Map Reduce based Analysis of Live Website Traffic integrated with improved Performance for Small files using Hadoop

Sushmitha R[1] Vaibhavi Shekar[2] Mrs. Swathi Baswaraju[3]
[1,2]B.E. Student [3]Assistant Professor
[1,2,3]Department of Information Science and Engineering
[1,2,3]New Horizon College of Engineering Bangalore, India

Abstract— Hadoop is an open source java framework that deals with big data. It has mainly two core components: HDFS (Hadoop distributed file system) stores large amount of data in a reliable manner and another is MapReduce which is a programming model processes the data in parallel and distributed fashion. Hadoop does not perform well for small files as a large number of small files puts a heavy burden on the Name Node of HDFS and results an increase in execution time for MapReduce. Hadoop is designed to handle large size files and hence suffers a performance penalty while dealing with large number of small files. This research paper gives an introduction about HDFS, small file problem and existing ways to deal with it along with proposed approach to handle small files. In the proposed approach, MapReduce programming model is used for merging small files on Hadoop. This approach improves the performance of Hadoop in handling small files by ignoring the files which have a size that is larger than the block size of Hadoop and also reducing the memory required by Name Node to store them. We also propose a Traffic analyzer with the combination of Hadoop and Map-Reduce paradigm. The combination of Hadoop and MapReduce programming tools makes it possible to provide a batch analysis in a minimum response time and memory computing capacity in order to process log in a highly available, efficient and stable way.

Key words: Log Files, Small Files, Hadoop, Hadoop Distributed File System (HDFS), MapReduce, Google Visualization

I. INTRODUCTION

As the data exploded from the web and grew beyond the ability to be handled by the traditional system, Hadoop [6] was created by Doug Cutting. As published in the papers by Google, Hadoop is capable in storing, processing and analysing hundreds of terabytes or petabytes of data. Hadoop is an open source framework provides distributed parallel processing of large amount of data across commodity computers. It can store and process the data. It can also scale without limits. For Hadoop, no data is too big. Hadoop uses MapReduce [3] programming model. Hadoop uses this model and queries the dataset, divides it into sub-parts and then run it parallel over multiple nodes. Log files are the one that consist of actions that have occurred. For example, web servers maintain log files that list every request made to those servers. These log files keep track of the number of visitors and time spent by them on the website. Thousands of petabytes or terabytes of log files are generated by a data centre in a day. It is a challenging job to store and analyse such large volumes of log files. The problem of analysing these log files is complex because of not only its volume but also its disparate structure.log files is one of the type of big data that is growing rapidly hence Hadoop is the best suitable platform for storing the log files and parallel implementation of MapReduce program for analyzing them.

MapReduce paradigms are designed to compute large volumes of log files in a parallel manner. Hence Hadoop-MapReduce is used in various areas for the analysis of Big Data. Hadoop enables applications to work with thousands of nodes and

Large number of data. Sometimes it becomes difficult to handle the data with a single machine. Distributing the computation solves this problem. Hadoop breaks the log files into numbers of blocks and these blocks are evenly distributed over a cluster of thousands of nodes.

Hadoop also includes a Hadoop Distributed File System (HDFS) [4], which divides the input data and stores that on the compute nodes. HDFS enables the data to be processed in parallel using all of the machines in the cluster. It is a portable filesystem written in Java for the Hadoop framework.

A small file is a one whose size is less than the HDFS block size. The design of the HDFS is such that it stores large files but lacks efficiency in storing a large number of small files because of high memory usage and unreliable access cost.

To overcome this, merging of small files is done using MapReduce programming model on Hadoop. This approach improves the performance of Hadoop in handling small files by ignoring the files that have a size that is larger than the block size of Hadoop and reduces the memory required by NameNode to store them.

This article tells how large volumes of log files of smaller size can be stored and handled using Hadoop and analysed them using MapReduce.

II. LITERATURE SURVEY

A. A Novel Approach to Improve the Performance of Hadoop in Handling of Small Files

This paper [1] gives an introduction to HDFS, problem involved in small files and an approach to handle these small files. It specifies the major categories of machine roles in Hadoop which are Name node, Data node and client machines.

