Journal of Engineering and Applied Sciences 12 (Special Issue 3): 6542-6545, 2017
ISSN: 1816-949X

Implementation of Big Data Analysis System to Prevent Illegal Sales in the Cable TV Industry

Young-Woon Kim and Hyeopgeon Lee
Department of Data Analysis, Seoul Gangseo Campus, Korea Polytechnic University,
Seoul, Republic of Korea

Abstract: Illegal sales, in which sales people ask their acquaintances to subscribe for a certain no-charge period and cancel their subscriptions to receive extra pay for service subscription, frequently occur in the cable TV industry. In order to rectify this illegal sales issue, a system that analyzes whether customers have accessed the service for a certain period is needed. However, the massive log data created from the frequent on/off access by millions of customers to the service is difficult to handle. In this study, MapReduce on Hadoop was used to process the massive amounts of data on a distributed processing system comprising a cluster of computers. After comparing the strong points of major NoSQLs, specifically, MongoDB, Cassandra and HBase, MongoDB was selected because of its strength in flexible search designation for real-time big data analysis in order to ultimately develop a big data analysis system to prevent illegal sales.

Key words: Big data analysis system, Hadoop, MongoDB, Cassandra, HBase, MapReduce

INTRODUCTION

In the cable TV industry, sales people frequently ask their acquaintances to subscribe to cable TV services for a certain no-charge period and subsequently cancel their subscriptions in order to receive extra pay for service subscription. Measures such as sampling research performed by putting inspectors in place in order to prevent such illegal sales have been attempted but the practice has not been exterminated and still occurs from time to time.

In order to resolve this problem, a system is needed to check whether customers have accessed the service for a certain period. However, a cable TV company usually has millions of customers who frequently access the service on an on/off basis thus, it is very difficult for them to process the massive log data generated.

In order to analyze the massive service access log resulting from the millions of customers, two processing technologies were applied. First, the distributed structure of MapReduce on Hadoop was chosen for the big data distributed processing system that uses a cluster multiple computers to handle massive amounts of data.

Second, existing RDBMS (Relation Data Base Management System) costs too much to process the massive amount of data and requires huge time investment to improve the hardware performance and hence has difficulties in developing a big data analysis system.

The table schema structure of an RDBMS makes it difficult to input nonstandard data before processing. Furthermore, output values have errors and it is expensive owing to duplicate queries during processing for analysis.

To make up for the problem of such relational database management system and to resolve the difficulty in handling atypical data, NoSQL was selected because it can be designed to collect and store the data on the basis of distributed processing by analyzing the characteristics of a typical data (Thangavelu and Manoharan, 2016).

MATERIALS AND METHODS

Hadoop is a distributed processing framework that uses clusters comprising multiple computers to process massive amounts of data. It comprises a middleware in the form of an engine and a software development framework. In contrast to transaction processing which responds instantly, it is designed to first collect the data to be processed and to respond upon completion of the request. Therefore, it is suitable for processing massive amounts of data which requires a certain amount of time.

Hadoop fundamentally uses the distributed processing structure of MapReduce. MapReduce is composed of a Map phase in which a single datum is divided into many pieces for processing and a Reduce phase in which the processed results are combined to extract a single result.

Corresponding Author: Young-Woon Kim, Department of Data Analysis, Seoul Gangseo Campus, Korea Polytechnic University, Seoul, Republic of Korea

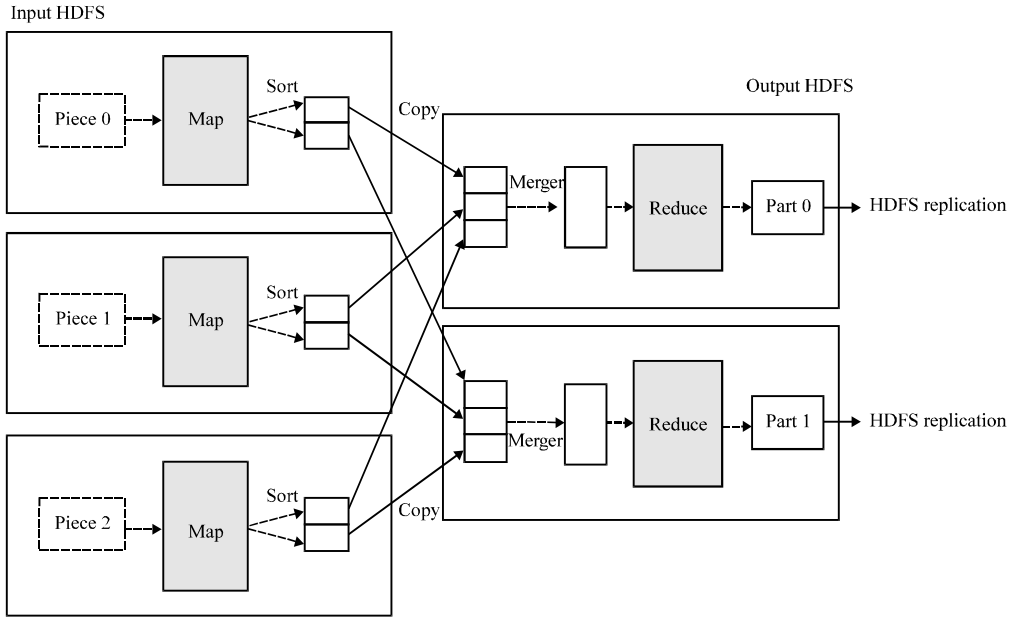

Fig. 1: MapReduce execuation phase

Table 1: Comparison of the features of major NoSQLs

Division	MongoDB	Cassandra	HBase
Features	Stores data in the BSON format, including data structure information. Using this data as a value pairs it with a key	Stores data in the key-value format with a row key as an index. The data must be stored in a column unit	Rows and keys are stored in alignment in a table and play the role of an index. multiple multiple columns are allocated to row keys
Advantages	Search conditions can be flexibly specified Used with no schema	Suitable for services that generate a considerable amount of writings	Ensures consistency in massive data in a distributed environment
Disadvantages	Join or transaction processing cannot be performed	Cannot be searched with complex conditions	Response time becomes slow when store requests are concentrated for a specific range of key values

In order to perform MapReduce, it must be accessible from the entire system and be able to save a large amount of data. Hadoop uses HDFS (Hadoop Distributed File System). In other words, it is composed of a MapReduce module and HDFS (Thangavelu and Manoharan, 2016; Greeshma and Pradeepini, 2016) (MongoDB) (Fig. 1).

HDFS is designed to save large files of over tens of Tera Bytes (TB) or Peta Bytes (PB) to the distributed server and to process many clients quickly. It can configure the storage using a low specification server.

Proposed model: The distributed structure of MapReduce on Hadoop was employed to handle the massive amounts of data. As shown in Table 1, after comparing the strengths of major NoSQLs like MongoDB, Cassandra and HBase, MongoDB was selected because it has the advantage of flexibly assigning searches for real-time big data analysis.

RESULTS AND DISCUSSION

The popularity of Hadoop has grown in the last few years because it meets the needs of many organizations for flexible data analysis capabilities with an unmatched price-performance curve. The flexible data analysis features apply to data in a variety of formats from unstructured data such as raw text, to semi-structured data such as logs, to structured data with a fixed schema (Cassandra) (Ramzan *et al*., 2016; Sung and Doo, 2015).

Hadoop has been particularly useful in environments where massive server farms are used to collect data from a variety of sources. Hadoop is able to process parallel queries as big, background batch jobs on the same server farm. This saves the user from having to acquire additional hardware for a traditional database system to process the data (assume such a system can scale to the required size). Hadoop also reduces the effort and time required to load data into another system you can process it directly within Hadoop. This overhead becomes impractical in very large data sets (Hadoop, 2014). NoSQL (Non-Relational Operation Database SQL) is a storage method that is able to accommodate the characteristics of big data such as large-sized, atypical and real-time and it stores and manages big data with high performance.

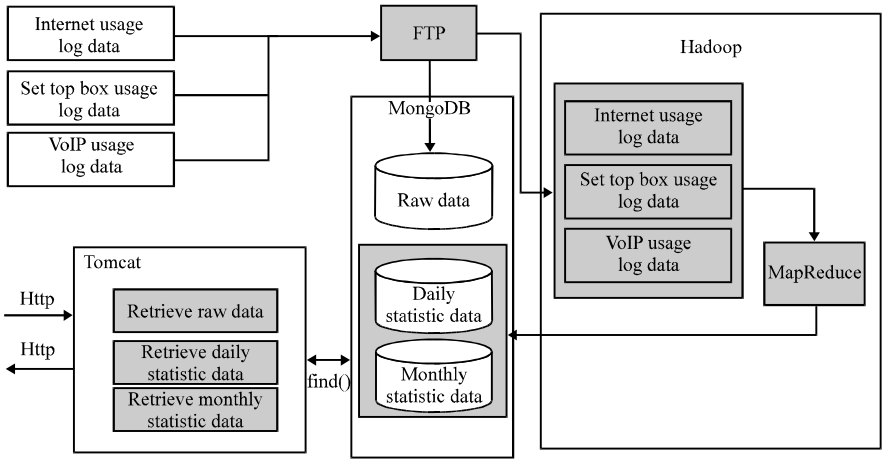

Fig. 2: Big data analysis system to prevent illagel sales

As a new data storage technology, NoSQL has the following three advantages that cause it to be more desired than RDBMS. First, it is open source and can be edited. In addition, it is freeware that can be developed at a low cost, hence, it is suitable for cloud computing environments which require flexible system architecture. Second, it is designed with an atypical data structure so that it can avoid joins and is able to guarantee effective management performance.

Third, NoSQL technology is mostly implemented by memory mapping, so, it is faster than RDBMS in reading/writing the big data. Because it can also be implemented on existing operating systems and hardware, it has greater flexibility and expandability. The major open source based NoSQL includes MongoDB, Cassandra and HBase.

MongoDB is a document-oriented database based on reliability and scalability. MongoDB which aims for low management cost and convenient usability with big data was developed by 10gen with open source and hence it can be commercially supported. The minimum storage unit in MongoDB is a document. Each document is collected at "collection" and each collection is managed at the database to support the scope query, secondary index, alignment operation and set operation of MapReduce. MongoDB collects the document by collection and does not need a schema. A MongoDB query is created in Java Script and document based query is conducted. This is a shell with real-time access and supports multiple program languages (Thangavelu and Manoharan, 2016).

MongoDB is able to expand distribution by using auto-sharding. Sharding is a process in which data are divided and stored on different servers separately. By storing data to multiple servers, more data can be managed and processed (Hbase, 2017; Parthiban and Selvakuma, 2016; Malleswari and Vadivu, 2016). Cassandra has been applied and used by Facebook, a representative social media network site and was distributed as open source by Google in 2008. Cassandra is a highly scalable, high-performance distributed database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It is a type of NoSQL database. Cassandra is a <Key, value> structure DBMS and operates similar to SQL by CQL (Cassandra Query Language) (Thangavelu and Manoharan, 2016; Jongmyeon, 2014).

It is suitable for storing and processing massive amounts of data, restricting properties used for alignment and frequent data changes. HBase is implemented on the Hadoop file system and is a distributed column oriented database made of a table structure using row, column group, row name and timestamp. A large number of columns can be stored in a single row. HBase supports more than 1,000 cluster nodes, speedy reading/writing the big data and guarantees high performance. HBase also supports compression and in-memory processing and provides input and output for MapReduce tasks (Thangavelu and Manoharan, 2016).

Hbase provides random, real time access to your data in Hadoop. It was created for hosting very large tables, making it a great choice to store multi-structured or sparse data. Users can query HBase for a particular point in time, making "flashback" queries possible. These following characterisitcs make HBase a great choice for storing semi-structured data like log data and then providing that data very quickly to users or applications integrated with HBase (Fig. 2).

CONCLUSION

In order to analyze the massive service access logs generated by millions of customers, big data such as customer's set top box usage log data were processed using MapReduce on Hadoop, a big data distributed processing technology. Big data comprising daily and monthly statistics were stored in MongoDB and the big data analysis system to prevent illegal sales by conducting analysis in real time using various search conditions.

REFERENCES

Greeshma, L. and G. Pradeepini, 2016. Big data analytics with apache hadoop mapreduce framework. Indian J. Sci. Technol., 9: 1-5.

Hadoop, 2014. Leading_big_data_technologies. Hadoop, Teradata IT Service Management Company, Dayton, Ohio, USA. https://www.thinkbiganalytics.com/

Hbase, 2017. Welcome to apache HBase™. Apache Software Foundation, Forest Hill, Maryland, USA. http://hbase.apache.org/

Jongmyeon, J., 2014. NoSQL and mongoDB. Deutsche Bahn Railway Company, Berlin, Germany.

Malleswari, T.N. and G. Vadivu, 2016. MapReduce: A technical review. Indian J. Sci. Technol., 9: 1-6.

Parthiban, P. and S. Selvakumar, 2016. Big data architecture for capturing, storing analyzing and visualizing of web server logs. Indian J. Sci. Technol., 9: 1-9.

Ramzan, M., F. Ramzan and S. Thakur, 2016. A systematic review of type-2 diabetes by hadoop/map-reduce. Indian J. Sci. Technol., 9: 1-6.

Sung, K.N. and S.L. Doo, 2015. Bigdata platform design and implementation model. Indian J. Sci. Technol., 8: 1-8.

Thangavelu, A. and N. Manoharan, 2016. Design and analysis of an effective channel distribution approach for agricultural commodities using mongoDB. Indian J. Sci. Technol., 9: 1-10.