Fundamental Survey of Map Reduce in Bigdata with Hadoop Environment

Maulik Dhamecha[1(✉)] and Tejas Patalia[2]

[1] Computer Engineering, Gujarat Technological University, Ahmedabad, Gujarat, India
mvdhamecha@gmail.com
[2] Computer Engineering Department, VVP Engineering College, Gujarat Technological University, Rajkot, Gujarat, India
pataliatejas@rediffmail.com

Abstract. The terminology "Big Data" was initiated for variety of industry processes, methods and technology to explore new field. Big organizations like Amazon, flip cart and also many government subsidiaries like ISRO, NASA and BISAG are considering Big Data to fulfill their analytical objectives with mapping technique and reducing technique. We can consider Big Data as key factor related large or small-sized data repositories and consortium which have been identifies the possible (which is random manner extensively) to make capital out of. And for that hadoop is very effective platform to shows the efficiency of map reduce technique.

Keywords: Mapreduce · Mapping technique · Reducing technique Hadoop

1 Introduction

Into the computational market the term "big data" is very usually used term. In a data world, Big Data means every one's data. It is the Knowledge or relevant fact considered by particular organization, gathered and deal with recent methods or procedure to generate best result in precise manner.

As we start the learning about Big Data, data analytics will elaborate the subject precisely likely to talking about "The three V's" - "volume, velocity and variety," this points which identifies the challenge of data analytics word. In brief, it is big amount of attribute processing in fast and various manner. User's executable processing records, databases of production for organization, influx of web log files, live video file, current media's user conversations and many more could be involved. Mark van Rijmenam told about main 3 V's that "Why the 3 V's Are Not Sufficient to Describe Big Data," also says that "veracity, variability, visualization, and value" to the definition. Rijmenam also says that "90% of all data ever created, was created in the past two years. From now on, the amount of data in the world will double every two years" [1, 4].

A software framework for distributed processing of large data sets on compute clusters of commodity hardware is known as Hadoop MapReduce (Hadoop Map/Reduce)" [2]. It is a partial part of the Apache Hadoop. Organizing endeavor,

S. Verma et al. (Eds.): CNC 2018, CCIS 839, pp. 215–222, 2019.
https://doi.org/10.1007/978-981-13-2372-0_19

surveil them and re-surveil any crashed tasks ware taking care about the framework. “According to The Apache Software Foundation, the primary objective of Map/Reduce is to split the input data set into independent chunks that are processed in a completely parallel manner” [1, 15]. The framework of Hadoop MapReduce can organize the output data for mapping class, which output becomes input of reduce class. So, all incoming data and the outgoing data relevant of tasks are saved in a resulted file system.

2 Hadoop Features and Characteristics

Apache Hadoop is the most powerful and very precise big data tool. “Hadoop provides the most reliable storage layer – HDFS, a batch processing engine – MapReduce and a Resource Management Layer – YARN” [4]. There are some important Hadoop Features which are stated as below-

Open source: Apache Hadoop is known as open source project. So, we can modified code according to business requirements.
Distributed processing: data is processed in parallel way on a cluster of different nodes because can be saved in a distributed manner in Haoop Distributed File System across that cluster of nodes [7].
Fault Tolerance: There are 3 replicas of each block was stored across that cluster in Hadoop in default manner and it can also be modified also as per the given requirement. So if any node will become down, data can be recovered from any other nodes in easy manner. [2, 11] By the framework, automatically data can be recovered if failures of nodes or tasks are occurred. This is how we can says this is one of the important feature of Hadoop.
Reliability: Data is satisfactory stored on the different nodes of cluster in a case machine failures due to replication. [15] If your machine will going to fail for work then also your data will be stored satisfactory.
High Availability: Due to number of copies of data is large, data can be available and ready to despite of hardware failure. If any hardware crashes of machine will happen, then data will be accumulate from another pathway also [15].
Scalability: whenever any new hardware can be easily accommodate to the given nodes we can say that hadoop is highly scalable. When any extra nodes can be accommodate on the urgent way without any downtime, we can says that it also provides horizontal scalability [7, 15].
Economy: As we apply it of commonly connected hardware, we can says that Apache Hadoop is not very expensive. Cluster have no need any specialized node for it. Hadoop gives us huge cost cutting and also as it is very simple to accumulate more machines on that cluster over here. [9] So whenever any requirement will increases in time manner, you can also put up the machines and also without any alteration and without so much pre-planning for that.

