Advances in Social Science, Education and Humanities Research, volume 176

2nd International Conference on Management, Education and Social Science (ICMESS 2018)

Research on NoSQL Database Technology

LI Jun-shan, LI Jian-jun

The institute of information science & technology, South China Business College Guangdong University of Foreign Studies, Guangzhou 510545, China

***Abstract*—Due to the demand for ultra-large-scale and high-concurrency purely dynamic social networking websites and big data management, traditional relational databases can no longer meet the storage and access requirements of massive data, so NoSQL database systems for specific applications have emerged. First, this paper introduces the background of the emergence of NoSQL non-relational databases. Second, it also introduces the concept and overall architecture of the NoSQL database. Then, the notable features of NoSQL databases relative to relational databases are given. Next, the classification of NoSQL databases by storage type is described. Finally, the development status and prospect of NoSQL database technology are given. It is expected that the research in this paper can further promote the research and application of non-relational database technology.**

Keywords—NoSQL; Non-relational databases; Massive data management; Data storage; Data logical model

I. INTRODUCTION

Since the commercialization of a large number of commercially available relational database management systems (RDBMS) in the 1980s and has been widely used, the relational database management system has not only become a mainstream database product, but also has applications in various fields of the national economy and data management. The field has an absolutely dominant position. However, with the rapid development and wide application of Internet technology, traditional relational databases have exposed many insurmountable problems when dealing with hyperscale and high concurrent and pure dynamic social networking website (SNS) type web2.0 [1]. In order to solve the challenges brought by large-scale data sets and multiple data types, especially the big data application problems, non-relational data storage system NoSQL emerged.

The full name of NoSQL is Not Only SQL, which means "not just SQL" or "structured queries," but narrowly refers to non-relational databases, and broadly refers to "non-relational data stores." There are also literatures interpreting their meaning as "When using a relational database, relational databases are used. When not applicable, it is not necessary to use non-relational databases. Instead, consider using more appropriate data storage [2]." Obviously, NoSQL is a data storage system that is different from a relational database (generally, NoSQL is also called a non-relational database system corresponding to a relational database). This is the case for relational database applications that currently cover almost all fields. For example, NoSQL's non-relational database concept is undoubtedly a new kind of thinking injection and an entirely new database technology revolution.

II. THE PRODUCTION OF NOSQL DATABASE

In the early days of the Internet application, the traffic of a website is generally not large. The pages of a website are more static webpages. That is, websites with dynamic interaction types have limited dynamic functions. Therefore, a single relational database can be easily handled. With the rapid development of the website, popular forums, blogs, sns, and microblogs are gradually leading the trend in the web domain. Especially after the emergence of many dynamic websites with strong functions, although RDBMS can tolerate a certain degree of irregularities and structural lack of data, however, in the face of mass sparse data of loose structure, RDBMS seems to be awkward. At the same time, starting from Inktomi, which can be regarded as the first search engine, to Google later, the widely used relational database management system leaked a series of problems of its own when applied to massive data [3]. To this end, Google has built a massively scalable infrastructure to support Google's search engines and other applications (including Google Maps, Google Earth, Gmail, Google Finance, and Google Apps), establishing a scalable infrastructure for parallelism. Handle massive data. With the release of Google related technologies and solutions, the inventor of open source search engine Lucene developed the first open source software that mimics some of the characteristics of Google's infrastructure, followed by Lucene's core developers who joined Yahoo, relying on many open source contributors The support created an open source Hadoop product and its subprojects and related projects that can replace all parts of Google's infrastructure.

In fact, the term NoSQL and its ideas came before the first release of Hadoop. NoSQL first came from the name of a small open source relational database developed by Carlo Strozzi in 1998 because the database stores all data as an ASCII file and uses shell scripts instead of SQL to access data, hence the name Nosql[4]. At a conference on distributed open source databases, data storage and processing topics held in San Francisco in June 2009, Eric Evans from Rackspace once again raised the concept of NoSQL [5]. In fact, at the conference, the term NoSQL originally did not have a deeper meaning, but the result was that it quickly spread across the Internet and became a new trend in the IT field. The reason is that Google's success has helped the public accept the concept of distributed computing in the new era, spurred people's interest in parallel large-scale processing and distributed non-relational data storage, and the emergence of Hadoop laid the foundation for the rapid development of NoSQL solid foundation. Further, NoSQL was supported by two leading Web giants Google and Amazon, and

