International Journal of Soft Computing 11 (5): 305-311, 2016
ISSN: 1816-9503

Big Data Optimization for Social Networking Tweet

[1]M. Senthikumar and [2]P. Ilango
[1]School of Infromation Technology and Engineering,
[2]School of Computer Sciences and Engineering, VIT University, Vellore, India

Abstract: Over trillions of active users are there in the social media network worldwide, generating insurmountable data and of dynamic structure. That is why social media is indeed mountains of big data to be explored and performance is a great issue to be improved. Apache Hadoop (a cloud service) is a well-known platform for its scalability, fault-tolerance and capability of processing big data. Hadoop MapReduce gives users full control on how input datasets are processed. Hence, Hadoop is the heart of big data analytics. However, there are several issues challenging Hadoop performance. Hadoop has a large set of configuration parameters which have an impact on performance and successful completion of loads of Hadoop jobs. This study aimed at setting up and tuning Hadoop clusters on performance metrics like workload balance, throughput, network bandwidth and response time. This is achieved by the iterative process of tuning the Hadoop parameters.

Key words: HDFS, hadoop, well-known platform, response time, trillions

INTRODUCTION

Processing the twitter dataset by iteratively tuning the performance parameters of Hadoop for optimized workloads. By using Hadoop, we tried to make the prediction of bursty events (called buzz) in the collected dataset easier with considerable improvement in the performance of Hadoop. Big data and its related technologies bring major benefits in the business. But organization feels difficult to control the vast and different collection of data to analyze and investigated. Many impacts using big data like competitions and growth of companies and individual its providing the support For the huge potential. Data becomes crucial now a days and evaluate the best approach in the ways of filtering and analyzing data hadoop optimized analytic processing with Map reduce programming used. The hadoop framework speeds up The process of data in large volume in a distributed environment. The simplicity and easy to extend of the framework gives the promise of tools for data processing (Rehg *et al*., 2016). Data and traditional machine learning algorithms generating new challenges in social media and networks. Social networking problem areas like data processing, staging, representation of data and the ways of data used in the pattern Mining, analyzing user behavior, etc. The data size keep on increasing and not properly scaled this reason the frameworks behind big data become popular in a large number of researchers (Dean and Ghemawat, 2008). Twitter is like micro blogging sites with popular platform expressing the opinions in an optimal way. Twitter are very small in nature and maintaining grammar and spelling becomes very important, otherwise difficult to understand the context. Statisticians and semantic approaches for tweet are followed for overcoming the problem and ensuring that context will give creating information and avoid inclusion of any non similar information is found in tweets (Ma *et al*., 2015). Map reduce programming working with large sets of data set and broad variety of real world tasks and user specify the computation in the form of Map and Reduce and work automatically parallelizes across large scale clusters. Example Google over the past years more than thousands distinct map reduce programs implemented internally (Uzunkaya *et at*., 2015).

Literature review: Tweets information classified as like or dislike on the topic depends on personal preferences feature selection. The experiment promising tweets with low number give quick personalization in next coming tweets. Personalization is playing a major role in twitter, for example, if the hottest topic discussed in twitter not all tweets interested by user. The tweets filtered by like and unlike by the user. The feature selection method is suitable when the tweets become small texts. Small number of tweets quickly trained and additional training dataset for like category model combined with external source texts for better tweets suggestion (Orgaz *et al*., 2016). HDFS proceed with commodity hardware have many advantages like low cost, high scalability and fault tolerance. HDFS storage efficiency improved and files transfer, reading increased three fold of original HDFS (Silva *et al*., 2014). Social networking, Twitter deals with mental health of human lives like depression. Two or more

Corresponding Author: M. Senthikumar, School of Infromation Technology and Engineering, VIT University, Vellore, India

tweets related to depression the health professional can send awareness message to the depressed user. Depression classified into Great, economic, era, tropical, real estate, etc (Feller *et al*., 2015). Tweets categorized into two item new tweets and retweets when tweets are now posted its transferred to original of which have already someone else posted. Total no of tweets is equal to the number of tweets with the new tweet with retweets (Heger, 2013). Information retrieval system identifies the facts like what a person is doing, what time, location, etc. Analysis of electoral tweets for sentiment, whether it's positive or negative and emotion joy or sadness style like simple or sarcasm, etc. This experiment automatically identifies emotion of the user and performing sentiment analysis (Ko *et al*., 2014). Twitter sentiment analysis with two classifiers ensembles and lexicons and tweets are classified into positive and negative this approach helps the customers for sourcing the products, for companies monitoring the public sentiment of the product or brand and Many more applications. By using multinomial bayes, SVM and Random forest will improve the classification accuracy in an optimal way (Maitrey and Jha, 2015). The twitter application programming interface is created as Java library and integrated Java application with all twitter services. Analyzing the collected tweets and transfer the data into graphical charts.

