Indian Journal of Science and Technology, Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/106497, December 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

IPB -Implementation of Parallel Mining for Big Data

Neha Mangla[1]*, K. Sushma[1] and Lithin Kumble[2]

[1]A. I. T., Bangalore - 560 056, Karnataka, India; Apj.neha@gmail.com, sushmakrishna2@gmail.com
[2]REVA University, Bangalore - 560064, Karnataka, India; lithin@revainstitution.org

Abstract

Data in this era is generating at tremendous rate so now it is need of today to handle the data to gain useful insight, this data can be useful for researcher and accommodation to do analysis. As we know traditional system cannot handle more than terabytes of data since it affects performance and also storage is very costly. Big data is an innovative technique analyze, store, manage, distributes and capture datasets. **Objective**: To achieve compressed storage, we implement a parallel mining algorithm called as Implementation of Parallel Mining for Big data. **Method**: Hadoop is a platform which enables the distributing processing using map reduces programming. This help in getting result at very fast rate as result in less time help in competing for growth of business. Unstructured datasets is taken for analysis which is real time is taken and converted to structured format and process in map reduces. It is found in literature existing mining algorithm for the real time datasets which always lacks in fault tolerance, load balancing, data distribution and automatic parallelization. To overcome these disadvantages we implement map reduce for association analysis. **Finding/Improvement**: In IPB we improve performance in the computing node the load is distributed. In our proposed solution we use real-world celestial spectral data. The graphical representation of traditional system comparison with Hadoop is shown in this paper.

Keywords: Big Data, IPB (Implementation of Parallel Mining for Big Data), Map Reduce, Parallel Mining, Hadoop Association Analysis

1. Introduction

Information Technology is growing rapidly and volume of data is increasing such as social media, balck box so on and it is reaching petabytes of data threshold and as increase in data also increases computational requirements which include fault tolerance, load balancing, data distribution and automatic parallelization. In the terms of business and academics big data is become the key role. Here efficient parallel mining algorithm techniques are used to easy and fast processing of data. In this we consider a data mining tool called as R tool to compare with the proposed system where I process our unstructured data perform the association analysis is performed in map reduce then it is being sent to represented by Intelligent graph system in R tool the time required to process and cannot process of huge amount of data is not easy which is great disadvantage

The real time data can be processed in Map reduce. Firstly we generate realistic data by creating developer account and by streaming the data in the flume and the unstructured data is taken to hue and particular data can be searched using the solaries can also change colour, highlights, and bolds. Now the data as to be converted to structured data using Hive and this structured data is fed into Map reduce is a programming model which consists of two phase:-

- Firstly, Mapper phase which is each separate line and produces a key value pair.
- Secondly, to do the association analysis and it is represented using graphical representation. After the all these task performed we compare the two systems, processing speed, availability data the time taken execution of data and many more criteria.

The contributions of this paper are:-

- We made complete overview about parallel mining on the realistic datasets and those datasets where produced to hive structured format was obtained

*Author for correspondence

- We developed a parallel mining method using Map reduce programming model
- We also gave complete overview about existing traditional system R tool and did the association analysis using the datasets.
- We also show the load balancing and how data is being distributed in the clustering nodes and processing is done.
- The comparison of r tool and IPB the system is showed and it is measured in turn of processing speed, scalability, availability, performance and which kind of real world and synthetic which can be processed in these tools.

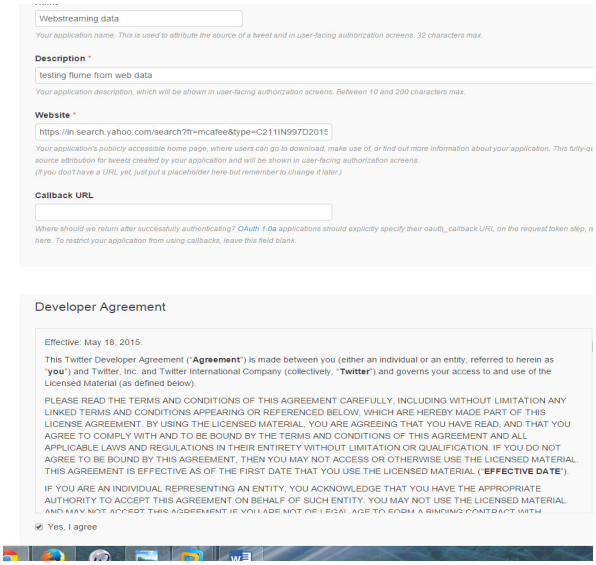

Figure 1. Snapshot of creation of developer account.