accordingly the two most important new systems emerged in the field: The first is Google's distributed, column-oriented, multidimensional, Sparse, multi-version table system – BigTable [6]; The system will be a large table of data according to the value of the row key segmentation, and distributed to multiple servers, is a strong consistency system. The second is Dynamo[7], Amazon's distributed storage system based on the P2P architecture; this system distributes data with consistent hashing algorithms, has better availability and failure recovery capabilities, and relatively poor consistency, is a The ultimate consistency of the system. Since then, a large number of developers have started to use, clone, or mix the two products in their own applications, and have emerged many different implementations. In less than five years, NoSQL and the concept of managing big data have been widely disseminated. Numerous well-known companies have begun to use a variety of use cases including Facebook, Netflix, Yahoo, EBay, and Hulu, many of which are the company also contributed their own expansion components and new products to the world through open source. With the further development and extensive application of web2.0, the development and application of NoSql database technology have been promoted.

III. NoSQL Database Concept

NoSQL non-relational database currently does not have a recognized authoritative definition. Sourav Mazumder, chief technology architect at InfoSys Technologies, gave a more comprehensive description [8]:

(1) Logically model data using an extensible, loosely-coupled data pattern.

(2) Designed to follow the consistency-availability-partition tolerance (CAP) theorem across multi-node data distribution models, supporting horizontal scaling.

(3) Have data persistence capabilities in disk and/or memory.

(4) Support multiple "Non-SQL" interfaces for data access.

Based on the above description and other related literature, NoSQL has unique connotations at least in the following aspects relative to traditional relational databases.

(1) NoSQL uses a loose type, extensible data model; the data model does not have a strict definition, does not require the data model to be determined before the data is stored, and the data model can be dynamically changed during system operation, so it is very beneficial to Store most of the semi-structured and unstructured data in web applications.

(2) NoSQL uses multi-node data distribution model to distribute records on multiple nodes through data partitioning. It can achieve horizontal scaling, supports horizontal expansion, and can adapt to the rapid growth of Web applications, and it has a large number of data in a distributed architecture. Better performance.

(3) The NoSQL database no longer supports the ACID features (for example, Atomicity, Consistency, Isolation, Durability) of the firms in the traditional relational database management system that have been formed for a long time, and does not require transaction management.

(4) Have persistent storage of data on disk or in memory, or both.

(5) Does not support JOIN operations, supports large-scale data processing, most of the technologies are open source.

For example, in a NoSQL database that stores data in the form of key-value pairs, the data structure is not fixed, and each tuple can have different fields. Each tuple can add some of its own Key-value pair. For another example, in a NoSQL database stored as a document, an application is allowed to store data of any structure in one data element; since it is not limited to a fixed storage structure, some unnecessary time and space overhead can be reduced; Data storage does not require a fixed table structure, there is no table connection operation. For another example, Sina’s Weibo system supports distributed data access by a large number of users through distributed computing over 400 servers. All of the above shows that NoSQL has the performance advantages unmatched by relational databases in big data access.

In addition, the concept of NoSQL can be further understood through the NoSQL overall architecture (see Fig. 1) given by Sourav Mazumder [8]. Sourav Mazumder divides the NoSQL database into four layers:

