Data Science CW2

Yuan Lou
Electronics and Computer Science
University of Southampton
United Kingdom
yl2m17@soton.ac.uk

ABSTRACT

Traditional databases handle big data is not appropriate. Some tools are different for big data processing. This article compares some of the storage methods in the world today, including relational database non-relational databases. Introduced some features and applications of relational databases. After comparing the performance of different databases, some conclusions were obtained to process ENRON data. MongoDB is the best e to process this data. Finally, the concepts and features of distributed storage and parallel storage are introduced.

Keywords

big data, relational, non-relational, MongoDB

1.Introduction

Big data research is the mainstream of today's society, and a large amount of data is generated in various industries. Therefore, the storage and processing of data is very important. However, traditional data processing methods are not applicable to big data. So there are some new ways of handling storage. This paper compares some of the storage methods in the world today, including relational database non-relational databases. Introduced some characteristics and applications of relational databases. After comparing the performance of different databases, some conclusions were obtained for processing ENRON data. Finally, the concepts and features of distributed storage and parallel storage are introduced.

2.Relational database and non-relational database

2.1.Relational database

A relational database is a database based on a relational model that processes data in a database by means of mathematical concepts and methods such as set algebra. The various entities in the real world and the various connections between entities are represented by a relational model. The relationship model was first proposed by Edgar Cod in 1970 and is in line with the "Cord 12 Law." [1]. Although there are some unsupported statements about this model, it is still the traditional standard for data storage. The standard data query language SQL is a language based on a relational database that performs retrieval and manipulation of data in a relational database. The relational model consists of three parts: relational data structure, relational operation set, and relationship integrity constraint. Simply put, a relational model refers to a two-dimensional form model, so a relational database is a data organization consisting of two-dimensional tables and the connections between them. The current mainstream relational databases are Oracle, DB2, PostgreSQL, Microsoft SQL Server, Microsoft Access, MySQL, and so on.

2.2.Non-relational database

The term NoSQL first appeared in 1998 and is a lightweight, open source, non-SQL-enabled relational database developed by Carlo Strozzi. In 2009, Johan Oskarsson of Last.fm launched a discussion on distributed open source databases, and Eric Evans from Rackspace again proposed the concept of NoSQL[2].NoSQL mainly refers to non-relational, distributed, database design patterns that do not provide ACID. NoSQL or non-relational databases are a system of supervised databases that stand in stark contrast to relational frameworks in a number of ways. NoSQL takes advantage of relational tables as its storage structure and distinguishes it from relational databases because it does not guarantee ACID attributes. In addition, NoSQL does not require SQL as its query language. In NoSQL, the join operation cannot be performed, but the data is scaled horizontally. According to the different structuring methods and applications, NoSQL can be divided into the following categories: Column-Oriented, Key-Value, and Document-Oriented. The representatives are HaBase, Redis, and MongoDB.

2.3.Relationship

Transaction reliability:

Relational databases guarantee very high transaction reliability because they fully support ACID unlike the NoSQL databases because they range from BASE to ACID.

cloud:

The relationship between the relational database and the cloud environment is limited because the cloud database does not conform to the ACID standard. However, the characteristics of the NoSQL database are very suitable for the cloud database. Its advantages are availability, scalability, performance and flexibility, while also handling unstructured, semi-structured data or structured data, so NoSQL database is the best solution for cloud databases.

Big data processing:

The main purpose of NoSQL databases is to handle big data because they include ways to improve the performance of storing and retrieving data. Relational databases are not good at handling big data because it distributes data to multiple servers in both vertical and horizontal forms, which is more complicated.

Safety:

Relational databases use a very secure mechanism to provide security services, many security threads such as SQL injection, cross-site scripting, Root Kit, weak communication protocols, and more. NoSQL database provides a powerful solution to solve big data storage problems and improve database performance, but security is a flaw.[3]

3.Different non-relational database systems

3.1.Column-Oriented

The search-oriented columnar storage has a storage structure of a columnar structure and a row-like structure of a relational database. This structure makes many statistical aggregation operations simpler and more convenient, and the system has higher scalability. This kind of database can also adapt to the increase of massive data and the change of data structure. This feature is consistent with the related requirements of cloud computing. For example, GoogleAppengine's BigTable and Hadoop subsystem HaBase with the same design concept are typical representatives of this kind. . In particular, Big Table is especially suitable for MapReduce processing, which is highly adaptable to the development of cloud computing.