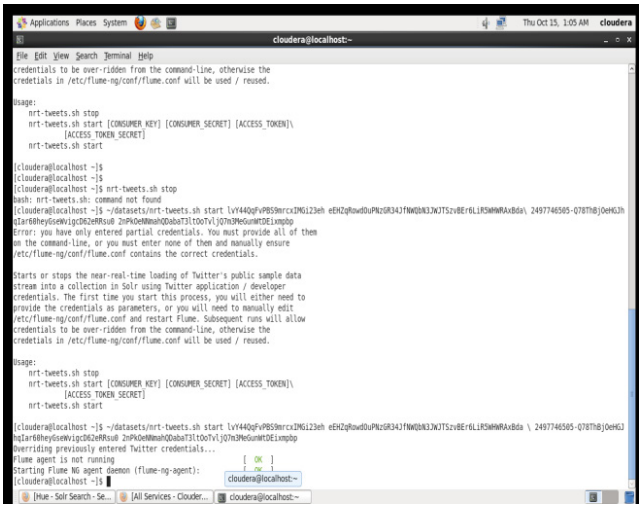

Figure 2. I have developer account with name Web Streaming data with keys and access tokens.

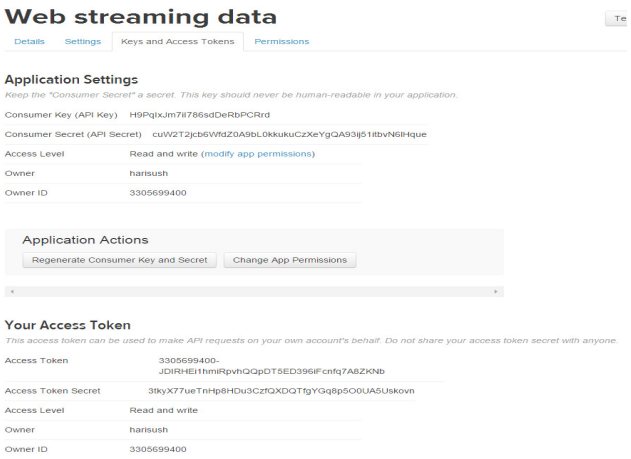

Figure 3. Coding in hadoop to connect twitter with flume.

Flume is a reliable, distributed and available service for efficiently aggregating, collecting and moving the large amount of celestial data and log data.

Flume components interact in the following way:

- Client starts the flow of the flume
- The transmission of the Event is done by the client to a source operating with the Agent
- This Event is received by the Source and delivered to one or more channels.
- Channels can be drained due the one more sink with the same Agent.
- The ingestion rate from the drain rate can be calculated using the producer consumer model of the Channels
- If the data is generated faster from spike in the client side activity if the limited has being exceeded the channel size can be extended and normally the task can be performed
- The source of one agent can be chained to the sink of the agent. The complex data flow can be created.

There is no central coordination point in flume in the distributed architecture flume can be easily scale up horizontally since because the agent runs independently with none of the failure of single point.

Initially we create a gmail account and followed by a Twitter account in this and developer account is opened and goes for the option which is at left corner called manage your app and generate the keys and access token.