Hadoop and HDFS: Hadoop is the open source software provided by apache foundation Hadoop providing for computing applications to become highly scalable distributed environment. The developer focuses on the dataset and its logic only and no need to worry about processing. The HDFS stores large no of files in many machines this help in achieving high reliability by data replication across many hosts and avoiding RAID storage for hosts. The HDFS built data nodes from the cluster and data over the network using block protocol. Data over HTTP will help the user to allow the access to a web browser or other client All data nodes connected to gether to rebalancing, copy/move the data and ensuring the replication of data is achieved. If any single node fails it will become a dead node and any new need added it will become a live node.

Issues: Data volume of social media to be processed by cloud applications are growing much faster than computing power. Optimizing such big data volume is always a challenge for hadoop performance. Enhancing the data processing speed has to be concerned more than reducing the latency of data. The workload has to be balanced among the map slots and reduce slots to reduce the network bandwidth. Hadoop performance tuning parameters should be identified for each issue which is really a time consuming task. A Map Reduce program for predicting buzz has to be done in analyzing the collected Twitter dataset. The hadoop cluster to process large datasets with an efficient throughput that leads a promising conclusion that Hadoop is the most beneficial framework to analyze and manage the huge social medial big data like Twitter, Facebook, LinkedIn, etc.

Data analysis: Twitter is used worldwide social networking and it will allow the users to update their status in the form of tweets and interested people retweet other posts and communicated to the user directly. The Twitter dataset used in this work is based on the classification of buzz and non-buzz from the given instance which is described by 77 features. Each instance, covers seven days of observation for a specific topic on which active discussion is made at time step t. This dataset is published using the UCI guidelines. The data is stored using a standard Comma Separated Value (CSV) format. Number of created discussions this feature measures the number of discussions created at time step t and involving the instance's topic. Attention level this feature is a measure of the attention paid to a the instance's topic on a social media. Buzz -This attribute is Boolean: 1 meaning buzz observed 0 meaning no buzz observed.

System design and implementation:

Proposed algorithm:

Algorithm:

Step 1: To perform mapping in hadoop with social Twitter data.
Input: Passing the parameters long writable key with text value output: Text, writable integer and output reporter

Step 2: Finding the Buzz in HDFS System: If the value 0 buzz not presen else if the value (Bello *et al*., 2015) buzz is present. Take the value from Hadoop Cluster in the form of String [] cols and increment the value String [I] cols = cols [I] I values from 1 to n if (cols [I] = = 0) then buzz not found else increment the counter then if (cols [I] ==1) buzz found.

Step 3: To perform reducing in hadoop with social twitter data.
Input: passing the parameters long writable key with text value output: text, writable integer and output reporter. Initialize display value to zero and increment the value till the end of string display key and the value with optimized output

Step 4: To Perform on driver: If the string length! = 2 then exits and perform job configuration for display with reducer output. Call file input format and output format and set combiner, reducer, output key, value, values and run the job with optimized configuration.

Step 5: Running a job optimized configuration: Configure the values in yarn-site, core-site, Hdfs-site, Mapread-site calling the configured file from map reduced program

MATERIALS AND METHODS

Proposed system for optimization: By this proposed design, we tune the parameters which contribute to balance workload, data transfer rate and response time before the allocation of job is made to the Map and Reduce slots. For this purpose, the Twitter dataset has been used. The dataset has various instances

