Analysis of the Big Data based on MapReduce

Zi-de Tian

Mathematics Department, Baicheng Normal College, Baicheng, 137000, China

Keywords: Big Data, MapReduce,SQL server

Abstract. Big Data are becoming a popular technology focus both in science and in industry and motivate technology shift to data centric architecture and operational models. Big Data is bringing a positive change in the decision making process of various business organizations.In this paper, MapReduce big data analysis methods, and with SQL server performance comparison, the experimental results show that, compared to SQL server, MapReduce method loads a small time, as the data set increases, the performance MapReduce approach is better. So MapReduce method has better scalability and speedup for large data processing applications.

1 Introduction

Graphs presented by Google since 2004, has been widely used in large data processing applications. Graphs environment biggest advantage is that encapsulates the underlying parallel details, has good fault tolerance and extensibility, at the same time provide the programmer with simple API interface. But the original graphs calculation mode is not designed to query data connection, so the connection algorithm based on graphs efficiency is not high, in recent years, in view of the graphs for the connection of Chad's support and optimization has received the widespread attention of academia and industry, already have a lot of research results and the corresponding system implementation, for example, in the query optimizer, conversion to join operation is top priority.This paper is on the basis of predecessors' continue to optimize equal join algorithm efficiency. Graphs computing environment without the support of the index, therefore in the treatment of the connection, the map function need to load all the required connection data sets and then put all the data transmission phase (shuffle phase) through the network to reduce side, in the shuffle phase will have a lot of useless data transmission. Therefore, this chapter connection algorithm optimization goal is to reduce the amount of network transmission equal join algorithm based on graphs, so as to improve the efficiency of the connection algorithm.

2 Related Work

MapReduce join operations are widely used log analysis, online analytical processing unit and data analysis. As we all know, the connection between operational data is essential to the operation data query, but also the most time-consuming operation, thus improving the efficiency of large data connection algorithm can effectively improve the efficiency of data query tasks.

MapReduce programming model. MapReduce is Google developed Java, Python, C ++ programming model, Hadoop MapReduce is open source implementation of Google MapReduce, primarily for large-scale (TB-level) data file processing, which is a simplified programming model and efficient distributed task scheduling model, programmers need to focus on the application itself, makes programming a cloud computing environment is very simple. MapReduce thinking is the use of "Map (mapping)" and "Reduce (simplified)" constitutes the basic unit of operation, the first cut is not related data blocks allocated to a large number of Map-tasking, then the intermediate results as a function of the input Reduce, Finally, the output of the final results are summarized.As shown in Fig. 1, input data is first split and then feed to workers in the map phase.

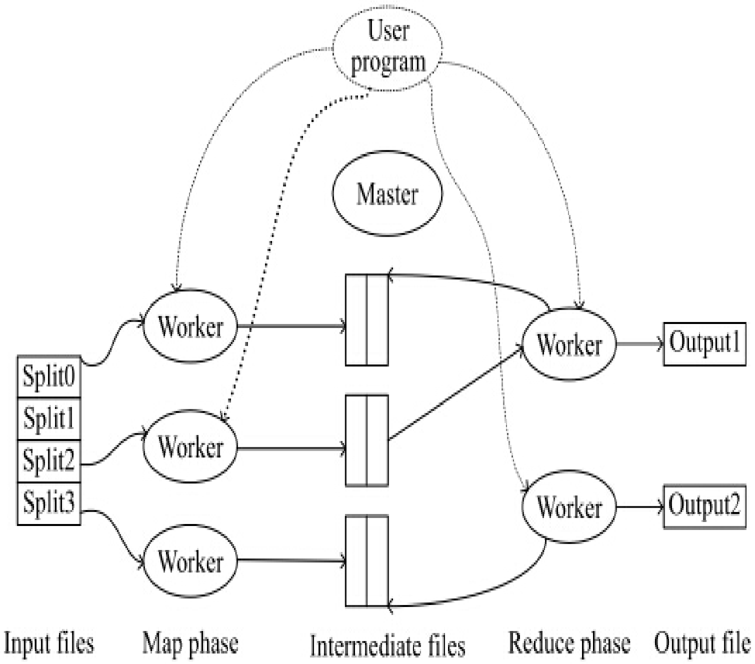

Fig. 1 A high-level illustration of the MapReduce programming model and the underlying implementation architecture.

In short, it can be seen from Figure 1, MapReduce programming mode the input data file is divided into M separate data slice (split); and M is assigned to a plurality of worker starts Map function is performed in parallel, the intermediate file write (local write), and the calculated results in key / value pairs in the form of the output of intermediate results. Intermediate results of key / value according to the key packet, perform the Reduce function, from the intermediate file location information obtained from the Master, the Reduce command is sent to the intermediate node to perform file, calculates and outputs the final result, MapReduce output is stored in output file R can further reduce the need for intermediate file transfer bandwidth.

