The introduction of TDengine

I. What is TDengine?  
 TDengine is an innovative big Data processing product launched by TAOS Data

in the face of the rapidly growing Internet of Things big Data market and technical

challenges. It does not rely on any third party software, nor does it optimize or package an open source database or streaming computing product. It is independently

developed after absorbing the advantages of many traditional relational databases,

NoSQL databases, streaming computing engines, message queues and other software, and has its own unique advantages in the processing of big data in sequential space.  
 One of TDengine’s modules is the sequential database. But in addition, in order to reduce the complexity of research and development, the difficulty of system maintenance, TDengine also provides caching, message queue, subscription, streaming computing and other functions, for the Internet of Things, industrial Internet big data processing to provide a full stack of technical solutions, is an efficient and easy to use

Internet of things big data platform.  
 TDengine is an efficient platform for storage, query and analysis of time series

big data, specially designed for the optimization of Internet of Things, Internet of

vehicles, industrial Internet, operation and maintenance monitoring, etc. You can use

it just like you use MySQL, a relational database, which is simple and convenient.

II. Why TDengine?

Firstly, we summarize the characteristics of Internet of Things data as follows:

1. The data is sequential and must have a timestamp;

2. Data is structured;

3. Data is rarely updated or deleted;

4. The data source is unique;

5. Write more than read less than Internet applications;

6. Users focus on trends over a period of time, not values at a particular point in time;

7. Data has a retention period;

8. Data query and analysis must be based on time periods and geographic regions;

9. In addition to storing queries, various statistical and real-time computing operations are often required;

10. Smooth flow, can be predicted;

11. Often need to have interpolation and other special calculations;

12. Huge amount of data, more than 10 billion pieces of data can be collected in a day;

After summarizing these characteristics, it can be found that the Internet of Things data is like log data, almost no update operation is possible, so the implementation of transaction processing in the database is completely redundant. Data is temporal, timestamp can be used as the primary key, there is no need for complex index structure;

Data in the Internet of Things is structured and stored by key-value like HBase and Cassandra, which greatly reduces the computing and storage efficiency. Therefore, structured storage should be used. Internet of Things data is hot or cold based on time, with the data just collected being the hottest, rather than the user clicking on it. Therefore, efficient caching can be achieved with simple FIFO memory management, without the need for Redis at all; Internet of Things data is a data flow from the perspective of a device. It is not the most natural thing to realize the flow calculation of sliding Windows. How can we use such a complex engine as Spark?

For data partition, simple partition by device partition by time period, it is easy to solve, there is no need for complex partition mechanism; The flow of IOT data is relatively smooth, and IOT devices are bound to have caching capabilities, so you can ditch Kafka and implement a simple message queue and data subscription to meet your needs. Then I found the timing database, looked at their documentation and code, and found that they made use of some characteristics of timing data, but still not fully utilized, and only positioned as a database. Behind and learn about the industry has a real-time database and found the real-time database are old products, basically or Windows on research and development, the price is expensive, and there is no standard SQL, extending little, almost no big data analysis ability, have no ability to cope with the increasingly large amount of data and large data analysis requirements, will be eliminated sooner or later.

Therefore, TDengine quickly positioned its product, that is, the big data platform of the Internet of Things. It should integrate the series of functions such as timing database, cache, message subscription and streaming computing to solve the big data problems of the Internet of Things in a one-stop way, so as to greatly reduce the complexity and cost of system development and maintenance.

After studying the characteristics of Internet of Things data, TDengine made two technical innovations: 1. The data model of "one table for one device" greatly improves the data insertion and query efficiency of a single device; 2. 2. Label each table with static label, and store the static label data and the collected dynamic data completely separately to solve the problem of multi-table aggregation query.

III. Applicable scenarios of TDengine

TDengine as a basic software, the application range and its wide, in principle, all the use of machines, equipment, sensors to collect data can be used.

