Acta Technica *61*, No. 4B/2016, 185–196

Research on big data processing and analysis architecture based on MongoDB[1]

Longge Wang[2,3], Tao Song[4], Junyang Yu[2,3]

Abstract. A novel big data processing and analysis architecture is proposed based on MongoDB. First of all, a data processing framework is present to guide the application of big data. Secondly, a new big data processing architecture consisting of non-relational database MongoDB and distributed framework MapReduce is put forward to satisfy the requirements of big data storage and parallel processing. Finally, the travel data of Beijing during the Spring Festival in 2014 is tested to verify the efficiency of the proposed big data processing architecture.

Key words. MongoDB, NoSQL, data processing, data analysis, big data.

1. Introduction

Over the past ten years, data has increased in a large scale in various fields. The application of internet technology, especially the rapidly development of the mobile internet technology, produces large amounts of data in the day-to-day use. And the novel data acquisition setups, such as GPS, new-type sensors, automatic tracking and monitoring system, also produce a large amount of data in everyday practice. Under the explosive increase of global data, the term of big data was used to describe enormous datasets. Big data is an evolving term that describes any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information.

[1]This work was supported by Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), project SKLNST-2016-2-23. Also, the authors greatly appreciate the reviewers' valuable comments on this paper.

[2]Software School, Henan University, Kaifeng, 475001, China

[3]State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), Beijing, 100876, China

[4]Zhengzhou University, Zhengzhou, 450000, Henan, China

Big data brings the new opportunities and challenges for the data storage, data analysis and data processing. Big data is often characterized by 5 Vs [1], [2].

1. Volume: organizations collect data from a variety of sources, including business transactions, social media and information from sensor or machine-to-machine data.
2. Velocity: Data streams in at an unprecedented speed and must be dealt with in a timely manner; for example RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time.
3. Variety: data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data and financial transactions.
4. Value: The critical problem in big data is how to discover values from datasets with an enormous scale, various types, and rapid generation.
5. Veracity. Big data veracity has an impact on the confidence reposed by the marketer to their database.

In a volatile big data environment, accuracy becomes an issue among digital marketers regarding the collected data for their business.

The application of big data has achieved very good results with the placement and use of a large number of sensors [2–4], but there are still some problems to be solved. Firstly, the traditional database is difficult to meet the needs of the data storage and management with the rapid growth of the sensor data. The sensor data has the high concurrency and large data volume characteristics. Secondly, the captured data has a certain degree of redundancy as the sensors usually collect data at intervals of several seconds. Therefore, finding a sensor data storage method with high data storage efficiency becomes an important research area for the application of big data.

In this paper, a novel big data processing and analysis architecture is proposed. Firstly, a data processing framework is presented to process and analysis the big data. Secondly, a MongoDB-based big data processing architecture is proposed to improve the efficient of storage and parallel processing, which composed of non-relational database (MongoDB) and distributed framework (MapReduce). Finally, the travel data of Beijing during the Spring Festival in 2014 is tested to verify the efficiency of the proposed big data processing architecture.

2. Processing of big data

Big data is basically consisted of the acquired data and exchanged data. The knowledge which was mined through the big data processing system can be used to support the upper decision or application. Five main phases [4] such as data preparation, storage management, calculation processing, data analysis and knowledge display are requisites. The processing of big data is shown in Fig. 1.

2.1. Data preparation phase

Big data requires some preprocessing as cleaning and sorting before storage and calculation. It is similar to ETL (Extracting, Transforming and Loading) in traditional data processing system. Compared with the traditional data analysis, the differences are not only in quantity and format but in data quality. In view of these, the format standardization and noise elimination are necessary steps in data preparation phase.