3 Experimental Simulation

Experimental data sets.8-year ground total factor mapping data as shown in Table 1. Wherein the data format (26 / line), the name of each property are as follows: District station number (long), longitude, latitude, altitude (both floating point), the site level (integer), total cloud cover, wind direction, wind speed, sea level pressure (atmospheric pressure or site), three hours transformer, a past weather, past weather 2,6 hour rainfall, low cloud-like, low cloud cover, low cloud high, dew point, visibility, present weather, temperature, cloud-like, high cloud, flag 1, flag 2 (both integer) variable temperature 24 hours, 24 hours transformer.

Table 1

Dataset	File name	Capacity	Matrix
1	ds1.txt	200M	1*752*365*20
2	ds2.txt	350M	2*752*365*20
3	ds3.txt	500M	3*752*365*20
4	ds4.txt	600M	4*752*365*20
5	ds5.txt	750M	5*752*365*20
6	ds6.txt	950M	6*752*365*20

7	ds7.txt	1100M	7*752*365*20
8	ds8.txt	1200M	8*752*365*20

Load data. First, create separate tables in the Hive and SQL server platform, and then the data set 1,4,8, respectively, to load data into SQL server, each table 1,4,8 nodes Hive platform.

As can be seen from Figure 2, the time is much less than the load data MapReduce SQL server, and the more clusters of nodes, a data loading time is shorter;

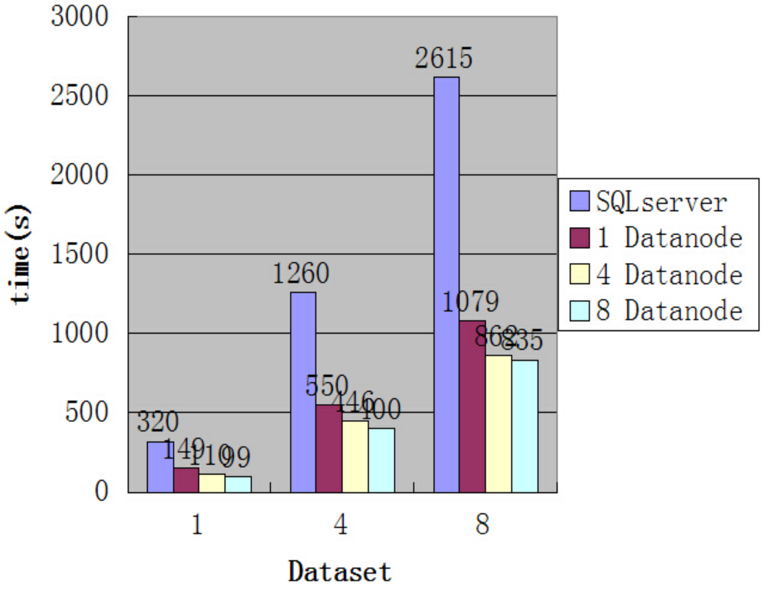

Fig. 2 Load data

4 Analysis of Large Data.

Find a district station total precipitation (summing Tasks)

Select a,select a,sum(m) from tab6 group by a;

Find visibility and temperature are equal area stations (multi-table query tasks)

Select a,from tab1 as a1,tab2 as a2 where a1.t=a2.t and a1.r=a2.r group by a1.a;

Table 2

Find a district station total precipitation (summing Tasks)
select a,sum(m) from tab6 group by a;
Find visibility and temperature are equal area stations (multi-table query tasks)
Select a,from tab1 as a1,tab2 as a2 where a1.t=a2.t and a1.r=a2.r group by a1.a;

Figure 3 can be seen in the implementation of the summation task, as the amount of data increases, SQL server execution time significantly increased, while MapReduce has increased very smooth. Figure 4 can be seen in the implementation of multi-table query tasks, MapReduce with the increasing amount of data, the execution time of slow growth, when dealing with large data sets, we can see that MapReduce is more suited to handle multi-table queries than SQL server. The text of the SQL server and compare the performance of MapReduce when dealing with large data sets, MapReduce has a very good advantage, when the amount of data increases, SQL server memory overflow problems occur, and high scalability of MapReduce, knot the more points, the better the performance data analysis; MapReduce users can customize the functions to handle the problem and Hive SQL server itself can not handle, more in line with the needs of large data processing.

