International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 12, December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

Data Load Balancing in a Big Data Heterogeneous Environment

Prof. R Vijay Anand[1], Prof. M N Rajaprabha[2], Prof. P Jayalakshmi[3]

[1,2,3] *School of Information Technology & Engineering, VIT (India)*

ABSTRACT

Data load balancing is one of the key problems of big data technology. As a big data application, Hadoop has had many successful applications. HDFS is Hadoop Distributed File System and has the load balancing procedure which can balance the storage load on each machine. However, this method cannot balance the overload rack preferentially, and so it is likely to cause the breakdown of overload machines. In this paper, we focus on the overload machines and propose an improved algorithm for balancing the overload racks preferentially. The improved method constructs Prior Balance List which includes overload machines, For Balance List and Next For Balance List by many factors and balances among the racks selected from these lists firstly. Experiments show that the improved method can balance the overload racks in time and reduce the possibility of breakdown of these racks. Thus in between these two phases along with the resource retrieval and process of resource allocation at proper cluster nodes might be a huge fraction that can be further enhanced to yield better results.

Keywords: Big Data,Hadoop,HDFS, Heterogeneous environment, Load Balancing, Overload

I. INTRODUCTION

Vast increase in data load on cloud communications plays a major role in the quality of service for Big data applications. Adding to this the data storage of ordered as well as unordered kinds are also a major issue in the direction of working with search engine or data discover applications. As an additional factor data concentrated computation for cloud services is also growing with immense speed. The major requisite for quality services in cloud communications are optimized resource utilization, efficient processing in terms of execution time and data portability. To build up such a cloud structure a number of studies have been done and still going on. The core of Apache Hadoop consists of a storage part Hadoop Distributed File System (HDFS) and a processing part MapReduce. Hadoop splits files into large blocks and distributes them amongst the nodes in the cluster. To process the data, Hadoop MapReduce transfers packaged code for nodes to process in parallel, based on the data each node needs to process. This is the subject of reality that Hadoop is a potential candidate to operate with hundreds of Peta bytes of data on cloud, but with increase in further loads

II. BIG DATA ANALYTICS

Gartner defines Big Data is an innovative and most cost effective form of processing the raw information collected from different resource , He classifies it as 3 V's Velocity , Volume and Variety of

information. IBM claims 80% of data are gathered through sensors used in climate and whether information, post in social networking like digital pictures and videos, audios, purchase transaction records, and also GPS signals, Also all these data are raw which don't have any structure. These unstructured data are called Big data.

Hadoop is undoubtedly the most preferred choice for processing and analyzing this huge volume of unstructured data which has its unique feature like hadoop is flexible to adopt variety of data, it is consistent, and also since it is open source no need for spending huge amount of money in it so it is relatively inexpensive, Hadoop is also scalable solution. It is capable of storing huge amount of unstructured data in its own file system which is HDFS (Hadoop Distributed File System), We have various technology and solution to support the processing of data in hadoop example we have MapReduce , Pig and Hive. The data collected after performing the above task are used for future use of the organization for their business or any other need

Basically Big data analytics is the process of analyzing and examining this large amount of data types, to find the hidden patterns, or unidentified correlations and other practical information.

III. HADOOP FRAMEWORK

Hadoop is the software framework which plays a major role in big data analytics, Hadoop is developed and distributed by apache software foundation which is a open source community, For storing huge volume of unstructured data hadoop uses special file system which is called HDFS (Hadoop Distributed file System), On the other hand we have Goolge file system Which is called GFS, Both has its unique feature and functionality, HDFS and hadoop follow master/slave architecture for distributing data to multiple node and monitor the processing of all the node.It contains one master node and multiple slave nodes, check out the fig below which shows the architecture of hadoop, There are 2 major components present in hadoop one is name node which contains all the information about the data node and which data is distributed in which cluster node, And second is data node where the actual data is present , This data node will be present in all the heterogeneous system connected to the master node which is name node. Similar to hadoop in GFS we have chuck server which is a data node in hadoop and we have central server which is called as NameNode in hadoop.

The below figure shows the data distribution among multiple node in hadoop with the help of name node.