2. Queries on Retrieved Data

- Create table sentiment (id int, text string) load data local inpath'/home/cloudera/Desktop/senti.txt' overwrite into table senti;
- Create External Table dictionary (string, stemmed type string, word string, pos, length into, string, polarity string)'\t' Stored as Text file Location '/user/cloudera/upload/upload/data/dictionary;
- Create view l1 as select id, words from senti lateral view explode(sentences(lower(text))) dummy as words;

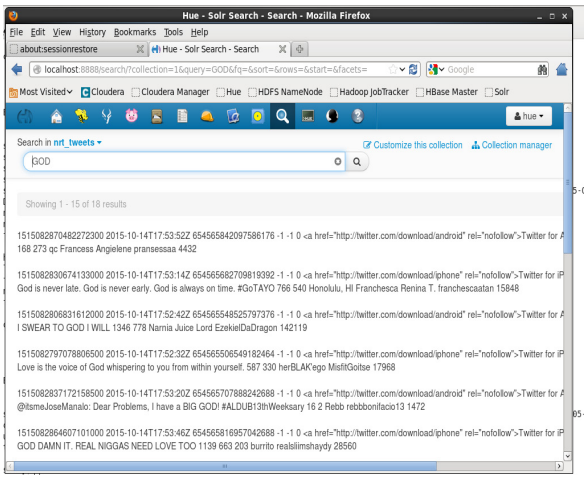

Figure 4. Twitter data retrieved in flume.

The data which is present now is unstructured data shown in Figure 5 which has to be

converted to the structured format where this job is done by the hive platform it can be processed to the map reduce and it is processed for the further analysis.

```
hive> use harish_db;
OK
Time taken: 0.046 seconds
hive> create table senti(id int,text string)
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.812 seconds
hive> load data local inpath '/home/training/Desktop/senti.txt' overwrite into table senti;
Copying data from file:/home/training/Desktop/senti.txt
Copying file: file:/home/training/Desktop/senti.txt
Loading data to table harish_db.senti
Deleted hdfs://localhost/user/hive/warehouse/harish_db.db/senti
OK
Time taken: 0.347 seconds
hive> CREATE EXTERNAL TABLE dictionary (
    >     type string,
    >     length int,
    >     word string,
    >     pos string,
    >     stemmed string,
    >     polarity string
    > )
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    > STORED AS TEXTFILE
    > LOCATION '/home/training/Desktop/dictionary';
OK
Time taken: 0.081 seconds
hive>
```

Figure 5. Query done in Hive platform.

2.1 Existing System

Here we are going to learn about the how the tweets can be extracted using the code in the R tool with the called as twitter R[1,2], we need to install this package first. Later the tweets are converted to data frame then corpus. Next step is used for stemmed words to retrieve their radicals from the twitter data. We need to install few packages they are Snowball, RWeka, Rjava and RWekajars. Now we have to convert this unstructured to structure in terms of matrix. Where here row means terms column means entity we build a corpus processed with function Term Document Matrix ()[7] the code is given below

rdata and file wherever it exsists set this to that

```
library(twitteR)
library(tm)
setwd("../Desktop/")
getwd()
# load dataset
data = load("neha2.RData")
# view length of dataset
n <- length(tweets)
# check out a few tweets
tweets[1:3]
# convert to data frame now
<- wordFreq(dictCorpus , "many")
n.mining= wordFreq(Corpus ,mining)
cat(n.miner, n.mining)
# replace oldword with newword
# we are taken exmple for the given 3 pairs.
#network plot of terms between frequrntly associated words
# strength of edge denotes correlation
library(graph)
library(Rgraphviz)
# change corThreshold to inlude more/less terms in network
plot(tdm, term = freq.terms, corThreshold = 0.08, weighting = T)
# topic modelling
dtm <- as.DocumentTermMatrix(tdm)
library (topic models)
```

It is now list frequent term and association. findFreq-Terms() this function is list the number of frequent less than 10(10 is just a example). Now we will plot the bar graph for the twitter data which is having the maximum tweets and retweets show in Figure 7

