Why do we need a time-serise database?

ID:2019141460470 Name:高子斌 Major: Software Engineering

**Why do we need time-series database?**

Before we discuss this problem, we need to know what is called time-series database. The full name of time-series database is time sequential database. Time series database is mainly used to process data with time tags (changing in time order also is called time serialization). Data with time tags is also known as time series data. Now that we know what time series database is, where does time series database need to be used in industry or production? Time series data are mainly collected and generated by various types of real-time monitoring, inspection and analysis equipment, such as power industry, chemical industry, meteorological industry, geographic information even more. The typical characteristics of these industrial data are: the generation frequency is fast (each monitoring point can produce multiple pieces of data in one second), it is heavily dependent on the collection time (each piece of data requires a unique time). There are many monitoring points and a large amount of information (the conventional real-time monitoring system has thousands of monitoring points, which generate data every second and dozens of GB of data every day).

Next, I will explain the characteristics and advantages of time series database from several aspects, and why we need to use time series database instead of the usual database.

**First:Data status changes over time**

Some simple example is easy to understand. If I give you 10 yuan, the traditional database of the bank will have an expense in my account and a receipt in your account. Then, if you give me 10 yuan, the same process will be reversed. Finally, our bank balance looks the same. For banks, there is no change this month. However, with the time series database, the bank will perceive that there may be a deeper reason why these two people have been transferring 10 yuan to each other. If you track this subtle difference, the month end account balance will have greater significance.

Another example is the average temperature of a place for several consecutive days. In the past decades, average temperature has been used as the main reference factor for building energy conservation. In any week, the daily average temperature in the same place may be only slightly different, but at the same time, the factors affecting the environment may have changed dramatically. On the contrary, understanding the temperature changes every hour of the day, together with the precipitation, cloud cover and wind speed during this period, can greatly improve the ability of property modeling and energy efficiency optimization.

From the above two examples, we can see a conclusion that the temporal data set tracks the changes of the whole system as INSERT operation rather than UPDATE operation. Our definition revolves around "change". We can determine the timing data that we haven't collected today and the timing of data collection from now on, so that we can make use of its value in the future. Under such operation, we can better see the change of data rather than just the beginning and end of data, which is more meaningful.

**Second: Are all data time series data?**

Although this era began with moderate computing technology, due to the main macro trends: Moore's law, Clyde's law, cloud computing, "big data" technology industry, our ability to collect, store and analyze data has increased exponentially. According to Moore's law, computing power (transistor density) doubles every 18 months, while Clyde's law assumes that storage capacity doubles every 12 months.

We are no longer satisfied with observing the state of the world, but need to measure how the world changes with time, accurate to the order of sub seconds. "Big data" is now dwarfed by another type of data, which relies heavily on time to preserve changing information.

Are all data derived from time series data? Recall the previous examples of web applications. We sometimes order data, but we don't realize that tracking user activity can help analyze user stickiness. Or think of any "normal" data set, such as the current account and balance of a large retail bank, or the source code of a software project, or the text of this article.

Usually, we only need to store the latest state of the system, but what if we store each change and calculate the latest state when querying? Is a "normal" data not just a view on the inherent timing data (cache performance reasons)? Does the bank have a transaction bill? (isn't blockchain just a distributed and immutable timing log?) Is there no version control (such as git submission) for software projects? Didn't this article review history (undo, redo)?

In other words: don't all databases have logs?

We recognize that many applications may never need timing data (preferably using the "current state view"). But as we continue to follow the exponential curve of technological progress, these "current state views" seem to become less necessary. On the contrary, we find that more and more data stored in the form of time series often help us better understand it.

So in general, all the data so far have time series data.

**Third: Time series data everywhere?**

Can you think of some common examples of time series data in daily work? Is there someone who asks you to help with the report or analysis, but lacks data fidelity?

Suppose you maintain a web site. Each time a user logs in, the "last\_login" timestamp of the user is updated in the "users" table. But what if you treat each login as a separate event and collect them over time? With this kind of time series data, we can analyze the historical login activities, see how the usage changes over time, divide users according to the frequency of users accessing the application, and so on.

Another example is crucial to every IT group around the world, including the operational indicators of servers, networks, applications, environments, etc. This kind of timing measurement data is very important to ensure the reliability of services. By tracking the changes in each indicator, it departments can quickly identify problems, plan for upcoming events, and diagnose whether application updates have led to changes in user behavior, for better or worse.

These examples illustrate a key point. Preserving the inherent timing characteristics of data allows us to retain valuable information, such as how the data changes over time. We note that both examples describe a common temporal data type - event data.

Of course, there is an obvious problem with storing data in this way: you will eventually get a large amount of data, and the growth rate is very fast. So this is the problem: being able to analyze the increased time series data is more valuable than ever before, but it accumulates very quickly.

Massive data will cause a series of problems, whether it is storage or fast query, which is why we are more inclined to use temporal database than ever before. The world requires us to make data-driven decisions faster and better. Traditional static data cannot solve this problem. To meet your needs, you need to collect data with the highest possible fidelity - this is what timing data provides: everything that happens in the system can be stored like a movie, whether it's software, physical power plants, games, or customers in applications.

**Fourth: Why do we need time series database?**

The fact is that you can, and some people have already done so. However, TSDB has become the fastest-growing database category today for at least two reasons: scale and availability.

Scale: time series data accumulates very quickly, while ordinary databases are designed to handle this scale (at least not in an automated way). Relational databases perform poorly on very large datasets, while NoSQL databases perform better on Scale (although relational databases that fine tune time series data can actually perform better, as we showed in the benchmark compared with influxdb, Cassandra, and mongodb). In contrast, the benefits of the introduction of temporal databases (whether relational databases or NoSQL based databases) can only be realized when you regard time as the first consideration. These benefits enable them to provide large-scale performance improvements, including higher throughput and faster large-scale queries, as well as better data compression.

Availability: TSDB usually also includes built-in functions and operations commonly used in time series data analysis, such as data retention strategy, continuous query, flexible time aggregation, etc. Even if you just start collecting this type of data, you don't need to consider the scale yet. These functions can still provide a better user experience and make data analysis tasks easier. Using built-in functions and features to analyze trends that are readily available in the data layer, you will often find unexpected value, whether your data set is large or small.

**Fifth: Mining value with time series analysis**

So far, you have begun to identify the applications or fields of time sequence data in your business, which are waiting for you to do something. So what should we do now? It is time to understand why time series databases are essential tools.

Let's take a look at an example of a web application based on our previous reference. As we discussed, we only record the last time the user logged in as a field in the "users" table, and always update the previously stored value with the new login information. Although this allows us to query how many people log in a week or a month, we cannot analyze the frequency and duration of their login, nor can we analyze other information that may tell us more about the user experience or usage.

We can quickly improve this by tracking each login, not just the latest one. To do this, we will start recording the time of each login and the type of device used to access our application (such as mobile phone, pad, PC). This small change (another attribute of tracking user login experience) provides immediate value, enabling us to answer questions such as "which device is the most commonly used (individual users and all users)?", And "when are users most active in the day?" From this, we can better know our priority functions, such as mobile device specific functions, the time to display some promotional information, and so on.

**Conclusion**

Time series database can help us analyze and find meaningful data and support rapid decision-making. It has made great contributions to our life, production, national security and national construction, so time series database is of great significance in reality.