Massive AIS data storage and query based on Hadoop platform

Taizhi Lv\textsuperscript{1,2}, Chenyong He\textsuperscript{2}, Juan Zhang\textsuperscript{1}, Zhiyang Song\textsuperscript{1}

\textsuperscript{1}Jiangsu Maritime Institute, Department of Information Technology, Nanjing, China
\textsuperscript{2}Nanjing Huihai Transportation Technology Company Limited, Nanjing, China
20060826@jmi.edu.cn

Abstract. There are nearly 50 million AIS data generated every day in the Yangtze River Basin. Efficient storage and query of AIS data become the key problem of smart maritime administration. To overcome this problem, a distributed storage and query scheme based on Hadoop platform is proposed. Firstly, the original AIS message is parsed, the abnormal data is removed and the parsed data is pushed to the Kafka message queue. Secondly, the messages in the queue are pushed to different servers to complete the storage of real-time cache, historical trajectory and regional information. Finally, the query interface is provided to realize the real-time query of ship status, trajectory and regional ship in a certain period of time. The real applications show that compared with the traditional relational database, the solution can meet the requirements of real-time query and has high scalability.

1. Introduction

Automatic identification system (AIS) is a digital navigation aid system composed of shore based and shipboard equipment [1]. AIS data includes ship type, name, maritime mobile service identification (MMSI), arrival time, speed, transit area and other information. It can be applied to ship identification, ship tracking, ship collision avoidance, maritime management, channel calculation, etc.

AIS data is ship trajectory information, and has the big data characteristics such as large-scale, high-speed, low value density. The massive data and high concurrent access of AIS bring great challenges to the traditional storage method based on the relational database. Some scholars propose some solutions of dividing relational database or table, which alleviates the load defects to a certain extent, but also leads to the complexity of expansion, and cross table query also increases the complexity of query [2]. With the advent of the era of big data, the development of distributed storage and NoSQL database technology provides an opportunity to solve massive data storage and query. Hadoop, a highly reliable, efficient and scalable distributed processing platform, and NoSQL such as HBase and Redis have been widely used in medical, transportation, intelligence and many other fields [3-4].

It is of great significance to use big data technology to manage AIS data and realize high performance, high reliability storage, high concurrency and low latency storage. HBase supports the data volume of a single table to reach Pb level. By setting partition and row key reasonably, it can read data in millisecond level. By using HBase to store AIS historical trajectory and historical region information, it can effectively solve the problem of inefficient reading and writing for massive data in relational database. Based on the high-performance caching mechanism, Redis can meet the application scenarios of high concurrency and high throughput. Ship real-time location information is often used in various business systems, which needs to meet the high concurrent access of users. AIS dynamic information, AIS static
information and ship basic information are stored in Redis, and it can provide high concurrent user access.

2. System architecture
In order to implement the real-time acquisition, storage and query of AIS data, the system architecture as shown in Figure 1 is designed, including data pre-processing module, data storage module and data query module.

The data pre-processing module collects data from each AIS based station or data center. By the three steps of data parsing, data de-duplication and data clean, the accuracy of data is improved. The pre-processing data is sends to Kafka message queue. The data storage module is divided into three sub-modules. The real-time data storage sub-module takes the AIS data from the message queue, and then updates the latest location and ship information to Redis. The historical trajectory storage sub-module stores the dynamic information in AIS into HBase database. The historical region storage sub-module stores the ships contained in a certain region in Redis in minutes, and stores the region information in HBase every minute. The data query sub-module provides users with unified access through the construction of micro service cluster, and realizes the query of ship, trajectory, region and other information.

3. Data pre-processing
For monitoring ships, the shore based network system is established to collect and store all the AIS information along the coast. The data pre-processing module receives the real-time AIS data from the network system and writes it to the Kafka message queue through three steps: parsing, cleaning and de-duplication.

