Timescale DB are the top time series databases .
- For NoSQL databases , mongo DB and Cassandra are the most common databases .
- So , these are the 4 databases I have chosen from the supported ones by the suite

include secondary database models

57 systems in ranking, March 2023

| Rank | | | DBMS | Database Model | Score | | |
|----------|----------|----------|-------------------------------|--------------------------|----------|----------|----------|
| Mar 2023 | Feb 2023 | Mar 2022 | | | Mar 2023 | Feb 2023 | Mar 2022 |
| 1. | 1. | 1. | MongoDB | Document, Multi-model | 458.78 | +6.02 | -26.88 |
| 2. | 2. | 2. | Redis | Key-value, Multi-model | 172.45 | -1.39 | -4.31 |
| 3. | 3. | 3. | Teradata | Relational, Multi-model | 63.74 | +0.71 | -5.11 |
| 4. | 4. | 4. | InfluxDB | Time Series, Multi-model | 29.15 | -0.29 | -0.54 |
| 5. | 5. | 5. | Informix | Relational, Multi-model | 21.75 | +0.05 | -1.64 |
| 6. | 6. | 6. | Vertica | Relational, Multi-model | 16.98 | -0.44 | -3.19 |
| 7. | 7. | 7. | ClickHouse | Relational, Multi-model | 14.24 | +0.25 | +1.75 |
| 8. | 8. | 8. | Kdb | Time Series, Multi-model | 8.03 | +0.73 | -0.98 |
| 9. | 9. | 11. | Prometheus | Time Series | 7.33 | +0.27 | +1.00 |
| 10. | 11. | 12. | Graphite | Time Series | 6.59 | -0.16 | +1.11 |
| 11. | 10. | 9. | SingleStore | Relational, Multi-model | 6.54 | -0.24 | -0.90 |
| 12. | 12. | 10. | Microsoft Azure Data Explorer | Relational, Multi-model | 6.51 | +0.04 | -0.84 |
| 13. | 13. | 16. | Trino | Relational, Multi-model | 5.21 | +0.41 | +2.24 |
| 14. | 14. | 13. | TimescaleDB | Time Series, Multi-model | 4.52 | +0.02 | +0.06 |
| 15. | 15. | 14. | RavenDB | Document, Multi-model | 4.04 | -0.17 | +0.43 |
| 16. | 16. | 17. | RRDtool | Time Series | 3.02 | +0.02 | +0.41 |
| 17. | 17. | 19. | DolphinDB | Time Series, Multi-model | 2.69 | -0.16 | +1.16 |
| 18. | 18. | 15. | Apache Druid | Multi-model | 2.63 | +0.14 | -0.63 |
| 19. | 20. | 26. | TDengine | Time Series, Multi-model | 2.43 | +0.02 | +1.67 |
| 20. | 19. | 18. | OpenTSDB | Time Series | 2.32 | -0.12 | +0.53 |
| 21. | 22. | 22. | QuestDB | Time Series, Multi-model | 1.89 | -0.06 | +0.73 |
| 22. | 21. | 21. | GridDB | Time Series, Multi-model | 1.83 | -0.13 | +0.49 |