

Research on Optimization of Big data Storage Structure in Distributed System

LU Zheng-Wu

Communication University of China, Beijing, China

e-mail: buaaspeed@gmail.com

Keywords: big data; distributed system; row—column store.

Abstract. In a distributed system, the storage structure of the data directly affects the storage efficiency and processing performance of big data. In the row storage structure, the data from the local read, loading speed, but the compression efficiency is low and there is data redundancy; in the column storage structure, the data compression efficiency is high, but the data cross-node access increased network transmission consumption. The Aiming at the shortcomings of the row storage structure and the column storage structure, a kind of storage method combined with rows and columns is proposed to improve the data storage structure. The experimental results show that the improved data storage structure is slightly lower than the row storage in the loading speed. In the data compression, the efficiency of the parallel storage and the column storage is high. The combined storage structure not only avoids the extra disk I / O overhead, but also reduces the unnecessary storage of the network, which greatly improves the storage efficiency and processing performance of the distributed system for big data.

Introduction

The current society has entered the era of data explosion, massive data processing and analysis is called "big data" [1]. Hundreds of millions of users around the world through the Internet interrelated, with the resulting data is also high growth. In the Internet field, applications related to big data have become very important. As the traditional relational database in the management of big data encountered difficulties and obstacles, based on distributed mass data management system has become the current research focus.

Hadoop [2] is the implementation of MapReduce [3] distributed computing framework, providing large scalable, high fault tolerant large-scale distributed clusters for big data processing. MapReduce-based data warehouse [4] can not directly manage the data storage in the cluster, but by the Hadoop distributed storage system HDFS (Hadoop Distributed File System) [5-7] to manage the massive data. How to design an efficient data storage structure in HDFS to encounter large numbers of data encountered a series of difficulties, and the impact of data warehouse performance is a key factor to meet the full use of MapReduce computing features to deal with big data storage structure. Based on the analysis of the limitations of distributed storage system HDFS, this paper makes some improvements to the big data storage and processing by changing the data storage structure.

Big data characteristics and processing requirements

big data characteristics

Big data (big data), or huge amount of information, refers to the size of the data involved can not be through the current mainstream software tools, in a reasonable time to capture, management, processing. The main features of big data are: large amount of data, rich data types, processing speed, accurate calculation.

Big data processing requirements

High speed data loading. High-speed loading of data is a key issue in big data processing, such as Facebook 20TB of data per day stored in the data warehouse. In the normal query, due to network bandwidth, disk I / O data transmission in the resource bottleneck, shorten the data load time becomes very critical.

High-speed query processing. To meet a large number of users at the same time to the system to submit real-time requests and high load query, requires the underlying data storage structure to meet the growing data at the same time, can efficiently handle query requests.

High utilization of storage space. Growing Internet users, resulting in a sharp increase in global data. This requires the system to have a good scalability in storage. How to store data to maximize disk utilization is the key issue.

Adapt to dynamic high load mode. For different applications, users analyze big data sets in different ways. Although some data analysis is performed in static mode cycles, most of the data analysis does not follow any of the normal patterns. Therefore, the system needs to use unpredictable data analysis requests in limited storage space, rather than running in specific modes.

Analysis of Advantages and Disadvantages of Existing Data Storage Structures

Analysis on the Advantages and Disadvantages of row Storage Structure

The row storage structure [8] is a traditional relational database storage structure, and the records are stored in a database relation table in the form of rows. When you add a row, all the columns in the record need to be stored and the records are stored consecutively in the page block of the disk. In the distributed system storage, the table is divided horizontally, and all data in each row is stored in the same HDFS block. Figure 1 shows the distribution of data structures stored in rows in HDFS blocks.