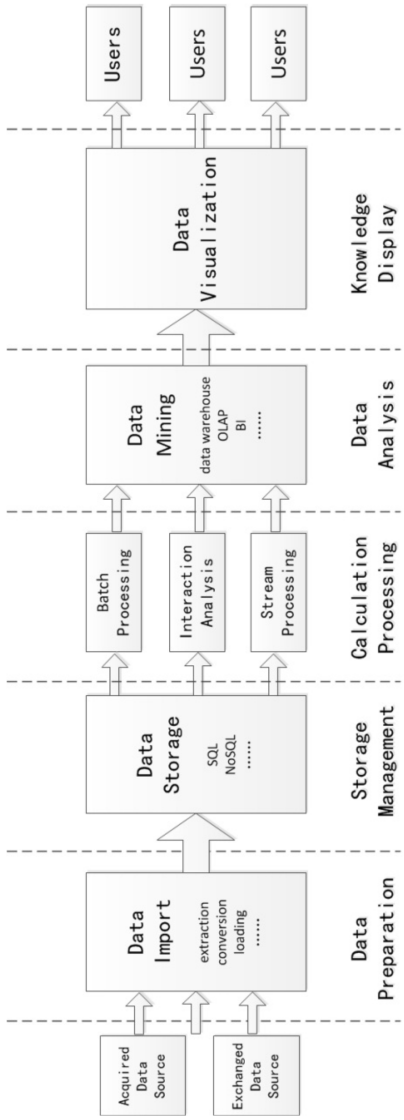

Fig. 1. Processing of big data

2.2. Storage management phase

The storage technology is facing great challenges in terms of performance and cost since the rapid growth rate of big data. Storage system of big data requires high adaptability of unstructured data management, high extensibility of various data formats and low storage cost of massive data.

2.3. Calculation processing phase

According to the data type and analysis target, it is necessary to introduce appropriate algorithm models to implement the fast data processing. Significant amounts of calculating resources are required in big data processing. Therefore distributed computation has become the mainstream architecture for big data processing.

2.4. Knowledge display phase

Presenting the visualized results to the user is an important process of big data analysis when the big data services engaged in supporting the decision application. In some closed loop big data services, visualized results generally are directly applied by the machine according to the algorithm without manual intervention.

3. Comparative analysis on Non-relational database

Storage technology is facing great challenges cause of the characteristics of big data such as heterogeneous, massive, real-time processing. Traditional relational database has been difficult to meet the performance requirements such as high efficiency accessing, high concurrent reading and writing, high availability and high scalability. Distributed storage technology of non-relational database [5], [6] has a greater advantage for massive data storing in each service node through the large-scale distributed structure correspondingly. Accordingly, the distributed storage based on the non-relational database provides an effective solution for the storage and management of big data. The comparative analysis of existing three Non-relational databases is shown in Table 1.

From three mainstream NoSQL databases characteristics analysis [5–7], MongoDB could better support the massive data sharding and quering compared to Hbase and Dynamo. Especially, the index function of MongoDB improves the query speed of acquired data.

4. Big data processing architecture based on MongoDB

According to the previous analysis, designing and constructing big data processing architecture which composed of non-relational database (MongoDB) and distributed framework (MapReduce) will effectively satisfy the core requirements of efficient storage and parallel processing.

The basic operation principle of MongoDB clusters system as follows [10], [11]. MongoDB clusters determine whether the data on a slice is more than a predetermined value when importing data occurred by users. Sharding mechanism will be activated if the value exceeded the limit of storage. Data set is divided into chunks and assigned to different shards. In this process, MetaData information in term of shard and chunk will be stored in Config Server. The architecture adopt with MongoDB clusters could be forming multi-partitions MongoDB servers. In the process of data storage, query and analysis, each partition node could be carried out concurrent processing due to distributed storage. With the increasing of the amount of data, data set can be added to the new shards. Furthermore, each data node is configured with the corresponding backup node. Consequently, the performance requirements of system such as availability, efficiency and scalability could be satisfied. Logical relationship of MongoDB clusters deployment is shown in Fig. 2.

