Performance Evaluation of TimescaleDB for Storage of Historical Data from WinCC OA SCADA Systems

September 2021

AUTHOR:
Mehant Kammakomati

BE-ICS-FT

SUPERVISORS:
Rafal Kulaga
Anthony Hennessey

CERN openlab

ABSTRACT

This project was completed in the scope of NextGen Archiver (NGA) for WinCC OA SCADA systems. The NGA is a new archiver for WinCC OA that uses a pluggable architecture to support multiple database technologies. This project was part of a wider effort to benchmark a range of database technologies to understand their limits in terms of functionality and performance in the context of CERN use cases. The benchmarking methodology involves producing realistic test data and performing write and read benchmarks on the database technologies under the test. The specific focus of this project was to perform ingestion benchmarking on TimescaleDB and PostgreSQL. We obtained an ingestion rate of 80K rows/second for TimescaleDB one-node and 150K rows/second for TimescaleDB two-node making them 2X and 3X higher than that of PostgreSQL which is around 40K rows/second. As the benchmark runs progressed we observed a considerable decline in the ingestion rate for PostgreSQL, but the ingest rate was stable for TimescaleDB. The work on the benchmarks will continue and focus on query performance and evaluation of different schema variants.

TABLE OF CONTENTS

1. INTRODUCTION

This project is part of the wider NextGen Archiver (NGA) development project, the NGA is a new archiver for WinCC OA. WinCC OA is the de-facto standard for creating Supervisory Control and Data Acquisition (SCADA) systems at CERN and is used in almost 700 systems. The SCADA systems monitor the states of sensors and actuators; some of these states are archived through the NGA. Each specific state value that is archived is called a signal, and in addition the system can register alarms based on rules defined against the value of a signal. For the purpose of this project, we have used an example system with 100,000 signals that can trigger around 10 million events per day in total across all the signals. Archiving is a critical function of every SCADA system because it allows operators and experts to see the history of signal changes and alarms. Furthermore, NextGen Archiver supports multiple database technologies through pluggable backends. It is important to know the limits of each of them both in terms of functionality and performance.

The project focuses on comparing the performance of PostgreSQL, which is a technology already supported by the NGA, with its very promising extension targeted specifically at storage of time series – TimescaleDB.

2. BENCHMARKING METHODOLOGY

This section explains the benchmarking workflow used, specifications of the benchmarking environment, and benchmarking objectives.

