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

With the continuous development of science and technology, our world is changing at a warp speed. We capture and analyze more data than ever before, and the speed is faster than ever before too. We have officially entered the era of big data. In the era of big data, various types of devices which are in various fields will produce massive time-series data. This new type of data is very different from the data we traditionally process. Time-series data is a series of data points indexed/listed in time order. It has the following characteristics: time stamped data，structured data，data source is like a stream，data rate is pretty stable，immutable，more write than read operations，data is rarely deleted or updated，there is always retention policy, Real-time data computing is desired, Query is always in time and space range.

But it is worth noting that time-series data does not simply take time as a measure. The more critical thing is to help us analyze data and obtain valuable information. The data analysis of time-series data is much deeper than the traditional one. For example, 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 changes in each indicator, we can quickly identify problems, plan for upcoming events, and diagnose whether application updates have led to changes in user behavior, for better or worse. We can see from the above example that preserving the inherent timing characteristics of data allows us to retain valuable information, such as how the data changes over time. The era of big data requires us to make data-driven decisions faster and better. Traditional static data cannot solve this problem. In order to meet the demand, we need to collect data with the highest fidelity possible - this is what time-series data provides: everything that happens in the system can be stored like a movie. In short, time-series data sets track changes to the entire system and constantly insert new data, rather than updating the original data. The reason why time-series data is so powerful is that every change in the system is recorded as a new line, so that changes can be measured: analyzing changes in the past, monitoring changes in the present, and predicting how they will change in the future.

Even though the time-series data described above has many advantages, everything has two sides. It also has certain defects, but the defects described here are more from the perspective of how to use it well. The amount of time-series data is too large, and the growth rate is very fast. Massive data will cause a series of problems. For us, the massive amount of data is not conducive to use, process and analyze, because there are Great difficulties in both storage and fast query. Therefore, how to better process and analyze the time-series data, making the time-series data be better used by our production and life, is now a very hot issue.

The development of human science and technology is carried out step by step, just like walking on steps. Therefore, if we want to find a solution to this problem, we need to focus on the existing solution. Our common tool for dealing with data is database. Time-series data is also data, so we can also make better use of it through the database. Of course, first of all, we need to analyze whether our existing tools can meet our needs. After all, human beings are all lazy and don't want to find something to do for themselves.

Considering general databases, relational databases perform poorly on very large data sets. As mentioned above, the amount of data in time-series data sets is very large, so those who use relational databases cannot meet our needs. It is worth noting that although NoSQL database performs better on large-scale datasets, its performance still cannot meet our expectations. When you take time as the first consideration, NoSQL database cannot provide ideal throughput, fast enough large-scale queries and sufficient data compression capacity. At the same time, due to the characteristics of time-series data I mentioned above, we need special time analysis methods such as down sampling, interpolation and time weight average when processing time-series data. Neither relational database nor NoSQL database provides these methods, so we have no way to deal with temporal data conveniently. And Some time series are irregular, corresponds to discrete events. These could be requests to an API, trades in a stock market, or really any kind of event that you'd want to track in time. Most general databases are inefficient in processing such time series. In the past, many people have tried to store time series data in common relational databases. Generally they find it works for a time, but things start to fall apart as the scale of the data increases. This is what happened in fact, not just fantasy. This is a conclusion reached after many times of practice. The general database can't deal with time series data well. After all, general databases were not designed to solve the specific time series issues, and trying to get them to solve them is impractical.

Therefore, we need a special database——time-series database to meet our needs. At the same time, it is worth noting that the purpose of building the database is to process and absorb the inherent and uninterrupted time-series data flow, reduce any negative performance impact or delay, query data in a meaningful way, and make all aspects easier and faster. Compared with general databases, temporal databases can provide large-scale performance improvements, including higher throughput, faster large-scale queries, and better data compression. Time-series database 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.

The above is a theoretical analysis of the reasons why we need time-series database. Next, I will analyze this problem with some specific examples.

First of all, we need to pay attention to the fields in which time-series data can be used. Monitoring software system: virtual machine, container, service, application; Monitoring physical systems: equipment, machinery, connecting equipment, environment, family, human body; Asset tracking applications: vehicles, trucks, physical containers, pallets; Financial trading system: typical securities, new cryptocurrencies; Event application: track user or customer interaction data; Business intelligence tools: track key indicators and the overall operation of the business. We can find that these examples are closely related to our production and life. The importance of time-series database is obvious.

Combined with the examples above, suppose we have 100 servers and each has 200 unique measurements to collect. That means we have 20,000 unique series. Further, let's say that we're collecting this data every 10 seconds. That means in the course of a day we're collecting 86,400 / 10 = 8,640 values per series for a total of 20,000 \* 8,640 = 172,800,000 values for each day. Such a huge amount of data has exceeded the processing capacity of the general database.

In addition, as many developers of time-series databases say, ”one of our goals we envisioned was optimizing for users’ or developers’ time to value.” Time-series database can give them a better experience by solving problems fast. After all, the faster users and developers get their problems solved, the better the experience will be. Time-series database can get better performance than more generalized databases while also reducing the developer effort to get a solution up by at least an order of magnitude. With the advantage of this, application developers focus on the code that creates unique value inside their app.

In the era of big data in the past decade, organizations of all sizes and types have collected a large amount of information about our world and understood it through calculation and analysis. According to Moore's law, computing power (transistor density) doubles every 18 months, while Kryder's law assumes that storage capacity doubles every 12 months, our ability to collect, store and analyze data has grown exponentially. 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. If we observe life carefully, we will find that many aspects of our work and study are related to time-series data. If we want to make good use of time-series data, the importance of time-series database is an understood thing. In my opinion, time-series database represents a part of the future. In the coming decades, human development cannot be separated from time-series data and time-series database

All in all, the objective superiority of time-series database is reflected in two aspects, namely, scale and availability. At the same time, for users and developers, it also has some hidden advantages. Because it is convenient and fast to deal with problems, time-series database can bring people a good mood.

And these advantages also answer the question well.” Why do we need a time-series database?” Because it can solve our problems and meet our production and living needs. Although we may achieve the same goal through other tools and means, the time-series database is more convenient, faster and easier to use.