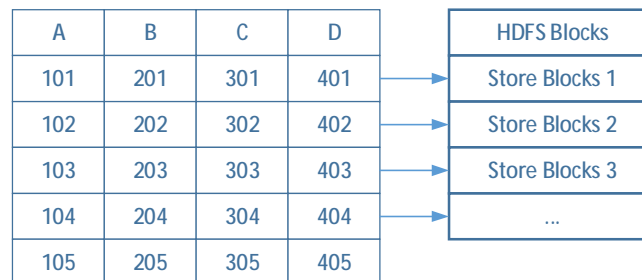


Fig. 1 The layout of raw store structure among HDFS blocks

The data distribution analysis of the rows stored in the Hadoop cluster DataNode [5]. When stored in a row structure, all the columns in each row are stored in the same HDFS block. In the distributed system HDFS, the data in the large table is divided horizontally, and each group of data may be distributed on different DataNode nodes. The data structure of the rows stored in the Hadoop cluster DataNode nodes is shown in Figure 2.

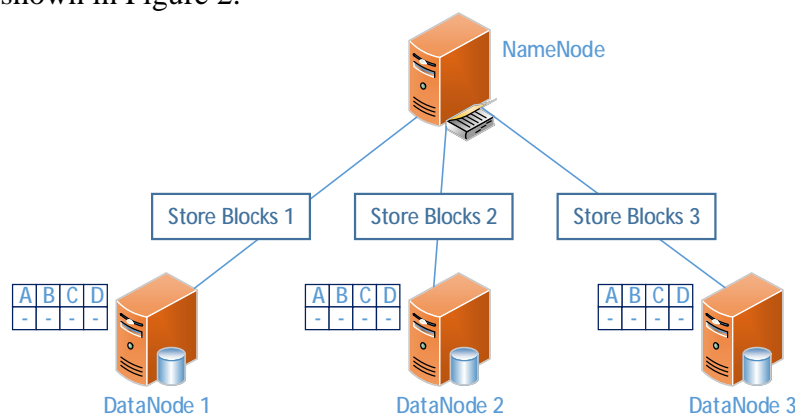


Fig. 2 The layout of raw store structure among DataNodes

Analysis of Data Reading Operation in Raw Storage. If you read columns A and C in the row, first read all the eligible rows on the local DataNode node, and then select the A and C columns from each row to filter out the unnecessary B and D columns. The schematic diagram of the raw data reading operation is shown in Figure 3.

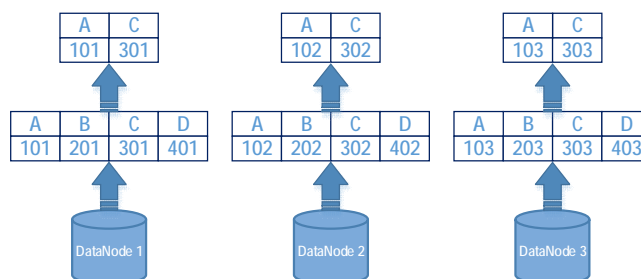


Fig. 3 The schematic diagram of read operation about row store

Advantages and Disadvantages of Row Storage Structure. Data is loaded fast, all data is read from the local, no additional network bandwidth consumption. However, all the columns in each row are stored in the same HDFS block. When the data reads one row of data, all the columns in the row are read from the disk and the unneeded columns are read. Disk I / O overhead.

Each column cannot store the same data type, different data types in the data compression compression is poor, this led to low disk space utilization, but also increased the disk I / O overhead.

Analysis on Advantages and Disadvantages of Column Storage Structure

The column storage structure [9-10] divides the relational tables vertically into multiple sub-relationship table. All the data in each set of subordinate relations tables are stored in the same HDFS block, and each column is stored independently. The distribution of the data structure in the HDFS block is shown in Figure 4.

