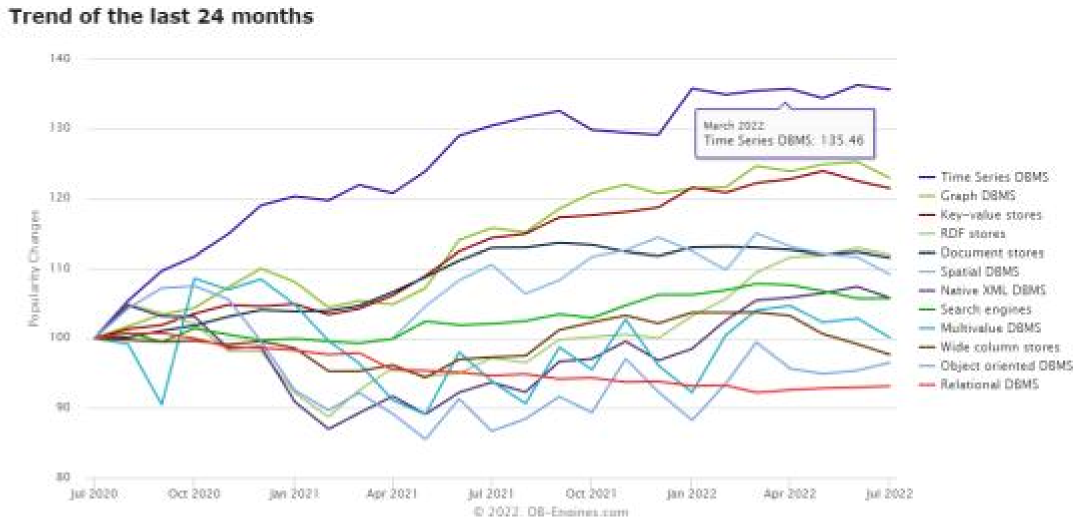
Why do we need a time-series database?

Looking at the statistics from *db-engines.com*, time series databases have become very popular in recent times, ranking first among various database models. So what is a time series database? Why do you need a time series database? This article will take you through the concepts, application scenarios and features of a time series database.



What is time series data?

By definition, it is a string of data that is indexed by a temporal dimension. To explain in descriptive terms what time series data is, it simply means that this type of data describes the measured values of a subject being measured at each point in time over a time horizon. There are three important parts to modeling time-series data: the subject, the point in time, and the measured value. Applying this model, you will find that you are exposed to this type of data all the time in your daily work life:

- If you are a stockholder, the stock price of a certain stock is a type of time-series data, which records the stock price at each point in time.
- If you are an operation and maintenance personnel, monitoring data is a kind of time-series data, for example, the CPU monitoring data of a machine, which records the actual CPU consumption value on the machine at each point of time.

The world is made up of data, and every object that exists in this world is generating data every moment. And the mining and utilization of these data is silently changing people's life style in this era. For example, the management of personal health through wearable devices is to continuously collect your personal health data, such as heartbeat, body temperature, etc., and then apply model calculations to assess your health after collecting the data. If your vision and imagination are large enough, you will find that the data you can tap into and use is abundant in the environment you live in. These objects that can generate data will include your phone, car, air conditioner, refrigerator, etc. The core idea of the current hot Internet of Things is actually to build a network that allows all objects to produce data and tap into its value. And the data collected through this network is typically the temporal data. Time-series data is used to describe the state change information of an object in the time dimension of history, and the analysis of time-series data is the process of trying to grasp and control the law of its change. With the development of IoT, big data and artificial intelligence technology, the temporal data also shows an explosive growth. In order to better support the storage and analysis of such data, a variety of emerging database products have emerged in the market. These database products are invented to solve the shortcomings and defects of traditional relational databases in storing and analyzing time-series data, and these products are uniformly categorized as time-series databases.

What are the application scenarios of time series database?

Time-series database is a vertical type database highly optimized for time-series data. There are a large number of application scenarios suitable for temporal database in manufacturing, banking and finance, DevOps, social media, health care, smart home, network and other industries:

- Manufacturing: For example, a lightweight production management cloud platform, using IoT and big data technology, collects and analyzes all kinds of timing data generated by the production process, and presents the production progress, target achievement status, and the utilization status of people, machines and materials at the production site in real time, making the production site completely transparent and improving production efficiency.
- Banking and Finance: The trading system of traditional securities and emerging crypto-digital currencies, collecting and analyzing the timing data generated in the trading process to realize financial quantitative trading.
- DevOps: O&M system for IT infrastructure and applications, collecting and analyzing equipment operation and application service operation monitoring indicators to grasp the health status of equipment and applications in real time.
- Health care: business intelligence tools, collecting health data in smart watches, smart bracelets, tracking key indicators and the overall health of the business
- Smart Home: Home IoT platform to collect data from home smart devices for remote monitoring
- Network: Network monitoring system to present network latency and bandwidth usage in real time.