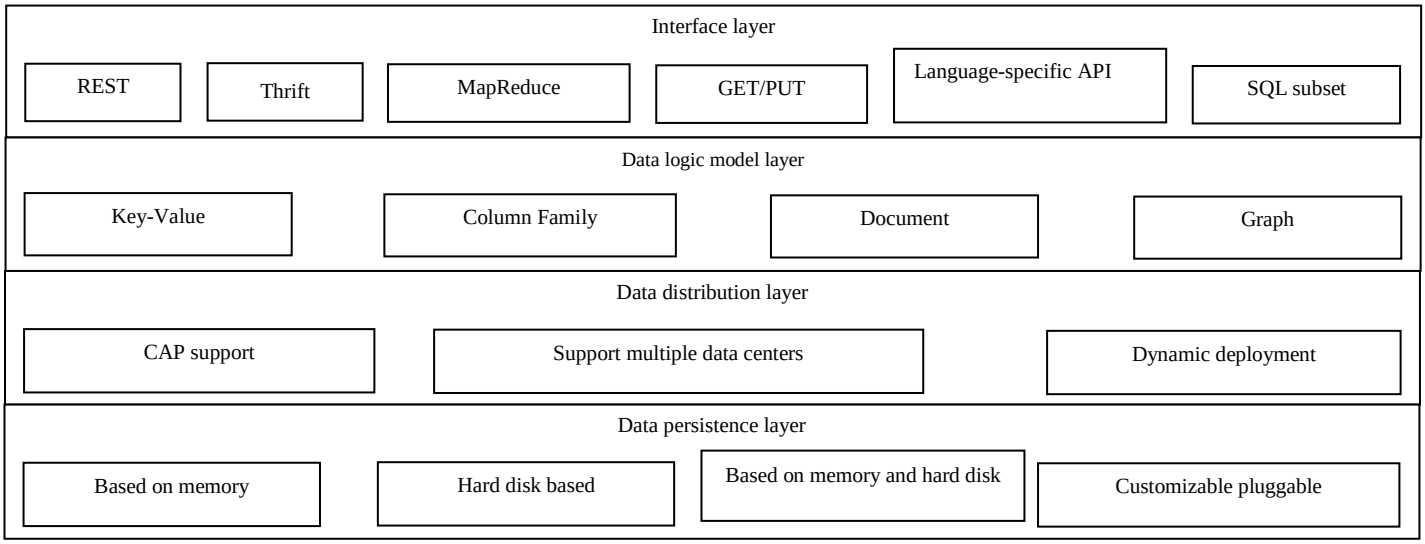

Fig. 1. The overall architecture of NoSQL

(1) Interface layer. Used to provide reasonable and convenient programming languages and data call interfaces for upper applications, mainly including REST (Representational State Transfer), RPC protocol Thrift from Facebook, Map Reduce for large-scale data processing, and Get/similar to Memcached. Put mode, language-specific APIs, etc.

(2) The data logic model layer. Used to describe the logical representation of data in a database, including key-value storage, column-cluster storage, document storage, and graph-structure storage.

(3) Data distribution layer. It is used to define the distribution of data, including the CAP mechanism for horizontal expansion, the multi-data center support mechanism for ensuring the smooth running of NoSQL databases across multiple data centers, and the dynamic deployment support mechanism.

(4) Data persistence layer. Used to define the form of data storage, including memory-based, hard disk-based, both memory and hard disk-based persistent storage, as well as custom pluggable persistence.

IV. NoSQL Database Features

Compared to relational databases, NoSQL databases have the following salient features:

A. Easy data dispersion

Relational databases are premised on JOIN operations. The existence of associations between data is implemented using JOIN. For JOIN processing, relational databases have to store data in the same server, which is obviously not conducive to the dispersion of data. Unlike relational databases, NoSQL databases originally do not support JOIN processing, and each data is independently designed so that it is easy to spread the data across multiple servers [2]. Since the data is distributed across multiple servers, the amount of data on each server is reduced, making it easy to write and read large amounts of data, satisfying the very large-scale and high-concurrency SNS type of web 2.0 pure Dynamic social networking sites require high concurrent reads and writes in the database.

B. No sharing operation

One of the reasons for the NoSQL database is to "make it easier to write large amounts of data." Starting from the implementation technology that enables the server to easily handle larger amounts of data, it is clear that only the performance-enhancing or scale-up options are available. A direct solution to increase processing power by improving the performance of the current server itself is to purchase a server that doubles in performance without changing the program, but generally requires up to 5 to 10 times more investment. The scale-up plan means that more inexpensive servers can be used to increase the processing capacity. Although it needs to make changes to the program, the use of inexpensive servers can control the cost, and can further increase the number of inexpensive servers to increase processing capacity as needed. Avoid the complexity and high cost of traditional commercial database sharing operations.