Table 1. Comparative analysis of Non-relational databases

Database	HBase	Dynamo	MongoDB
Data models	Columnar	Key	Document
Single query only	Single query only	Key query only	Support most query
R/W performance	Write complex	Always write, read complex	Write complex
Usability of interface	Support most programming languages	Support simple interface only, RESTFUL	Support most programming languages
Expansibility	Add table service	Add node, table migration	Add shard, chunk migration
Data version	Time stamp	Vector clock	Real time
Data version	Not support	Not support	Support

Big data processing architecture mainly involves control and interaction among the MongoDB distributed cluster, Hadoop cluster, and master server [8–10]. The data processing procedure is as follows.

4.1. Control of Hadoop cluster

The master server is responsible for the coordination control of task assignment. according to the request of query or update on the data shard node, the task and information returned are distributed in a unified way to MongoDB configuration and routing node.

4.2. Control of mongo routers

The mongo routers are responsible for operations of routing and coordinating, thereby control clusters as a whole system. The mongos inquire about information of

MongoDB configuration and routing firstly, and then find out forward the request to the shard node for storing corresponding data. Furthermore, the processing results are returned to mongos separately when sharding node completed the operations. Mongo routers summarize all the results and return to master server at last.

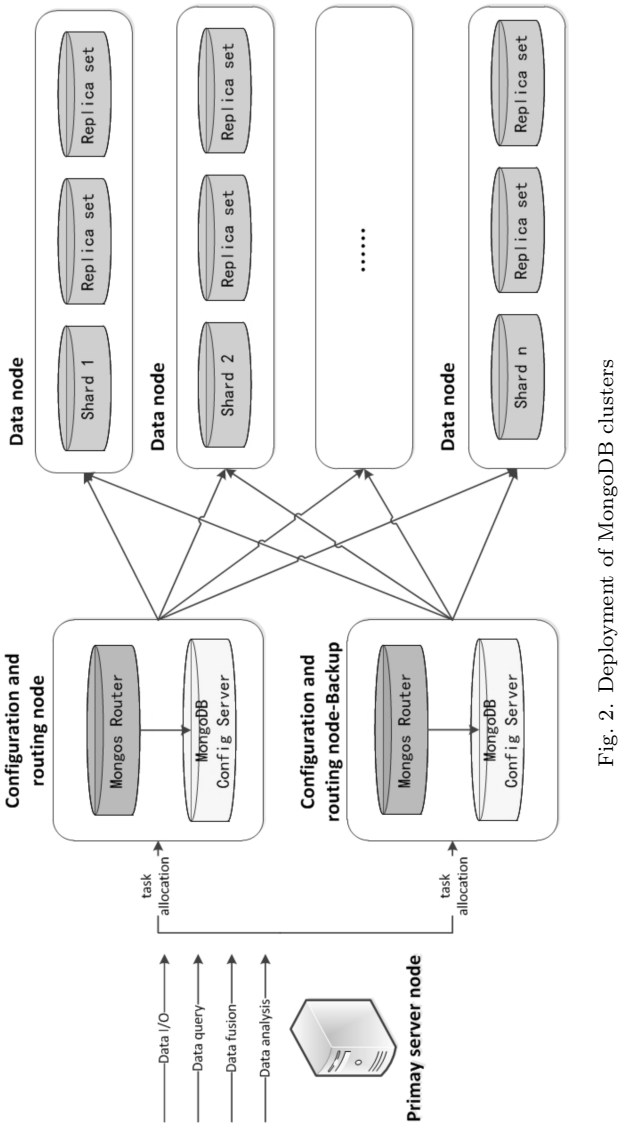

Fig. 2. Deployment of MongoDB clusters

4.3. Data source sharding of MongoDB

Many Chunks is creating as input data for parallel processing from MongoDB. If data is not sharding by MongoDB, then Hadoop is reading the data through the

route. On the contrary, Hadoop is reading the information of data Chunks from MongoDB router and configures server firstly, and then it reads and inputs the storing data on the shards.

4.4. MapReduce procedure

The system node calls MapReduce application to implement the Map and Reduce procedure through the above 3 processes. The final query results will be presented to the user or parallel write to the data shard on the MongoDB.

