Project Proposal

Exploring Computational Capabilities of InfluxDB and TimescaleDB

CS550 – Advanced Operating Systems

February 8th 2021

Elliot Tan

A20382909

Zhenghao Zhao

A20429116

1. Introduction

Time series data has increasingly attracted attention from both research areas and industry companies for the past few years. With the development of Artificial Intelligence, time series data became widely used in machine learning and data mining cases. Financial data is one of the most interesting and lucrative data, such as stock market prices and index prices. Also, time series data plays an important role in forecasting the trend of COVID19 pandemic. Time series databases, which are able to handle time series data efficiently, therefore also became essential. In this project, our team will take two most representative time series databases, InfluxDB and TimescaleDB, as examples to learn about the capabilities, operations and architecture of such systems. In addition, we will process real time series data on these two databases, analyzing their performances and comparing the differences and similarities between them.

2. Background/related works

Some properties of time series databases are as follows[1]. Related data is co-located by a time series database and stored in the same physical location, allowing for better I/O performance. Because of this, queries over a range of data over time tend to also be very fast, and some databases also implement ways to easily make such queries. Time series databases also have to be able to handle a large number of write requests. Take for example continuously recording stock market prices, or measurements from a network in an Internet of Things. This project will take a closer look at how much faster and more efficiently time series databases handle time series information as supposed traditional databases.

There has been some research done on comparing different time series databases. Sanaboyana Tulasi Priyanka conducted a performance evaluation on time series databases and compared the energy consumed by both OpenTSDB and InfluxDB on the same set of machines with the same data. She compared them under 3 conditions: no load, synchronization, and during read write operations. She concluded that InfluxDB used less energy than OpenTSDB in the same scenarios[5]. Another research project by Bader, Kopp, and Falkenthal was a survey and comparison between different open source time series databases. They found 83 TSDBs from their systematic search and grouped and compared them by their scalability, load balancing, and others[3]. Piotr Grzesik and Dariusz Mrozek did a comparative analysis of the time taken to insert and read data for 5 different databases: TimscaleDB, InfluxDB, Riak TS, PostgreSQL, and SQLite. They concluded that while PostgreSQL performed the best on smaller datasets, InfluxDB and Timescale DB were more applicable as they provided[2]. An article by Rob Kiefer shows his comparison between PostgreSQL and TimescaleDB and showed that TimescaleDB far outperforms PostgreSQL when handling time-based data. TimescaleDB was able to perform 20 times better at inserts and 2000 times better at deletes than PostgreSQL.

3. Problem Statement

In this project, we will be looking into influxDB and TimescaleDB , and will develop a deeper understanding of their computational capabilities, their unique operations, and their architecture. We will also be evaluating these time series databases and their differences from traditional

databases such as PostgreSQL. We also hope to uncover the differences in capabilities between influxDB and TimescaleDB.

4. Proposed Solution

In order to reach our objectives, we will be researching deeply into influxDB and TimescaleDB in order to understand how they work, and what they are good for. We will be getting a list of operations on how they work and will be learning how they are used, and evaluating the latencies for each system. We will be using the cloud system Chameleon to compare the databases in order to get a fair comparison.

5. Evaluation

The evaluation of this project is comprised by the following:

- The time series data we will use for this project is from kaggle[7], which includes stock prices of 30 companies from 2016 to 2018. We will use AAPL and AMZN stock data, and maybe process with stock data of other companies if it is possible.
- During evaluation, we will use Google benchmark[8] to evaluate the computation capabilities performances on the time series databases we select (InfluxDB and TimescaleDB). Google benchmark is mainly used for benchmarking functions in C++. With Google benchmark, we can get processing time and CPU iterations of a program, so that we can measure the computation capabilities of the databases.
- Depending on the results we get, we will analyze the performance differences between different datasets.
- Also, we will compare two time series databases, concluding their similarities and comparing their differences theoretically.

6. Conclusions

Time series data has become a great interest in many industry companies. Its capability to store and query time series data efficiently is a considerable assistance for machine learning and data mining. Therefore, it is necessary for us to get to know the architecture and performance of time series databases and get familiar with related operations. We select two most representative time series databases to learn how time series databases are able to handle time series data more efficiently than normal databases. In addition, stock market prices are very interesting time series data, and a lot of research about mining stock prices has been proposed in the last few years, so we decided to use stock price data to evaluate the performance of the databases.

7. Additional information

7a. Timeline

Week	Date	Task (Elliot)	Task (Zhenghao)
Week 1	02/08/2021 - 02/14/2021	Research the architecture of InfluxDB.	Research TimescaleDB. Download AMSN stock data.Create a hypertable to store time series data.
Week 2	02/15/2021 -	Research InfluxDB and data	Insert data to the hypertable. Install

	02/21/2021	sets from kaggle, find out how to create tables in influxDB	and configure TimescaleDB.
Week 3	02/22/2021 - 02/28/2021	Insert data into table. Configure InfluxDB	Process queries using Timescale DB.
Week 4	03/01/2021 - 03/07/2021	Learn how to Query with InfluxDB	Research Google benchmark. Install and set up Google benchmark.
Week 5	03/08/2021 - 03/14/2021	Research Google benchmark. Install and set up Google benchmark.	Test queries on TimescaleDB using Google benchmark.
Week 6	03/15/2021 - 03/21/2021	Test queries on InfluxDB using Google benchmark.	Test other functions on TimescaleDB using Google benchmark.
Week 7	03/22/2021 - 03/28/2021	Continue testing functions on InfluxDB	Research the architecture of TimescaleDB.
Week 8	03/29/2021 - 04/04/2021	Result analysis.Compare the performance between two databases.	Result analysis.Compare the performance between two databases.
Week 9	04/05/2021 - 04/11/2021	Buffer week	Buffer week.
Week 10	04/12/2021 - 04/18/2021	create final report/presentation	Final report and presentation

7b. Deliverables

- Final pdf
- Final ppt presentation
- Source code (github repository)

8. References

[1] Naqvi, S. N. Z., Yfantidou, S., & Zimányi, E. (2017). Time series databases and influxdb. Studienarbeit, Université Libre de Bruxelles, 12.

[2] Grzesik, P., & Mrozek, D. (2020, June). Comparative Analysis of Time Series Databases in the Context of Edge Computing for Low Power Sensor Networks. In International Conference on Computational Science (pp. 371-383). Springer, Cham.

[3]Bader, A., Kopp, O., & Falkenthal, M. (2017). Survey and comparison of open source time series databases. Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband.

[4] Ganz, J., Beyer, M., & Plotzky, C. (2017). Time-series based solution using InfluxDB. no. i.

[5] Sanaboyina, T. P. (2016). Performance evaluation of time series databases based on energy consumption.

[6] Kiefer, R. (2017). TimescaleDB vs. PostgreSQL for time-series: 20x higher inserts, 2000x faster deletes, 1.2 x-14,000 x faster queries. Timescale Blog.

[7] https://www.kaggle.com/szrlee/stock-time-series-20050101-to-20171231

[8] https://github.com/google/benchmark