Analysis and Research on Big Data Query Technology Based on NOSQL Database

Shuiquan Guo

Wuhan City Polytechnic, Wuhan, China

Keywords: NOSQL database, big data, query technology, big data query

Abstract: Based on the application of NOSQL, the classification of NOSQL database and its big data query technology, as well as the big data operation in the comprehensive analysis system of real-name railway ticketing information, this paper applied big data query technology based on NOSQL database.

1. Introduction

With the rapid development of science and technology, mankind is generating a large amount of various types of data at an unprecedented rate. Especially with the cloud computing industry landing, big data has also attracted the attention of the whole society, has become a well-known concept for a people. Go to the national government agencies, research institutions, technology companies, under the football commentator, tabloid reporter, are talking about big data. This shows that big data is not only a data science community problem, but also all human problems, all walks of life are profoundly understand the opportunities and challenges brought by big data.

NOSQL database using big data query technology can make the data search time to a great extent, and can make the data read and write efficiency and horizontal scalability greatly improved for all areas of the query provides the technical basis for the following Combining with the big data operation in the comprehensive analysis system of real-name ticket information of railway passenger ticket.

2. NOSQL concept

The definition of NOSQL given on NOSQL-databases' official web site is: Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source and horizontally scalable.

NOSQL considers it shorthand for "Not Only SQL," and today NOSQL refers broadly to such a class of databases and data stores that do not follow the classic RDBMS principles and are often associated with Web-scale big data sets. That is, NOSQL does not simply mean a product or a technology that represents a family of products, as well as a series of different, sometimes interrelated, concepts about data storage and processing. Its significance lies in: The relational database is used when using the relational database, when not applicable, it is not necessary to use a relational database, and more suitable data storage can be considered [1].

NOSQL non-relational database technology uses loosely coupled data patterns, supports horizontal scaling, data persistence on disk and / or in-memory, and supports multiple "Non-SQL" interfaces for data access. NOSQL's data model includes Key-Value pairs, document-oriented storage, columnar storage, and graph structure storage. NOSQL supports complex queries, weak transaction mechanisms, support for redundant backup to ensure stand-alone reliability, a variety of data synchronization to achieve multi-machine reliability, support for hash partitioning and range partitioning for distributed expansion, emphasizing the final consistency [1].

3. NOSQL related theory

3.1 CAP theory

The theory of CAP was first proposed by Professor Eric Brewer and later proved by Seth Gilbert and Nancy Lynch. CAP theory first of all three characteristics of distributed systems were summarized as follows [2]:

Consistency: All data in the system is backed up and is the same value at the same time.

Availability: Each operation always returns within a certain amount of time, meaning the system is always available.

Tolerance to network Partitions: In the case of a network partition, such as disconnecting a network, a separate system can operate normally.

CAP theory points out that it is not possible for a distributed system to achieve all three of these characteristics at the same time. The system must make the trade-off, at least at the expense of the other two. Fig.1 focuses on different database products.

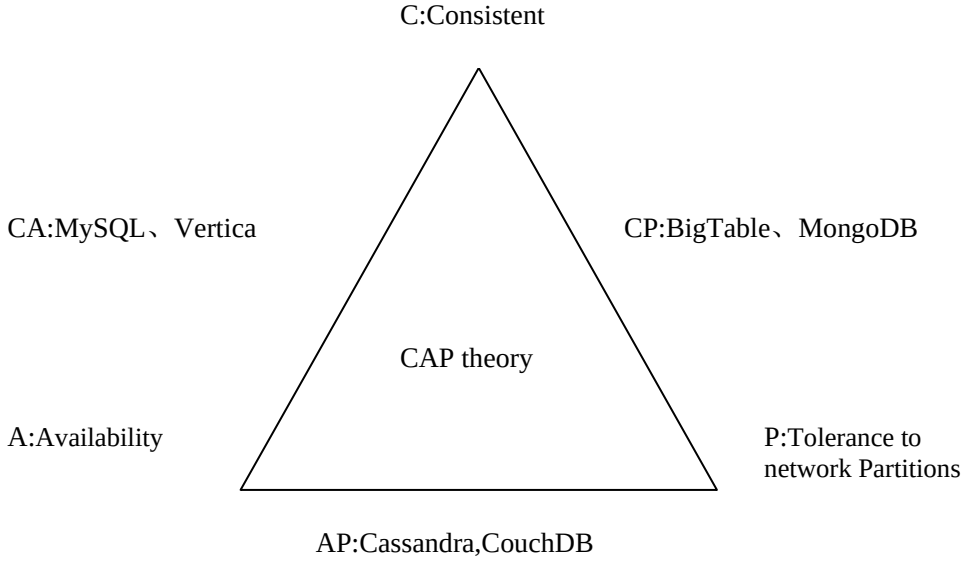

Fig.1 CAP theory

3.2 Final consistency

There are usually two levels of consistency in NOSQL: the first is strong consistency, that is, all machine states in a cluster are in sync; the second is final consistency, which allows short-lived data to be inconsistent, but the data eventually is consistent. According to the CAP theory, strong consistency cannot be achieved simultaneously with availability and partition tolerance. The final consistency is the compromise solution to consider the user experience, but also the biggest difference with the traditional RDBMS [3].

