Why do we need a time-series database?

To understand this question, we need to know what time-series database is first.

Time-series data is a series of data points indexed/listed in time order. There are many characteristics of time-series data: 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.

Understanding those characteristics about time-series data, we are able to understand that it could afford the following functions. First of all, it could be used in real time alert, including fire alarm system. It could also be used in forecasting and insights of operation. What’s more, identifying problems could also be completed.

Why is Time-Series Database needed?

Today’s data volume is huge (in most cases, it is generated automatically). Data ingestion rate becomes a challenge. Query latency becomes a challenge. Storage becomes a challenge. Special data analysis is required such as down sampling, interpolation, time weighted average We need a purpose-built database to process time-series data.

While other databases can also process time series data to some extent when the data is small, TSDB can handle data ingestion, compression, and aggregation over time more efficiently. Taking the Internet of Vehicles scenario as an example, 20,000 vehicles, 60 metrics per car, assuming that it is collected once per second, then 20,000 \* 60 = 1200,000 metric values will be reported per second, that is, 120W data metric values per second, and each metric value is 16 bytes (assuming that only 8 bytes of timestamp and 8 bytes of floating-point numbers are included), then about 64G of data will be generated per hour. In fact, each indicator value will also be accompanied by additional data such as labels, and the actual storage space will be larger.

"The so-called time series data is data with strong correlation with time and time stamp, which is a repetitive time series, and the time point is not same, the data fields are the same, but the values are different. From a market point of view, the past CNC factory PLCs or grid SCADA data are time series data, but the amount of data is not so large. Since 2015 and 2016, Smart Door Locks, Sweeping Robots, Smart Air Conditioners and other IoT smart devices have begun to rise, Industry 4.0 Intelligent transformation has gradually been done more, coupled with the scene of public cloud and private cloud monitoring, servers and other devices from three or five thousand to hundreds of thousands of other new scenarios, the concurrency of time series data has suddenly risen, from second to second.

A few thousand have reached the scale of tens of thousands, hundreds of thousands per second. Zheng Bo introduced, "Relational databases, including Hadoop, cannot get these numbers. According to good storage and compression, the disk occupancy rate is relatively high, and there is no optimization and support for IoT devices and these scenarios. As a result, foreign manufacturers (such as Facebook) and emerging manufacturers (such as InfluxDB) began to consider whether to develop new databases to support. China's Internet of Things is developing relatively fast, and now it is also a period of vigorous development, and China also needs to have such a database to meet this demand. Arguably, it was demand that drove the market, the pressure on data storage with the cost of compression and the pressure on the availability of some uses, triggered the invention of the time series database.

The data of the Internet of Things and the Industrial Internet is streaming data, like video streaming, and the value of a single data point is very low, and even the loss of data for a short period of time does not affect the conclusion of the analysis, nor does it affect the normal operation of the system. However, seemingly simple things, due to the huge number of data records, the real-time writing of data has become a bottleneck, and query and analysis are extremely slow, which has become a new technical challenge. Traditional relational databases, NoSQL databases and streaming computing engines have limited performance improvement due to the lack of full use of the characteristics of Internet of Things data, and can only rely on cluster technology to invest more computing resources and storage resources to deal with, and the operation and maintenance costs of the system have risen sharply.

In the face of this high-growth Internet of Things data market, in recent years, a number of companies focusing on time series data processing have emerged, such as InfluxData in the United States, whose financing has exceeded 130 million US dollars, and its product InfluxDB has a considerable market share in IT operation and maintenance monitoring. OSIsoft, an established real-time database company in industrial control, received a $1.2 billion investment from SoftBank in May 2017, hoping to become a leader in databases in the emerging IoT space. Open source communities are also very active, such as OpenTSDB, which is based on HBase. In China, Alibaba, Baidu, and Huawei all have Products based on OpenTSDB.

