Load Balancing Method for Tilting Data Connection on MapReduce Platform

Cheng Junjing[1,*], Cheng Shengyang[2], Ben Nnegjun[3]

[1]Yancheng Polytechnic College, Yancheng, Jiangsu, China

[2]Department of Computer Science University of Sheffield, Sheffield, South Yorkshire, UK, S10 3DN

[3]Yancheng Polytechnic College, Yancheng, Jiangsu, China

*Corresponding author

Keywords: MapReduce, Tilt data, Load balancing, Solutions.

Abstract: The rapid development of Internet platform has accelerated the speed of data growth. There are some difficulties in screening and utilizing massive data. If the inclined data connection calculation is carried out on MapReduce platform, the process of data screening can be effectively increased and the work efficiency can be improved. However, the current research is limited to the theoretical research stage, not from the perspective of empirical analysis. In order to effectively improve this situation, this paper will develop a load balancing method for inclined data connection on MapReduce platform, in order to provide some reference for solving the current problems.

1. Research background

1.1 Literature review

At this stage, one of the most popular parallel computing frameworks is the MapReduce programming model proposed by Google. This model is based on high-performance computing platform of cluster, which allows parallel computing platform with common market servers. Generally speaking, in the process of computer data processing, connection operation is one of the most important processing steps, but it can not support all kinds of operations such as connection (Zhou, et al, 2013). Therefore, the connection algorithm on MapReduce has always been the focus of research in the field of large data. Some scholars have found that many of the current content is based on uniformly distributed data, the data analysis is optimized (Han, et al, 2013). However, the uneven distribution of data is the norm in life. MapReduce programming model can increase the time gap between tasks and reduce the efficiency of resource recycling. According to the principle of binary equivalence join operation, some scholars put forward a new join algorithm (Li and Dong, 2012) which combines data pre-partitioning and sampling technology. The author uses the method of reservoir sampling to obtain samples, and gets the I/O cost of connection operation according to the distribution of data in the sample space. The results show that the algorithm performs well in processing skew data connection operations. In view of MapReduce's inability to handle data skewed connection operations effectively, some scholars specially proposed frequency classification join algorithm, which is based on MapReduce model (Tao, et al, 2017). Generally speaking, according to the frequency partitioning principle of data in connected data sets, data set classification mainly includes three types, which can realize data redistribution, mainly for partitioning algorithm and broadcasting algorithm, in order to eliminate the adverse effects of data skew (Tao, et al, 2016). In addition, some scholars have clearly pointed out that the redistributed data can complete data connection operations in very short nodes, which can effectively avoid the transmission cost of connection operations under MapReduce framework (Wang and Sun, 2013). In this case, MapReduce model framework can effectively balance the load of each node, and further optimize the efficiency of connection operation under data skew. To some extent, the above literature explores an optimization algorithm for skew data in MapReduce model, but based on the existing situation, this algorithm still has room for upgrading. For this reason, this paper will tilt the load balancing algorithm of data connection on MapReduce platform to get the best solution.

Copyright © (2019) Francis Academic Press, UK DOI: 10.25236/aisct.2019.008

1.2 Purposes of research

How to deal with these data efficiently and make use of them is the focus of people's attention at present. Tens of thousands of enterprises, individuals and machines can generate and acquire new data anytime and anywhere. Memories, computers, mobile phones and other media can be the carriers of data sources. Take domestic Internet companies as an example, as of the latest data, Baidu's total data has exceeded 1000 PB, and the data processed in a single day has exceeded 100 PB; Alibaba keeps nearly 100 PB of data. In addition, Tencent's total storage data volume is close to 100PB after compression processing, and increases at a 10% growth rate. In this case, the traditional parallel database is challenged in the extension method, and it is difficult to approach the current task of large data processing. Therefore, MapReduce can process large data effectively because of its good concurrency and scalability. However, in the use of data processing algorithms, how to create a specific operation process, as well as the verification process need to be studied in depth. Therefore, this paper presents a load balancing method for tilting data connection on MapReduce platform and verifies it.