C. Flexible expansion

In the past, when the load of relational databases needed to increase, the relational database management system could not easily scale-out on commercial cluster machines (that is, by connecting multiple low-cost servers together. With increased load, data administrators always maximize the use of resources by scaling-up (that is, purchasing larger and more powerful servers to carry the increased load). However, as large data analysis requires the use of a large amount of computing power to handle the target data set, the high scalability and availability requirements of the database, and the need to migrate the database to the cloud or virtual environment, the design of the new NoSQL database can be used low. The cost of commercial hardware transparently scales out with new nodes [9]. In other words, the NoSQL database system is composed of databases that are distributed on different nodes to form a storage system.

You can dynamically add (or delete) nodes without downtime maintenance, and data can be automatically migrated.

D. Flexible data model

Change management is a difficult task for large relational DBMS products. Even minor changes to the relational DBMS data model may require system downtime or service degradation. The NoSQL database is loose in data model constraints. Its key-to-store and document database allows applications to store any structure of data in a data element. Even a relatively strict BigTable-based NoSQL database (eg, Cassandra, HBase) is usually not too restrictive when creating new columns. Therefore, in a NoSQL database, changes to the application or database schema do not need to be managed as a complex change unit; in theory, applications can be allowed to iterate faster.

E. Asynchronous replication

Early relational DBMSs run on a single CPU, and read and write operations are performed by a single database instance. NoSQL replication technology allows database read and write operations can be dispersed on separate servers running on different CPUs. That is, NoSQL database replication refers to the one-way information propagation behavior that occurs between different database instances. The copying party and the copying party form a network connection between the copying party and the copying party [10]. The copying method is usually the copying party actively sends the data to the copying party, and the copying party stores the received data in the current instance, so that the data is essentially backed up on different instances, the main library focuses on the write request, and reads from the library. Requests to improve the system's query service capabilities with high availability, high read performance, and horizontal expansion.

Replication in NoSQL is often log-based asynchronous replication so that data can be written to a node as quickly as possible without network delays. The disadvantage of asynchronous replication is that the write data sent by the master server is not necessarily received from the server. This may not always ensure the consistency of master-slave server data because there may be data loss.

V. CLASSIFICATION OF NOSQL DATABASES

According to different storage types of databases, NoSQL databases are divided into key-value pair storage databases, column-type storage databases, document storage databases, and graphical storage databases [5].

A. Key-value Store Database

The key-value store database is usually implemented as a hash table with a specific key and a pointer to specific value. Therefore, a key-value pair storage database is a NoSQL database that organizes and stores data in the form of key-value pairs, and queries the data by a completely consistent query of keys [11-12]. Key-value storage does not need to consider the storage format of data, and directly uses the key value to quickly query the required data. It is very suitable for data that does not involve too many data relationships and business relationships. It can effectively reduce the number of read-write disks and has extremely high reading and writing performance. When key-value pairs store database key-value pairs to save and read data values, system efficiency is very high because it does not have many limitations such as SQL processor, indexing system, and analysis system. The key-value storage scheme not only provides efficient access performance, but also has low implementation cost and scalability. The ability to satisfy extremely high read/write performance is the most significant feature of key-value pairs for storing NoSQL databases.

The key-value pair storage is divided into three storage modes: temporary key-value pair storage, permanent key-value pair storage, and both having a key-value pair storage according to different data storage modes.

(1) Temporary key-value pairs are stored. The so-called temporary nature is that "data may be lost." Memcached is a temporary key-value pair stored NoSQL database. Memcached keeps all the data in memory, which saves and reads very quickly, but when Memcached stops, the data does not exist. Since the data is stored in memory, data beyond the memory capacity cannot be operated. A typical feature of temporary key-value storage is that the data in memory is stored and read very quickly, and data may be lost.

(2) Permanent key-value pair storage. In contrast to the temporary, the so-called permanent is "data will not be lost." Tokyo Tyrant belongs to this type of NoSQL database. Unlike the temporary, a permanent key does not store data in memory like Memcached, but instead stores the data on a hard disk. Since the IO operation of the hard disk must occur when data is saved to the hard disk, there is a gap between the performance and Memcached, but the data is not lost is its greatest advantage. The typical characteristic of permanent key-value storage is to save data on the hard disk, save and read processing speed is very fast, data will not be lost.

