IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

EFFICIENT MAP STORING AND REDUCE READING FOR RELATED BIG DATA BATCH TASKS

Ravneet Kaur Sidhu*, Charanjiv Singh

* Research Scholar, Department of Computer Engineering, Punjabi University, Patiala, India.
Assistant Professor, Department of Computer Engineering, Punjabi University, Patiala, India.

ABSTRACT

Big Data has come up with aureate haste and a clef enabler for the social business. Big Data is bringing a positive change in the decision making process of various business organizations. With the several offerings Big Data has come up with several issues and challenges which are related to the Big Data Management, Big Data processing and Big Data analysis. In Big Data definition, Big means a dataset which makes data concept to grow so much that it becomes difficult to manage it by using existing data management concepts and tools. To store and process such huge data the Hadoop framework uses MapReduce algorithms that work on the files distributed over cluster of computers in HDFS. Map Reduce is playing a very significant role in processing of Big Data. This paper includes a brief about Big Data and its related issues, emphasizes on role of MapReduce in Big Data processing; aim at finding a solution to improve the processing time and proper utilization of resources. MapReduce is elastic scalable, efficient and fault tolerant for analysing a large set of data.

KEYWORDS: - Big Data, MapReduce, Hadoop Distributed File System, Hadoop.

INTRODUCTION

Large Volume of Data is growing because the organizations are continuously capturing the collective amount of data for better decision making process**.** Volume of data increases by online contents like blogs, posts, social networking sites, photos created by the users and servers continuously record the messages about what the online users are doing. Today's Business is unexpectedly affected by this growth of Data. Every day 2.5 quintillion bytes of data are created according to the estimation done by IBM and it is very large amount so the 90% of data in the world has been created in last 3 years [3]. It is a mind boggling figure.

Various industrial organizations intend to make sense from the massive arrival of Big Data to develop the analytic platforms for producing the traditional structure data which includes semi-structured and unstructured sources of information. So in this manner Industrial organizations can take the advantage of Big Data processing for better decision making process. Success of Big Data depends on its analysis. Big Data analysis is an ongoing process for this required a set of activities instead of an isolated activity. Therefore for finding the Data and determining the new insights then integrating and presenting those new insights properly to achieve unique goals required a unified set of solutions.

MapReduce has come up as a highly effective and efficient tool for Big Data analysis; Reasons behind the popularity of MapReduce is its unique features which includes simplicity and communicative manners of its programming model as MapReduce has mainly two functions map ( ) and reduce ( ) even though a large number of data analytical tasks can be expressed as a set of MapReduce jobs. MapReduce can also become more efficient by making proper tuning of various performance affective parameters.

The rest of this paper is organized as follows: section II represent applications of big data analytics, section III challenges related to Big Data, section IV and section V, talks about HDFS and the contribution of MapReduce in Big Data Processing. Section VI highlights on Literature Review of the paper. Finally section VII concludes the paper.

APPLICATIONS

Big Data have various different applications. Taking examination of information on social networking sites for instance, gigantic measure of interpersonal organization information is being created by Twitter, Facebook, LinkedIn and YouTube. This information uncovers various singular's qualities and has been abused in different fields. What's more, the online networking and Internet contain monstrous measure of data on the purchaser inclinations and confidences, driving monetary markers, business cycles, political demeanours, and the financial and social conditions of a general public. It is foreseen that the informal organization information will keep on blasting and be misused for some new applications.

Several other new applications that are becoming possible in the Big Data analytics include:

Customized services: With more personal data collected, commercial enterprises are able to provide customized services adapt to individual preferences. For example, Target (a retailing company in US) predicts customer's need by analysing the transactions.

Online Security: When a network-based attack takes place, historical data on network traffic may allow us to efficiently identify the source and targets of the attack.

Health and Medicines: More and more health related metrics such as individual's molecular characteristics, human activities, human habits and environmental factors are now available. Using these pieces of information, it is possible to diagnose an individual's disease and select individualized treatments.

Digitization: Nowadays many archives are being digitized. For example, Google has scanned millions of books and identified about every word in every one of those books.