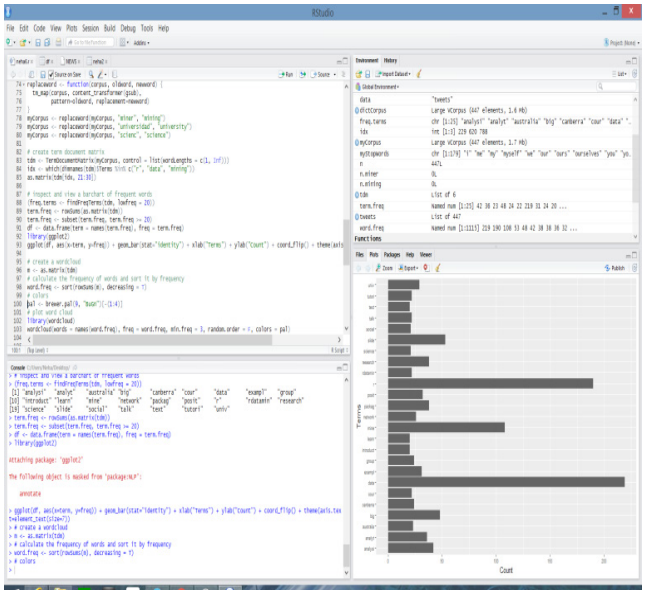

Figure 6. It shows the bar graph the maximum tweets that is happened.

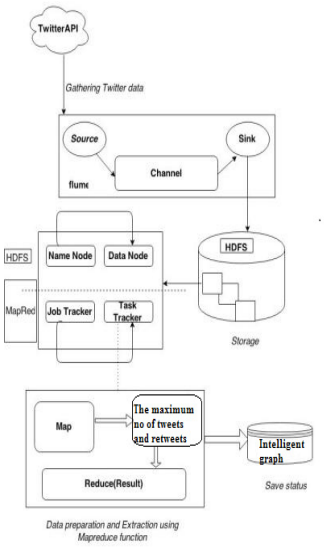

Figure 7. Represents the graph the maximum number of repeated words.

We can also generate the word cloud for the datasets. We can generate the word cloud for those maximum words with maximum frequency. We can generate the networks of the terms for the particular data. Later it is converted to adjacency matrix. This matrix contains the maximum frequencies and it is in the form of table. Figure 7 shows the graph of the maximum number of words repeated in the table and we can build a graph with graph.adjacency()[6] from package igraph.

3. Results and Discussion

In this proposed system we will process the structured data which we had retrieved from the flume platform and it should be processed into the map reduce for the getting the maximum of tweets and retweets and pictorial presentation of it. The complete architecture about how it is working as shown in Figure 8 It is giving the complete as I discussed in early chapter how the data is extracted from the flume the unstructured data later it is converted structured data using the hive platform and it is saved in Hadoop Distributed File System (HDFS)[11] and later it is processed to the map reduce job as show in the Figure 8. In the mapper stage it processes the structured data and the maximum of tweets and retweets done is saved and after the map reduce job is completed successfully then it is saved in HDFS and it is sent to the Intelligent graph system for the result. I have already discussed about fluming of the data in the before chapter itself.

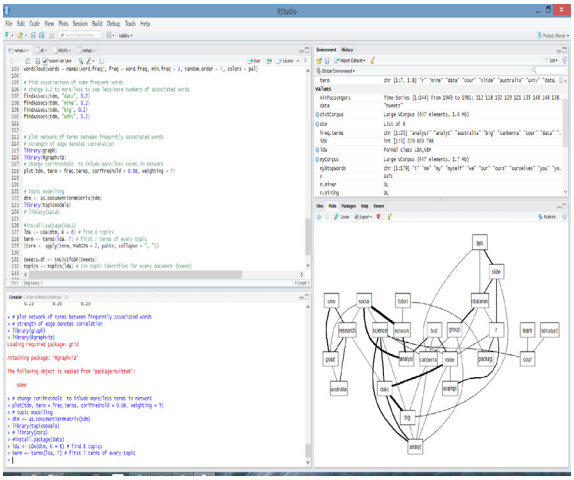

Figure 8. The complete overview about IPB mechanism.

3.1 Map Reduce Model

This model is used in parallel data paradigm. This model consists of 2 stages:- one is the map phase and other is the reducer phase. The data structure which is present in map reduce[7,8] model is Tuple. The data which is sent inside the map reduce task is converted to the tuple format. Each line is taken as the separate chunk into the map reduce task. The map function is a list of <k1,v1> pairs. This is again processed to the reduce stage here it accepts only key value pair. The output from the reduce function is in the form of <k2,v3> pairs. The equation for this process can be represented as:

Map<k1,v1> = list <k2,v2>
Reduce (<k2,list(v2)>) = <k2,v3>

To conclude their will 3 stage in map reduce one is splitting stage, Mapper stage and later is reduction stage:-