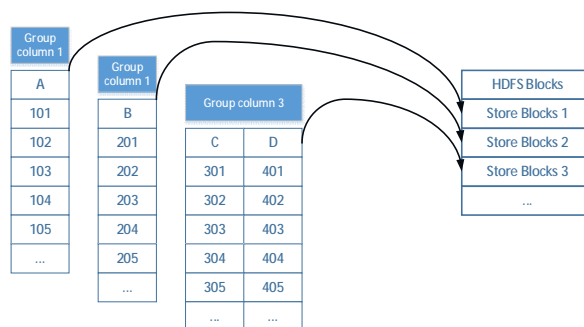


Fig. 4 The layout of column store structure among HDFS blocks

The analysis of the data distribution stored in the Hadoop cluster DataNode nodes. When stored in a column structure, all data in each column is stored in the same HDFS block. In the distributed system HDFS, the data in the large table is divided vertically by column, and each group of data may be distributed on different DataNode nodes. The data structure of the column data stored in the Hadoop cluster DataNode is shown in Figure 5.

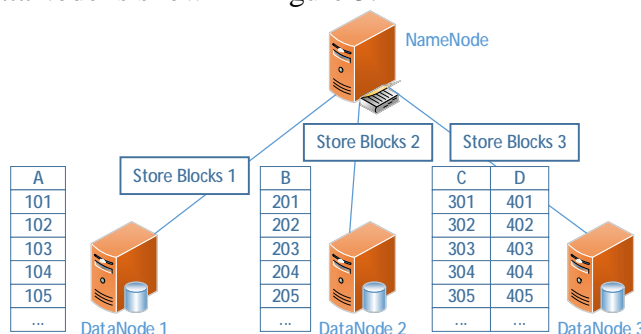


Fig. 5 The layout of column store structure among DataNodes

Analysis of Data Reading Operation in Column Storage. If the rows A and C are read in rows, the A and C columns are stored on two different nodes. First, all data of column A is read from DataNode1, C column data is read from DataNode3, and finally, Data is transferred to the same machine. Figure 6 shows the operation of the column data read operation.

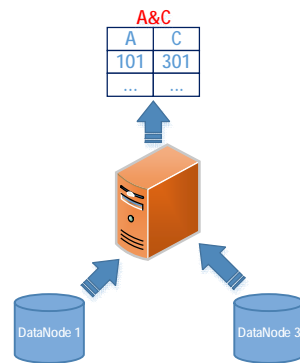


Fig. 6 The schematic diagram of read operation about column store

Analysis of the Advantages and Disadvantages of Column Storage Structure. Column storage only reads useful columns, which can avoid additional disk I / O overhead. At the same time, the same column in the same data type, in the data compression has a good compression ratio, improve the disk space utilization. However, the column storage is divided vertically by column, and different columns may be distributed across different data nodes. Reading different columns of data can be accessed across nodes, increasing the time it takes for network traffic.

Optimization Analysis of Data Storage Structure in Distributed System

Through the analysis of the advantages and disadvantages of the above row storage and column storage structure, the row storage mainly consumes additional disk I / O in the unnecessary columns, and the column storage mainly consumes extra data across the nodes of the network transmission time. The best case is to avoid additional disk I / O consumption and network transmission consumption. Suppose that the i -column valid data is read from the n -line data, and the column storage structure is compared with the data consumption and the network consumption.

Table. 1 The efficiency of row store and column store

	Raw storage	Column storage	Optimal structure
Data read	n	i/n	i/n
Network consumption	0	β ($0\% \leq \beta \leq 100\%$)	0

Improvements to existing data storage structures

The combination of rows and columns of storage structure analysis

Respectively, in combination with the advantages of row storage and column storage nodes, the data in the relational table is divided into a row group by row, and each row group is stored in the same HDFS block. In each row group, the table is divided vertically by a plurality of child relations tables, each column being stored separately in the same HDFS block after data compression. The data distribution between row groups in the Hadoop cluster DataNode nodes is shown in Figure 7.