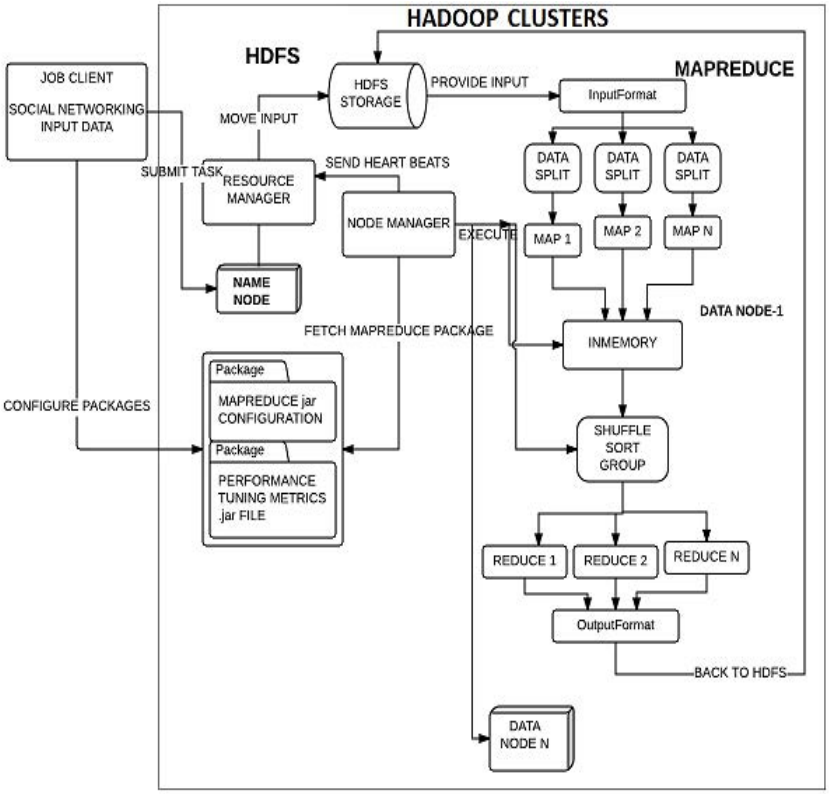


Fig. 1: Proposed design hadoop social media system

where each instance, covers seven days of observation for a specific topic for 1 year. When this dataset is processed using an efficient MapReduce program, the prediction of the total amount of buzz and non-buzz over that year can be done. And the result of job execution with tuned parameters will show considerable enhancement of performance of Hadoop Fig. 1.

RESULTS AND DISCUSSION

From the data distribution mechanism, the CPU and IO workload are always underutilized when working on a data-intensive application. The big block size can shorten the seeking latencies however because of the large block size, the transfer time of data access dominates the whole process time and the IO stall becomes a significant factor in the execution time. Perfecting strategies are needed to parallelize these workloads to avoid this idle time. The original MapReduce just randomly assigns tasks in computing nodes and loads data from local or remote disk when it is required. The CPU will not process the new task until all the resources are loaded into the memory. However, this waiting period will downgrade performance. The perfecting mechanism will help the MapReduce preparing the required data before the task is launched. Figure 2 setting the environment for Hadoop cluste after Hadoop environment ready data loaded with HDFS and data split into different blocks. Figure 2 shows the various steps before executing the Hadoop job. Its explain the step by step process from installation of Java JDK to configuring the Hadoop node.

Figure 3 shows that the Hadoop cluster ready for performing the tasks and sample I/O processed and Hadoop cluster status become live and map reduce done successfully. Figure 4 shows that the cluster summary and configuration of the node and DFS and Non DFS used. Figure 5 shows the summary of file system counte, number of job counters and map reduce framework before optimizing. Figure 6 shows the summary of file system counters, number of job counters and map reduce framework after optimizing

Figure 7 Hadoop ouput with buzz and non buzz. Figure 7 shows the final output of the twitter dataset number of buzz and non buzz detected by Hadoop cluster. Number no of buzz 112932 and non buzz 27775 detected as per the twitter dataset in optimal way using Hadoop cluster environment.

