# Research on real-time data processing technology for Internet of things

Jia Wu [1,a], Dan Su[1], Chao Liu[1], Bing lv[1], ShengPeng Ji[2], Xianhui Li [2,b], Gang Li[2]

[1]JiBei Electric Power Corporation, Beijing 100025, China;

[2]China Realtime Database Co.,Ltd , Nanjing 210000, China.

[a]alice.0729@163.com, [b]lixianhui@sgepri.sgcc.com.cn

**Keywords:** Sensor real-time data, distributed processing-data partition

**Abstract.** In the age of internet of things, the storage and real-time processing requirements of massive amounts of real-time data generated by sensors, so that the traditional database or data processing architecture has been unable to deal with. Analyzing the processing characteristics of the real-time data of the things and studying the data processing technology of the Internet of things. A distributed storage and computing architecture for data partition and multi node distribution is proposed. The architecture can achieve linear expansion of the processing performance and storage size of the real-time data. For different real-time data processing scene, the design of different data partitions in different formats, on the one hand to ensure data replication redundancy, on the one hand to provide the optimal data query efficiency for different scenarios.

## Introduction

Basing on Gartner forecast, Until 2020, there will have 26 billion equipment for internet of things around the world, Management of networking equipment (dynamic) data is more complicated than management of traditional(static)data. The most important difficulty is that the quantity of substantial increase and treatment of the real-time requirements strengthens significantly.

On the one hand, In the process of evolution of the Internet of things application, Sensor number increasing, data sampling frequency increasing, data accumulation time is becoming more and more long, thus, the amount of produced data is very large, it can be the trillions of records, the generation velocity of data is also very fast, it can be millions of record per sec [2].

On the other hand, The real-time data, which is produced by sensor, is using for abnormal warning, trend prediction. The request is that it according to the data to make the respond immediately. Thus, data must do the real-time query, real-time analysis.

Facing great deal of real-time data, the traditional database is not only “overflow” but also “cannot check out”. So the current industry use the compromise plan: only store the recent data (drop the old data), or only store some sampling record (e.g. down-sampling to record the hourly data). These solutions are clearly not be able to avoid a discarded valuable data. [3]At the age of the year, data is the king. Who owns the data more, who has stronger competitiveness, and it is a pity to abandon data.

## Real-time Data Processing Architecture

According to the Internet of things mass time-series data processing requirements, Combining the theory of distributed storage and parallel computing theory, [6] With reference to the distributed file system or distributed non-relational database, designing the distributed real-time data processing of the technical framework, As show Fig.1.

The management server is the manager of the distributed real-time data processing, mainly used for store the metadata information, including each nodes division, node status, Data partition type, data blocks position, task scheduling, safety management, and other key information.

Storage computing node is responsible for the mass fragmentation of the sequential data storage, complete different kinds of calculation at the same time. The number of the storage computing node is limited by Ethernet bandwidth, computer room physical conditions and other hard conditions, it

 

can support several to hundreds of scale during designing. Each storage computing nodes logic is equal, deploy the same process to finish the same logic calculation, according to the management server to the data regionalization principle, only store the data which belongs to the corresponding partition data, So as to achieve the distributed storage purpose.

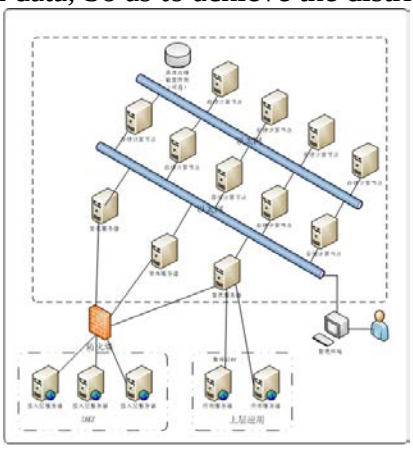

Fig.1 processing technical framework

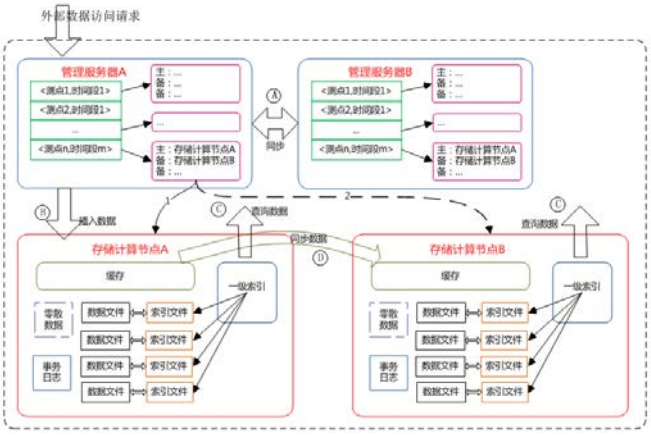


Fig2 the distributed real-time data storage mechanism

As show Fig.2, the management server A and B are two matchable management servers, storage the same content of the metadata information to finish the same function.