Table 1. Column-Oriented non-relational database

Example	**Cassandra, HBase, Riak**
Application scenarios	Distributed file system
The data model	Store in column clusters, the same column of data exists together
Strengths	Fast find, scalable, and easier for distributed expansion
Weaknesses	Relative limitations of function

3.2.Key-Value

Cache storage for high-performance concurrent read/write, its structure is similar to the hash table in the data structure. Each Key corresponds to a Value, which can provide very fast query speed, big data storage and high concurrent operation. The primary key queries and modifies the data. The key feature of the Key-Value database is its high concurrent read/write performance, making it ideal for use as a caching system.

Table 2. Key-Valuenon-relational database

Example	**Redis, Voldemort, Oracle BDB**
Application scenarios	Web applications, mainly used to handle high access load of large amounts of data, also used in some log systems and so on.
The data model	Key points to the value of the key value pair, usually implemented by hash table
Strengths	Fast find
Weaknesses	Data is unstructured and is usually only treated as a string or binary data

3.3.Document-Oriented

For the storage of documents for massive data access, the structure of this kind of storage is very similar to Key-Value, and each Key corresponds to a Value, but this Value is mainly stored in documents such as JSON (JavaScriptObjectNotations) or XML. This storage method can be easily used by object-oriented languages. Such a database can quickly query data in a large amount of data, typically represented by MongoDB, CouchDB, and so on.

Table 3. Document-Oriented non-relational database

Example	**CouchDB ,MongoDB**
Application scenarios	Web application (similar to Key-Value, Value is structured, the difference is that the database can understand the contents of Value)
The data model	Key-value pair corresponding to Key-Value, Value is structured data
Strengths	The data structure requirements are not strict, the table structure is variable, and there is no need to define the table structure as a relational database.
Weaknesses	The query performance is not high and there is a lack of uniform query syntax.

4.Why NoSQL should be used to process Enron data?

4.1.Enron Email Dataset and NoSQL

The Enron Email Dataset (1.5GB) contains approximately 500,000 emails exchanged by Enron employees prior to the crash. It is considered big data. The NoSQL system is designed to handle big data and large transaction rates, so it is suitable for NoSQL. Non-relational databases have obvious advantages. There are different performance manifestations when using different types of non-relational databases to process mail data.

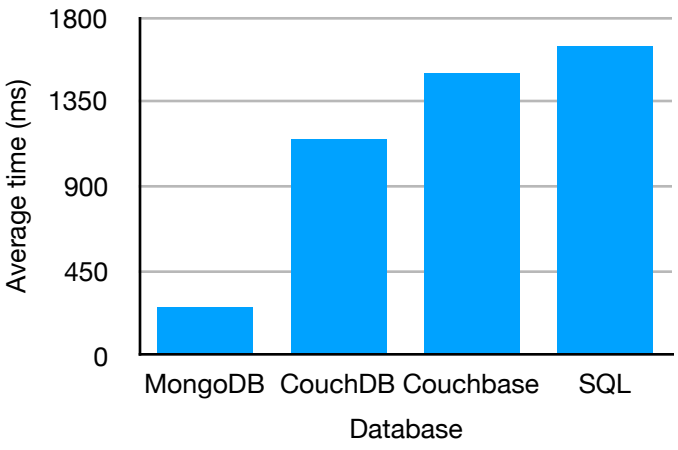
4.2.Performance comparison of different non-relational databases

Some experiments provide performance comparisons between different non-relational and relational databases, and these

experimental data provide strong evidence for research. [4] I

mainly describe performance in three ways. Instantiate , Read , Write.

Instantiate:



We noticed that MongoDB provides the fastest database bucket creation. CouchDB, Couchbase, and SQL Express are much slower to create buckets than mongoDB

Read:

Table 4. Instantiate Performance comparison

Database	Number of operations					
	10	50	100	1000	10000	100000
MongoDB	8	14	100	138	1085	10201
CouchDB	23	101	100	1819	19508	176098
Couchbase	15	22	100	86	811	7244
SQL	13	23	100	277	1968	17214

The figure shows the reading time. Sorted according to the speed of reading data, in order: Couchbase, MongoDB, SQL Express, CouchDB. Couchbase, MongoDB, and CouchDB are all document-oriented databases, which means that the performance is different under the same data model. SQL Express has better read performance than some NoSQL databases.