In the above scenarios, especially in the IoT IoT and OPS operation and maintenance monitoring areas, there exists a huge amount of monitoring data that needs to be stored and managed. Take Huawei Cloud Eye Service (CES) service as an example, a single Region needs to monitor more than 70 million monitoring indicators, and 900,000 reported monitoring indicators need to be processed per second, assuming 50 bytes per indicator, there is 1PB of monitoring data in a year; 80G of various sensor monitoring data in a day for self-driving vehicles. The traditional relational database is difficult to support such a large amount of data and such a large writing pressure, Hadoop big data solution and the existing temporal database will also face a very big challenge. For large-scale IoT and public cloud scale operation and maintenance monitoring scenarios, the demands on the temporal database mainly include:

- Continuous high-performance writing: monitoring indicators are often collected at a fixed frequency, and some industrial IoT scenarios sensor collection frequency is very high, some have reached 100ns, and public cloud operations and maintenance monitoring scenarios are basically second-level collection. Time-series database needs to support 7*24 hours of uninterrupted continuous high pressure writing.
- High-performance query: The value of the time series database lies in data analysis, and there are high real-time requirements, typical analysis tasks such as anomaly detection and predictive maintenance, such time series analysis tasks require frequent access to a large amount of time series data from the database, in order to ensure the real-time analysis, the time series database needs to be able to quickly respond to massive data query requests.
- Low storage cost: The data volume of IoT IoT and O&M monitoring scenario has grown exponentially, and the data volume is more than a thousand times of the typical OLTP database scenario, and is very

sensitive to cost, so it needs to provide low-cost storage solutions.

- Support massive timeline: In large-scale IoT IoT and public cloud scale operation and maintenance scenarios, the indicators that need to be monitored are usually in the tens of millions or even billions, and the timing database should be able to support the management capability of billions of timelines.
- Elasticity: monitoring scenarios also exist for sudden business growth, for example: Huawei Welink service's O&M monitoring data surges 100 times during the epidemic, the time series database needs to provide enough sensitive elastic scaling capability to quickly expand capacity to cope with sudden business growth.

Features of time-series databases

Features of data writing:

- Smooth, continuous, high concurrent and high throughput writes: The writes of temporal data are relatively smooth, which is different from application data, which is usually proportional to the amount of application access, which usually has peaks and valleys. Time-series data is usually generated at a fixed time frequency and is not constrained by other factors, and the rate of its data generation is relatively smooth. Time-series data is generated by each individual independently, so when the number of individuals is large, the write concurrency and throughput are usually high, especially in the IoT scenario. Write concurrency and throughput can be simply calculated by the number of individuals and the frequency of data generation, for example, if you have 1000 individuals generating data at a frequency of 10 seconds, your average concurrency and write volume per second is 100
- Write More Read Less: 95%-99% of operations on timing data are write operations, which is typical of write more read less data. This is related to its data characteristics, for example, monitoring data, you may have a lot of monitoring items, but you may actually read less, usually only care about a few specific key indicators or in a specific scenario to read the data.
- Real-time writing of recently generated data, no updates: The writing of temporal data is real-time, and each writing is the most recently generated data, which is related to the characteristics of its data generation, because its data generation advances with time, and the newly generated data will be written in real-time. Data writing without update, in the dimension of time, as time advances, each time the data is new, there will be no update of the old data, although it does not exclude artificially doing revisions to the data.

Features of data query and analysis

- Read by time range: Normally, you don't care about data at a specific point, but rather over a period of time. So temporal data is read, basically, by time range.
- High probability of recent data being read: The more recent the data is, the more likely it is to be read. Taking monitoring data as an example, you will usually only care about monitoring data from the last few hours or the last few days, and very rarely from a month or a year ago.

Features of data storage:

- Large amount of data: Take monitoring data as an example, if the time interval of our collected monitoring data is 1s, then one monitoring item will generate 86400 data points per day, and if there are 10000 monitoring items, it will generate 8640000 data points per day. In the IoT scenario, this number will be even larger. The scale of the whole data is terabytes or even petabytes.

- Time-sensitive: Time-series data is time-sensitive, and data usually has a retention period beyond which it can be considered invalid and can be recycled. On the one hand, it is because the more historical the data is, the lower the value available; on the other hand, the low-value data can be cleaned up in order to save storage cost.
- Multi-precision data storage: In the characteristics of the query mentioned in the temporal data for storage costs and query efficiency considerations, will need a multi-precision query, the same need for a multi-precision data storage.