Figure 8 the dark color bars represent the number of map and reduce slots allocated for the dataset which has an execution time of 32824 and 11305 ms, respectively. The light color bars represent the number of map and

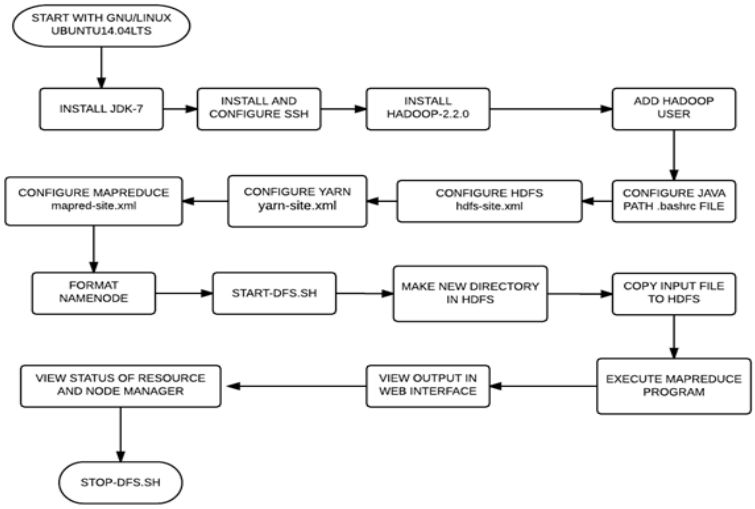
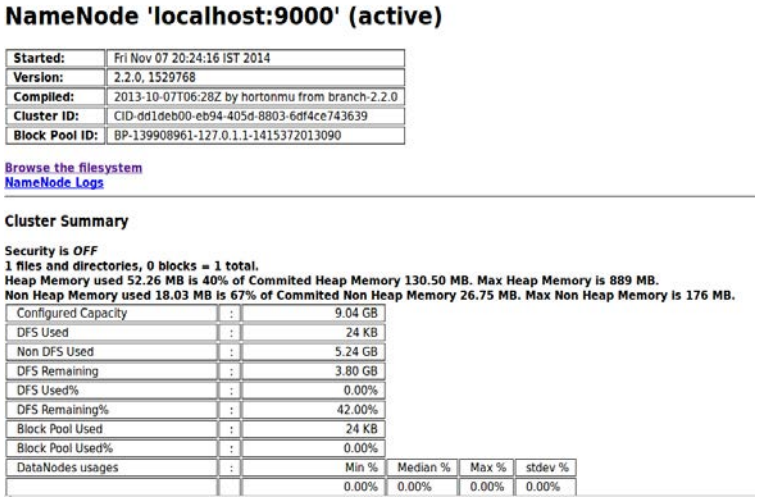


Fig. 2: Setting the environment for Hadoop cluster

```
    File System Counters
            FILE: Number of bytes read=1257416
            FILE: Number of bytes written=3067201
            FILE: Number of read operations=0
            FILE: Number of large read operations=0
            FILE: Number of write operations=0
            HDFS: Number of bytes read=68187557
            HDFS: Number of bytes written=694558
            HDFS: Number of read operations=21
            HDFS: Number of large read operations=0
            HDFS: Number of write operations=10
    Job Counters
            Launched map tasks=2
            Launched reduce tasks=5
            Data-local map tasks=2
            Total time spent by all maps in occupied slots (ms)=37244
            Total time spent by all reduces in occupied slots (ms)=90717
    Map-Reduce Framework
            Map input records=140707
            Map output records=140707
            Map output bytes=975972
            Map output materialized bytes=1257446
            Input split bytes=180
            Combine input records=140707
            Combine output records=140707
            Reduce input groups=3026
            Reduce shuffle bytes=1257446
            Reduce input records=140707
            Reduce output records=140707
            Spilled Records=281414
            Shuffled Maps =10
            Failed Shuffles=0
            Merged Map outputs=10
            GC time elapsed (ms)=2359
            CPU time spent (ms)=16760
            Physical memory (bytes) snapshot=1009299456
            Virtual memory (bytes) snapshot=3423612928
            Total committed heap usage (bytes)=754974720
```

Fig. 3: Mapreduce process



NameNode 'localhost:9000' (active)

Started:	Fri Nov 07 20:24:16 IST 2014
Version:	2.2.0, 1529768
Compiled:	2013-10-07T06:28Z by hortonmu from branch-2.2.0
Cluster ID:	CID-dd1deb00-eb94-405d-8803-6df4ce743639
Block Pool ID:	BP-139908961-127.0.1.1-1415372013090

Browse the filesystem
NameNode Logs

Cluster Summary

Security is *OFF*
1 files and directories, 0 blocks = 1 total.
Heap Memory used 52.26 MB is 40% of Commited Heap Memory 130.50 MB. Max Heap Memory is 889 MB.
Non Heap Memory used 18.03 MB is 67% of Commited Non Heap Memory 26.75 MB. Max Non Heap Memory is 176 MB.