5. Experiment and discussion

In this section, the travel data of Beijing during the Spring Festival in 2014 are tested to verify the efficiency of the big data processing framework proposed by this paper. The obtained traffic data during the Spring Festival period are more than 2 TB, only part of the most important data are selected to test the proposed big data processing framework in this example.

5.1. Data preparation

In the road traffic data collection, the minimum time granularity, 5 minutes, traffic data are captured by using the various sensors located on the road, such as the traffic flow detector, induction coil, microwave sensors, video surveillance systems, and other equipment. Then the basic parameters of road network traffic flow can be obtained by using these sensor data in combination with the toll road data. Up to now, 693 sets of traffic volume collection equipment has been built in Beijing to cover 593 ordinary road sections in order to obtain road traffic data, where 110 sets in national road, 342 sets in provincial road, and 241 sets in country road. In addition, 18 highways have realized the collection of video, traffic flow and statistical data of import and export toll stations for key sections in Beijing.

In the inter-city passenger flow data collection, intercity passenger flow of real-time and expected travel data can be obtained through the railway ticketing system, inter-provincial passenger ticketing system, as well as civil aviation passenger flight dynamics information.

In this example, 17 days travel data between January 16, 2014 (the first day of the Spring Festival travel) and February 1 (the 17th day of the Spring Festival travel) are selected for experimental validation. Eight access to Beijing's national roads traffic data are selected as the ordinary road data, which captured by the cross-section traffic detection equipment. 11 Beijing's external (excluding urban) highway toll data are selected as highway data. And railway, civil aviation and inter-provincial passenger transport data are derived from the passenger data collected by their respective information systems.

5.2. Data calculation processing

The Map Reduce distributed batch computing method is adopted in order to meet the requirements of high efficient storage and parallel processing of large-scale traffic data in this example. MongoDB server cluster contains MongoDB server with multi-partition node, where each partition holds a portion of the data. Each node can be processed in parallel to ensure the efficiency of the system in the process of saving, querying and analyzing data.

Master acts as the master server to control the task distribution and process coordination of the entire system. Mongos is responsible for routing and coordinating operations, which achieves the overall control of the cluster. And MongoDB cluster creates many chunks from the source data as input data for parallel processing. The system node calls the MapReduce application to implement the Map and Reduce process of the data, and presents the final query result to the user, and writes the data to the MongoDB fragments.

5.3. Data analysis and mining

The proposed data processing framework is used to analyze the traffic data during the Spring Festival in this paper. First of all, the neural network algorithm [11] is applied for feature extraction of traffic data during the Spring Festival in Beijing, and then the key features of the key indicators are got. Secondly, the association rule algorithm [12] is used to analyze the relationship between different features, and the more valuable traffic features and models are obtained. Finally, the obtained models and information are used to guide the urban transport services, planning and management.

5.4. Knowledge display

The use of knowledge display to show the boring traffic data becomes intuitive and visible. The results of the feature analysis in this example are shown below.

- The effect of departure from Beijing is increasing sharply, and the urban passenger transport is the first way for residents to leave Beijing.

  The main feature of urban traffic is a large number of people leaving the city which led to a sharp drop in demand for travel within the city during the Spring Festival, as shown in Fig. 3.

  As can be seen from Fig. 3, the volume of intercity traffic and road traffic began to leave Beijing more than entering Beijing since the launch of the Spring Festival travel. The number of people leaving Beijing increased day by day with the cumulative effect of leaving Beijing, and reaching the maximum on February 1 (the second day of the Spring Festival). Intercity passenger transport (railway, civil aviation and highway inter-provincial passenger transport) is the most important way for residents to leave Beijing during the Spring Festival. The number of people leaving Beijing gradually increased with the Spring Festival approaching. The number of single-day leaving Beijing in January 29 reaches

Fig. 3. Visualization of changes in number of people leaving Beijing during Spring Festival

the peak, 800.000 people. This is because January 29 is the day before Chinese New Year's Eve, and the tradition of home New Year contributed to the increase in the number of people leaving Beijing.