Founded in 2017, Beijing Taosi Data Technology Co., Ltd. is optimistic about this market, does not rely on any third-party software or open source software, and independently develops TDengine Database, a complete time series big data processing engine, after absorbing the advantages of many traditional relational databases, NoSQL databases, streaming computing engines, message queues and other software. TDengine Database's performance far exceeds that of InfluxDB, and its simple installation, deployment, and maintenance, using SQL interfaces, and learning costs are almost zero, which is expected to become a dark horse in the time series data processing market.

Time-series database hopes to provide customers with stable distributed clusters to meet the application requirements of the Internet of Things 24-hour uninterrupted operation, even the energy industry and power industry with extremely high stability requirements can be perfectly supported; The second is to provide customers with a powerful computing engine to meet the needs of customers for historical data, real-time data calculation and modeling, mining value from data, and then controlling equipment, and optimizing processes and processes to influence transactions and decisions; The third is to continue to cooperate with industry customers and integrators to create a function library, or algorithm library, in various industries, only in this way can all walks of life be used and create more value; The fourth is to provide lightweight one-stop solutions to reduce the comprehensive application cost of enterprises using big data, so that a large number of traditional enterprises can also use big data.

Here are few databases using time-series database:

First is the InfluxData.

InfluxData is the creator of [InfluxDB](http://www.influxdata.com/products/open-source/" \l "influxdb), the open source time series database. It is purpose-built to handle the massive volumes of time-stamped data produced by IoT devices and sensors, applications, containers, VMs and networks. With InfluxDB, developers can build real-time applications for IoT, analytics and cloud-native services quickly and at scale.

InfluxData is focused on developer happiness and meeting developers where they are – in their language of choice, using their preferred tools and wherever they are building applications: in the cloud, on-premises, or locally. InfluxData is also committed to helping developers get up and running faster, so they can focus on other things.

InfluxDB customers range from startups to Fortune 500 enterprises with use cases that span every industry vertical, such as consumer and industrial IoT, security, fintech, renewable energy, and more. There are currently over 600,000 daily active instances of InfluxDB OSS and over 50,000 registered developers using InfluxDB Cloud globally; the community continues to grow rapidly every year.

The second is the MongoDB:

MongoDB’s application data platform provides developers a unified interface to power operational and transactional database requirements plus search, real-time and data lake applications needs. This enables developers to move fast and simplify how they build with data for almost any class of application.

In its most recent developer survey, Stack Overflow names MongoDB as [the database developers most wanted](https://www.mongodb.com/blog/post/stack-overflow-research-most-wanted-database) to use. This was the 4th consequetive year MongoDB had topped this poll. MongoDB was also named a leader in the [Forrester Wave™: Big Data NoSQL, Q1 2019](https://www.mongodb.com/collateral/forrester-nosql-wave) and in the [Forrester Wave: Database-As-A-Service, Q2 2019](https://www.mongodb.com/collateral/forrester-wave-database-as-a-service).

MongoDB is designed to meet the demands of modern apps with a technology foundation that enables you through:

1. The [document data model](https://www.mongodb.com/document-databases), giving developers the easiest and most intuitive way to work with data.

2. The [MongoDB Query API](https://www.mongodb.com/mongodb-query-api), giving developers the fastest way to innovate in building transactional, operational and analytical applications.

3. Delivered as a [multi-cloud, global platform](https://www.mongodb.com/cloud/atlas), giving developers the freedom to run their applications anywhere with the flexibility to move data across private and public clouds as requirements evolve – without having to change a single line of code. Atlas enables customers to take advantage of MongoDB’s capabilities across AWS, Azure, and GCP without needing to deploy, operate, and scale the software or underlying infrastructure themselves.

The future of time-series database is bright, reasons are as follows:

First of all, the current domestic talent resources in the database field are very scarce, and it will be more difficult to find relevant talents such as database kernel research and development; Second, under the wave of localization, there may be some enterprises that take advantage of policy dividends and participate in market competition too little. The excessive use of policy dividends is not conducive to the continuous improvement of product quality, which needs to be avoided by enterprises as much as possible; Third, enterprises should go out and compete with the top products and teams in the world.