Configured Capacity	:	9.04 GB
DFS Used	:	24 KB
Non DFS Used	:	5.24 GB
DFS Remaining	:	3.80 GB
DFS Used%	:	0.00%
DFS Remaining%	:	42.00%
Block Pool Used	:	24 KB
Block Pool Used%	:	0.00%

DataNodes usages	:	Min %	Median %	Max %	stdev %
		0.00%	0.00%	0.00%	0.00%

Fig. 4: HDFS cluster summary

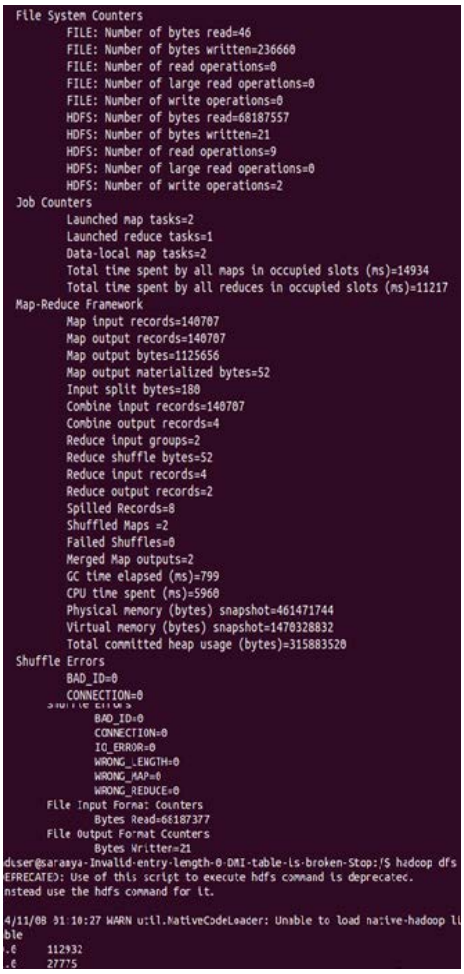


Fig. 5: Mapreduce before applying tuned configuration

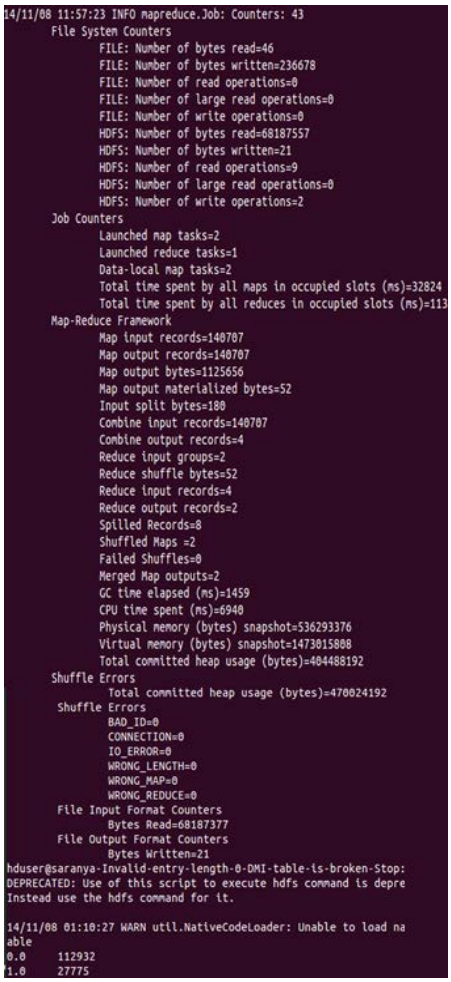


Fig. 6: Mapreduce after applying tuned configuration

reduce slots allocated for the dataset which has an execution time of 14734 and 11217 ms, respectively. This depiction shows clearly that when input, output parameters of the map and reduce tasks are tuned, it reduces the response time considerably. Figure 9 shows the CPU time spent and CPU time elapsed for processing the dataset which has 6940 and 1459 ms, respectively in dark blue color bars represent. The light color bars represent the CPU time spent and CPU time elapsed for processing the dataset which has 5960 and 799 ms, respectively.