- The traffic pressure distribution in the city is balanced, and the travel time constraint of citizens is obviously weakened.

  Compared to the average hourly traffic index during the Spring Festival and other normal periods, the result is shown in Fig. 4.

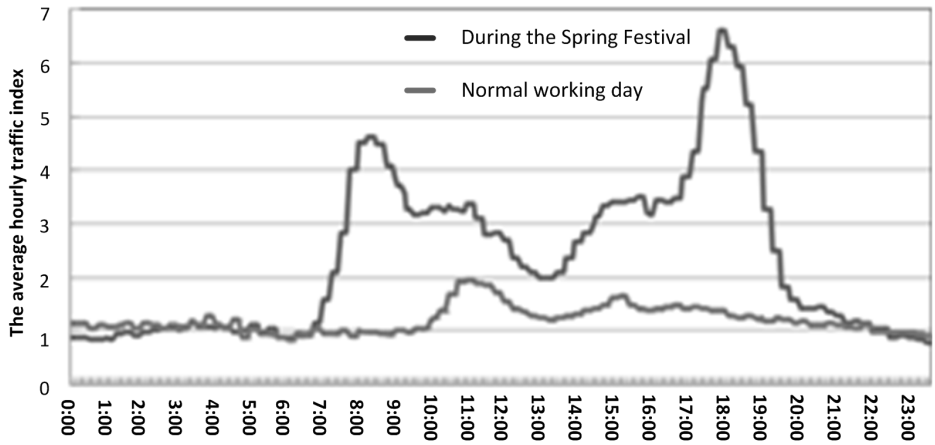

Fig. 4. Comparison of average hourly traffic index between Spring Festival and ordinary working day

It can be seen from Fig. 4 that the travel time constraints were significantly reduced during the Spring Festival, and the elastic travel time effectively balances the traffic pressure over time. There is no obvious morning and evening traffic peak during the Spring Festival in compare with the normal working days. The traffic pressure was balanced throughout the day, and the traffic index at all times was 2.0 or less during the Spring Festival. In other words, city traffic is at the unimpeded level all day. Therefore, the urban road network operation was good because of that a large number of people from Beijing to avoid the daily concentration of morning and evening peak pressure.

- The population is positively correlated with the total demand in the city, and is positively related to the operation of urban road network. igure 5 shows that the sharp drop in the total traffic demand with the return passenger flow, student traffic, and family visits and other traffic generated a large number of people and vehicles from Beijing. The average daily traffic index and population in Beijing are generally declining except on January 18 and 25 (two rest days before the holiday), and both the trends of them are basically consistent. The overall average daily traffic index decreased and the residents travel smooth traffic with the reduction of population in Beijing. All in all, the urban population has a significant impact on the operation of urban road network, and there was a strong positive correlation between them. Therefore, in the normalized congestion control work, we not only through policy-oriented to reduce the frequency of motor vehicle use, through the price mechanism to guide the peak-shifting time-sharing travel, but also should actively explore new management methods, reasonable planning of urban layout and function, strict control of city size and population. The proposed MongoDB-based big data processing architecture is verified through the realization and analysis of this example.

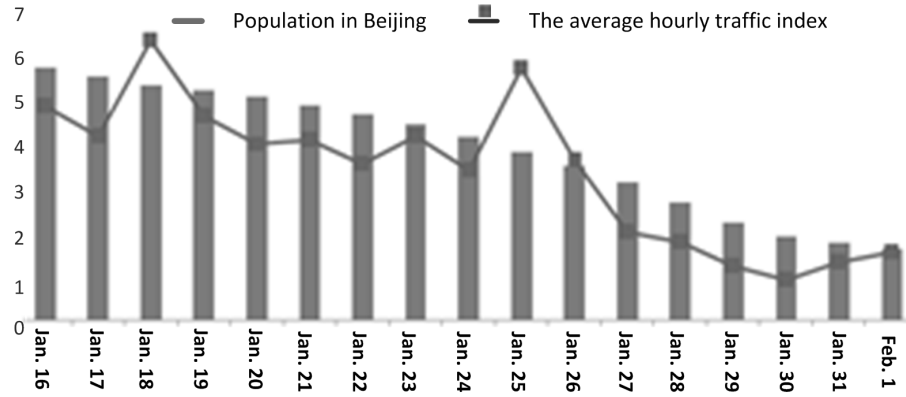

Fig. 5. Comparison between average daily traffic index and population changes in Beijing during Spring Festival