- Splitting stage: Here the input is taken from the map reduce and each line is split into multiple chunks and can be processed parallel[6,7]. I can also fix the of the chunk if necessary and their exists a function splitter where I can define my own split rules for the given datasets but by default Hadoop only completes this step.
- Mapping stage: In this stage it will read the data chunk and convert the data into tuples. As I told for the text the input for map function will be the each single line in this file. This will emit out in tuple format and here it is shuffled and sorted and stored in the local file and processed to the reducer stage.
- Reduction stage: After all the map tasks done all aggregation happen in this stage only

List of values is accepted with the same key in the reduce function. We define our own process for the given value. The output is obtained which is in tuple format.

This model can be processed huge amount of datasets and map function and reduce function can parallel on the cluster without draining the performance of the system. Figure 9 shows the complete overview about map reduce and these are performance we are going to measure and discussed in before chapter.

Here the aggregation stage where in this process the given dataset is sent to mapper stage for processing and the association analysis is taken where it accepts the data key value pair in the given system it count the maximum of all tweets present in the table and later it is processed in intelligent graph to obtain the excepted result

Now we will the processing in the map reduce before we start with map reduce we should all the services using the command called as start-all.sh. using jps command for all services started or not and we can also check in hadoop administration shown in Figure 10 and Figure 11. I can create a new file in the HDFS here we have created a file called as fi. Using the command show in the Figure 12 we will copy the twitter data in the file fi and we can weather the data is copied or not in hadoop adminstartion as show in Figure 13. Now it is time to start the map reduce process and in the data their only and wait the reduce to finish 100% as shown in the Figure 14. If the map reduce successful then the fie called success will be created in fi inside the output file as shown in Figure 15. If the process is unsuccessful then it gives error message then and their only.

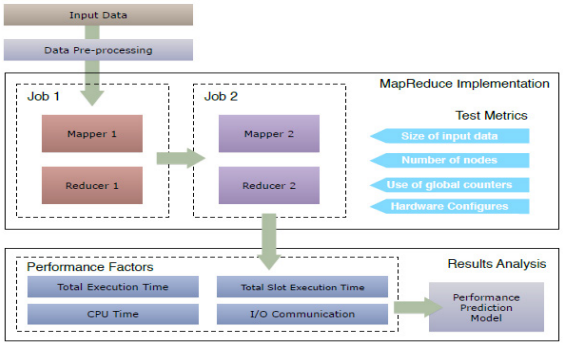

Figure 9. The map reduces structure overview.

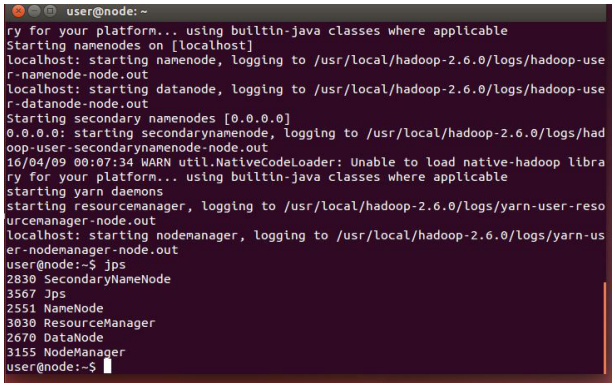

Figure 10. Start all the services.

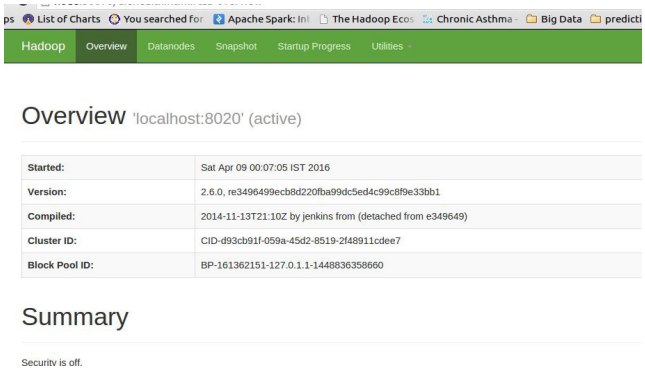

Figure 11. Checking in the hadoop administration services are started or not.

