Why do we need a time series database

2020141461173 高宝硕

As 5G technology matures, the Internet of Things will make everything connected. Before the Internet of Things era, only mobile phones and computers can be connected to the Internet, but in the future, all devices will be connected to the Internet. These devices will spit out a large amount of time-organized data every moment, which needs to be stored for query, statistics and analysis. In the current outbreak, people need to track daily COVID-19 statistics. Besides, autonomous driving and smart home are getting closer and closer to our life, and they may become an integral part of our daily life in the near future.

All of these apps requires a special kind of data:

* self-driving cars constantly collect data about changes in their environment
* smart home systems constantly monitor changes in the house, adjust temperatures, identify intruders
* The epidemic surveillance system collects daily confirmed cases and recovered cases

All of these apps rely on a form of data that measures how things change over time, called Time-Series Data, and it's coming to play an important role in our world.

What is Time-Series Data?

Time-series data is a series of data points indexed/listed in time order. The data were collected at different times and used to describe the changes of phenomena over time. All types of devices, such as the Internet of Things and the industrial Internet, generate vast amounts of temporal data.

This kind of data has the following characteristics：

* Time series data can be accumulated quickly，so it has a huge scale
* High concurrent write and data is rarely deleted or updated
* There are obvious hot and cold data, and generally only the recent data will be queried

What Can We Do with Time-Series Data?

1. Consistently generating huge amounts of data without peaks and troughs. To take a simple example, a sentry monitoring system that monitors 100 metrics per second on 1W servers would generate 100 WATTS of TPS per second. For example, if 100 million people are wearing a popular sports wristband and each wristband only collects 3 metrcis (heartbeat, pulse and step count) per second, it will also generate 300W TPS per second.

2. Data is inserted, and there is no update or deletion. Based on the fact that data generated by sequential services is rarely updated or deleted, there can be considerable simplification in the design of sequential database architectures.

3. More attention will be paid to recent data, and streaming processing will be paid more attention in the future. Time-old data are rarely accessed or even discarded. It's easy to understand, sentry systems we tend to care most about the last hour, most about the last three days, very little about the last three days. With the advent of streaming computing, sequential data will inevitably pay more attention to the value of real-time data in the future development, and this part of data is undoubtedly the most valuable. It is a very common and important scenario that the data can be generated and the alarm can be generated according to certain rules. The more timely the alarm is, the better for the business.

4. Labels of multiple dimensions exist in data, which often requires multi-dimensional joint query and statistical query. Another very important function of time series data is multi-dimensional aggregated statistical query. For example, the business needs to calculate the click-through rate and total revenue of advertisements published by Google in USA in the last hour, which is a typical multi-dimensional aggregated statistical query demand. This requirement is usually low on effectiveness, but high on query aggregation performance

Why is Time-Series Database needed?

Now we have a lot of mature database tools, they are very powerful in data processing. For example, ORACLE, MySQL, Microsoft SQL Server, PostgreSQL in relational databases and mongoDB,Neo4j, Redis in NoSQL databases. Can't we just use what we have to deal with time series data? Of course you can. We can use range partitioning. Range partitioning maps data to partitions based on ranges of partition. And there are people who do this:

They used Oracle to store data in 366 single-table partitions

• Each partition only included one table

• A single table stored the total time-series data for a day

• It could only store data for one year (delete data by partitions)

• Oracle posed certain difficulties in operations & maintenance (O&M).

• They partitioned tables by date and numbered the partitions from 1 to 366 for queries.

• If they failed to delete the historical data in time, for instance, 2020.7.21 and 2021.7.21 would be assigned into the same partition.

Thus we can see the disadvantages of using traditional database tools to deal with temporal data:

• As the volume of data increases, processing becomes increasingly complex

• Poor query/write performance

• It is very complicated to manage

• Not very convenient to use

• Potentially increasing storage costs significantly by having to keep a redundant copies of the data

Therefore, we need Time-Series Database to handle time-series data. It also turns out that Time-Series Database have been the fastest growing category over the past few years.

Characteristics of Time-Series Database

Time-Series Database is a kind of database specially used to store and manage time series data. It has the characteristics as following:

• more write and less read

• have clear hot and cold data

• high concurrent write

• no transaction requirements

• massive data continuous write

Time series databases process this large amount of data more efficiently and bring performance improvements, including higher throughput, faster large-scale queries, and better data compression.

TSDB also typically includes some common capabilities and operations for analyzing time series data: data retention policies, continuous queries, flexible time aggregation, and so on. Regardless of size right now, these features can still provide a better user experience and make your life easier. This is why developers are increasingly adopting time series databases and using them for a variety of usage scenarios.

Time series database application scenarios

As a powerful tool for processing timing data, the application of Time series database is very broad.

Common business requirements for time series database are as follows:

• Obtain the latest status and query the latest data (such as the latest sensor status).

• Display interval statistics, specify the time range, query statistics, such as average, maximum, minimum, count, etc.

• Obtain abnormal data and filter abnormal data based on specified conditions

Common Application Scenarios for time series database are as follows:

• Monitoring software systems: VMS, containers, services, and applications

• Physical monitoring system: hydrologic monitoring, equipment monitoring in manufacturing plants, national security-related data monitoring, communication monitoring, sensor data, blood glucose meter, blood pressure change, heart rate, etc

• Asset tracking applications: cars, trucks, physical containers, shipping pallets

• Financial trading systems: traditional securities, emerging cryptocurrencies

• Event applications: Track user and customer interaction data

• Business intelligence tools: Track key metrics and the overall health of the business

• In the Internet industry, there is also a lot of time series data, such as the behavior trajectory of users visiting websites, log data generated by applications and so on.

An example of a time series database, Prometheus.

Nowadays，the Most Popular Time-Series Database are Prometheus, InfluxDB, TDengine, TimeScaleDB and OpenTSDB. What I'm going to introduce to you is Prometheus.

Prometheus is an open-source systems monitoring and alerting toolkit originally built at SoundCloud. Since its inception in 2012, many companies and organizations have adopted Prometheus.

Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels.

Prometheus is written in Go.

Prometheus’s features

• a multi-dimensional data model with time series data identified by

metric name and key/value pairs

• PromQL, a flexible query language to leverage this dimensionality

• no reliance on distributed storage; single server nodes are

autonomous

• time series collection happens via a pull model over HTTP

• pushing time series is supported via an intermediary gateway

• targets are discovered via service discovery or static configuration

• multiple modes of graphing and dashboarding support

Prometheus’s components

• the main Prometheus server which scrapes and stores time

series data

• client libraries for instrumenting application code

• a push gateway for supporting short-lived jobs

• special-purpose exporters for services like HAProxy, StatsD,

Graphite, etc.

• an alertmanager to handle alerts

• various support tools

In addition，Prometheus use PromQL(Prometheus Querying Language), It is a DSL built upon Go . It uses the Prometheus data representation model in the form of

"Key&Value" and returns the results as vectors.

Summary

Time series database will be a very marketable and challenging database in the future. Although there are such and such services, most of them have such and such problems, and it is difficult to talk about maturity now. In order to occupy a certain position in the era of Internet of Things and industry 4.0, Time series database is a technology that must be expanded. I believe that time series database will become the mainstream of database in the future. It has advantages that traditional databases do not have.