why do we need a time-series database

2019141090379 毛泓霖

**1、What is a time series database?**

With a time series database (TSDB), it's possible to add, process, and track massive quantities of [time series data](https://aiven.io/blog/time-series-or-event-data-get-less-confused). They do this efficiently and continuously, with lightning speed and precision. Other types of databases also work for these workloads, and have indeed been used in the past, TSDBs haven specific algorithms and architecture to meet the requirements of speed and high volumes.

A time series database stores data as pairs of time(s) and value(s). Storing data this way makes it easy to analyze sequences of points recorded in order over time. A TSDB can handle concurrent series, measuring many different variables or metrics in parallel.

Early time series databases were mostly used for processing volatile financial data and streamlining securities trading. The world’s changed a lot since they were first introduced, and many new use cases have emerged as technology has continued to evolve.

For example, the Internet of Things (IoT) concept employs sensors that constantly collect and stream data. IoT technology is used for multiple purposes, from powering industrial applications to predicting sales demand, from weather monitoring to wearable fitness devices. The amount of data they produce is staggering.

Well, it can be staggering for more traditional databases, which were designed for different purposes. Luckily, there are time series databases, and they are getting better and better at dealing with the growing demand of this data type.

**2、What are the benefits of a time series database?**

There’s a reason more and more developers and organizations are using time series databases. They deliver a number of benefits, which we’ll now briefly explore.

**1. More accurate and meaningful time series measurement**

A time series database makes it easy to measure how datasets change over time. You can concurrently view past, present, and future datasets for reporting that is more accurate and meaningful.

**2. Resource-efficient data storage**

By the very nature of the data type, processing it can require massive amounts of storage, which can be difficult to manage. It's also very expensive. Time series databases have tooling to aggregate data into predetermined time periods and to eliminate any data streams as needed. There are also compression algorithms that optimize data storage.

**3. Lightning-fast data queries**

A TSDB can also make it easy to query and retrieve data based on specific periods. Imagine that you can't remember the title of a book you recently read, but you know it was three months ago. Time series databases can help you figure out what the book was without having to use a bunch of wildcard searches.

Using a time series database, you can quickly find information based on timeframe.

**3、When to Use a Time Series Database**

Lots of companies and individuals store their time series data in other types of databases (relational, noSQL) successfully. If you’re one of those, you’re happy, and you have no current issues, far be it from me to demand you change. You do you.

However, there are definite benefits to using a database designed for your time series data.

Scalability

Scalability is one of those magical words that we hear often and is used correctly *sometimes.*The general problem with time series and scale outside of a [Time Series Database](https://www.influxdata.com/time-series-database/) is this: if Skynosaur flies for 1,500 hours (the minimum number of hours for a commercial pilot’s license), we’ve already reached over a million data points for one device. The makers of Skynosaur (Skynosaurus Rex, Inc.) could have thousands of devices sending data home. Querying by timestamp would involve millions of rows of data in a relational database.

People often claim that SQL databases don’t scale well while NoSQL databases do, but it was easier for me to understand in terms of [ACID](https://en.wikipedia.org/wiki/ACID) versus BASE. To unfairly summarize, ACID-compliant databases are concerned with guaranteeing validity — data should be atomic, consistent, isolated and durable. The BASE model allows us to give up some of the ACID principles for the sake of speed, or scale, or whatever we want to prioritize. To decide which system works, we need to establish the main purpose of our database.

If we don’t care about durable data, we can write commands without flushing to disk (meaning the data probably won’t survive a reboot). If we don’t care about atomicity, we can shorten the duration that data sets are locked. Time series databases balance the ACID/BASE relationship by offering principles that suit time series data.

For example, time series data is more valuable as a whole than as individual points, so the database knows it can sacrifice durability for the sake of a higher number of writes. Skynosaur sends data home every five seconds, so if we lost some data points in 1,500 hours of flight time, our overall trends would still be intact.

Scalability, in this case, means that a time series database specializes in a higher number of writes with eventual consistency, even across distributed storage, and that specialty means less worry for the people that care about that data.