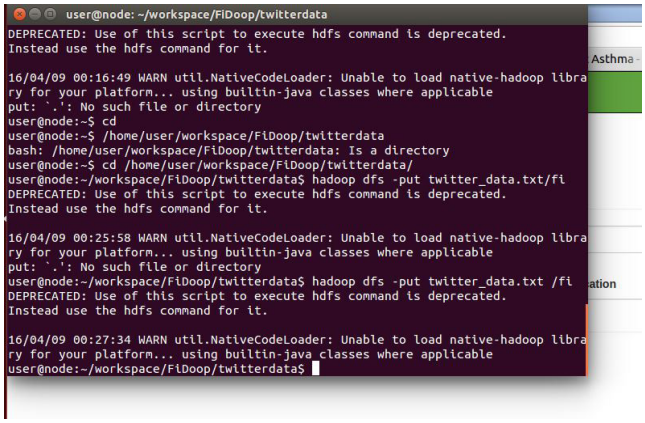

Figure 12. The data is copied to the file.

Figure 13. The copied file in the hadoop administration.

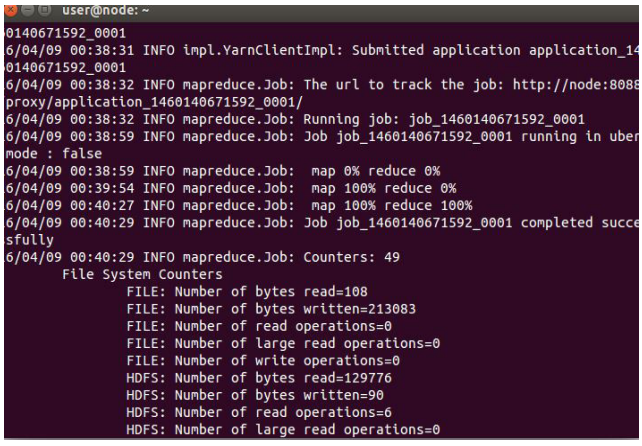

Figure 14. The map reduces process for the file.

The obtained output must be put into the intelligent graph which is written in php it is processed as shown in the Figure 16. The result obtained by the map reduce is shown in term of line graph, bar graph, pie graph as shown in the Figure 17, 18 and 19. Even other types can also be implemented.

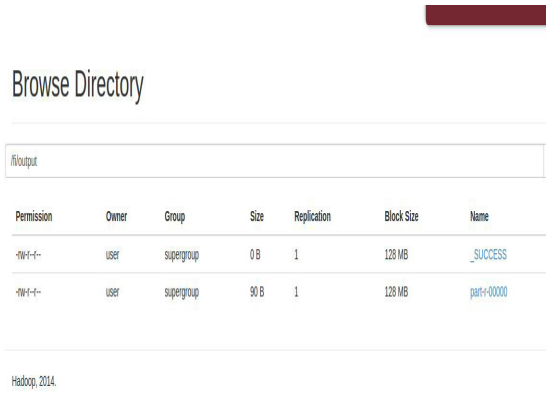

Figure 15. The success file created in file if map reduce process complete successfully.

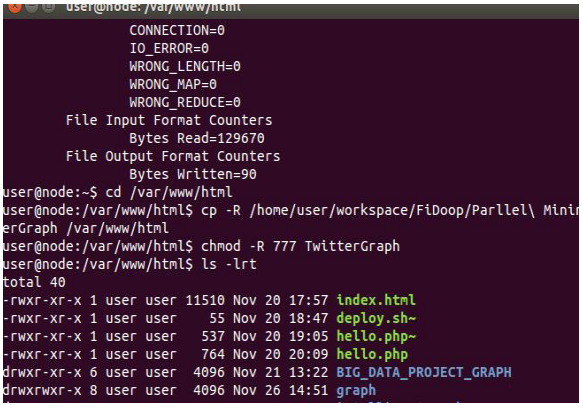

Figure 16. Processing the output to the intelligent graph system.