BIG DATA CHALLENEGS

This is the world of Big Data, Web logs, sensor network, Internet texts and Documents, Internet search indexing, CRD (call records details) etc. these all includes Big Data. For making more effective decisions and taking the full advantage of available information, it is required to process Big Data efficiently but Big Data includes problems in capturing, searching, storing, sharing, analysing and visualizing of huge amount of data.

Here is a glance on Big Data challenges:

BIG DATA MANAGEMENT AND STORAGE

In Big Data Big means the size of data is growing continuously, on the other hand storage capacity is much less than the rising amount of Data. The reconstruction of available information framework is needed to form a hierarchical framework because Researchers has come up with the conclusion that available DBMSs are not adequate to process the large amount of data. Architecture commonly used for processing of data uses the database server, Database server has constraint of scalability and cost which are prime goals of Big Data.

Different business models has been suggested by the providers of database but basically those are application specific for example Google seems to be more interested in small applications. Big Data Storage is another big issue in Big Data management as available computer algorithms are sufficient to store homogeneous data but not able to smartly store data comes in real time because of its heterogeneous behaviour. So how to rearrange Data is another big problem in context of Big Data Management. [6]

SPEED OF CALCULATION

Speed is a prime requirement when a process query in Database but the process may take time because in short time it cannot traverse all the relevant data in database, Use of indices is most favourable solution but Index is only intended simple type of data in Big Data but Big Data is not only having simple type of data as it is facing complications. So to solve the same can take the combination of Index for Big Data and advanced pre-processing techniques. For the same to solve Big Data problems can use application parallelization and divide-and conquer both are natural computational process but require to add some extra computational resources which is not that easy.

BIG DATA PRIVACY PROTECTION

IT companies are using online Big Data applications for sharing information and reducing the cost, in this way security and Privacy badly affect the Big Data storage and processing as third party services are involved to host private information and to perform computation tasks on this available data. Big Data is growing in continuation and brings lots of challenges of dynamic data monitoring and security protection. Security in Big Data means to process the data mining without exposing the sensitive information of individuals. Data is always dynamically changed as an example Variations of attributes, addition of new data, thus it is challenging to implement effective privacy protection for this complicated situation while current technologies of Privacy protection is mainly based on static data.

Re-PROCESSING OF SAME DATA FOR REALTED TASK

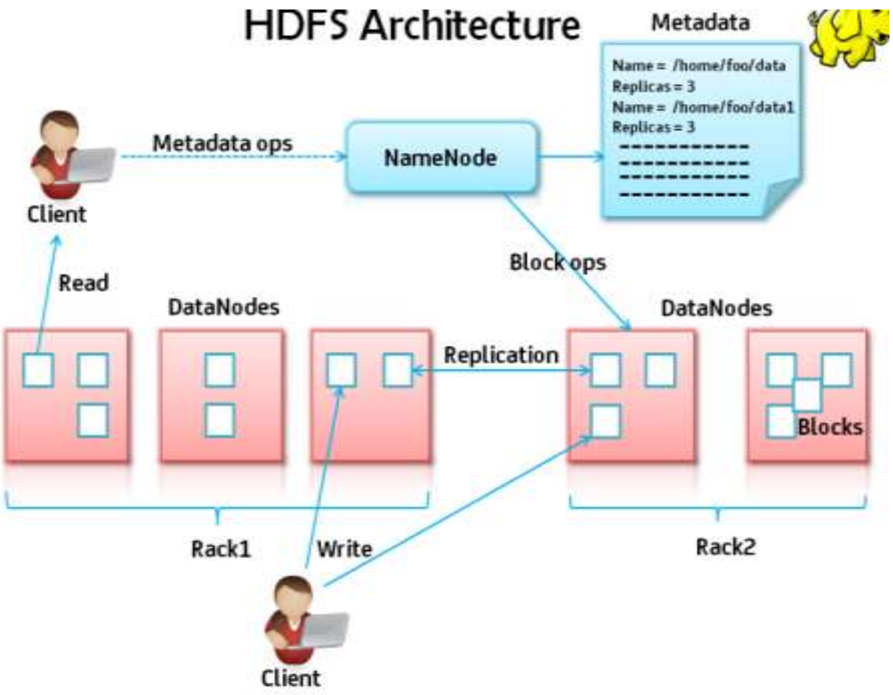
Hadoop offers a framework where the big data stored in HDFS is sorted and then analysed by mapper and reducer function discussed later in the paper. When the user has to work on similar data, it results in additional increase of processing time and resources to sort same data again and again.