3.3 BASE thinking

BASE theory is based on CAP theory, which is actually an extension of AP in CAP theory. It is a generalization of consistency. This is because in more and more applications and practical cases, usability and zoning tolerance are considered to be more consistent Sex needs more rigorous design. More of these application designs tend to reduce conformance, while emphasizing availability and data redundancy mechanisms (i.e., the orderly dissemination of data across disparate nodes) [3].

3.4 Distributed Systems

Distributed System is a software system based on the network with a high degree of cohesion and transparency. Cohesion refers to the highly autonomous node of each database distribution, a local database management system. Transparency means that each database distribution node is

transparent to the user's application, cannot see is the local or remote. In a distributed database system, the user does not feel the data is distributed, that is, the user does not need to know whether the relationship is split, whether there is a copy, where the data resides, and at which site the transaction is performed [2].

4. NOSQL data model

A data model is a model that defines how data is input and output. Its main role is to provide the information system definition and format of the data. The data model is the core and foundation of the database system. The existing database system is based on some data model.

4.1 Key-value-based data model

Key-Value storage is the simplest NOSQL store that will map "keys" to the appropriate "values" (data) in an algorithm, regardless of the content of the data. It is an unstructured data storage model. Application developers need to organize and define their own "value" data format and resolve. The key-value storage system does not support any non-"key" queries [4]. Dynamo is a typical key-value storage system.

The main application scenarios: content cache, mainly for processing large amounts of data high access load, but also for some log systems.

4.2 Document-oriented data model

Representatives of document-oriented storage systems are CouchDB and MongoDB. Document storage data generally json or similar json format, the storage content is document type. MongoDB's document data is stored in bson format, CouchDB's document data is stored in json format, and documents can store lists, key-value pairs, and hierarchically complex documents. The flexibility and complexity of document-based storage is a double-edged sword: on the one hand, developers can organize document structures at will; on the other hand, query requirements at the application level can become more complex.

4.3 Column-oriented data model

The data model is characterized by columnar storage, that is, each row of data is stored in different columns, the collection of these columns called cluster, it can read a large number of rows and updates. Representative system has BigTable and HBase and so on. BigTable is Google's distributed storage system designed to efficiently manage massive, large-scale structured data that can be used to process petabytes of mass data and distributed across thousands of common servers [4]. Both HBase and Cassandra's data models draw on Google's BigTable.

4.4 Figure structural data model

Figure Structure Storage is another form of storage for NOSQL. Diagram-based graph databases use the concepts of nodes, attributes, and edges. Nodes are similar to the concept of objects in object-oriented programming and represent entities similar to people, businesses, accounts, or whatever [1]. Properties store node-related information, such as using Wikipedia as a node, then its properties can be web pages, reference materials, or words beginning with W, whichever one you choose depends on the actual application. Edge is used to connect nodes and nodes or nodes and attributes, indicating the relationship between the two, the most important information stored in the edge. Neo4J and HyperGraphDB are currently the most popular graph structure databases.

5. NOSQL application advantages

5.1 Flexible data model

In the traditional RDBMS area, you must analyze the data and build the data model before storing it. Build the data model is to create a data table to determine the various fields in the table, the field's data type and the relationship between the fields between the tables. However, in real business,

demand is constantly changing over time, and while some degree of refactoring is supported in traditional relational databases, refactoring cannot do much if applications change too much. The common thinking of NOSQL databases is to break the limits of this data model [5]. NOSQL database allows applications to store any structure they want in a data unit, the data unit is generally no model restrictions, and the link between them is flat. But pay attention to the integrity of data management while paying attention to the freedom of the mode.

5.2 High performance at low cost

NOSQL's simple data model makes it extremely scalable and easy to scale nodes, and due to its distributed architecture, its design philosophy is based on low-cost, volatile machines, Cost for high performance. NOSQL databases often use inexpensive server clusters to manage the proliferation of data and transactions [5]. Cheap server clustering solutions, clustering with RDBMS on a relatively high-performance machine, have more data nodes and provide cheaper, more reliable, more backed-up services.