2. MapReduce connotation and main stages

2.1 MapReduce connotation

MapReduce is a kind of computing model, framework and platform, mainly for large data processing. Generally speaking, MapReduce contains the following three meanings. First, MapReduce exists on a cluster-based high-performance parallel computing platform. MapReduce can use ordinary commercial servers to form tens of thousands of parallel computing clusters. Secondly, MapReduce exists based on parallel computing and runtime software frameworks (Yao and Zhang, 2018). It can automatically divide computing data and tasks, and complete parallel processing tasks in time. It provides a well-designed parallel computing software framework. In this case, MapReduce can deliver data distributed storage, data communication, fault-tolerant processing and other parallel computing content to the system for processing, which can significantly reduce the pressure of software developers (Zhou, et al, 2018). Finally, MapReduce is a method of programming. The principle is that MapReduce uses the idea of function programming language and two functions of Map and Reduce to execute computing tasks, so that batch data programming and computing tasks can be accomplished efficiently and conveniently.

2.2 Main stages

Generally, MapReduce consists of two phases of tasks: Map and Reduce. In the corresponding function framework, it is expressed as Map function and Reduce function. Both input and output formats are $< key, value >$.

The Map phase is responsible for task decomposition and execution. Data resources to be processed are obtained from $HDFS$, and then key-value pairs in $< key_1, value_1 >$ form are generated. By default, $value_1$ is text content and needs to be computed through a programmed Map function, while key_1 is an unrealistic identifier, and then the output of $< key_2, value_2 >$ is obtained. After output, the intermediate results are temporarily saved to the local disk (Zhang, et al, 2017).

Reduce phase task execution. The input form of the Reduce function is $< key_2, list\_value_2 >$. Usually the records of the same key will be processed by the same Reduce and merged into $list\_value_2$. Finally, the exported results will be stored in $HDFS$.

The basic forms of Map function and Reduce function are as follows:

$$Map: \quad < key_1, value_1 > \rightarrow < key_1, value_1 >$$

$$\mathrm{Re}duce: \quad < key_2, list\_value_2 > \rightarrow < key, final\_value >$$

3. Connection algorithms and tilt problem in MapReduce

3.1 Connection algorithms in MapReduce

Joint analysis or processing of multiple data sets usually uses join operation, which has broad application prospects in the field of large data. Among them, the join algorithm is very easy to implement when analyzing and processing small-scale data sets. It can be realized by combining the join keys (Zhao, et al, 2017). When analyzing and processing massive data, people usually use MapReduce to operate. MapReduce can support group aggregation and selection operations, and has certain advantages. The disadvantage is that MapReduce lacks indexing mechanism, which makes the rules of join operation more cumbersome. Therefore, the problem of data skew in distributed environment makes it difficult to implement connection operation in MapReduce. In connection operations at this stage, the main research is divided into two aspects: one is to modify the MapReduce framework to make it easier to connect operations; the other is to build and load the connection algorithm ports from the MapReduce framework.

3.2 Tilt problem in MapReduce

A complete MapReduce job execution cycle is from the start of the task to the completion of all node tasks. Generally speaking, the ideal state is that the tasks of multiple sub-nodes can be accomplished simultaneously. In this case, the operation execution time is the shortest and the efficiency is the highest, which meets the requirements of cost saving. In fact, however, it is very difficult to do so. Due to the emergence of unexpected situations, the execution efficiency of many sub-nodes' hardware will be different, and the size of the cut data will be different, resulting in a certain gap in the types of tasks. Therefore, these reasons may lead to more execution time of some nodes than usual, seriously affecting the overall efficiency of the job. As a result, the tilt problem in MapReduce will lead to inefficient operation. The so-called data skew problem is that in the data set, some values appear too frequently, which is higher than other values. Normally, the data distribution is skewed, which is applicable to the Pareto‘s principle. For example, 20% of users of data information share 80% of website visits and so on. When data skew occurs, there are usually two situations. First, the frequency skew, that is, the amount of data in some areas is much more than in other places; second, the size skew. When a few values incline, and there are more records of the inclination values, the inclination is high