HDFS

HDFS [5], the Hadoop Distributed File System, is a distributed file system designed to hold very large amounts of data (petabytes or even zettabytes), and provide high-throughput access to this information. Records are put away in an excess manner over various machines to guarantee their strength to disappointment and high accessibility to extremely parallel applications. Specifically, it guarantees Big Data's strength to failure and high accessibility to parallel applications.

Figure 1 shows the master/slave architecture of HDFS. An HDFS [5] cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The size of these blocks is same and is set prior to the conversion of a file to HDFS.

Figure 1:



HDFS architecture

The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.

HDFS is built using the Java language; any machine that supports Java can run the NameNode or the DataNode software. An HDFS cluster is comprised of a NameNode which manages the cluster metadata and DataNodes that store the data. Files and directories are represented on the NameNode by inodes. Inodes record attributes like permissions, modification and access times, or namespace and disk space quotas.

The file content that is split into large blocks (typically 128 megabytes), is independently replicated (each block) at multiple DataNodes. The blocks are stored on the local file system on the datanodes. The Namenode actively monitors the number of replicas of a block. When a replica of a block is lost due to a DataNode failure or disk failure, the NameNode creates another replica of the block. The NameNode maintains the namespace tree and the mapping of blocks to DataNodes, holding the entire namespace image in RAM.

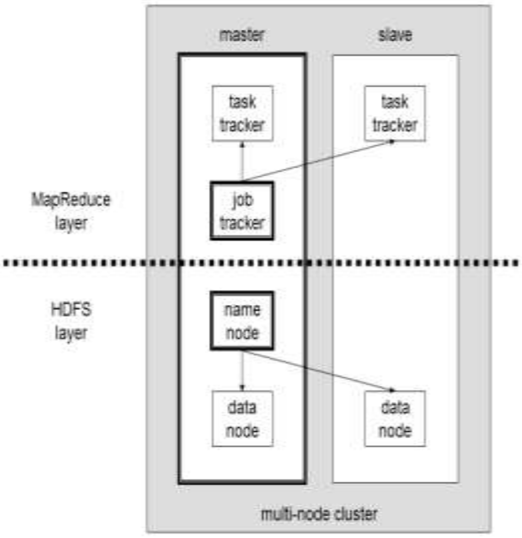
The NameNode does not directly send requests to DataNodes. It sends instructions to the DataNodes by replying to heartbeats sent by those DataNodes. The instructions include commands to: replicate blocks to other nodes, remove local block replicas, re-register and send an immediate block report, or shut down the node.

CONRIBUTIONS OF MAP-REDUCE IN BIG DATA PROCESSING

The business and scientific applications need to process and analyse a large volume of data for making better decision. There is required to process terabytes of data in efficient manner on daily bases as explained before in the introductory part of paper, Industries faced the Big Data problem due to the inability of conventional database systems and software tools to manage and process this large amount of data in tolerable time limits. MapReduce is a playing a promising role for processing and managing the Big Data. MapReduce system is well known because of its elastic scalability and fault tolerance behaviour.

MapReduce[14] is the open source software that allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster. It is a simple programming model for processing huge data sets in parallel. MapReduce have master/slave architecture this is shown in figure 2. The basic notion of MapReduce is to divide a task into subtasks, handle the sub-tasks in parallel, and aggregate the results of the subtasks to form the final output. Programs written in MapReduce are automatically parallelized: programmers do not need to be concerned about the implementation details of parallel processing. Instead, programmers write two functions: map and reduce. The map phase reads the input (in parallel) and distributes the data to the reducers. Auxiliary phases such as sorting, partitioning and combining values can also take place between the maps and reduce phases.