a. Workflow

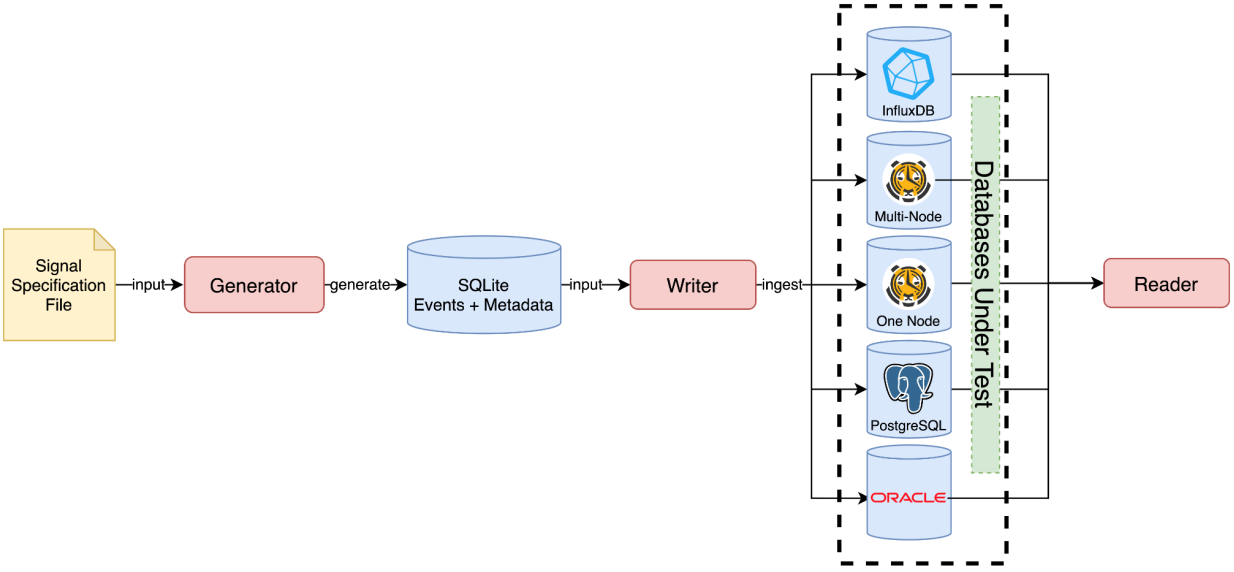

Figure 1. *Workflow for the ingestion benchmark*

Benchmark workflow as shown in Figure 1. can be divided into two stages. The first stage involves the initial step which is to write a Signal Specification File. This file defines how random signal

and alarm data should be generated such that the generated test data resembles real data generated by the systems at CERN. The generated test data is written to an SQLite database. By persisting the test data it can be reused for benchmarks across different databases for a fair comparison. SQLite is chosen because it provides the flexibility to perform SQL-based analysis on the testing data.

In the second stage, each writer component pulls data from the SQLite database and writes them to the target database (database under test) in batches. On every successful write, the writer component records the time taken. These details are logged and these logs can be filtered and used for analysis and visualization. Going forward the reader component will be used to perform query benchmarks on the data written during the write tests.

b. Environment

The initial test setup used for benchmarks consisted of six virtual machines, each with 16GB of memory, 1 TB of IO3 storage (300 MB/s read/write speed), and 8 CPU cores. The target databases and the benchmarking tools were run on separate machines to ensure that they do not compete for resources.

c. Objective

The objective of the project is to compare data ingestion rates across PostgreSQL and TimescaleDB (one-node and multi-node) databases and observe their changes with increasing amounts of data stored in the databases.

3. TIMESCALEDB

TimescaleDB is a new, open-source time-series database that is developed on top of PostgreSQL. The database is architected for time-series data and claims that it allows fast ingest and complex time-based queries.

TimescaleDB claims that the ingest, queries and deletes are many times faster than PostgreSQL and can easily scale to petabytes scale of data. They also provide new time-centric functions for easy time-series data manipulation. TimescaleDB uses the concept of hypertable for single-node version and distributed hypertable for multi-node version.

TimescaleDB is backed by an active open source community and gets regular updates and feature additions making it a reliable service.

