Why do we need a time-series database?

**1. What is a sequential database?**

Time series database is called time series database, which is one of many types of database. Now IT has been used in various industries, such as electric power industry, chemical industry, meteorological industry, energy industry, Internet of vehicles, finance, IT operation and maintenance, etc. Timing database is mainly used to process time-tagged data, usually with a large number of measurement points and a large amount of stored data.

**2. Sequential database characteristics**

All the collected data are unique data sources of each collection point in time series. Usually, data will not be deleted or modified. Data storage is generally set with retention period. Query operations are performed to query and analyze data in a specified period of time. Statistics and aggregation of stored data are required for real-time calculation. A huge amount of data is stored.

**2.1 High throughput write capability**

This is tailored to the feature that sequential services continuously generate massive data. Currently, to achieve high throughput write of the system, two basic technical requirements must be met: horizontal scalability of the system and single-node LSM architecture. It is easy to understand the horizontal scalability of the system, which cannot be supported by a single machine. The system must be clustered, and it is easy to add nodes to expand. In the final analysis, it is not aware of the business when expanding. The LSM architecture is used to ensure high throughput write on a single machine. In the LSM architecture, data writing only needs to be written to the memory and written to the log, so there is no need to write data randomly to the disk. Currently, HBase, Kudu, Druid and other systems that require high write performance adopt this architecture.

**2.2 Storage tiering /TTL**

This is a technical feature customized for the hot and cold nature of temporal data. Data tiering requires that the latest hour data can be stored in memory, the latest day data can be stored on SSDS, and the older data can be stored on cheaper HDDS or TTL expiration.

**2.3 High compression rate**

There are two reasons to provide high compression rate. One is cost saving, which is easy to understand. If you compress 1 TB data to 100 GB, you can reduce the disk cost of 900 GB, which is a great temptation for businesses. Another aspect is the compressed data can be easier to ensure that stored in the memory, such as the recent data is 1 t 3 hours, I now only 100 gb of memory, if without compression, there will be 900 gigabytes of data are forced into the hard disk, it will query overhead is very large, and will use compression this 1 t data into memory, the query performance will be very good.

**2.4 Multi-dimensional query capability**

Temporal data usually have labels of multiple dimensions to describe a single data, which is the dimension column mentioned above. How to efficiently query according to random dimensions is a problem that must be solved, which usually needs to consider bitmap index or inverted index technology.

**2.5 High efficiency polymerization capacity**

A common requirement for sequential services is aggregated statistical report queries, such as the sentinels need to see the total number of exceptions on an interface in the last day, or the maximum elapsed time for an interface to execute. Such aggregation is actually simple count and Max. The problem is how to efficiently query and aggregate the original data that meets the conditions on the basis of such a large amount of data. You should know that the original value of statistics may not be in memory because of a long time ago, so this may be a very time-consuming operation. Currently, the most mature solution in the industry is to use pre-aggregation, which is to complete basic aggregation operations as data is written in.

**3. Characteristics of sequential business**

**3.1 Continue to produce huge amounts of data, with no peaks and troughs**

For example, if the sentinel-like monitoring system were to monitor the various metrics of 1W servers and take 100 metrics per second for each server, there would be 100W 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.

**3.2 Data are all insert operation, basically no update deletion operation**

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

**3.3 In the future, more attention will be paid to streaming processing. Long-lived 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.

**3.4 Data has labels of multiple dimensions, 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 usually has a low requirement for effectiveness, but a high requirement for query aggregation performance.

**4. Differences between a sequential database and a regular database**

The first difference is the amount of storage used to store large amounts of data. Sequential database generally has a high data compression rate. Even in the case of massive data storage, data can be compressed and stored with low storage cost. This is due to the storage mode of sequential database, which generally adopts column storage.

In addition to lower storage costs, the second significant difference is better write performance compared to a normal database, even in the case of a large number of devices, measurement points, can still have excellent performance.

The third difference is that sequential databases collect data more frequently and store a large amount of data. Users can set the retention period of data based on their service requirements, for example, 10 years or 50 years. Relational databases generally do not have specific data retention times.

Fourth, relational libraries use different index data structures than sequential databases. The advantages of BTree structure and B+Tree in the relational library are obvious, providing excellent retrieval performance for a large range of queries. However, in sequential database, write is significantly higher than query, and most of them use LSM Tree (Log Structured Merge Tree), which can withstand higher write throughput than Btree. LSM Tree achieves higher write performance through in-memory writes and sequential writes to subsequent disks, avoiding random writes, which is more suitable for sequential databases.

The fifth difference is the use of sequential database. There are some differences between sequential database and common database. In relational database, SQL statements are generally used for query and write operations, but not all sequential database uses SQL statements. Each sequential database also develops its own unique uses that do not fully reference the SQL language of the relational library.

**5. Application scenario of sequential database**

**5.1 Real-time monitoring of system operation and peacekeeping services**

In business deployment of various script on the server, client index data real-time acquisition server (IO, CPU, memory bandwidth, etc.), business related data (abnormal method call number, delayed response, the JVM GC data, etc.), database related data (read, write, delay delay, etc.), obviously, These data are all time series dependent. After collection and real-time calculation by the client, it will be sent to the sentinel server, which will store the data and display the monitoring and analysis for users to query.

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.

**5.2 Iot device status monitoring storage analysis**

In the foreseeable future 3-5 years, with the arrival of the Internet of Things and industry 4.0, all devices will carry sensors and network, and the timing data collected by sensors will rely heavily on TSDB's real-time analysis ability, storage ability and query statistics ability.