6. Conclusion

In this paper, a novel big data processing and analysis architecture was proposed to satisfy the core requirements of big data storage and parallel processing. Firstly, a data processing framework was used to guide the application of big data. Secondly, a variety of non-relational databases were compared to select the appropriate database. Thirdly, a big data processing architecture was proposed in order to improve storage efficiency, which composed of non-relational database MongoDB and distributed framework MapReduce. Finally, the travel data of Beijing during the Spring Festival in 2014 was tested in this paper. Experimental results illustrate that the proposed big data processing architecture is effective.

References

[1] S. Sagiroglu, D. Sinanc: *Big data: A review*. Proc. IC Collaboration Technologies and Systems (CTS), 20–24 May 2013, San Diego, CA, USA, 42–47.

[2] A. Katal, M. Wazid, R. H. Goudar: *Big data: Issues, challenges, tools and good practices*. Proc. IEEE Sixth IC Contemporary Computing (IC3), 8–10 Aug. 2013, Noida, India, 404–409.

[3] A. Zaslavsky, C. Perera, D. Georgakopoulos: *Sensing as a service and big data*. Proc. IC Advances in Cloud Computing (ACC), 26–28 July, Bangalore, India, 2012, CD-ROM.

[4] C. H. Lee, D. Birch, C. Wu, D. Silva, O. Tsinalis, Y. Li, S. Yan, M. Ghanem, Y. Guo: *Building a generic platform for big sensor data application*. Proc. IEEE IC Big Data, 6–9 Oct. 2013, Santa Clara, CA ,USA, 94–102.

[5] J. Han, E. Haihong, G Le, J. Du: *Survey on NoSQL database*. Proc. IEEE 6th IC Pervasive Computing and Applications (ICPCA), 26–28 October 2011, Port Elizabeth, South Africa, 2853–2857.

[6] B. G. Tudorica, C. Bucur: *A comparison between several NoSQL databases with comments and notes*. Proc. IEEE 10th Roedunet International Conference, 23–25 June 2011, Iasi, Romania, 1–5.

[7] V. Abramova, J. Bernardino: *NoSQL databases: MongoDB vs Cassandra*. Proc. IC Computer Science and Software Engineering, 10–12 Juli 2013, Porto, Portugal, 14–22.

[8] M. A. Khan, Z. A. Memon, S. Khan: *Highly available Hadoop namenode architecture*. Proc. IEEE IC Advanced Computer Science Applications and Technologies (AC SAT), 26–28 Nov. 2012, Kuala Lumpur, Malaysia, 167–172.

[9] E. Dede, M. Govindaraju, D. Gunter, R. S. Canon, L. Ramakrishnan: *Performance evaluation of a MongoDB and Hadoop platform for scientific data analysis*. Proc. 4th Workshop Scientific Cloud Computing, 17–21 June 2013, New York City, NY, USA, 13–20.

[10] L. Vasyliuk, V. Teslyuk: *Software model to MEMS data access based upon MongoDB*. Proc. IEEE XIXth International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), 22–25 Sept. 2014, Tbilisi, Georgia, 184–186.

[11] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, C. Shahabi: *Big data and its technical challenges*. Commun. ACM *57* (2014), No. 7, 86–94.

[12] G. H. Kim, S. Trimi, J. H. Chung: *Big-data applications in the government sector*. Commun. ACM *57* (2014), No. 3, 78–85.

Received November 16, 2016