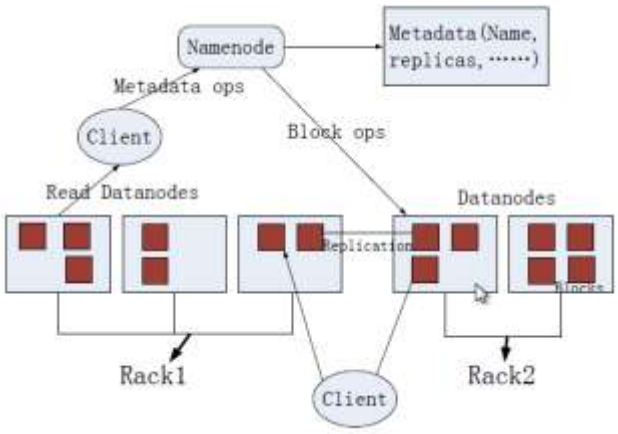

Fig 1

Steps in load distribution:

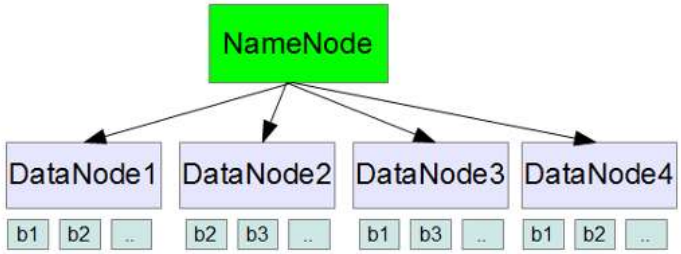

Fig 2

- In the above figure which refers the connection between NameNode and DataNode , there is one master NameNode which has details of multiple DataNode b1, b2 ,b3 etc …
- In HDFS when the data is uploaded all the data will get split into several blocks and by default the block size is 128MB in Hadoop, But the block size is scalable from 64MB to 128MB, Each bloc will get replicated to multiple node basically to recovery the data in terms of failure , The data will get replicated into minimum of 3 different node.
- Hadoop will take care of performance of all the node and it will make sure that any node failure will never affect the results in data loss.
- Hadoop will be the one which is completely responsible for maintaining the directory of file maintaining the directory of file system and keep track of all the data in the hadoop cluster, NameNode is very critical since it is single failure point , if NameNode fails then entire cluster will go down.
- There are multiple DataNodes which are cheap servers which will hold the actual data to be process.
- The query from client will fist taken by NameNode and get the metadata file from NameNode , with this metadata information the query will be passed to the respective DataNode
- Hadoop also provides CLI (Command Line Interface) for administrators to work with the HDFS System.
- Name Node comes with web server from where we can access all the HDFS in cluster of node.

Evaluation of node performance:

The important task and the core funtinality of Hadoop is to perform MapReduce task , MapReduce task containis three major operation which is Mapping , Shuffling and sorting and Reduce, We use Reducer which is a java progress execution function that process the key/value pair, The main idea behind is to transfer the data to Reduce tasks node purely based on the efficiency and performance of the node where Reducer is performed , We can calculate the performance of the node from the rate in which the data flows and the rate in which the data comes out , Which is executing velocity of the node which is running the Reducer task.

The execution velocity vi of the Reducer task is calculated from below equation:

$$v_i = \frac{r_i}{t_i}$$

Where r_i is the data proporation and of particular Reducer task, And t_i is the execution time of the it^h Reducer task, The node performance is giving for score of 100, Initially all the cluster node will be given score of 100 and depends on the execution of Reducer task, the score will get reduced or set stable, So now the relative weight of the node P is calculated which executed the Reducer task , The pi is calculated with the below equation:

$$p_i = \frac{v_i}{\sum_{i}^{n} v_i} \times 100 \times ReduceNum$$

Finally, the performance of all the nodes will be renewed effectively.