4. Data tilt connection algorithms on MapReduce platform

The connection algorithm under MapReduce computing framework is similar to the traditional database connection algorithm, mainly based on the same key values in two Tables. Sometimes, the connection operation is carried out in a non-equivalent state, but this paper only studies the connection operation under the same key value in the introduction. At this point, suppose that two relational Tables $S(A, jk)$ and $T(B, jk)$ are connected. Among them, A and B represent other attributes of Tables S and T, while jk is the join key. After joining, a new relation Table R is generated in the form of $S \infty T = R(A, B, jk)$. In other words, the SQL language is represented as follows:

$$\begin{aligned}&SELECT\ S.A, S.jk, T, B\\&FROM\ S, T\\&WHERE\ S.jk = T.jk\end{aligned}$$

The data skew connection algorithm adopted at this time is mainly aimed at the results of Reduce end connection. The specific steps include two steps. Firstly, the connection keys in the relational Table are counted and calculated, from which the number of results generated by inclined connection keys is obtained. In addition, the corresponding proportion of Reduces is allocated according to the results, and then sent to reduce to complete the connection operation.

4.1 Data statistics and computation of relational Tables

The frequency of connection keys in relation Table S and T is counted, and the data with the most frequency is defined as inclined data. Data calculation, on the other hand, is based on statistical data to calculate the number of results generated by each connection and the number of allocations. In order to calculate the cost of the algorithm, it is necessary to record the distribution of the original data clearly in advance. Because it is time-consuming and laborious to analyze data in massive data, and extracting a certain amount of samples can reflect the distribution of data, which will further reduce costs. Therefore, this study is mainly through MapReduce tilted data connection operation solutions.

4.2 Statistical-based connections

By analyzing and statistic the frequency of connection keys in relation Tables, we can fully understand the specific situation of inclined data, and further estimate the proportion of inclined data and the amount of data generated. Since the MapReduce number starts from 0, the tilted data connection is handled by MapReduce in the case of 0 to k_1-1. Then, the number of MapReduces is processed according to the calculated specific gravity. The formula for calculating the number of MapReduces (k) for processing skew data is as follows:

$$k = \frac{|S_{skew}| \times |T_{skew}|}{|S_{skew}| \times |T_{skew}| + |S_{others}| \times |S_{others}|}$$

4.3 Data transmission rules

Since the residual parameter of HASH function is the number k of cluster MapReduce, then there is $HSASH(jk) = jk\%k_1$. After file fragmentation, the skewed data is sent back to multiple MapReduces, which can avoid the frequency of some data appearing more than others. For ease of illustration, it can be assumed that Table T skewed data is lower than Table S, so that the data used in relational Table T can be copied into each MapReduce. After sending, the MapReduce end performs connection operations. At this point, these MapReduces receive the amount of skewed data from relational Table T as P. Relevant numerical values are obtained through specific formulas, and the following formulas are obtained: $P = |S_{skew}| / (k_1)$.

4.4 Examples

Next, this paper will illustrate Table S and Table T connection calculation data through specific examples. Assuming that the tilted connection key is “1”, the order corresponding to the size is:

$$S = \{1,1,1,1,1,2,3,4\}$$

$$T = \{1,1,2,2,3,3,4,4,\}$$

Firstly, the frequency of connection keys is counted. From this, we can get that the frequency of key value 1 in S is 8, the comment of key value 2 is 2, and the frequency of key value 3 to key value 5 is 1. Then, according to the statistical calculation, the number of connection results of key value 1 is 8*2=16; the number of connection results of key value 2 is 2*2=4; the number of connection results of key value 3 to key value 5 is 1*2=2. Then the total connection result is 16 + 4 + 2 + 2 + 2 = 26. Secondly, according to the statistical data obtained from the above steps, the calculation results are matched with the number of MapReduces in the cluster. Therefore, the number of tilted data processed is (16/24)*4=3.