Twitter dataset analysis result: Figure 10 shows that total number of instances, attributes, Buzz predicted and non-predicted from the twitter dataset. The MapReduce output, this classification has been made Positives instances (Buzz): 27775 (19 %) negative instances (Non Buzz): 112932 (81%).

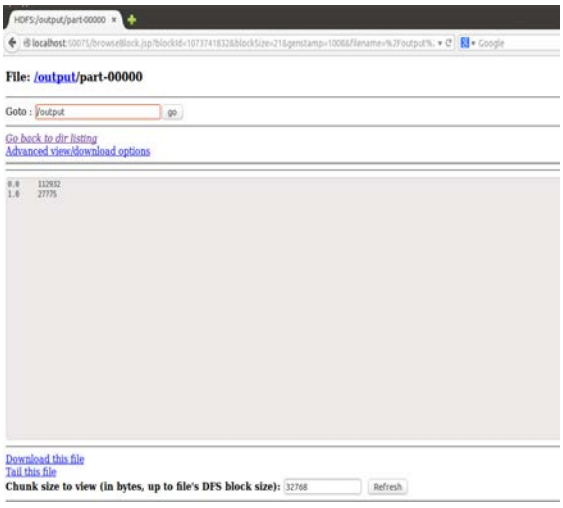


Fig. 7: Hadoop ouput with buzz and non buzz

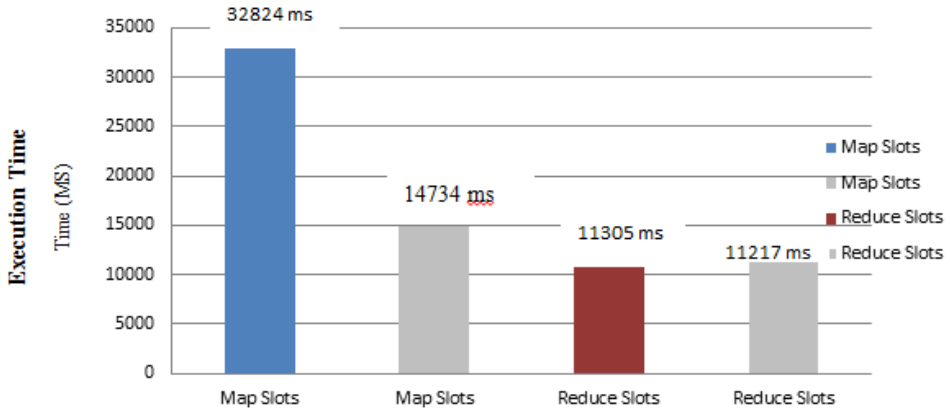


Fig. 8: Execution of map and reduce slots in hadoop

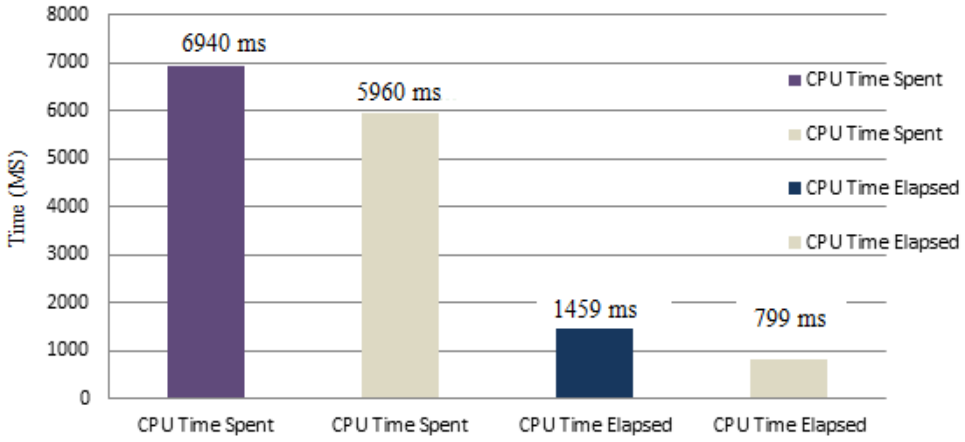


Fig. 9: Overall process execution time for mapreduce

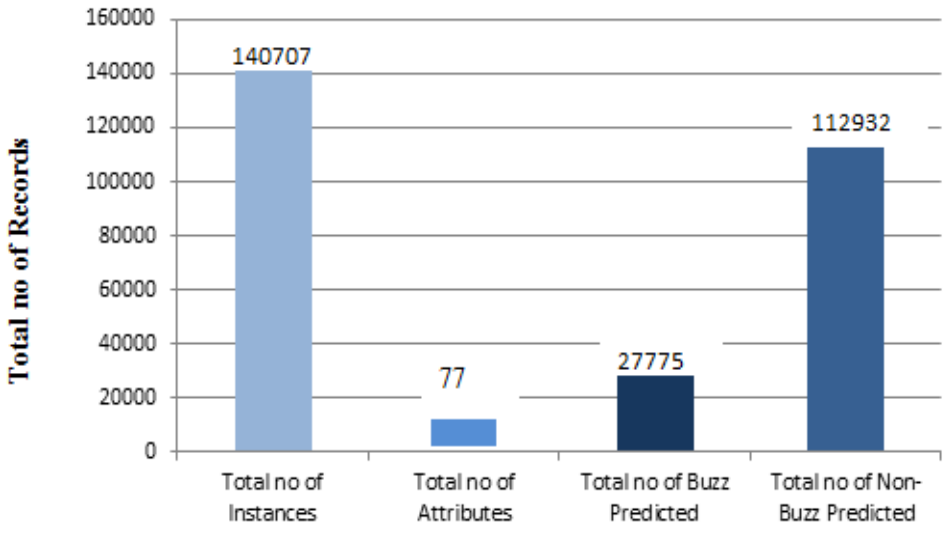


Fig. 10: Result of hadoop cluster

CONCLUSION

Improving the performance of Hadoop yaddressing metrics like workload balance, throughput and etworkbandwidth. However, it will be efficient in the heterogeneous Hadoop cluster. We have processed Twitter dataset with a Hadoop cluster with our MapReduce environment and a sample tuned Hadoop configuration file. When this analysis is done with the large data sets, Hadoop will surely be a great tool in the prediction of buzz in an efficient manner. The performance of Hadoop has proved by results. The enhancement will be the performance of Hadoop for multi-node cluster with homogenous by addressing all the performance metrics.

REFERENCES