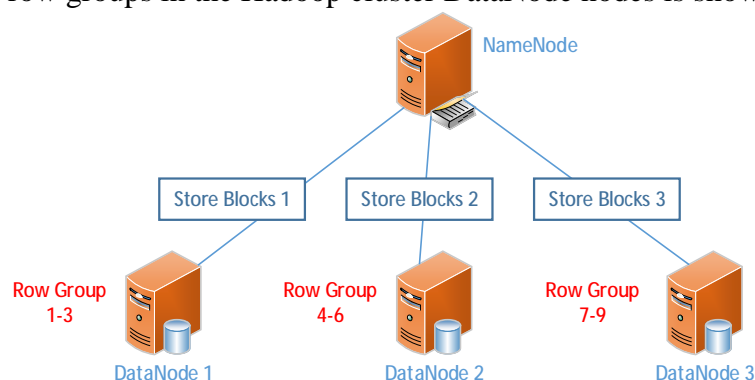


Fig. 7 The layout of row-column store structure among DataNodes

Row group is divided into columns by vertical column, and each column is stored separately in the HDFS block after data compression. All the columns in each row are stored in the same HDFS block. The data in the row group is divided vertically by column The storage structure is shown in Figure 8 in the HDFS block.

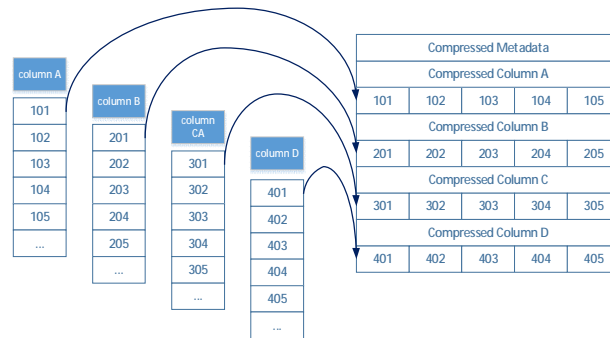


Fig. 8 The layout of row-column store structure among HDFS

Analysis of the Advantages of Storage Structures Combined with Row and Column

Data read operation analysis. The relational tables are horizontally divided into multiple rows, and the rows are vertically divided by columns and stored in the same HDFS block. If columns A and C are read, the local row group is read first, and then the required columns A and C are selected in the row group. This avoids the need to read unnecessary columns in the row storage structure and consumes additional disk I / O, and also avoids the extra network transfer time due to cross-node access in the column storage structure.

Data compression analysis. As the row group is divided vertically by column, each column has the same data type, so there is a good compression ratio in the data compression, thus improving the disk space utilization.

Analysis of segmentation size of Row group

The block size of the Hadoop Distributed Storage System (HDFS) defaults to 64MB, which can be modified in the configuration file. If the storage space of the group is larger than the HDFS block, a row group is stored in multiple HDFS blocks. The main problems are:

1) By the characteristics of distributed storage systems, different blocks may be allocated in different data nodes. Therefore, in the data access, it is possible to cross-node access, which consumes unnecessary network transmission time.

2) HDFS high fault tolerance is the default of each data in the cluster has three backup. If the data node fails, because the ranks of the combination of storage structure constraints, the need for longer recovery time, but also need additional maintenance.

So the maximum size of each row group cannot exceed the size of the HDFS block, you can change the size of the HDFS block to change the size of the row group.

Performance evaluation

The data storage structure of row storage, column storage, and combination of row and column performs performance comparison between data loading time and data compression efficiency. The comparison results are shown in Figure 9.