5. Empirical analysis and result analysis

At this time, experiments and results are compared according to the proposed MapReduce connection algorithm. In this paper, the default hash partition algorithm is used for comparison. Finally, the authenticity and reliability of the algorithm are obtained through experiments. The test environment is mainly set to three nodes. Among them, one node is used to complete task scheduling assignment, and two nodes are used to complete task connection operation. At this time, the device hardware is the same, CPU is I5-4460, 3.0GHz, 8G memory, 6G display memory, 500G mechanical hard disk. The software environment used in this project is Eclipse neon, Centos 6.5, etc. The experiment was carried out from three aspects: inclination, data scale and MapReduce. Through the experimental control variable method, two parameters are controlled unchanged, one parameter is changed, and the relevant results are finally obtained.

First, the influence of slope on the algorithm. Increased tilt increases the execution time of hash partitioning algorithm. As a result, it means that more skewed data will increase the connection results, which in turn will increase the operation time. With the number of MapReducers unchanged, the tilt rate increases, so that the tilt data processing can be divided into more numbers, so that the algorithm execution remains efficient. Secondly, the influence of the amount of data on the algorithm. With the increase of the amount of data, the time of operation results presents an incremental state. This is because when the inclination is fixed, the increase of data volume directly leads to the increase of inclined elements, resulting in faster growth of connection results. Each MapReduce receives the tilt data of T Table is fixed, and the tilt data of S Table is split data, so the result of connection will increase.

Acknowledgements

(1) In 2017, Jiangsu Province Education Informationization Project, “Exploration and Practice of Hybrid Teaching Model Based on ’Rain Classroom’”, (20172184).

(2) Jiangsu Province Modern Education Research 2017 Project, “Practical Research of Hybrid Teaching Model Based on “Rain Classroom” in Higher Vocational Project Course Teaching”, (2017-R-56372).

References

[1] Zhou J.H., Wang Q., Gao J. (2013). A MapReduce load balancing method based on dynamic partitioning. Computer Research and Development,50 (s1): 369-377.

[2] Han L., Sun X.Z., Wu Z.C., et al. (2013). Research on Optimization of data partition based on sampling on MapReduce. Computer Research and Development, 50 (s2): 77-84.

[3] Li Y.L., Dong J. (2012). Research and Improvement of MapReduce Model Based on Hadoop. Computer Engineering and Design, 33 (8): 3110-3116.

[4] Tao Y.C., Ding L.D., et al. (2017). MapReduce Online Sampling Zoning Load Balancing Research. Minicomputer System, 38 (2): 238-242.

[5] Tao Y.C., et al.(2016). MapReduce load balancing research based on Maxdiff histogram. Minicomputer system, 37 (3): 417-421.

[6] Wang D.L., Sun Z.L..(2013). Large Data Flow Processing Method Based on MapReduce. Computer Applications, 33 (s2): 57-59.

[7] Yao Li, Zhang X.H. (2018). An improved random forest algorithm for processing multi-class unbalanced data in MapReduce environment.Microelectronics and Computer, 35 (11): 145-150.

[8] Zhou H.P., et al. (2018). MapReduce Cluster Load Balancing Strategy Based on Index Migration. Computer Science, 45 (5): 303-309.

[9] Zhang J.W., et al.(2017). MapReduce join query algorithm optimization under non-uniform data distribution. Computer Science and Exploration, 11 (5): 752-767.

[10] Zhao M.L., et al. (2016). Research on the method of creating spatial weights based on MapReduce in big data analysis. Computer Applied Research, 28 (7): 2068-2070.