Data Locality: "move computation to data instead of data to computation" - Hadoop is working on this fundamental principle of data locality. [8] Whenever any user will submits the MapReduce process, this process will moved to data in that cluster in place of transferring data to that place where the process is submitted.

3 Functionality of Map-Reduce

We can say that Apache Hadoop, Mapreduce is the heart. It will behave like the programming structure which permit for extendibility into the big number of clusters of a Server. We can say that MapReduce paradigm is comparatively easy to catch for the known users about the scaling processing of cluster.

Understanding for new users can be difficult about this topic, because it is not generalize concept for people to have been expand as n former times. If you are new for the Hadoop MapReduce tasks, don't be hesitate, we will try to get knowledge in it that you will pick up it quickly.

"This term MapReduce is refers two different and distinct tasks that Hadoop programs platform. [9]"As shown in Fig. 1, Map job is initial task, which gets a cluster of attributes and transfer it into different cluster of attributes, into that every attributes are tumbledown into attributes like key pairs or value pairs. Outgoing data going to the reduce job through a map function as incoming and segregate particular data attributes like key pairs or value pairs into a small chunks of attributes. Name of MapReduce itself says that, every time reduce job is occurred only after the completion of the map job.

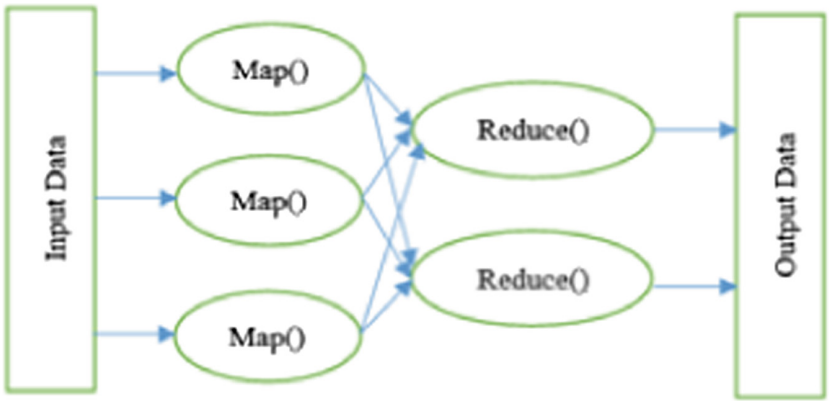

Fig. 1. Basics of Map-reduce

The fundamental of Hadoop says that "The MapReduce framework works mainly on <key, value> pairs, that is, the framework views the input to the task as a pair of <key, value> set and produces a pair of <key, value> sets as the output of the task, producing of different types [15]".

The key classes and value classes have been put in sequentially by the Hadoop framework and that's for that reason necessity to implant the Writable interface. In Addition, the set of key values will implant the accessible pathway to ordering by the framework.

Here is the Input & Output of a MapReduce job:

"(input) <key1, value1> -> map -> <key2, value2> -> combine -> <key2, value2> -> reduce -> <key3, value3> (output)" [4].

Commonly the MapReduce concept is concern about the data sending to the node at which place the result is placed. MapReduce processed in three ways, first is mapping, shuffling and reducing.

Mapping Stage: The input data process is known as map or mapper's task. Data which are file or directory typed which are stored in the Hadoop distributive file system (HDFS) as input. The input data of file is going through the mapper function interpretably. This data would be processed by the mapper stage and creates into many small parts of this data.

Reducing Stage: "This stage is the combination of the two stage: the Shuffle stage and the Reducing stage" [9]. The main task of Reducer's is to get the data from mapper stage and process that data. After completion of this process, it produces a new group of output, which will saved in the Hadoop distributive file system HDFS.

Into the MapReduce task, Hadoop pass the Map and Reduce tasks into the cluster to the significant servers. The knowledge of data-processing such as allocation of tasks to different nodes, verifying that task was completed or not, copying data into the cluster of different nodes and all this will be managed by the framework. Most of this processing or computing takes can place on local space with data that compares the traffic between nodes. After that cluster will gather and compares the data into an appropriate form and sends this resultant data to the server after completion of this given tasks.