Dean, J. and S. Ghemawat, 2008. MapReduce: Simplified data processing on large clusters. Commun. ACM, 51: 107-113.

Feller, E., L. Ramakrishnan and C. Morin, 2015. Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study. J. Parallel Distrib. Comput., 79: 80-89.

Heger, D., 2013. Hadoop performance tuning-a pragmatic and iterative approach. CMG. J., 4: 97-113.

Ko, J., H.W. Kwon, H.S. Kim, K. Lee and M.Y. Choi, 2014. Model for twitter dynamics: Public attention and time series of tweeting. Physica Stat. Mech. Appl., 404: 142-149.

Ma, Y., Y. Zhou, Y. Yu, C. Peng and Z. Wang *et al*., 2015. A novel approach for improving security and storage efficiency on HDFS. Procedia Comput. Sci., 52: 631-635.

Maitrey, S. and C.K. Jha, 2015. MapReduce: Simplified data analysis of big data. Procedia Comput. Sci., 57: 563-571.

Orgaz, G.B., J.J. Jung and D. Camacho, 2016. Social big data: Recent achievements and new challenges. Inf. Fusion, 28: 45-59.

Rehg, P.A.C., M.J. Krauss, S. Sowles, S. Connolly and C. Rosas *et al*., 2016. A content analysis of depression-related tweets. Comput. Hum. Behav., 54: 351-357.

Silva, N.F.D., E.R. Hruschka and E.R. Hruschka, 2014. Tweet sentiment analysis with classifier ensembles. Decis. Support Syst., 66: 170-179.

Uzunkaya, C., T. Ensari and Y. Kavurucu, 2015. Hadoop ecosystem and its analysis on tweets. Procedia Social Behav. Sci., 195: 1890-1897.