3.1. Data parsing
The original message received adopts compression coding. In order to obtain the real information, it needs to be decoded according to the specified format. The AIS information is classified as two main groups, dynamic information and static information. Dynamic information includes MMSI, speed, longitude and latitude, and static information includes MMSI, international maritime organization (IMO) number, name, captain, width, destination port, category, etc. Data parsing is to find the corresponding start byte by the VDM protocol, continuously take the data with the specified location width, and then convert and merge it into the required information.
3.2. Data cleaning
The abnormal data mainly includes the unreasonable or vacant data such as longitude and latitude, speed and MMSI. The abnormal records will be deleted by the pre-processing module.

3.3. Data de-duplication
In order to avoid signal blind area, there is AIS receiving overlap area in the shore based network system. It causes one AIS message sent by a ship to be received by multiple AIS shore based stations. A large number of redundant AIS data will increase the cost of storage, transmission and processing. Data de-duplication is to delete duplicate record. AIS does not have a complete time stamp, it only contains UTC seconds. In most cases, the time to obtain AIS from shore base is less than 1 minute, so time stamp can be labelled for each AIS data.

The calculation of time stamp is as follows. If the second of receiving time is less than the value of second in AIS data, the labelled time is the receiving time minus 1 minute, and the second is set the value of second in AIS data. Otherwise, the labelled time is receiving time, and the second is set the value of second in AIS data. If the record is found in Redis, it will be discarded directly. If no record is found, this record is stored in the Redis cache and written to the message queue. Each AIS data is cached in Redis for one minute.

3.4. Message Queue
Kafka is a distributed, high throughput, high scalability message queuing system, including producers, consumers, topics, partitions, instances and consumers [5]. Data pre-processing module is the producer for Kafka, and sends AIS data to server cluster continuously. The message subject is divided into static data and dynamic data. In order to realize the load balance of AIS data processing, AIS data is stored in different partitions. Each Kafka instance (broker) is responsible for processing the read and write requests of messages. Multiple brokers form a Kafka cluster to realize AIS data asynchronous communication, peak pressure buffering and decoupling between systems.

4. Data storage
In order to satisfy various applications, AIS data is divided into two categories. The first category is real-time information, which stores real-time information of ships and provides millisecond level query through Redis cache mechanism. The second type is historical trajectory information and historical region information of ships, which provides second level query.

4.1. Real-time information storage

![Figure 2 Writing process of AIS data](image-url)
Redis is a high-performance distributed memory database based on key value pairs, which is suitable for high concurrency and low latency application scenarios. By applying Redis system to AIS real-time data, the system performance is improved and the real-time requirements is satisfied. Real time information includes two parts by the MMSI and region information respectively.

The MMSI code is used as the index to store the basic information of the ship. Ship basic information comes from AIS dynamic data, AIS static data and ship basic information table. AIS data writing process is shown in Figure 2.

When a ship AIS data is received, the MMSI code exists in Redis cache is determined. If there is no record, a new record is added, and from the ship basic information table, ship name, registration number and other information is updated to this record. If there is the record, the record will be updated according to the data content.

The region information by decoded from GeoHash is used as the index to store the all MMSI codes in this region. GeoHash is an address coding method, which can encode two-dimensional spatial longitude and latitude data into a string [6]. The nearby target can be obtained by comparing the similarity of GeoHash values. The function of Redis Geo is based on geohash coding, and uses the ordered set (ZSet) to save the coordinates of geographical location. The ships in a certain region by the coordinates and radius can be quickly retrieved.

4.2. Historical information storage
HBase is a highly reliable, column oriented, scalable NoSQL distributed database. HBase uses Hadoop HDFS to store files and map/reduce to process data. HBase is based on column family storage, and the same column family are stored in a file. The row-key is the unique identifier of the HBase table and serves as the index of the key record. With reasonable row-key setting, HBase can query information in hundreds of millions of data in milliseconds. The historical information of ship is stored in HBase. By reasonably row-key and column families, it can make the massive AIS data is distributed orderly, and meet the needs of real-time query.