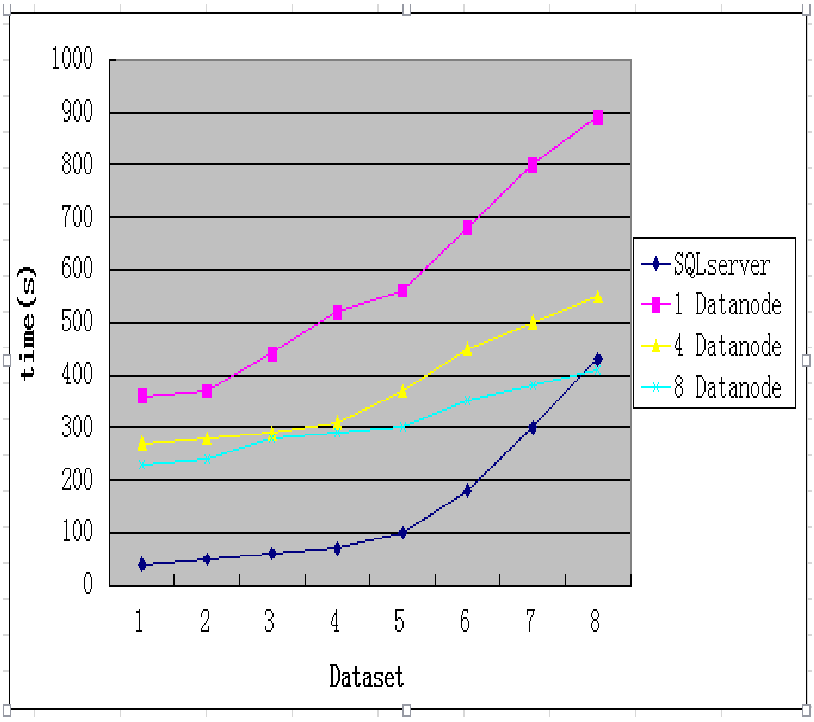

Fig. 3 Sum task

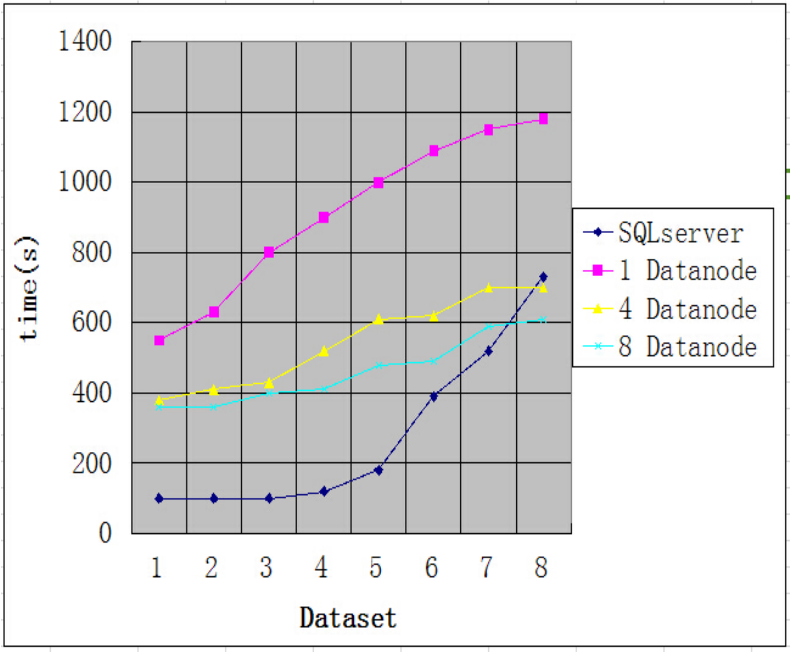

Fig. 4 Multi-table query task

5 Conclusion

With the development of cloud computing technology-based, MapReduce data analysis methods are more and more attention. To this end, this paper-based, MapReduce big data analysis methods, and, MapReduce and SQL server performance comparison conducted experiments show that, based on, MapReduce analytical method is feasible and effective in large data analysis. Next we will, MapReduce combined with the advantages of other database systems, develop, MapReduce and relational databases dual advantages of big data analysis system.

References

[1] J. Dean and S. Ghemawat, Mapreduce: Simplified data processing on large clusters, Commun. of ACM, vol. 51,no. 1, pp. 107-113, 2008.

[2] L. Wang, J. Zhan, W. Shi and Y. Liang, “In cloud, can scientific communities benefit from the economies of scale,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 2, pp. 296-303, 2012.[3] Java programming language, http://www.java.com/, 2013.

[4] P. Th. Eugster, P. A. Felber, R. Guerraoui, and A.-M.Kermarrec, The many faces of publish/subscribe, ACM Comput. Surv., vol. 35, no. 2, pp. 114-131, 2003

[4] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. In OSDI, pages 137-150{ 2004}.

[5] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J.DeWitt, S.Madden, and M.Stonebraker. A comparison of approaches to large-scale data analysis. In SIGMOD,pages 165-178{.ACM,2009}.

[6] Dean and S. Ghemawat. Mapreduce: a flexible data processing tool. CommunIn. ACM, pages53(1):72-77{ 2010}.

[7] B. Fung, K. Wang, L. Wang and P.C.K. Hung, “Privacypreserving data publishing for cluster analysis,” Data and Knowledge Engineering, vol. 68, no. 6, pp. 552-575, 2009.