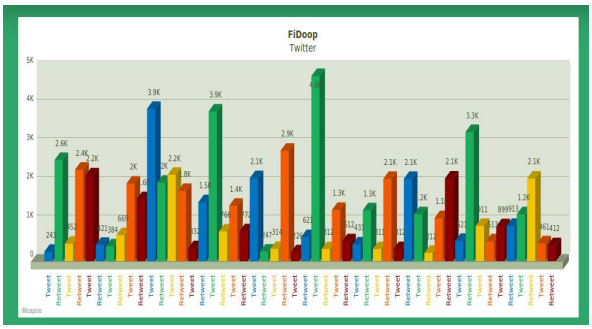

Figure 17. The excepted result obtained in bar graph.

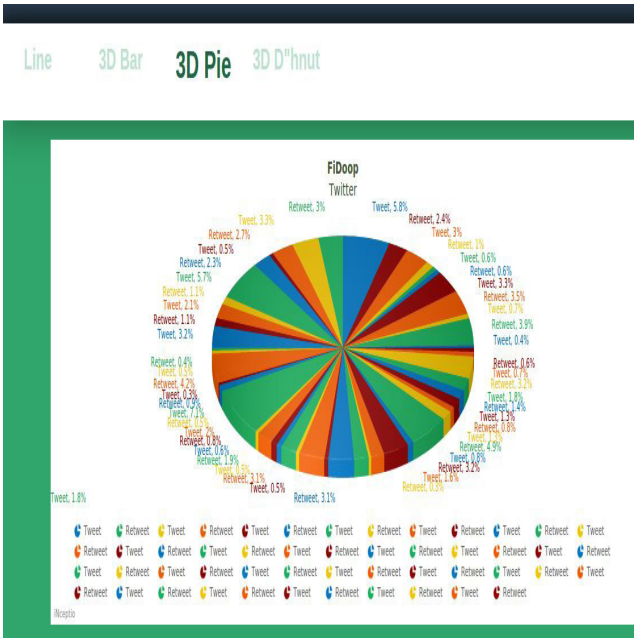

Figure 18. The result obtained in pie graph.

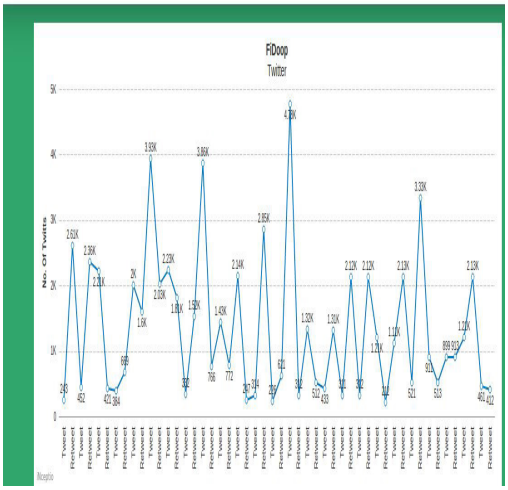

Figure 19. The result obtained in line graph.

3.2 IPB over Existing Tool

In this thesis I have work on the traditional system that is R tool now in this section. According to the programs done by me I have taken few factors for the comparison with the system. Few factors are:-

- Parallelism: The map reduce supports the parallelism processing data parallel we can process even 1TB of data in just minute (minimum of 4 nodes) but when consider in the R tool it doesn't provide any parallelism in its structure.
- Datasets: As the data increases we can also include nodes in the cluster which increase the performance of system and so large datasets are processed in the system but In the increase in data the R tool performance is drained.
- Fault tolerance: In map reduce iif some task is being interrupted then task tracker sends the heartbeat signal to the job tracker and the re-execution is done automatically but in R tool no such facilities.
- CPU Time: This is a little harder to understand, because this time will not change if a job does not change. This is the time consumed by CPU, which is used for processing instructions of a computer program.
- Availability: when we store the data in HDFS if there is 1 cluster present but by default 3 replication is created in the HDFS. So if one datasets is duplicated or deleted we can retrieve the data but when compared with the traditional system there is no such advantage is provided.
- Scalable: Hadoop is highly storage platform distributes the data across the many nodes it is not necessary that we install expensive servers but any kind of servers can be installed by this way the storage can be expanded storage.
- Load balancing: The datasets on the cluster is automatically scattered between the nodes in the cluster without any user instruction if there is more load on single load then it is automatically transferred to the other node where this cannot be done in the traditional tool.
- Data distribution: By default the name node distribute the data between the cluster in the node so user did not any kind instruction for the distribution of the data and in the data mining tool it is not possible.