Usability

If all of our data lived in a secure, durable black box, we could breathe easy. But how we access the data can be just as important as its storage. Every database has its query language, designed to access the contents as efficiently as possible. Keep that in mind because as we mentioned earlier, time series data is special. It’s a double rainbow with a timestamp.

Think of the army of Skynosaurs sending data to Skynosaurus Rex headquarters again. There are millions of data points to search, but now we have a query language that is built for the task at hand—not to view data as it relates to other pieces of the schema but to view data in the context of time in order to aggregate, set windows or see trends. This isn’t about whether other databases are capable of doing such a thing, it’s about how we choose to spend our resources.

Trade-offs

Database architecture is about trade-offs and priorities. Do you need speed or accuracy or volume or predefined schemas? The proof is in the benchmarks. Measure everything. Don’t choose a tool or a product—choose a solution to your problem. Specialty tools are made for special problems, so time series databases are optimized for time series problems.

**4、Common time series database use cases**

1. Store and access IoT data

Most IoT deployments — like connected water, energy, and temperature meters — involve constant data collection and reporting at regular intervals. Time series analysis can provide timestamped data points for identifying patterns, average usage, and inefficiencies.

For example, a connected pH meter connected to a TSDB might tell a technician tasked with maintaining a specific pH level that a certain vat of water is becoming too acidic. IoT endpoints also collect massive amounts of data, requiring highly scalable time series databases.

2. Monitor web services, applications and infrastructure

TSDBs can measure the performance of a company's applications and services. For example, the open source monitoring system Prometheus is a time series database that enables developers to keep tabs on performance trends over time. This enables them to easily detect when problems are occuring, which then allows them to plan maintenance and rapidly respond to incidents to sustain an optimal user experience.

Some web and mobile applications store the events within their app in a TSDB (such as a button click, playing a video or sharing some content). With this data they can map a user’s journey, identify frustrations or performance bottlenecks and enhance the user experience.

3. Understand financial trends

Using time series data to accurately predict financial trends is very difficult. However, a TSDB can provide a wealth of contextual data to help analysts. Let’s take the stock market as an example; a sudden increase in airline stock may coincide with holiday travel. Or an executive leadership purge may spook investors, causing a stock to temporarily tumble. Time series databases make it easy to cross-reference data, providing a richer, clearer picture.

4. Process self-driving vehicle data

Self-driving cars typically collect about 4,000 GB of data per day. This is beyond the scope of what a typical relational database can process. Time series databases ingest data and queries faster, and compress data more strongly.

As a result, they are ideal for processing massive volumes of real-time data that can be used to improve the safety of self-driving vehicles.

5. Sales forecasting

Retail stores must continuously predict future sales based on past ones, so that they can restock appropriately. Thanks to time series databases, retailers can use statistical models combined with historical data. Cross-referenced with consumer behavior trends, they can predict future patterns and make informed decisions about which products to keep in stock and when.

For instance, retailers are now using forecasting to plan ahead and restock bicycles, which are now experiencing a shortage due to the pandemic. Retailers are using data to predict when new products will become available again, what the demand will be like, and what alternative transportation options consumers are buying in lieu of bicycles e.g., trikes, rollerblades, etc.

**5、Importance of Time Series**

Given below are some of the importance mentioned:

* It is used by many organizations to forecast their business profit or loss trends and thus important business decisions can be taken for development.
* It is used to compare the present trend with the past trend that has already happened so the future trend can be estimated and prepared.
* The cycle variations over a period using time series will allow us to understand the business cycle quite effectively.
* It is used to understand the correlated seasonal trends of the data.
* It is also used to understand how an event can change its feature over a period of time and hence the reliability, flexibility, and other important features can be predicated.
* It is also used in the quality control process where the quality trend is predicated over time.
* If we receive the complex signal pattern for it then we can apply some transformations such as Fourier analysis to denoise the graph and break the complex pattern into a series of simpler patterns and hence a better understanding can be achieved.