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

陈昶旭 2020141461054

In order to explain the question that why do we need a time-series database, it’s learning about the time-series database that plays an important role in understanding it.

First and foremost, let’s take notice of the time-series data, which is a sequence of data points collected over time intervals, giving us the ability to track changes over time. Time-series data has characteristics that are really unique. It has incredibly high volume, sometimes in the millions or billions of points. It has a natural time order which means that time is always one of the axes that you would measure for this data. In the meanwhile, time-series data can track changes over milliseconds, days, or even years. For instance, the temperature of a classroom during this week or the price of a stock during this year might be the time-series data. And when it turns to the time-series database, it is a software system that is designed to store and retrieve data records that are part of a “time series”, which is a set of data points that are associated with a conception named timestamps. The timestamps provide a critical context for each of the data points in how they are related to others. Time series data is often a continuous flow of data like measurements from sensors and intraday stock prices. A time-series database lets you store large volumes of timestamped data in a format that allows fast insertion and fast retrieval to support complex analysis on that data. As a result, this practice of recording each and every change to the system as a new, different row is what makes time-series data so powerful. It allows us to measure and analyze change: what has changed in the past, what is changing in the present, and what can we forecast changes may look like in the future. What counts most is that the system fucus on the changes and the trend instead of the concrete data. And the unique properties of time series datasets mean that time series databases can provide significant improvements in storage space and performance over general-purpose databases.

Now we have learned something basic about the time-series data and the time-series database. So, what’s the reason why we should take a time-series database? Well, the whole world is changing at an incredible speed with the rapid development of the modern society. As a consequence, it’s nature to draw the conclusion that we need something new to meet the new mode of the development. And then, the time-series database is just the solution for that. When the COVID-19 strikes the world, the time-series data is becoming more valuable to study and more necessary to study. So, the truth is that you can, and some people do use normal database. But there are at least two reasons why time-series databases are the fastest-growing category of databases today: scale and usability. From where I stand, the scale counts more. Here I will explain the two reasons as follows.

On the one hand, time-series data accumulates very quickly, and normal databases are not designed to handle that scale (at least not in an automated way). Traditionally, relational databases fare poorly with very large datasets, while NoSQL databases are better at scale. Here I’d like to emphasize that although a relational database fine-tuned for time-series data can actually perform better, as we’ve shown in benchmarks versus Cassandra and versus MongoDB. In contrast, time-series databases - whether they’re relational or NoSQL-based - introduce efficiencies that are only possible when you treat time as a first-class citizen. These efficiencies allow them to offer massive scale, from performance improvements, including higher ingest rates and faster queries at scale (although some support more queries than others) to better data compression. The huge amount of data means the 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. One of the most important things to me about time-series data is that it's more valuable as a whole data set than as individual points. Because we learn from the data set. One point especially when we're talking about millions or billions right of points. It doesn't tell us that much about the whole dataset. More importantly. If we were gathering this data and we lost the point which happens sometimes. Our data set would still be intact, all of the patterns that we care about would be unaffected by losing one point. Let’s turn back to the scale itself. Scale can mean a lot of things you probably hear people saying things like but does it scale. And scale how right like a lot of people say scale but not that many people have a specific meaning in mind, does it scale horizontally vertically scaling complexity, there are all these different dimensions. We really need a database that specializes in higher rates with eventual consistency even across distributed systems right and that's something that time series does really well. Storing timeseries data as a flat relational table is massively inefficient. Every timepoint row will have to duplicate the object key plus a timestamp. Informix's timeseries specific data type, which is very similar to what dedicated timeseries databases use, saves about 70% of storage over storing the same data in a relational table structure for regular timeseries and about 50% for irregular timeseries. In addition, since at the relational level there is only one row per object key rather than thousands of indexes consume 90% less storage and there are no indexes to track the timepoints, so more storage savings. Processing timeseries queries the analysis across time for each object key is hugely efficient since all required timepoint data are fetched into memory in a single operation with a single index lookup.

On the other hand, the time-series database also typically includes built-in functions and operations common to time-series data analysis, such as data retention policies, continuous queries, flexible time aggregations and so on. Having built-in functions and features to analyze trends readily available at the data-layer often leads you to discover opportunities you didn’t know existed, no matter how big or small your dataset is. The work that time-series database has done is to let people spent less time on optimizing or adapting to the tool. Also, time-series is a really fast-growing category especially. The land of databases right time series data isn't new but. As we sort of see growth in IoT and develops we start to see people wanting to handle the data more efficiently. There's an explosion of connected things, everything in the IoT world is essentially time series data if it has a sensor, all sensor data is time series data because they're very much focused on the time stamp. This kind of loose structure in the database. Allows for more innovation because again. No one knows their data better. Then the people making it right Tesla knows what they need out of their data better than we do, but we can give them the tools to get after. If we use the electric to give an example, they can forecast you know roughly how much they use a night and then they can use that data to say well if they use it for the next three nights as they have for the past three nights for the past three months. Then we know that they'll be out of battery or we know that it will cost this much so. They can make smarter decisions because they have the data there to analyze. That’s what we called the usability of the time-series database.

For all the reasons we talked about time-series data and the time-series database. Except for the reason as mentioned above, time-series database has a resource-efficient data storage and lightning-fast data queries. 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. For another reason, a time-series 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.

Significantly different than other types of data and we think that in order to be. Using a time series database, you can quickly find information based on timeframe. Really accurate in the way that we query and also easier to read and understand that it kind of called for a new language to query with. And as the team built this new language. Its use ability and potential become something more obvious. To put it in a nutshell, the time-series database is a new way to solve the new problems with the rapid development of the society, which is linked to the future.