(3) Both have a key-value store. That is, both temporary and permanent key-value pairs, Redis belongs to this type of NoSQL database. Redis combines the advantages of temporary key-value pairs with storage and permanent key-value pairs. Redis first saves the data into memory and writes the data to the hard disk when certain conditions are met. This not only

ensures the speed of data processing in the memory, but also guarantees the data's permanence by writing to the hard disk. This type of database is particularly suitable for processing array-type data. Both temporary and permanent key-value pairs store typical data at the same time in the memory and hard disk, save and read very fast, data stored on the hard disk will not disappear, suitable for processing array type data .

B. Columnar Storage Database

A columnar storage database is a NoSQL database that stores data in the same column and then stores the next column of data, storing, retrieving, and controlling permissions in units of column clusters (each column belongs to a cluster of columns). It facilitates the storage of structured and semi-structured data for data compression [13]. Physically speaking, a table is a collection of columns. Each column is essentially a table with only one field. Therefore, there is a very large I/O advantage for a column or columns of queries. Columnar storage database is highly scalable, even if the data does not reduce the corresponding processing speed (especially the writing speed), columnar storage database is usually used for batch data processing, ad hoc query and business intelligence and analysis type the storage of data. Cassandra is a column-stored NoSQL database. Data storage does not require a fixed table structure and there is no restriction on the columns between each record is the most typical characteristic of the columnar storage NoSQL database.

C. Document Storage Database

A document storage database is also referred to simply as a document database, and is a type of NoSQL database stored in a key-value pair, and is a document data (a semi-structured data stored in a specific form) that is not mandatory. The document database is mainly oriented to use the storage engine's ability to divide different documents into different collections of storage. A document is equivalent to a record in a relational database. Multiple documents form a collection, and multiple collections are logically organized together as a document database. Unlike key-value storage, document storage is concerned with the internal structure of the document, which allows the storage engine to directly support secondary indexes, allowing efficient querying of any field. The document storage model supports nested storage capabilities, which means that the "values" of the fields can be nested to store other documents. The most significant feature of document-stored NoSQL databases is their ability to meet massive storage requirements and high query performance. MongoDB is a NoSQL database for document storage.

D. Graphical Storage Database

A graph storage database is also referred to as a graph database for short, and is a graph-based structure that represents and stores graph data NoSQL databases through nodes, edges, and attributes. In a graph store database, each element contains a pointer to an adjoining element directly, and is a storage system that adjoins each other without an index. Graphical storage databases are the best storage for graphical relationships and can be naturally extended to larger datasets without the need for concatenation operators, and have faster speeds for querying associated datasets. Graphical databases can be used to model things and their relationships, such as relational graphs, social networks, and recommendation systems. AllegroGraph belongs to graph storage NoSQL database.

Finally, it should be noted that although there are more than 100 kinds of NoSQL databases, they do not have a unified architecture. Different NoSQL have their own strengths. At present, successful NoSQL must be particularly suitable for certain applications or occasions, and its performance must be far better than relational databases and other NoSQL databases.

VI. NoSQL Database Development Status and Prospects

Under current technology conditions, computer architecture requires a large level of scalability in data storage, and NoSQL is working to change this. At present, Google, Yahoo, Facebook, Twitter, and Amazon all apply a large number of NoSQL databases [14]. In many areas, NoSQL has achieved success not only in the industry but also in academic fields. The university began to realize that the standard relational database alone is no longer enough, and it is necessary to add NoSQL to the curriculum; from a technical point of view, NoSQL is a very important supplement to relational databases.

In the short years since the NoSQL concept was introduced in 2009, NoSQL-type databases have exploded to produce more than 100 new databases. With the development of technology and the popularity of applications, some interesting mergers are taking place, such as CouchBase generated by CouchDB and Membase transactions. These are synchronized with the explosive growth of the Internet, big data, sensors, and many technologies in the future. This has also led to more data and different needs for their processing.