Algorithm for load balancing during node failure:

HDFS architecture is having efficiency of balancing data in the cluster of nodes, It will move the data automatically from one DataNode to other in case of failure happen in any node. But every time when failure occurs hadoop will create a new node and transfer all the left over data in failure node to newly created node, This will increase the cluster size when there are more number of failure, Our proposal is to monitor the performance of all the node in the cluster and transfer the left over data to the high performance and underutilized node, In the existing system user have command to control the average rate of all the nodes storage space, and the performance will vary according to the user specification, We can categories the node into four major types : I) Average performance node II) Over performance datanode III) Below Average performance node IV) Under performance data node

There will be one special balancing node which will perform the load balancing among the node by identifying the node which performance is high performance or underutilized node , In case of any node failure following steps will get carried out:-

1) First, data is moved from under performance Data nodes list (source) to over performance Data nodes list (Target).
2) Second, data is moved from under performance Data nodes list (source) to above Average performance Data nodes list (Target).
3) Finally, if data is remaining then data is moved from under performance Data nodes list (source) to below Average Utilized Data nodes list (Target).

The balance method in the above three steps are as follows:

1) We select one node marked as S(failure node) from the Source list and one node marked as T from Target list. When selecting one node, the principle is traversing the entire data until the failure node data is empty.
2) The load of S is moved to T. How many bytes can be moved depends on the bytes that T can receive and the bytes that S needs to move. Procedures record the number of bytes (max Size to Move) needed to move and the number of bytes (scheduled Size) that have been moved from the source node. Procedures also record the number of bytes (scheduled Size) that have been received and the maximum number of bytes (max Size To Move) that can be received.
3) After a round of balance, if the system still has not reached equilibrium, then jump to step 1 to repeat the balance procedure.

IV. CONCLUSION

Hadoop load balancing algorithm cannot optimize overload racks preferentially, But in this paper we proposed an algorithm which will calculate the performance of the each node in hadoop cluster with the result of velocity of each node we proposed simple algorithm which will distribute load among the different heterogeneous system. The same is applicable in case of any node failure the load allocated for the failure node will get distributed among different node depends upon the performance evolution algorithm and load balancing algorithm, In future need to focus more on optimizing the algorithm to get maximum percent of result in balancing the load

REFERENCES

[1] A Load Balance Algorithm Based on Nodes Performance in Hadoop Cluster by Zhipeng Gao , Dangpeng Liu , Yang Yang ,Jingchen Zheng , YuwenHao

[2] An Improved Hadoop Data Load Balancing Algorithm Kun Liu, Gaochao Xu , and Jun'e Yuan , JOURNAL OF NETWORKS, VOL. 8, NO. 12, DECEMBER 2013

[3] QoS oriented MapReduce Optimization for Hadoop Based BigData Application by Burhan Ul Islam Khan , RashidahF.Olanrewaju Burhan Ul Islam Khan et al. Int. Journal of Engineering Research and Applications www.ijera.com ISSN : 2248-9622, Vol. 4, Issue 4

[4] A Dynamic Load Balancing Method On A Heterogeneous Cluster Of Workstations by Alessandro Bevilacqua

[5] Big Data": Big Gaps of Knowledge in the Field of Internet Science Chris Snijders , Uwe Matzat , Ulf-Dietrich Reips ,International Journal of Internet Science 2012

[6] Profile-Based Load Balancing for Heterogeneous Clusters * M. Banikazemi, S. Prabhu, J. Sampathkumar, D. K. Panda, T. W. Page and P. Sadayappan

[7] Decentralized Load Balancing for Heterogeneous Grids Issam Al-Azzoni and Douglas G. Down Department of Computing and Software

[8] Analysis of Load Balancing in Large Heterogeneous Processor Sharing Systems by Arpan MUKHOPADHYAY and Ravi R. MAZUMDAR

[9] Performance Comparison of Adaptive and Hierarchical Load Sharingm in Heterogeneous Distributed Systems, Conf. Parallel and Distributed Computing Systems, New Orleans, 1997.

[10]A MODEL FOR RESOURCE-AWARE LOAD BALANCING ON HETEROGENEOUS AND NON-DEDICATED CLUSTERSBy Jamal Faik

[11] A Load Balancing Policy for Heterogeneous Computational Grids Said Fathy El-Zoghdy, Journal of Advanced Computer Science and Applications,Vol. 2, No. 5, 2011