4 Proposed Algorithm

The basic function of Map and Reduce of MapReduce paradigm are known related to data organization in form of key and value pairs. Into the Map task, it will get single set of data of single data domain and give it back as a list of key and value pairs in a multiple data domain:

$$\text{Map}\,(\text{key1},\ \text{value1}) \rightarrow \text{list}\,(\text{key2},\ \text{value2})$$

This mapper function is applicable in a serialized way to each key1 and value pair into the input datasets. This will generates a series of key2 and value pairs for every of the call. Completion of this process, this framework gathers all data pairs with the original key2 from the all list, make groups of those data and generate single group for every key.

After then applied the Reduce function in serialized way to every group, which produces the values in collective manner the common domain:

$$\text{Reduce}\,(\text{key2},\ \text{list}(\text{value2})) \rightarrow \text{list}(\text{value3})$$

Though one massage will permit to giving back multiple value of data, every single Reduce massage generate either the value v3 or a none return. This returns of that all massages are gathered as the expected result of the list.

The proposed algorithm is as below.

```
function map (String name, String document):
 // name: document or file name
 // document: document or file contents
 for each word w in document or file:
  emit (w, 1)

function reduce(String word, Iterator partial counts):
 // word: a word
 // partial Counts: a list of aggregated partial counts
 sum = 0
 for each pc in partial Counts:
  sum += pc
 emit (word, sum)
```

4.1 Example

Here we can try to understand this topic with an example. Suppose user contain four files of data and particular file have two columns data (“a key and a value in Hadoop terms”) that shows the value as name of city and relevant temperature of particular city for the different days. As we are taking this data just as an example so it is simple to understand. We can understand that any real time application is not so simple because we are talking about big data as it was containing record numbers of rows and may be it was not be preprocessed this data. Here no matter that what amount of data we are analyzing, the unique principles we are highlighting here is remain as it is. And for this example, “we had taken city is the key and temperature is the value.

Ahmedabad, 21
Vadodara, 26
Rajkot, 23
Surat, 33

Here, we have to find highest temperature of every city from every collection from all the data we have collected. We can cut down this in 5 major mapping tasks, at there every single map task works on 1 out of 5 given files, also map task going into the data and gives back with the maximum temperature value for every particular city using the MapReduce framework. The final value is look like as below:

$$(\text{Ahmedabad}, 21)(\text{Vadodara}, 26)(\text{Rajkot}, 23)(\text{Surat}, 33)$$

Think that given four other mapper tasks generate the below interim results:

(Ahmedabad, 18) (Vadodara, 27)
(Rajkot, 32) (Surat, 37)
(Ahmedabad, 32) (Vadodara, 20)
(Rajkot, 33) (Surat, 38)
(Ahmedabad, 22) (Vadodara, 19)
(Rajkot, 20) (Surat, 31)
(Ahmedabad, 31) (Vadodara, 22)
(Rajkot, 19) (Surat, 30)

Into the reduce tasks, feed all output stream of these files which added the results of all input data and single value of output data of particular every city, produced the executable output set as below:

$$(\text{Ahmedabad}, 32)(\text{Vadodara}, 27)(\text{Rajkot}, 33)(\text{Surat}, 38)$$

Similarly, you may also find an example of "census was conducted in Roman where the census bureau would dispatch its people to each city in the empire for map and reduce tasks. Work is like each census taker in each city would be tasked to count the number of people in that city and return the results to the capital city."

"Thus, the results from each city will be reduced to a single count (sum of all cities) to calculate the overall population of the empire. This mapping of people of cities, in parallel way and then combining the results is more efficient than sending an alone person to count every person in the empire in a serial" [16].

Here in Table 1, some basic commands are given for implementation of this basic map-reduce functionality.