For most of the NoSQL database systems that have been deployed today, there are many challenging issues that need to be addressed. In terms of generality, existing NoSQL database products are mostly application-specific solutions, resulting in their application has certain limitations, lack of global system considerations and versatility, and limited functionality. In terms of technology and theoretical maturity, no series of technical achievements have been formed, and there are no strong theories (such as relational computing theory, function dependency theory, Armstrong axiom systems, relational pattern normalization methods, etc.) and technologies that are similar to relational databases (such as query optimization strategies, two-stage blocking protocols, etc.), standard specifications (such as the SQL language), and built-in security mechanisms. In terms of system performance, the maturity,

stability, and functionality of RDBMS can be reassuring; in comparison, most NoSQL databases still have many features to be implemented. In terms of technical support, all RDBMS vendors have spared no effort to provide good corporate support. In contrast, most NoSQL systems are open source projects. Although each database has several companies to provide support, these companies are mostly small. The start-up companies have no global support resources, and there is no reassuring public trust like Oracle and Microsoft. In terms of management support, NoSQL's design goal is to provide zero-management solutions, but today's reality is still far from this goal. In terms of professional support, many global business units will have people who are familiar with RDBMS concepts and programming; in contrast, almost every NoSQL developer is in a learning mode, although this situation will change with the passage of time. It is not easy to find a NoSQL programmer with rich experience now. In terms of its unique open source advantages, NoSQL now requires a lot of skills to use it, and it requires a lot of human and material resources to maintain it. In terms of applications, NoSQL is difficult to achieve data integrity, and data integrity is essential in enterprise applications, so the current NoSQL project is difficult to popularize in the enterprise. In addition, it takes a long process from the emergence of the NoSQL database to acceptance by various users. In summary, NoSQL has great room for improvement and development, both from the technical level and from the application level.

VII. CONCLUSION

The emergence of big data has promoted the development of NoSQL database technology. NoSQL database has created an environment for the storage, transmission and processing of big data, which further promotes the application of NoSQL database. With the development of big data processing, cloud computing, Internet and other technologies, as well as the emergence of new applications in many cloud environments such as social networking, mobile services, and collaborative editing, new demands are placed on massive data management systems. With the expansion of scalability, flexibility, fault-tolerance, self-management, and "strong consistency", the design goals of the massive data management system in the era of cloud computing have provided a good opportunity for the NoSQL database. With the further increase of demand and the passage of time, NoSQL database system will gradually mature and gain wider application.

ACKNOWLEDGMENT

Supported by Teaching Quality Improvement Project and Teaching Reform Project for Guangdong Undergraduate Universities (No.296); Educational Innovation Project of Educational Department of Guangdong (Education Research) (No.2017GXJK243)

REFERENCES

[1] YAO L, ZHANG Y K. Solution of No SQL Distributed Storage and Extension [J]. Computer Engineering, 2012，38 [6]:40-42. (In Chinese)

[2] Basic knowledge of nosql database. http://www.ituring.com.cn/article/1069

[3] Shashank T. Professional NoSQL[M]. Birmingham: Wrox, 2011.

[4] Lith, Adam, Jakob M. Investigating storage solutions for large data:A comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data [D]. oteborg Sweden:Chalmers University of Technology，2010.

[5] SHEN D R,YU G,WANG X T,et al. Survey on No SQL for Management of Big Data[J]. Journal of Software, 2013, 24(8):1786-1803. (In Chinese)

[6] Fay C, Jeffery D, Sanjay G, et al. Bigtable: A Distributed Storage System for Structured Data [A]. //7th Symposium on Operating System Design and Implementation[C]. Seattle, WA, USA: 2006.

[7] Giuseppe D, Deniz H, Madan J, et al. Dynamo: Amazon's Highly Available Key-value Store [A]. SOSP'07[C]. Stevenson, Washington, USA: 2007.

[8] Sourav M. NoSQL in the Enterprise [J]. Architect, 2010, (8):62-64.

[9] You have to know about NoSQL database 10 key characteristics. http://www.xue163.com/exploit/184/1842623.html. (In Chinese)

[10] Redis drill (6) redis replication (Active-backup). http://www.itnose.net/detail/6637390.html. (In Chinese)

[11] Emmanuel G. Implementation of key-value pairs store(1): Why is the key value of storage, Why do you want to achieve it. http://. www. codecapsule.com

[12] Emmanuel G. Implementation of key-value pairs stored (2): To the existing key value is stored as a model. http://. www. codecapsule.com

[13] Data analysis tool—— Column type storage database. http://blog.csdn.net/physicsdandan/article/details/51988172. (In Chinese)

[14] Get the NoSQL database——Domestic application case inventory. http://www.dataguru.cn/thread-42932-1-1.html. (In Chinese)