5.3 Easy to expand

As the load on the database grows, RDBMSs typically scale to scale up performance, which means buying better-performing servers instead of legacy servers is more efficient for slow load increases. However, the load may have been increasing, it is impossible to replace the server every time [2]. In this way, the idea of scale-out is proposed, load balancing the different host. Although the RDBMS provides a scale-out feature, it is translucent to the program and can be extensively modified and shut down even when scale-out. NOSQL database in the design of the beginning to consider the horizontal expansion, it is transparent to the program, you can add nodes at any time, delete nodes.

6. Application of NOSQL Database Big Data Query Technology

NOSQL is the general term of non-relationship database to meet the growing application requirements of Internet. When traditional relational databases face a variety of pressures and challenges, such as mass data storage and management, high availability and high scalability, NOSQL becomes a very popular field of study, and obtain attention and many research achievements. NOSQL database breaks the traditional relational model, stores data as a free style instead of dependent of fixed table structure, and overcomes the shortcomings of traditional relational database, so as to have great performance. NOSQL database are designed to be deployed on inexpensive hardware to support distributed storage and to transparent extension node, their availability are powerful and they can maintain in low cost.

6.1 System Technology Architecture

As a new type of database, NOSQL is not intended to replace the traditional relational database. It is based on the characteristics of applications and combines with the relational database to make up for the deficiencies of the traditional relational database. This is a coexistence of opportunities and challenges field [3, 5]. I believe in the near future, major software vendors will provide better technical support NOSQL based on the actual application requirements, the theoretical research results will be more abundant. To use the Java design pattern of the railway passenger ticket real-name information system analysis of the technical architecture of the design, the technology architecture has four functions:

Data layer. Ticket information and real-name identity information and other data through the system workflow process to schedule, to use the open source ETL tool Kettle on the data regularly extracted and converted into the database for use by the service layer data.

Service layer. Service layer is the use of Java technology, combined with the workflow mechanism, the data layer data calls for the application layer to provide some related queries, statistical analysis and other services, this function can be achieved business applications.

Application layer. Interface functions for the service layer, the real-name information management, identification and push release.

Show layer. Is used to establish a user interface in the browser, you can provide navigation for application systems and management systems, and you can query a series of information above.

6.2 Query strategy and process

Some of these NOSQL databases are written in C / C ++, some in Java, others in Erlangen, each with its own uniqueness. Query process is:

Query strategy. From the real-name ticketing business data and real-name passenger identification information to find the corresponding key information, and then the associated operation can be found want to find the real-name car information. This strategy according to the conditions of the query, the list of real-name ticketing business data and real-name passenger information reverse index.

Query process. According to the established system architecture, the systems use the query strategy to reverse-index and design the process of data query.

On the loading information to judge, to see whether it is the ticket stub, if so, this information will be transferred to the car information base. If not, we must make judgments to see whether it is a waste ticket, a change of sign or a refund stub, which kind of implementation of which kind of process.

The reverse index passenger information and identity information is updated for use in the query process.

Submit real-name system car query request.

The request split, divided into car information and identity information, the reverse index.

The requested inspection results to operate, query records, and then return the query results.

7. Summary

To sum up, combining the information query of real-name railway ticket system to analyze the technology structure of NOSQL database big data query technology, the reverse index design is used to realize the real-name system related information query process and provide related steps, all this fully shows that NOSQL data has become an effective way to query and analyze big data.

Acknowledgement

Subject source: the topic of this paper is original from one of key projects of Hubei Planning Office, Study on Effective Approaches of Students' Education and Management Based on Data of Smart Campus (2017GA086), and the project of Wuhan Program Office, Analysis of Students' Behavior and Decision Support Based on Data of Campus Card (2016C251).

References

[1] CCh.Ch. Chen, NOSQL-based cloud storage and service of mass data, Journal of the Earth Science, 2013, vol.2, pp.32-35.

[2] R.H. Zhang and Sh.M. Hu, CODATA China Physical and Chemical Database, Important Science and Technology Achievements during the Tenth Five-Year Plan" and Exchanges of Significant Prospecting Results, 2006, pp.89-92.

[3] X.P. Wang and F.Ch. Wei, Maintenance and Optimization of AFC System Database for Changchun Rail Transit, Proceedings of 2012 China (Changchun) International Rail Transit Forum, 2012, pp.118-122.

[4] J.J. Cai, and X.X. Xin, Discussion on Financial and Economic Data Warehouse and Service System Architecture Based on Business Metadata Standardization, 2012 Annual Conference of China Journalists and Workers in Technology, 2012, pp. 155-157.

[5] Y.X. Zhang and J.L. Feng, NOSQL-based file-based data storage technology research, Manufacturing Automation, 2014, vol.6, pp.43-46.