4. Conclusion

In the paper I applied and worked both on data mining and hadoop data mining result could not manage the, fault tolerance, load balancing, data distribution and automatic parallelization and so many problems is faced in the data mining tool so this solved by hadoop platform in the

map reduce when compared with the data mining tool map reduce is advantageous in all the terms. IPB incorporates the parallel mechanism for the twitter data where I can achieve compressed storage and it is necessity not to build any conditional pattern. Firstly in this processes I retrieved the data from flume which is in unstructured format and later stored in the HDFS and from the HDFS it is then processed to the hive platform and converted to the structure.

Here in the system all the services are started data present in web is obtained in the flume and saved in HDFS the data present in hdfs which is obtained from flume. The important stage of the paper is the map reduce algorithm which was obtained where their three algorithm implemented mapper, reducer and driver config code was implemented where the data format specified in the for inputting the data is taken and the splitting of the data is done and in map stage it is converted to key value pair and it is sent reduce stage in the reduce stage the aggregation is done calculating the frequency of the tweets data present then later it is processed to the Driver function and aggregation is done. Later I run the map reduce process as the process gets completed a success file gets created in the HDFS and later we connect the result with the intelligent graph system and result is obtained in term of bar graph, line graph and 3D graph and so on having the maximum tweet and retweet in the table.

5. Acknowledgement

We are grateful to my institution, Atria Institute of Technology, for having provided me with the facilities to successfully complete this work "Implementation of Parallel Mining for Bigdata" and providing us all the necessary facilities for successful completion of this paper. Deadlines play a very important role in successful completion of the academic Project work on time, efficiently and effectively. We take this opportunity to express our deep sense of gratitude to our guide for her valuable guidance and help throughout the course of the academic report. They have always been patient with me and helped immensely in completing the task on hand. We also thank her for her immense support, guidance, specifications & ideas without which work would have been completed without full merit. We also thank the management. Finally, we thank my parents and friends for their motivation, moral and material support

6. References

1. Mehta T, Mangla N. A survey paper on big data analytics using map reduce and hive on hadoop framework. IJRAET. 2016; 4(2).
2. Mangla N. Machine learning approach for unstructured data using hive. International Journal of Engineering Research. 2016; 5(4).
3. Mangla N, Khola RK. Optimization of IP routing with content delivery network. Interntional Conference of IEEE Explore; 2010 Jun 11-12. p. 424-8.
4. Agarwal A, Biadsy F, Mckeown K. Contextual phrase-level polarity analysis. Proceedings of the 12th Conference of the European Chapter of the ACL (EACL); 2009 Mar. p. 24–32.
5. Barbosa L, Feng J. Robust sentiment detection. Conference on Computational Linguistics: Posters; 2010. p. 36–44.
6. Mangla N, Khola RK. A way to implement BGP with geographic information. 2010; 2(2):349–53.
7. Mangla N, Khola RK. Application based route journal. IOSRJEN. 2012 Aug; 2(8):78-82.
8. Bermingham A, Smeaton A. Classifying sentiment in microblogs: Is brevity an advantage is brevity an advantage? ACM. 2010:1833–6.
9. Fellbaum C. Wordnet, an Electronic Lexical Database. MIT Press; 1998.
10. Lammel R. Google's Map Reduce Programming Model - Revisited. Science of Computer Programming 70. 2008; 1–30.
11. Hadoop: Open source implementation of Map Reduce. Available from: http://lucene.apache.org/hadoop/
12. Ghemawat S, Gobioff H, Leung S. The Google file system. Symposium on Operating Systems Principles; 2003. p. 29–43.
13. MacQueen J. Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability; 1967; 1:281–97.
14. Borthakur D. The file system for hadoop: Design and arechtiture. 2007.