4. RESULTS OF THE DATA INGESTION BENCHMARK

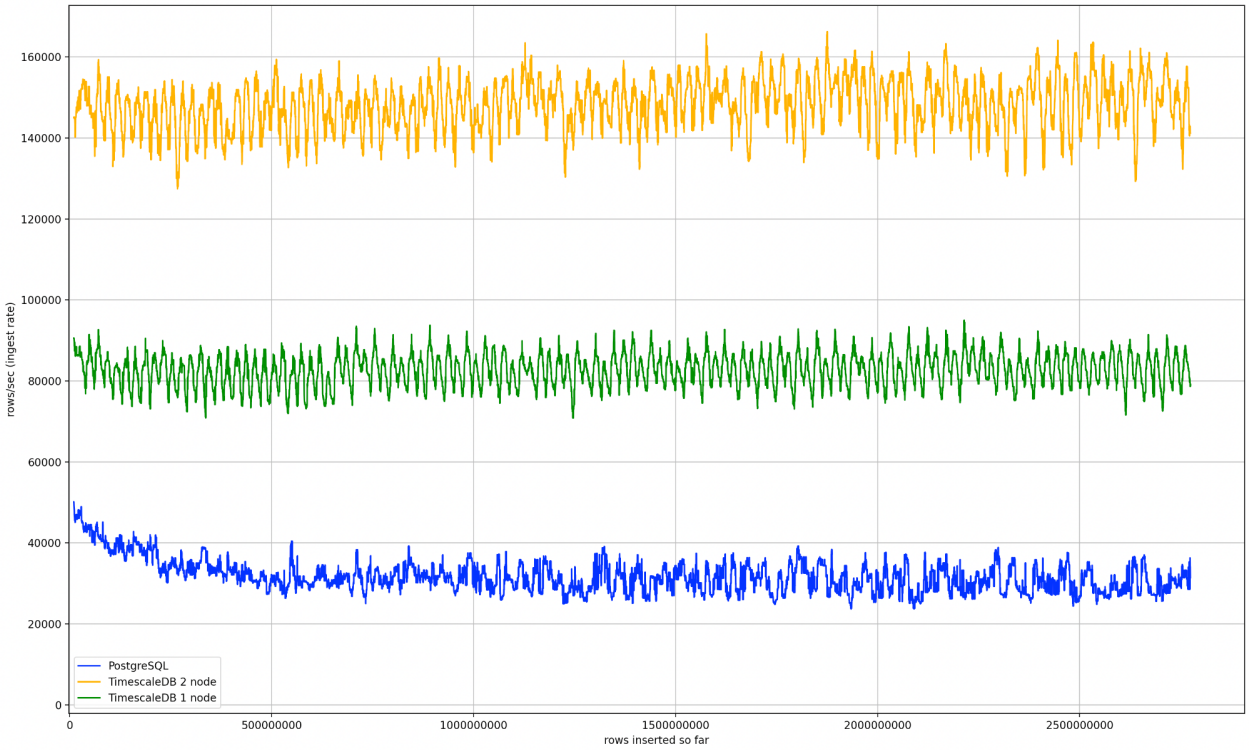

Figure 2. *Ingestion Rate: PostgreSQL vs TimescaleDB One-Node vs TimescaleDB Two-Node*

The ingestion rates of PostgreSQL, TimescaleDB one-node, and TimescaleDB two-node differ largely right from the start of the benchmark. Both TimescaleDB one-node and TimescaleDB two-node show a constant ingestion rate untill the end of the test run. On the other hand, the ingest rate of PostgreSQL starts to decline at the beginning of the test, falling from ~50K rows/sec to ~30K rows/sec. This makes the TimescaleDB one-node setup approximately two times faster than the PostgreSQL setup and the Timescale two-node setup approximately three times faster. The TimescaleDB one-node and two-node setups have stable ingest rates of ~80K rows/sec and ~150K rows/sec respectively.

5. DISK SPACE CONSUMPTION

Database Under Test	Disk Space Consumption
PostgreSQL	451GB
TimescaleDB one-node	427.8GB
TimescaleDB two-node	430.5GB

Table 1. *Disk space consumption of the test dataset on different databases*

Table 1. presents the disk space consumed by each database used in the benchmark. TimescaleDB one-node and two-node have consumed a similar amount of disk space, whereas PostgreSQL consumed a little more, but the differences are almost negligible.

6. SUMMARY AND FUTURE WORK

It can be deduced that TimescaleDB (two-node) is considerably faster than PostgreSQL in terms of ingestion rate. It scales well and provides a stable ingest rate, unlike PostgreSQL that showed a decline. The initial results show that TimescaleDB is a promising time-series database and should be considered for further benchmarking on other database operations. It should be noted that the ingestion performance of all tested databases and configurations is sufficient even for medium-sized systems at CERN. Therefore, query performance will play a much important role in the final choice of database technology.

Future work will consist of performing tests using more performant hardware, performing a query benchmark, and simultaneous ingestion/query tests that mimic production workloads. Performance and functionality of some time series-focused features such as continuous aggregates will also be tested.