In proposed approach, MapReduce programming model is used for merging the small files on Hadoop. This approach efficiently improves the performance of Hadoop in handling small files by ignoring the files which have a size that is larger than the block size of Hadoop and reduces the memory required by NameNode to store them.

B. An approach for MapReduce based Log analysis using Hadoop

This paper [2] speaks about log files and MapReduce. Hadoop MapReduce based log file analysis tool will provide us graphical reports showing hits for web pages, user's page

All rights reserved by *www.ijsrd.com*

view activity, in which part of website users are interested, traffic, attack etc. From these reports business communities can evaluate which parts of the website need to be improved on behalf, which are the potential customers, from which IP or area or region website is getting maximum hits, etc. which will be help in designing future business and marketing plans.

III. EXISTING SYSTEM AND DRAWBACKS

A small file can be defined as the file whose size is significantly smaller than the HDFS block size. The default block size of HDFS is 64MB. HDFS is primarily designed for accessing large files. To read or access the data from a small file causes hopping from DataNode to DataNode and a lot many seeks are required in search of data from the small file. In case there are large number of small files, then each map task processes very little input and for every data there is a map tasks. It imposes a high overhead in the system. In the NameNode, capacity of namespace of the system is limited by physical memory. Files, directories and blocks are name objects in HDFS and each of which requires 150 bytes, as a rule of thumb in Namenode's memory. There are large numbers of small files stored in the system, metadata occupies large portion of the namespace.

Speaking with respect to the application, A website tells a lot about a business. It shows how much thought a business puts into its brand and if it deserves to have a website. Unfortunately, many companies don't value their websites and don't get the complete benefit out of them. They neglect the design, website copy, and other important aspects. They focus only on making sales.This results in a bad website and leaves its visitors in a confusion if the company is really best one to do business with. This is why it's important to have a value proposition. A website needs to tell visitors why their business is the best choice for the visitor, instead of sending a lot of different messages that won't be received.

IV. PROPOSED SYSTEM

According to this problem analysis, the proposed approach was merging small files into larger ones. This will reduce the number of files. This is similar to "Merging" solution. But this technique was time consuming. So in order to reduce the time consumption, small files can be combined in parallel using the MapReduce paradigm where the Mapper will fetch the file, consider the key as file size and value as filename, pass it to Reducer. The Mapper would keep on adding the files until the default block size is reached and then pass it to Reducer as shown in fig. The Reducer will merge the files. This process will then be carried out parallel till all the files are combined. This will reduce the time in merging and executing the files. In this approach, it ignores those files to merge which are already larger than threshold and default threshold for this algorithm is set to (0.8) 80% of block size of Hadoop. It is also possible to give threshold as parameter to input to the algorithm as per requirement. It must be any integer between ranges of 0 to l

Coming to the application, we also propose a system which calculates the website worth of all the websites an user has subscribed to.

A web ranking metric, web analytics or simply web measurement refers to a system used to measure factors that affect a website's exposure and traffic on the web. The measurable factors generally correlate to web traffic, as web traffic itself has a direct effect on how others see a website's value.

Some notable factors, as noted by blogging webmasters, include page views, unique visitors and user sessions. Although some webmasters consider hits as measurable factors, they're generally unreliable as an indicator of true web traffic.

The web ranking metric 'system' serves a variety of different purposes to webmasters. Some factors are important enough to more or less prove the effectiveness of their corresponding marketing and/or advertisement goals.