As shown above, when insert the data, through the management server to search the storage data zone is the main storage for the calculate node A, in this node A to insert the data and record in the buffer and record this transaction log. According the metadata information can obtain this data storage for the calculate node B, then the storage calculate node A in charge of to sync the transaction into node B, when the node B finish the data insertion that will feedback the insert success information to node A. Till now, the data insert transaction is finish. If there have any fault during this process, the node A will roll back the transaction to ensure the transaction consistent and completely. Each storage calculate node should charge of the own data filing operation, from persisting data to the data file and establish the related indexes.

In data processing, searching all storage computing nodes form this partition data by using management server, and according to the metadata to choose the free node to complete data query process, then realize the Load balancing in the process of query.

**Real-time Data Saving Format**

The basic format of sensor real-time data includes time, sensor ID and value. Value can be a simple format (such as the temperature sensor), and also can be the numbers of the simple format combination (such as the spatial position sensor).

The main feature of Real-time data process is the accurate、fuzzy and condition retrieval on the time section. For example: Tocalculate the housing vacancy rate and etc. We will conclude the time-series data process for the two scenarios: Curve of the query and the section of the query. The Curve of the query means the single sensor of timing data query during the temporal interval. The section of the query is mean at a certain moment time the multiple sensors data query.

From the data processing dimension point of view, timing data processing have two dimension, one is the time dimension that means the Curve of the query. Another is the sensor ID dimension that means the section of the query. In the single factory, due to the installation of the sensor is limited, the design of timing data storage format more focus on the timing dimension process. During the extensive property linked wisdom city and smart grid, installation the sensors less than the million and more than a hundred million, the design of timing data storage format need to consider these two dimension process at the meantime.

Due to the limited of the disc IO, the data storage format heart aim is to make the timing data on the disc is "Ordinal Storage", then as per the interval scanning head can be ensure the high speed moving sequential. If can be realized, there is no doubt the best for the section of the query which can locate to

a certain time and remove the continuous values from the different sensors at the same time. But, the Curve of the query efficiency under this format is lower, due to the single sensor at the different time data storage is disconnected, the Curve of the query will make a lot of disk skip, the time consumption on head addressing. Meantime, if the timing data base format is according the sensor ID sort, the Curve of the query efficiency is better. But the section of the query efficiency is the lowest.

In order to meet the above scenario demand, reference the design of the distributed documents system, in this paper which present a data redundancy copy for timing storage format. According to the Curve of the query demand to design the sensor ID sorting format,T-Format. According the section of the query demand to design the time sorting format, T-Format. Considering to consideration to the query for the Curve and section demand, the design as per the time division, then according the sensor ID sorting format, IT-Format.

Using the following sample data to be the sample, According to the three different kinds of storage format to organizes time-series data, show the difference between several formats by comparing each other. For example, there are three sensor ID1、ID2、ID3,from 2015-05-12 10:00:00 to 2015-05-12 11:00:00, every 15 minutes to produce a data in an hour interval, totally produced 12 time-series data, Content respectively as follows table 1:

Table1 time-series data

| 2015-05-12 10:00:00 ID1, V1 (c1, c2, c3, ....) |
|---|
| 2015-05-12 10:00:00 ID3,V2 |
| 2015-05-12 10:00:00 ID2,V3 |
| 2015-05-12 10:15:00 ID1,V4 |
| 2015-05-12 10:15:00 ID2,V5 |
| 2015-05-12 10:15:00 ID3,V6 |
| 2015-05-12 10:30:00 ID3,V7 |
| 2015-05-12 10:30:00 ID2,V8 |
| 2015-05-12 10:30:00 ID1,V9 |
| 2015-05-12 10:45:00 ID2,V10 |
| 2015-05-12 10:45:00 ID1,V11 |
| 2015-05-12 10:45:00 ID3,V12 |

Table 2 I-Format time-series data

| Sensor ID | Time | Value |
|---|---|---|
| ID1 | 2015-05-12 10:00:00 | V1 |
| ID1 | 2015-05-12 10:15:00 | V4 |
| ID1 | 2015-05-12 10:30:00 | V9 |
| ID1 | 2015-05-12 10:45:00 | V11 |
| ID2 | 2015-05-12 10:00:00 | V3 |
| ID2 | 2015-05-12 10:15:00 | V5 |
| ID2 | 2015-05-12 10:30:00 | V8 |
| ID2 | 2015-05-12 10:45:00 | V10 |
| ID3 | 2015-05-12 10:00:00 | V2 |
| ID3 | 2015-05-12 10:15:00 | V6 |
| ID3 | 2015-05-12 10:30:00 | V7 |
| ID3 | 2015-05-12 10:45:00 | V12 |

**Format.** First of all,I-Format is sorted by ID. Then it is sorted by timing while ID is the same. The specific format is shown in table2:

Table 3 T-Format time-series data

| Time | Sensor ID | Value |
|---|---|---|
| 2015-05-12 10:00:00 | ID1 | V1 |
| 2015-05-12 10:00:00 | ID2 | V3 |
| 2015-05-12 10:00:00 | ID3 | V2 |
| 2015-05-12 10:15:00 | ID1 | V4 |
| 2015-05-12 10:15:00 | ID2 | V5 |
| 2015-05-12 10:15:00 | ID3 | V6 |
| 2015-05-12 10:30:00 | ID1 | V9 |
| 2015-05-12 10:30:00 | ID2 | V8 |
| 2015-05-12 10:30:00 | ID3 | V7 |
| 2015-05-12 10:45:00 | ID1 | V11 |
| 2015-05-12 10:45:00 | ID2 | V10 |
| 2015-05-12 10:45:00 | ID3 | V12 |

Table 4 IT-Format time-series data

| Time Interval | Sensor ID | Time | Value |
|---|---|---|---|
| 2015-05-12 | ID1 | 2015-05-12 10:00:00 | V1 |
| | ID1 | 2015-05-12 10:15:00 | V4 |
| | ID1 | 2015-05-12 10:30:00 | V9 |
| | ID1 | 2015-05-12 10:45:00 | V11 |
| | ID2 | 2015-05-12 10:00:00 | V3 |
| | ID2 | 2015-05-12 10:15:00 | V5 |
| | ID2 | 2015-05-12 10:30:00 | V8 |
| | ID2 | 2015-05-12 10:45:00 | V10 |
| | ID3 | 2015-05-12 10:00:00 | V2 |
| | ID3 | 2015-05-12 10:15:00 | V6 |
| | ID3 | 2015-05-12 10:30:00 | V7 |
| | ID3 | 2015-05-12 10:45:00 | V12 |
| 2015-05-13 | | | |

Firstly consider using I-Format to retrieval performance by given ID, if the efficiency for the statistical calculation is low in the given time interval or a certain time, it must scan all data and find out record meet time interval requirement to do the statistics.

**T-Format.** First of all, T-Format according to the event occurred time sorting, when the same time, it sort by the sensor ID, The specific format is shown in table3:

Firstly consider using T-Format to retrieval performance for a given time or interval without specified ID, for the given time with the specified ID, the efficiency is low, because this format is no aggregated data by ID, retrieve a given ID needs to scan all the data within a given period of time.

**IT-Format.** First of all, IT-Format is sorted by time interval, the time interval is available in hours, days, months, years, etc. It similar to the partitioning concepts of traditional relational database. In the same time interval, it is sorted by the sensor ID. in the same sensor ID, it is sorted by the event time. The specific format is shown in table 4:

IT-Format query efficiency is worse than T-Format if there has not ID in specify a time interval or a moment, because it needs to scan all the data period across partitions. IT-Format retrieval efficiency is worse thanT-Format if it has the given ID, because within the period across each partition, successively scanning the corresponding data of a given ID. But IT-Format can balance two types of queries, and performance loss is not big.

If there has both curve and cross section query application scenario at the same time, using IT-Format to consideration to the both of the requirement. If the scene determined. Firstly consider usingT-Format or T-Format. For example, Investigator traced suspected vehicle trajectory, or the suspect of the phone-record and so on, belongs to the curve of the standard query requirement, Priority should be given to the I - Format. Calculation of indicators, such as road conditions, traffic flow and so on, belong to the range of the statistical analysis of demand. Firstly consider using

T-Format； If user wants to check the Trajectory of the given car, and also wants to statistical the road condition information. Firstly consider using IT-Format.

## Conclusion

In this paper, basing on different time-series data application scenarios, using different format to do the shard storage and distributed to the different storage calculation nodes by Unified coordination processing, Through this distributed architecture to realize large scale temporal data processing, guarantee the efficiency of the time-series data in different application scenarios and also ensure the high reliability and scalability of the time-series data storage, it has the extensive application value.

## References

[1] Robert Badalian. Internet of things:Gate to Sensor Data Analysis[J]. China Integrated Circuit, 2015, 8:38–40.

[2] Ming A. Big data and Internet of things behind the smart device [J]. Microcomputer, 2015, 1:14.

[3] Yaqi,Song. Guoliang,Zhou. YongLi, Zhu.. Present situation and challenge of big data processing technology in smart grid [J]. Grid technology,2013,4:927 - 935.

[4] Yanxian,He. Tao,YU. YanZhao, Chen. Etc. Research on the mechanism of data storage and query in the Internet of things[J]. Computer Science,2015,3:185–190

[5] Xiaoming,Ling. Yusheng, Hao. Research on a disk history database model [J]. Computer Engineering,2014,5:26 - 30.

[6] Xiaoping,Feng.Jun,Gao. Distributed database HBase[J]. Information communication,2015,7:84-85.