4.2.1. Historical trajectory storage.
The dynamic information obtained from message queue is regarded as reasonable and available AIS historical trajectory data. The dynamic AIS includes 16 items of information. MMSI code, speed, heading, longitude, latitude and time stamp are the information which are used in maritime management, and they are classified into the same column family. For navigation status, position accuracy, communication status and other information are not commonly used, and they are classified into the same column family. The historical trajectory query is mainly based on the MMSI code and date. Because most ship sailing on the Yangtze River Basin belongs China, and the MMSI code starts with 412 and 423. If the MMSI code is used as the row-key directly, the row-key cannot be evenly distributed. For AIS data can be allocated to different regions, the reverse MMSI code concatenate time stamp is used as the row-key.

\[
\text{RowKey} = \text{Reverse(MMSI)} + \text{TIME} \quad (1)
\]

where reverse (MMSI) represents the inverted string of MMSI code, and time represents the time stamp of message receiving (accurate to seconds).

4.2.2. Historical region storage.
In practical application, it is necessary to query the ship information of a certain region at a certain time in real-time. The information can be found directly through the historical trajectory table, but the whole table information needs to be scanned, and the time required is in hours, and it cannot meet the real-time requirements. for overcome this problem, the historical region information table is designed, and the region information and time information are stored as row keys to support spatiotemporal multidimensional data query.
RowKey=GeoHashBy1KM (longitude, latitude) + TIME  \hspace{1cm} (2)

Where GeoHashBy1KM (longitude, latitude) is a function which divides the Yangtze River Basin into equal regions of a kilometre. The longitude and latitude are encoded by GeoHash. The column family stores MMSI codes of all ships which appear in this region at one minute. After receiving AIS data, the historical region storage module calculates the region in which ship locate according to longitude and latitude. If this ship is not in Redis, this ship is added to Redis according the region information. In each minute, the Redis cache is synchronized to the HBase.

5. Data query

In order to provide high concurrent using of AIS data for all kinds of applications, a micro service cluster is constructed, and the service load balancing is used through Nginx service gateway. In the case of the above data storage design, it mainly provides the following queries:

The real-time data query includes two ways. One way is to query the basic information of the ship by MMSI code. The results include the location, navigation, ship name and other information of one ship. The other way is to query all the ship information by a position and a certain radius. The result is all ships in the certain radius.

Historical data query is divided into two ways. One way is to query the historical trajectory of a ship by the MMSI and time interval. By startRow and endRow functions, the range is limited. The historical trajectory of a ship in a certain time interval can be quickly obtained. The other way is to query all ships in a certain region and time. By limiting the range of row-key, the information of all ships is obtained in one second.

6. Conclusion

AIS data is of great significance to the realization of intelligent shipping. A storage and query scheme based on Hadoop platform for massive AIS data collected from shore based is proposed. By Kafka message queue, AIS data is forwarded efficiently and stable. Redis provides real-time data query, which can achieve millisecond level query. The real-time query of historical information is realized by HBase, which can realize second level query and provide support for data analysis.

Acknowledgments

This work was financially supported by the funding of China postdoctoral science foundation (2019M651844), key project of natural science research of universities and colleges in Jiangsu Province(20KJA520009), postdoctoral science foundation of Jiangsu Province (2018K035C), six talent peaks project in Jiangsu Province (XYDXX-149), young academic leaders of QingLan project of college and university in Jiangsu Province, and the shipping big data collaborative innovation centre of Jiangsu Maritime Institute.

References

[1] Zhang, Y., et al. "The Significance of Incorporating Unidentified Vessels into AIS-based Ship Emission Inventory." Atmospheric environment 203. APR (2019):102-113.
[2] Liu, Z., Z. Wu, and Z. Zheng. "Modelling Ship Density Using a Molecular Dynamics Approach." Journal of Navigation 73.3(2019):1-18.
[3] Qin, J. L. Ma, and Q. Liu. "A Distributed Framework for Trajectory Similarity Query Based on HBase and Redis ". Information, 10.2(2019): 1-24.
[4] Ma, T., et al. "MHB: A Distributed Real-Time Query Scheme for Meteorological Data Based on HBase." Future Internet 8.1(2016):1-14.
[5] Du, Y., et al. "A Distributed Message Delivery Infrastructure for Connected Vehicle Technology Applications." IEEE Transactions on Intelligent Transportation Systems PP.99(2017):1-15.
[6] Guo, N., et al. "A Geographic Meshing and Coding Method Based on Adaptive Hilbert-Geohash." IEEE Access 7(2019): 99428-99440.