Table 1. Hadoop basic commands [8]

Options	Description
"namenode –format"	"Formats the DFS filesystem"
"secondarynamenode"	"Runs the DFS secondary namenode"
"namenode"	"Runs the DFS namenode"
"datanode"	"Runs a DFS datanode"
"dfsadmin"	"Runs a DFS admin client"
"mradmin"	"Runs a Map-Reduce admin client"
"fsck"	"Runs a DFS filesystem checking utility"
"fs"	"Runs a generic filesystem user client"
"balancer"	"Runs a cluster balancing utility"
"oiv"	"Applies the offline fsimage viewer to an fsimage"
"fetchdt"	"Fetches a delegation token from the NameNode"
"jobtracker"	"Runs the MapReduce job Tracker node"
"pipes"	"Runs a Pipes job"
"tasktracker"	"Runs a MapReduce task Tracker node"
"historyserver"	"Runs job history servers as a standalone daemon"
"job"	"Manipulates the MapReduce jobs"
"queue"	"Gets information regarding JobQueues"
"version"	"Prints the version"
"jar < jar>"	"Runs a jar file"
"distcp < srcurl > < desturl>"	"Copies file or directories recursively"
"distcp2 < srcurl > < desturl>"	"DistCp version 2"
"archive -archiveName NAME –p"	"Creates a hadoop archive"
"classpath"	"Prints the class path needed to get the Hadoop jar and the required libraries"
"daemonlog"	"Get/Set the log level for each daemon"

5 Conclusion

When we are talking about the processing of large amount of data sets, Hadoop MapReduce paradigm accessible for the scheduling such large amount of data in a protective and efficient way. Hadoop is a highly scalable platform and behind scalability, the main reason is to save and distribute huge amount of data in different servers. While every addition of data servers it can increase more processing throughput.

Acknowledgement. First of all I would like to thank the VVP Engineering College, Rajkot, Gujarat, India for providing me suitable working environment and because of this I am able to find the scope of my research work. With the use of the practical laboratory environment of VVP Engineering College, Rajkot, Gujarat I was able to get the results of my research work.

I am also very great full to Dr. Tejas Patalia to guide me throaty in my research. He is guiding me time to time for improvement in my research work and motivate me to work deep in this keen

area of research. Without his kind support it's not possible for me to continue my research journey. In last, I am thankful to all my colleague for supporting me and encourage me in my research.

References

1. Godhani, G., Dhamecha, M.: A study on movie recommendation system using parallel MapReduce technology. IJEDR **5**, 7683–7692 (2017)
2. Song, G., Meng, Z., Huet, F., Magoules, F., Yu, L., Lin, X.: A hadoop MapReduce performance prediction method. In: IEEE (2013)
3. Shvachko, K., Kuang, H., Radia, S., Chansler, R.: The hadoop distributed file system. IEEE (2010)
4. Subramaniyaswamy, V., Vijayakumar, V., Logesh, R., Indragandhi, V.: Unstructured data analysis on big data using map reduce. Science direct (2015)
5. Dean, J., Sanjay, G.: MapReduce: simplied data processing on large clusters. In: OSID (2004)
6. Dhamecha, M., Ganatra, A., Bhensadadiya, C.K.: Comprehensive study of hierarchical clustering algorithm and comparison with different clustering algorithms. In: CiiT (2011)
7. Cuzzocrea, A., Song, Y., Davis, K.C.: Analytics over large-scale multidimensional data: the big data revolution!. In: ACM (2011)
8. Tungkasthan, A., Premchaiswadi, W.: A parallel processing framework using MapReduce for content-based image retrieval. In: IEEE (2013)
9. Xu, W., Luo, W.: Analysis and optimization of data import with hadoop. In: IEEE (2012)
10. Chandarana, D., Dhamecha, M.: A survey for different approaches of outlier detection in data mining. In: IEEE (2015)
11. Maitrey, S., Jha, C.K.: Handling big data efficiently by using map reduce technique. In: IEEE (2015)
12. Agarwal, P., Shroff, G., Malhotra, P.: Approximate incremental big-data harmonization. In: IEEE (2013)
13. Wang, G., Salles, M.V.: Behavioral simulations in MapReduce. In: IEEE (2010)
14. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. In: Proceedings of the 6th Symposium on Operating Systems Design and Implementation, San Francisco CA (2004)
15. Shvachko, K.V.: HDFS scalability: the limits to growth. In: IEEE (2010)
16. Acharya, S., Chellappan, S.: Big Data and Analytics. Wiley, Hoboken (2015)
17. https://en.wikipedia.org/wiki/Big_data
18. www.bigdatahadoop.info