Here are some typical scenarios:

1. Public safety: Internet access records, call records, individual tracking, interval screening

2. Power industry: centralized monitoring of smart electricity meters, power grids and power generation equipment

3. Communication industry: detailed bill, user behavior, base station/communication equipment monitoring

4. Financial industry: transaction records, access records, ATM, POS machine monitoring

5. Travel tools: train/car/taxi/plane/bike real-time monitoring

6. Traffic industry: real-time road conditions, intersection flow monitoring, bayonet data;

7. Petroleum and petrochemical: real-time monitoring of oil Wells, transportation pipelines and transportation fleets

8. Internet: server/application monitoring, user access log, advertising click log

9. Logistics industry: tracking and monitoring of vehicles and containers

10. Environmental monitoring: weather, air, hydrology, geological environment monitoring;

11. Internet of Things: elevators, boilers, machinery, water meters, gas meters and other networked equipment

12. Military industry: data collection and storage of various military equipment

Manufacturing: production process control, process data, supply chain data collection and analysis

IV. the characteristics of TDengine

1. Designed specifically for Internet of Things data, the function that each collection point corresponds to one tab le is realized by utilizing the timing characteristics of Internet of Things data. But it is not suitable for handling general Internet data.

2. The use of column storage + compression, to save hardware costs. (High compression efficiency: it is highly efficient to use the characteristics of little fluctuation of Internet of Things data, compress after DIF interpolation, and then compress second order.)

3. Support high availability and divide each physical node into multiple virtual data nodes and virtual management nodes. Virtual data nodes store data, and virtual management nodes manage MetaData. Virtual data nodes and virtual management nodes are distributed on different physical nodes to achieve high availability of data set applications.

4. In storage structure, each collection point is used to create an independent table for storage. In this way, the data of each collection point can be stored continuously and the reading efficiency can be improved. Since each table has only one data source, lockless write is implemented to improve the write rate.

5. The concept of super table is introduced for changeable aggregation. The same type of collection device can create a super table. When creating a super table, you can specify labels for these tables to filter the tables in the database during query. In this way, you can achieve fast multi-table aggregation even if there are many tables in the database

6. The installation package is very small, easy to install and use. Support SQL, syntax is similar to MySQL.

V. Advantages of TDengine

1, More than 10 times of performance improvement: define the innovative data storage structure, single core can process at least 20,000 requests per second, insert millions of data points, read more than 10 million data points, more than 10 times faster than the existing general database.

2. The cost of hardware or cloud service is reduced to 1/5: Due to super performance, the computing resources are less than 1/5 of the general big data scheme; With column storage and advanced compression algorithms, storage takes up less than 1/10 of a generic database.

3. The whole stack temporal data processing engine: the database, message queues, caching, streaming computing functions such as integration, application without having to integrate Kafka/Redis/HBase/Spark/HDFS software, greatly reduce the complexity of the application development and maintenance costs. Seamless connection with third party tools: Integration with Telegraf, Grafana, Matlab, AND R without a line of code. MQTT, OPC, Hadoop, Spark, etc. will be supported in the future, and BI tools will be seamlessly connected.

4. Powerful analysis function: whether it is ten years ago or a second ago, the specified time range can be queried. Data can be aggregated on a timeline or across multiple devices. AD hoc queries can be made at any time by Shell, Python, R, and MATLAB.

5. High availability and horizontal scaling: With distributed architecture and consistent algorithms, TDengine ensures high availability and horizontal scaling to support mission-critical applications through multiple replication and clustering features.

6. Zero operation and maintenance cost, zero learning cost: cluster installation is simple and quick, no need for sub-database sub-table, real-time backup. Similar to standard SQL, support RESTful, support Python/Java/C/C++/C#/Go/Node.js, similar to MySQL, zero learning cost.

7. Core Open Source: TDengine's core is open source, except for some ancillary features. Enterprises are no longer tied to databases. This makes the ecosystem stronger, the product more stable, and the developer community more active.