V. SUPPORTING DIAGRAM FOR REFERENCE

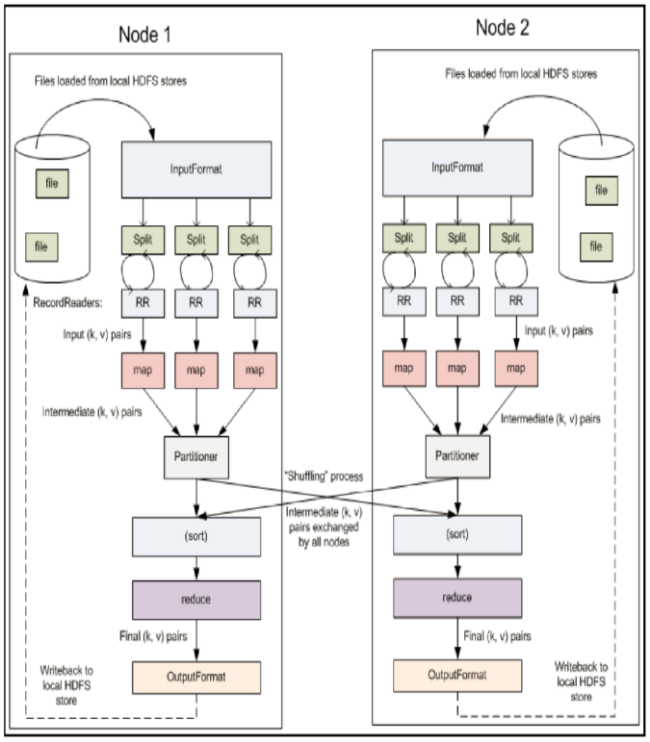

Fig. 1: Map reduce Data flow

VI. MAJOR ADVANTAGES OF PROPOSED SYSTEM

- The System can handle both sequence file and text file efficiently and also avoid files whose size is greater than threshold.
- The system performs really well even for small files
- Heavy burden on the name node is reduced and the execution time of MapReduce is minimized
- This system is capable of providing batch analysis with minimum response time.
- System has memory computing capacity in order to process the traffic estimation in high available.

VII. CONCLUSION

Hadoop is a wide area of research and one of the topics of research is handling of small files in HDFS, the following research focuses on a MapReduce approach to handle small files, considering mainly two parameters viz., the execution time to run file on Hadoop Cluster and the memory utilization by NameNode. Considering these parameters, proposed algorithm enhances the result compared to existing approaches. Hadoop MapReduce framework provides

All rights reserved by *www.ijsrd.com*

parallel distributed computing and reliable data storage by replicating data for large volumes of log files. Firstly, data get stored block wise in rack on several nodes in a cluster so that access time required can be reduced which saves much of the processing time and enhance performance. A Hadoop system capable of handling small files with minimum performance penalty and to analyze the website traffic with minimum response time and high availability.

ACKNOWLEDGEMENT

This research was supported by Department of ISE, New Horizon College of Engineering. We are thankful to our guide Mrs. Swathi.B, Faculty, Dept. of ISE, who provided the necessary facilities for the preparation of the paper.

REFERENCES

[1] Parth Gohil, Bakul Panchal, J.S. Dhobi,"A Novel Approach to Improve the Performance of Hadoop in Handling of Small Files" Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference.
[2] Hemant Hingave, Prof. Rasika Ingle, "An approach for MapReduce based Log analysis using Hadoop" Electronics and Communication Systems (ICECS), 2015 2nd IEEE International Conference.
[3] Jeffrey Dean and Sanjay Ghemawat, MapReduce: Simplied Data Processing on Large Clusters, Google InC.2004
[4] Dhruba Borthakur, The Hadoop Distributed File System: Architecture and Design, ACM 2006 Radia, Robert Chansler "The Hadoop Distributed File System" IEEE 2010
[5] Bo Dong, Jie Qiu, Qinghua Zheng, Xiao Zhong, Jingwei Li, Ying Li "A Novel Approach to Improving the Efficiency of Storing and Accessing Small Files On Hadoop: a Case Study by PowerPoint Files " 2010 IEEE International Conference on Services Computing
[6] Apache hadoop framework, http://hadoop.apache.org July 2008
[7] Small-file-problem-in-hadoop.html, http://amilaparanawithana.blogspot.in/2012/06/small-file-problem-inhadoop.html, June 2012
[8] "Analysis of execution log files". Internet: http://www.orzota.com/wpcontent/ uploads/2014/04/loganalysis-paper.pdf [April. 08, 2014].

All rights reserved by *www.ijsrd.com*