Write:

Table 5. Write Performance comparison

Database	Number of operations					
	10	50	100	1000	10000	100000
MongoDB	61	75	84	387	2693	23354
CouchDB	90	374	616	6211	67216	932038
Couchbase	60	76	63	142	936	8492
SQL	30	94	129	1790	15588	216479

The table shows the writing time. Sorted by written performance: Couchbase, MongoDB, SQL Express, and CouchDB. CouchDB's write performance is worse than SQL Express. But other NoSQL databases perform better than SQL express.

After some experiments, not all NoSQL databases perform better than the SQL database we tested. We noticed that even in NoSQL databases, the performance based on the type of operation (such as read and write) is quite different. We also observed performance and analysis, and the correlation between different data models is small. Couchbase and MongoDB are the two fastest read, write, and delete operations. However, Couchbase does not support getting all keys (or values). If the data doesn't need to traverse keys and values, then Couchbase would be a good choice, otherwise MongoDB would be a good fit. So for Enron data, MongoDB's schema design allows for the introduction of new CDR types and a fast replication set that meets the DRP and HA solution specifications. [5] At the same time, the analysis process needs to be traversed. Considering that, mongoDB is the best choice for processing this data.

5.Hadoop

Hadoop is a distributed system infrastructure developed by the Apache Foundation. Users can use the concept of clustering to perform high-speed operations and storage of data. Hadoop implements Hadoop Distributed File System, or HDFS. HDFS is highly fault-tolerant and designed to be deployed on low-cost hardware, while providing high throughput to access application data for large data sets (large data) Set) application. The core design of the Hadoop framework is: HDFS and MapReduce. HDFS provides storage for massive amounts of data, and MapReduce provides calculations for massive amounts of data.

5.1.Application

Data Lake:

Because Hadoop provides linear scalability, it is easy to add new data nodes for processing and storage in a cluster architecture, so it provides a very natural platform for capturing and managing raw data files. Based on this feature, Hadoop is suitable as a platform for capturing all data.

Enhanced data warehouse platform:

Hadoop distributed storage capabilities can also be used to extend data for data warehouse environment access for analysis. The most frequently used "hot data" is stored in the data warehouse, while less frequent "cold data" can be submitted to higher latency storage, such as the Hadoop distributed file system. This approach relies on the tight coupling of the data warehouse to Hadoop integration.

Large-scale batch computing engine.:

If data and compute nodes are configured, Hadoop becomes a massively parallel processing platform that can be used for batch processing applications for data manipulation and analysis. Data standardization is a good example of applying a transformation task to a dataset to prepare for analysis. Algorithm-driven analysis applications (such as data mining, machine learning, pattern analysis, and predictive models) can use Hadoop's batch processing capabilities because they all need to parallelize massively distributed data files and overlay the parallel processing results to provide the final Result set.

Event flow analysis processing engine:

The Hadoop environment can also be configured to process received data streams in real time or near real time, running in parallel on a Hadoop cluster.

5.2.MapReduce

MapReduce is a programming model that encapsulates details such as parallel computing, fault tolerance, data distribution, and load balancing. MapReduce first starts the mapping operation, maps the operation to each document in the collection, then groups them according to the generated keys, and puts the generated key value list into the corresponding keys. Reduce is to simplify the value in the list into a single value, the value is returned, and then the key grouping again, until the list of each key has only one value. The advantage of this is that after the task is decomposed, parallel computing can be performed by a large number of machines, reducing the time of the entire operation.

The MapReduce program is implemented in three phases, the mapping phase, the shuffle phase, and the reduction phase.

Mapping phase: The job of a map or mapper is to process input data. The general input data is in the form of a file or directory and is stored in Hadoop's file system (HDFS). The input file is

passed to the line by the line mapper function. The mapper processes the data and creates several small pieces of data.

Reduction phase: This phase is a combination of the Shuffle phase and the Reduce phase. The job of the reducer is to process the data from the mapper. After processing, it produces a new set of outputs that will be stored in HDFS.

6.Parallel distributed computing and centralized solution

6.1.Parallel distributed computing

Parallel computing and distributed computing are methods that take advantage of parallelism in computing to achieve higher performance, as opposed to centralized computing. By working with multiple elements to solve a problem, they can do it faster and solve larger problems. If the processing element shares memory, it is called parallel computing, otherwise it is called distributed computing. Distributed computing is a special form of parallel computing.

