Introduction to time series database

Author: 朱盛华 2020141461175 Software College

Absrtact: a brief introduction to the knowledge of time series database, including some simple concepts, the characteristics of time series data and the requirements of database, the introduction of traditional tools to realize time series database, and a brief description of some popular databases.

Key words: Time Series Database

Time series data are some column points indexed / listed in chronological order.

Related concepts of time series database：

\*Metric: the table in metric similar relational database represents a set of similar time series data, such as establishing a table for air quality sensors to store the monitoring data of all sensors.

\*Tag: tag describes the characteristics of the data source, which usually does not change over time. For example, sensor devices contain tag information such as deviceid and region where the device is located. The database will automatically index tags, and support multidimensional retrieval and query based on tags; Tag consists of tag key and tag value, both of which are of string type.

\*Timestamp: timestamp represents the time point of data generation, which can be specified when writing or automatically generated by the system.

\*Measured value field: field describes the measurement index of the data source, which usually changes over time. For example, the sensor equipment includes fields such as temperature and humidity.

\*Data point: a field value generated by the data source at a certain time is called a data point, and the data points are used as statistical indicators when querying and writing the database.

\*Timeline time series: a certain indicator of the data source changes with time to form a timeline. The combination of metric + tags + field determines a timeline; The calculation of time series data, including downsampling, aggregation (sum, count, Max, min, etc.), interpolation, etc., are based on the timeline dimension.

Its features include:

1. Stable writing, high concurrency and high throughput: timing data is usually generated at a fixed time frequency, which is not restricted by other factors, and its data generation speed is relatively stable. Timing data is generated by each individual independently, so when there are a large number of individuals, the concurrency and throughput of writing are very high.

2. Large amount of data: there may be terabytes and petabytes of data to be stored every day, and the data is often generated automatically.

3. Write more and read less: there are many indicators monitored, but usually only a few specific indicators and specific scenarios are concerned.

4. Real time writing: the writing of timing data is real-time, and each writing is the most recently generated data. Because its data generation advances with time, there are few operations of updating and deleting.

5. The attention of recent data is higher, and the data with a long history is rarely accessed, which is clear between hot and cold.

6. Multidimensional query and analysis.

7. Data with timestamp is structured data.

Requirements of time series data on time series database:

1. High throughput and high concurrency writing ability: sequential data has the typical characteristics of writing more and reading less. In terms of reading and writing, the first trade-off is the ability to write. There are high requirements for the high concurrency and high throughput write capability of the database.

2. High availability: distributed architecture, the system should have the ability of horizontal expansion.

3. Hierarchical data storage: put the latest hour level data into memory, the latest day level data into SSD, and the more distant data into cheaper HDD or directly use TTL to expire and eliminate.

4. High compression ratio: on the one hand, it saves costs, on the other hand, the compressed data can be more easily stored in memory.

5. Multi dimensional query and aggregation: the query latency at the interaction level is very low even when the data base (TB level) is large. On the basis of a large amount of data, the original data that meets the conditions will be queried and aggregated. The original value may not be in memory because of the long time.

Traditional types of time series database:

Relational database; NoSQL database; Real time database; General big data platform.

Relational database implements a time series database:

Range partitioning maps data to partitions based on ranges of partition key values that you establish for each partition. It is the most common type of partitioning and is often used with dates. For example, you might want to partition sales data into monthly partitions.

Each partition has a VALUES LESS THAN clause, which specifies anon inclusive upper bound for the partitions. Any binary values of the partition key equal to or higher than this literal are added to the next higher partition.

advantage:

Leverage existing computers and resources.

The team is familiar with relevant knowledge.

Disadvantages:

Large storage cost and poor compression of time series data.

High maintenance cost single machine system.

The write throughput is poor, and the write throughput of a single machine is low, which cannot meet the write pressure of tens of millions of levels.

Poor query performance, suitable for transaction processing, poor aggregation ability of massive data.

NoSQL database introduction:

NoSQL (originally referred to as "non SQL" or "non relational") databases provide a mechanism for storing and retrieving data. The modeling method of this data is different from the tabular relationships used in relational databases.

NoSQL databases are increasingly used for big data and real-time network applications. NoSQL systems are sometimes called not just SQL to emphasize that they may support SQL like query languages, or coexist with SQL databases in a multilingual persistence architecture.

Generally, large-scale processing and storage of timing data are allowed.

The advantages and disadvantages are the same as above.

Real time database:

Real time database is a kind of database system that uses real-time processing to deal with the changing workload.

Real time processing means that transactions should be processed fast enough to return results and take action immediately.

General big data platform:

Technology companies use a series of open source big data tools to build their own big data processing platform.

The most popular time series database Prometheus

• InfluxDB

• TDengine

• TimeScaleDB

• OpenTSDB

Prometheus：

Prometheus is an open-source systems monitoring and alerting toolkit originally built at SoundCloud. Since its inception in 2012, many companies and organizations have adopted Prometheus.

Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels.

InfluxDB

InfluxDB is an open-source time series database (TSDB) developed by InfluxData.

Written in Go

In InfluxDB, a point represents a single data record, similar to a row in a SQL database table. Each point:

• has a measurement, a tag set, a field key, a field value, and a

timestamp;

• is uniquely identified by its series and timestamp.

TDengine

TDengine is an open source time-series database developed by TAOS Data :

• High-Performance

• Scalable

• SQL-support

Written in C

TimescaleDB

• TimescaleDB is an open-source relational database for timeseries and analytics.

• Built on top of PostgreSQL.

• Written in C

• TimescaleDB supports full SQL and adds features for timeseries analytics.

OpenTSDB

• OpenTSDB is a distributed, scalable Time Series Database (TSDB) written on top of HBase.

• Written in Java

In OpenTSDB, a time series data point consists of:

• A metric name.

• A UNIX timestamp (seconds or milliseconds since Epoch).

• A value (64 bit integer or single-precision floating point value), a JSON formatted event or a histogram/digest.

• A set of tags (key-value pairs) that describe the time series the point belongs to.

OpenTSDB offers a built-in, simple user interface for selecting one or more metrics and tags to generate a graph as an image. Alternatively an HTTP API is available to tie OpenTSDB into external systems such as monitoring frameworks, dashboards, statistics packages or automation tools.

Data Model for TSDB

There are two types of data inside TSDB:

• Time-series data: the inserted time-stamped data points, it will grow with time, huge volume.

• Static Data: attributes for a time-series(device), like model, brand of a vehicle, location of a temperature sensor. They are called labels or tags in TSDB. They are used to identify a unique or a set of time-series for analysis.

Two data models:

Data Model – Tag-Set Model:

• Prometheus, InfluxDB and OpenTSDB adopt this model

• Every time series is uniquely identified by a metric name and a set of labels (measurement/tags in InfluxDB nomenclature )

• Schema is not required to be defined, easy to pick up. But no data validation, and slower query.

• If the label is changed, a new time-series

Data Model – Relational Model

• TDengine, TimeScale, QuestDB adopt this model

• A time-series is identified by a unique ID or table name

These are some simple time series data introductions. I hope you have some new understanding of time series database, and you can continue to query materials to deepen your understanding, and finally practice creating a database and doing some operations yourself.