Figure 2:



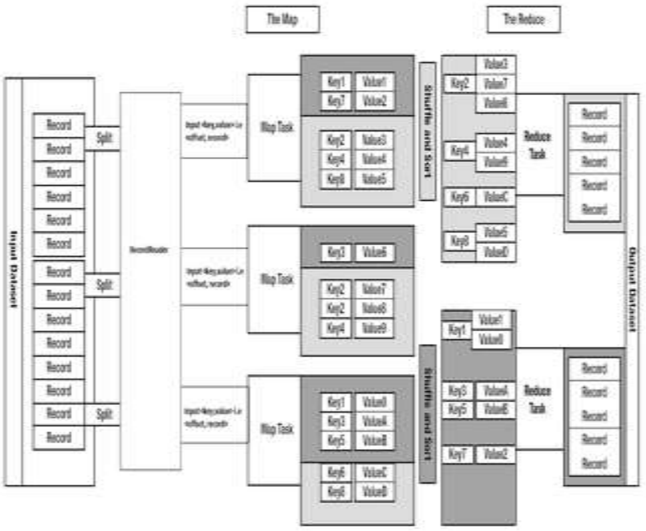
MapReduce Master/Slave

The map job takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job. Refer figure 3

Mapper function

Mapper maps input key/value pairs to a set of intermediate key/value pairs. Maps are the individual tasks that transform input records into intermediate records. The transformed intermediate records do not need to be of the same type as the input records. A given input pair may map to zero or many output pairs. The Hadoop Map/Reduce framework spawns one map task for each Input Split generated by the InputFormat for the job. In short it involves sorting of data that need to be analysed.

Figure 3:



MapReduce Architecture

Reducer function

Reducer reduces a set of intermediate values which share a key to a smaller set of values. The number of reduces for the job is set by the user via JobConf.setNumReduceTasks(int). Overall, Reducer implementations are passed the JobConf for the job via the JobConfigurable.configure (JobConf) method and can override it to initialize themselves. The framework then calls reduce (WritableComparable, Iterator, OutputCollector, Reporter) method for each <key, (list of values)> pair in the grouped inputs. Applications can then override the Closeable.close () method to perform any required cleanup. [7]

LITERATURE SURVEY

McKinsey's Business Technology Office and McKinsey Global Institute(MGI) has conducted a study on BIG DATA in 5 Domains – healthcare in US , the public sector in Europe, retail in the United States and manufacturing, personal –location data globally. And their study proves that Big Data can generate value in each listed domain. For example by use of BIG DATA a retailer can increase its operational margin by 60% [17].In a study by McKinsey's Business Technology Office and McKinsey Global Institute
(MGI) firm that the U.S. faces a shortage of 140,000 to 190,000 people with analytical expertise and 1.5 million managers and analysts with the skills to understand and make decisions based on the analysis of Big Data [9].MapReduce plays a significant role for solving the issues related to Big Data. MapReduce was proposed by Dean and Ghenmawat in [4] as a programming model for processing large amount of Data. MapReduce can also use as analytic tool for processing relational queries, It has been confirmed [10] [11] [12] [13] [14].
Hellerstein, Joe in 2008 worked on parallel programming in the age of big data choosing MapReduce. According to him, rather than requiring the programmer to unravel an algorithm into separate threads to be run on separate cores, parallel databases let them chop up the input data tables into pieces, and pump each piece through the same single-machine program on each processor. This "parallel dataflow" model makes programming a parallel machine as easy as programming a single machine. And it works on "shared-nothing" clusters of computers in a data centre [8]. The machines involved can communicate via simple streams of data messages, without a need for an expensive shared RAM or disk infrastructure. The MapReduce programming model has turned a new page in the parallelism story. In the late 1990s, pioneering web search companies built new parallel software infrastructure to manage web crawls and indexes. As part of this effort, they were forced to reinvent parallel databases –- in large part because the commercial database products at the time did not handle their workload well. Like SQL, the MapReduce framework is a parallel dataflow system that works by partitioning data across machines, each of which runs the same single-node logic.
SQL provides a higher-level language that is more flexible and optimizable, but less familiar to many programmers. MapReduce largely asks programmers to write traditional code, in languages like C, Java, Python and Perl. In addition to its familiar syntax, MapReduce allows programs to be written to and read from traditional files in a file system, rather than requiring database schema definitions. MapReduce is such a

compelling entryway into parallel programming that it is being used to nurture a new generation of parallel programmers.