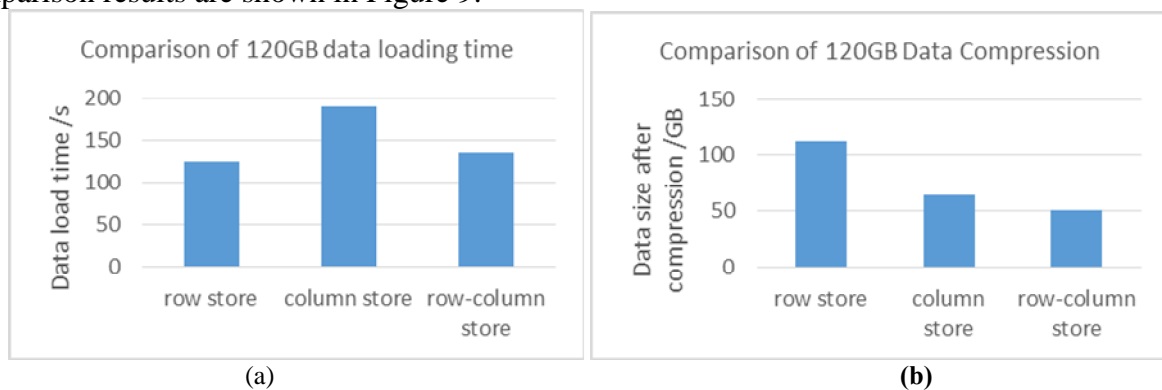


Fig. 9 The comparison result of three data storage structures

Through the test comparison, the line type is stored on the data load, the column storage takes the longest time, and the row storage is slightly better than the row and column storage. In the data

compression effect, the line storage compression effect is the worst, the ranks of the best storage compression.

Conclusion

The row data storage structure has better effect on data loading, but because of the different data types of the columns, the efficiency of data compression is very low, and the disk utilization rate can not be improved effectively. Although the data storage structure has a good compression effect, but because of cross-data node access makes the data load time is longer, thus reducing the system throughput. The combination of the ranks of the data storage structure not only the line structure of the storage of efficient data loading, while a good data compression effect. In a distributed system, the rows and columns of data storage structure greatly improve the big data storage and processing performance.

ACKNOWLEDGMENT

This work is supported by Research project of National Development and Reform Commission (No. [2013]1309), Thanks to the reviewers for the valuable comments helping to improve the quality of the manuscript.

References

- [1] MANYIKA J, CHUI M, BROWN B, et al. Big data: The next frontier for innovation, competition, and productivity [J]. *Communications of the ACM*, 2011, 56 (2) : 100— 105.
- [2] [WHITE T. Hadoop: the definitive guide [M]. O ' Reilly, 2012.
- [3] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters [J]. *Communications of the ACM*, 2008, 51(1) : 107 — 113.
- [4] THUSOO A, SARMA J S, JAIN N, et al. Hive: a ware-housing solution over a map — reduce framework[J]. *Proceedings of the VLDB Endowment*, 2009, 2 (2) : 1626— 1629.
- [5] BORTHAKUR D. The Hadoop distributed file system: Architecture and design [J]. *Hadoop Project Website*, 2007(11): 21.
- [6] KAUSHIK R T, BHANDARKAR M, NAHRSTEDT K. Evaluation and analysis of greenhdfs: A self — adaptive, energy — conserving variant of the Hadoop distributed file system [C] // *Cloud Computing Technology and Science (CloudCom)*, 2010 IEEE Second International Conference on. IEEE, 2010: 274 — 287.
- [7] SHVACHKO K, KUANG H, RADIA S, et al. The Hadoop distributed file system[C]// *Mass Storage Systems and Technologies (MSST)*, 2010 IEEE 26th Symposium on. IEEE, 2010: 1 — 10.
- [8] LIN, RAO J, SHEKITA E, et al. Leveraging a scalable row store to build a distributed text index [C] // *Proceedings of the first international workshop on Cloud data management*. ACM, 2009: 29 — 36.
- [9] IVANOVA M G, KERSTEN M L, NES N J, et al. An architecture for recycling intermediates in a column — store [J]. *ACM Transactions on Database Systems (TODS)*, 2010, 35 (4): 24.
- [10] ABADI D J, BONCZ P A, HARIZOPOULOS S. Column — oriented database systems [J]. *Proceedings of the VLDB Endowment*, 2009, 2(2): 1664 — 1665.