The distributed task packages are independent of each other. The result of the previous task package is not returned or the result processing error has little effect on the processing of the next task package. Therefore, distributed real-time requirements are not high, and calculation errors are allowed because each computing task is calculated for several participants. After uploading the results to the server, the results are compared, and then the results are greatly different. The speed of distributed computing is fast, but the real "efficiency" is low. [6]

There is a great connection between the task packages for parallel processing in parallel, and each task block for parallel computing is necessary, there is no waste of segmentation, and the calculation results affect each other. It is required that each calculation result be absolutely correct and time synchronized. Therefore, the number of task packages for parallel computing is relatively limited and should be possible in a limited time.[7]

6.2.Amdahl's Law

When we do parallelization or distributed computing, add computing resources for higher performance. However, the final code cannot be faster than a single-sequence (ie, non-parallel) component running on a single processor. Amdahl's law has the following formula. Consider a partially parallel algorithm, where P is a parallel component and S is a sequence component (ie, a non-parallel component), P+S=100%. T(n) is the running time and the number of processors is n:

$$T(n) \geq S * T(1) + \frac{P * T(1)}{n}$$

If the algorithm cannot be fully parallelized, the final run time depends on the performance of the sequence components. The higher the degree of parallelization, the less obvious the acceleration performance. I assume that the algorithm takes 100 seconds on a single processor and 99% on a parallel component. Add the number of processors to increase the speed, and after calculation we get the results.

$$T(1) = 100s$$

$$T(10) \approx 0.01 * 100s + \frac{0.99 * 100s}{10} = 10.9s \Rightarrow 9.2X \ speedup$$

$$T(100) \approx 1s + 0.99s = 1.99s \Rightarrow 50.2X \ speedup$$

$$T(1000) \approx 1s + 0.099s = 1.099s \Rightarrow 91X \ speedup$$

We see that as n increases, the result of the acceleration does not rise linearly. Using 10 processors, it is 9.2 times faster. With 100 processors, it is 50x. Using 1000 processors, it is only 91x. Amdahl's Law is equally applicable to distributed systems and hybrid parallel-distributed systems. At this time, n is equal to the total number of processors of all computers.

6.3.Cluster computing

Computer clusters connect a set of loosely integrated computer software or hardware to work together highly closely. In a sense, they can be seen as a computer. A single computer in a clustered system is often referred to as a node, usually connected via a local area network, but there are other possible ways of connecting. Cluster computers are often used to improve the computational speed and/or reliability of a single computer. In general, cluster computers are much more cost-effective than a single computer, such as a workstation or supercomputer. Cluster computers can be divided into functions, structures, high-availability (HA) clusters, load balancing clusters, high-performance (HPC) clusters, grid computing.

7.Conclusions

ITraditional databases handle big data is not appropriate. Some tools are different for big data processing. This article compares some of the storage methods in the world today, including relational database non-relational databases. Introduced some features and applications of relational databases. After comparing the performance of different databases, some conclusions were obtained to process ENRON data. MongoDB is the best e to process this data. Finally, the concepts and features of distributed storage and parallel storage are showed.

8.REFERENCES

1. Lourenço J R, Cabral B, Carreiro P, et al. Choosing the right NoSQL database for the job: a quality attribute evaluation[J]. Journal of Big Data, 2015, 2(1):18.
2. Lämmel, Ralf. Google's MapReduce programming model — Revisited[J]. Science of Computer Programming, 2008, 70(1):1-30.
3. Lee K H , Lee Y J , Choi H , et al. Parallel Data Processing with MapReduce: A Survey[J]. Acm Sigmod Record, 2012, 40(4):11-20.
4. Stonebraker, Michael. SQL databases v. NoSQL databases[J]. Communications of the ACM, 2010, 53(4):10.
5. Gudivada V N , Rao D , Raghavan V V . NoSQL Systems for Big Data Management[C]// Services. IEEE, 2014.
6. Shvachko K , Kuang H , Radia S , et al. The Hadoop Distributed File System[C]// 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST). IEEE, 2010.
7. Valiant L G . A bridging model for parallel computation[J]. Comm Acm, 1990, 33(8):103-111.