The Velocity dimension, as one of the Vs used to define Big Data, brings many new challenges to traditional data processing approaches and especially to MapReduce. To overcome the inherent limitations of the traditional MapReduce platforms, other authors have been leveraging the familiar MapReduce programming paradigm but additionally providing a different runtime environment. For instance, Logothetis and Yocum [15] proposed a continuous MapReduce in the context of a data processing platform that runs over a wide-area network. In this work, the execution of the *Map* and *Reduce* functions is managed by a data stream processing platform. In order to improve the processing latency, the mappers are continuously fed with batches of tuples (instead of input files), and they push their results to reducers as soon as they are available. This approach is similar to the one adopted by the StreamMapReduce [16] project, which uses these ideas to implement a fully-fledged event stream processing (ESP) implementation.

CONCLUSION AND FUTURE SCOPE

Here concluded that Big Data is playing a very significant role in industries for making better business decisions but the main issue with Big Data is, its analysis, MR is playing a significant role in Big Data analysis and giving an equal competitive edge to the parallel DBMS. The performance of MR can improve by the proper tuning of its performance affecting parameters but still the research work is required in the same direction .Researchers need to provide the proper grouping algorithms, scheduling algorithms to make big data more effective. While running MapReduce on dataset for accomplishing different task it is found that the data produced by Task 1 Mapper is same as needed for Task 2 Reducer. So instead of again running Mapper for Task 2 it is much better to initially store Task 1 Mapper result which can be directly used by Reducer. We need to find efficient way to store the data produced by mapper. We need to change the default behaviour of Reducer to be able to fetch the data stored by our new approach. We finally need to do performance analysis of our approach to previous approach.

REFERENCES

[1] http://hadoop.apache.org.

[2] http://developer.yahoo.net/blogs/hadoop/2008/09/.

[3] Improving Decision Making in the World of Big Data http://www.forbes.com/sites/christopherfrank/2012/03/25/improving-decision-making-in-the-world-of-big-data/.

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

[5] Hadoop Distributed File System (HDFS), http://hortonworks.com/hadoop/hdfs/

[6] Y. Bu, B. Howe, M. Balazinska and M. D. Ernst, "HaLoop: Efficient iterative data processing on large clusters," Proc.VLDB Endow, 3(1-2), pp. 285-296, 2010

[7] Jeffery Dean ,Sanjay Ghemawat .An Article on MapReduce:A Flexible Data Processing Tool.In SigMOD pages3(1):72 75{ACM 2010 }.

[8] https://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-144.html

[9] Big Data: The next frontier for competition http://www.mckinsey.com/features/big_data.

[10] H.-C. Yang, A. Dasdan, R.-L.Hsiao, and D. S. Parker. Map-Reduce-Merge: simplified relational data processing on large clusters. In SIGMOD, pages121-134{2007}.

[11] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A.Tomkins. Pig latin: A note on foreign language for data processing. In SIGMOD, pages 1099-1110{ ACM,2008}.

[12] R. Chaiken, B. Jenkins, P.-A. Larson, B. Ramsey,D. Shakib, S. Weaver, and J. Zhou. Scope: Easy and efficient parallel processing of massive data sets. In.PVLDB, 1(2):pages1265-1276{2008}.

[13] D. J. DeWitt, E. Paulson, E. Robinson, J. Naughton, J.Royalty, S. Shankar, and A. Krioukov. Clustera: An integrated computation and data management system...In Proc. VLDB Endow., 1(1):pages28-41{2008}.

[14] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wycho®, and R. Murthy. Hive – A warehousing solution over a map-reduce framework. In PVLDB, 2(2):1626-1629{2009}.

[15] D. Logothetis and K. Yocum, "Ad-hoc data processing in the cloud," Proc. of the VLDB Endowment, 1(2), pp. 1472-1475, 2008.

[16] A. Brito, A. Martin, T. Knauth, S. Creutz, D. Becker, S. Weigert and C. Fetzer, "Scalable and low-latency data processing with stream MapReduce," IEEE Third International Conference on Cloud Computing Technology and Science, pp. 48-58, 2011.

[17] Big Data: The next frontier for innovation, competition and productivity. http://www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation.