**Indian Journal of Science and Technology,** *Vol 9(39), DOI: 10.17485/ijst/2016/v9i39/85585, October 2016*
*ISSN (Print) : 0974-6846*
*ISSN (Online) : 0974-5645*

# A Quantitative Review on Introducing the Election Process with Cloud Based Electronic Voting and Measuring the Performance using Map Reduce

**R. K. Nadesh, K. Arivuselvan, and Srinivasan Pathanjali**

School of Information Technology and Engineering, VIT University, Near Katpadi Road, Vellore - 632014, Tamil Nadu, India; rknadesh@gmail.com, arivupsg1@gmail.com, getpathanjali@live.com

## Abstract

The measure of information created consistently in an advanced domain is expanding in an exponential rate. The increased accessibility of the system is strongly affected by the structure and high volume of data. The advancement in the web technologies gave rise to many applications and among them is online voting system which makes the voting process very easy and efficient. In developing countries, with millions of people voting through the system creates a huge amount of data which makes the data processing more complex. To make effective voting process the primary need is infrastructure and maintaining the infrastructure is one of the key task. However there exist different cloud service providers, choosing the appropriate is always challenging. The efficiency of the system is in the counting process, where a large set of data is to be processed. In this review, we discuss the election process using cloud and vote counting using Map Reduce. Hadoop gives better devices and methods to handle this colossal measure of information. The performance is measured by taking input from the voting application and then analysed. The parameters considered are the measure of bytes composed, read by the framework and the current state of the Hadoop system on increased file size. The proposed system for processing voting data shows that time and processing speed do not increase proportionally to the file count.

**Keywords:** Hadoop, Online Voting, Web Technologies

## 1. Introduction

Voting style has changed from the world counting papers to recording the votes electronically[1]. The online voting system provides the voter to vote from anywhere in the country[1]. The system provides many advantages from the traditional voting system by reducing the voting process time and provides more flexibility and accuracy. But with the advantages comes the drawback where the large volume of data takes a lot of time to process which affects the system performance. This huge amount of data can be stored, processed and analysed in many ways but they require fast retrieval technique. Hadoop is considered as the best solution for handling big data which uses parallel computing techniques[2]. Hadoop gives a complete administration apparatus to manage huge data[2]. An online voting system is an experimental study by running the MapReduce Job and measure the performance of the Hadoop System.

## 2. Related Works

Different research works has been carried in conducting the election process in cloud and to implement secure and fear free voting, Usage of Aadhar Card or UID shall be introduced[3]. The behaviour of the systems is unpredictable, to focus more on feasibility two fold systems are used[4]. To improve the scalability, the authors[5] introduces some scalability solutions for web applications. Identity management and access control are introduced to protect from the fraud behaviours by group accessing scheme[6,7] which is to secure online voting system. Resource allo-

*Author for correspondence

cation efficiency is the other area to be focussed when number of request is generated randomly[8]. The behaviour of the Map Reduce Job is analysed by increasing the number of files which contains the dataset[9]. Variables like consolidated time spend by the maps are analysed for storage and data processing[10]. A MapReduce concept is implemented that scales to large clusters of machines and the working of MapReduce and HDFS when they are put together.

## 3. Proposed Method

The demand for online voting system is ever increasing and the system creates a huge amount of data. The results that are produced should be processed in an efficient way. Traditional data storage system and their data processing techniques are not really effective in handling BigData. These data can take different forms like structured, unstructured and semi structured. The processing power of the machine is influence by the large data size. The scope of this paper is that we run sample voting data and check the efficiency and the flexibility of the system by using Hadoop Framework. The system should be fully automated and should be able to handle extremely large volumes of data. The datasets are created using the application and are used to analyse the MapReduce task. Performance of the system is measured by increasing the file size.

## 4. Architecture

Hadoop consists of two segments HDF framework and the MapReduce. Hadoop structural engineering follows the general architecture of the HDFS. Primary node termed as Name contains the Meta information. The Secondary node will update the primary node at every interval Information is stored in the node named Data Node. When the JT receives the job from client, make partition and allocate job to TT that keeps running the Data node. When they receive information from data node, they execute the assignment given to them. They consistently converse with the JT arranges and coordinates the job given by the client. With JT, TT, Data node the the read operation is faster.

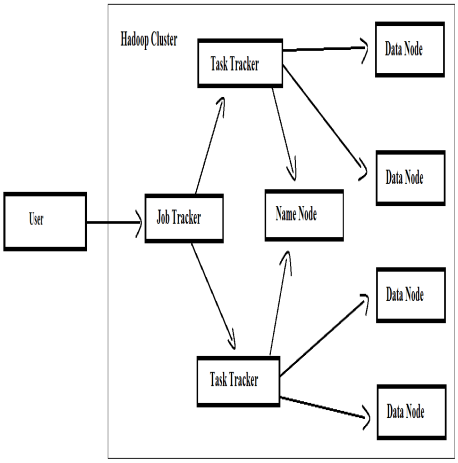


**Figure 1.** The General HDFS.

## 5. Vote Count Algorithm

The program to count the vote is created based on vote count allotted in the map reduce model. Whenever the vote is registered in favour of a party, it is redirected to the Input set. The input is a collection of characters and finally mapper is executed.

For user input of the following
Leo
Capricorn
Leo
The reduced operation will be
(<Leo>, 2)
(<Capricorn>, 1)

## 6. Experimental Setup

The experiment was conducted with Acer Aspire 5742 system with, Intel(R) Core i3-370M processor, with 2GB. Hadoop is installed on the Linux file system.

In this model, the preferred architecture is single node and includes all components.

To estimate the system usage and attraction of the Map and Reduce, for the copy of the files from the local file system to the HDFS we have used the datasets from the voting system. The output of the system is redirected to local file processing unit. The paper focus is on the file count and not the content size. Metafile is maintained separately to track the different files. As the data size increase, the metafile is also updated. We had analysed the behaviour of system to HDFS and the task running on these files.

The online voting system is developed using HTML, CSS and the validation is done using PHP. The XAMP server and phpmyadmin is installed using command lines. The tables are created inside phpmyadmin.

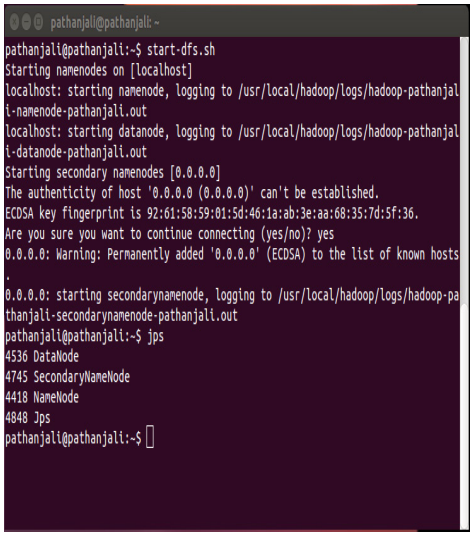


**Figure 2.** jps running.

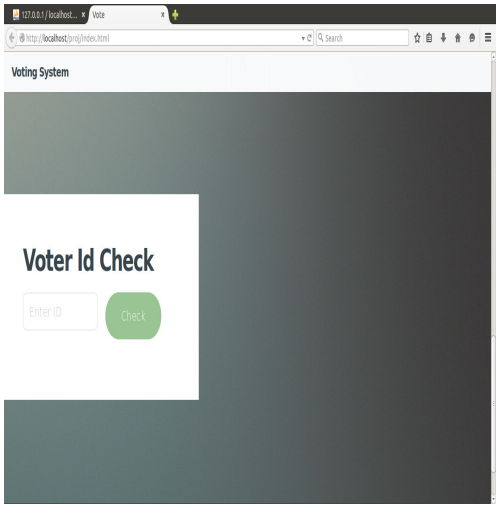


**Figure 3.** Identity check.

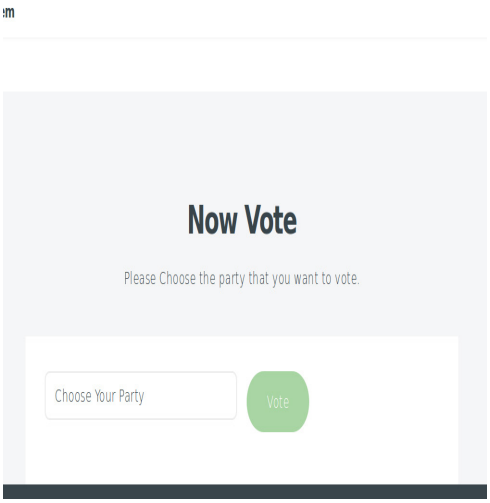


**Figure 4.** Party Selection.

## 7. Results and Analysis

The execution environment is byte composed, file bytes read, reduce input records. Table 1 demonstrates the outcome of the experiment. The number of files is increased each time. The files are copied from local file system to HDFS and the whole task is made to run on single node cluster environment. We started our experiment with a single file and thus increased gradually.

**Table 1.** Bytes Usage by MapReduce Task

| No. of files | 1 | 5 | 10 | 20 | 40 | 60 |
|---|---|---|---|---|---|---|
| Bytes read | 45 | 336 | 672 | 1406 | 2812 | 4218 |
| Bytes Written | 57 | 58 | 63 | 63 | 63 | 67 |
| Map Input records | 6 | 45 | 90 | 190 | 384 | 576 |
| Map output records | 6 | 43 | 86 | 153 | 368 | 552 |
| Combine Input records | 6 | 43 | 86 | 153 | 368 | 552 |
| Combine Output Records | 6 | 16 | 32 | 76 | 116 | 174 |
| Split bytes | 117 | 585 | 1171 | 3227 | 5002 | 7493 |
| Reduce Input records (b) | 6 | 16 | 32 | 58 | 116 | 174 |
| Reduced output records (b) | 6 | 6 | 6 | 6 | 6 | 6 |
| CPU time spent (ms) | 2200 | 7280 | 25710 | 29030 | 59180 | 76960 |
| GC time spent (ms) | 233 | 1514 | 38969 | 32909 | 70630 | 83533 |
| Heap usage (b) | 160239616 | 690573312 | 135349 0432 | 267932 4672 | 533099 3152 | 798266 1632 |

Figure 4 demonstrates the situation of the byte composed as for the number of documents in the System. Number of bytes composed with the rate at which the number of documents is expanding.

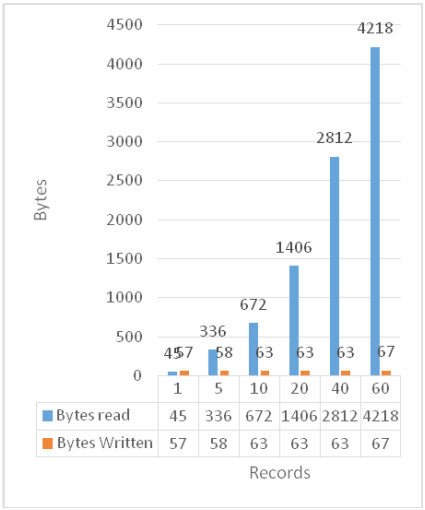


**Figure 5.** Bytes Read and Written.

The following graph shows the mapped input records and the output records of the MapReduce Task.

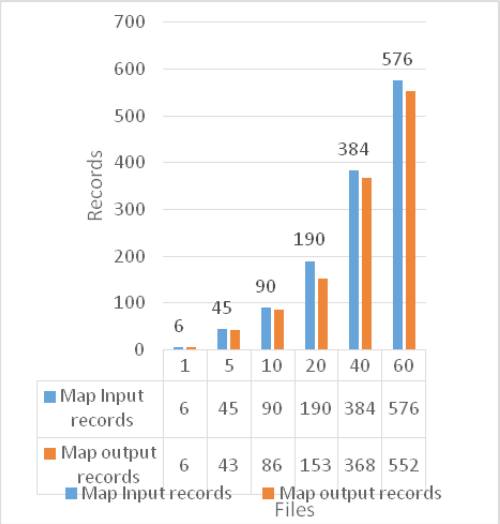


**Figure 6.** Mapped Records.

In the next graph we show the time spent by the CPU to process the data. It shows that the processing time does not increase proportionally to increase in number of files.

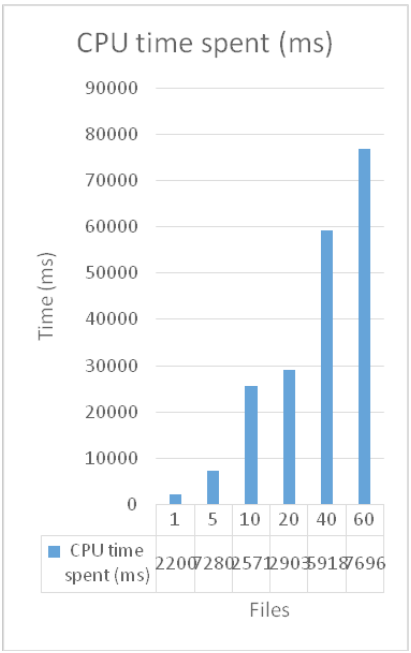


**Figure 7.** Time Spent for processing.

## 8. Conclusion

The strength of the MapReduce task with increasing the amount of files interms of number have been analyzed. We have created a program for counting the vote and made it into JAR file for running. The output shows that the time spent and the bytes composed don't increase in the same extent as compared to the increase in amount of records. Our proposal of data processing and management for the voting system is highly flexible and efficient. In addition to its reliability it can handle multiple clusters and parallel processing and provides better scalability for large election.

## References

1. Anand Ankit, Divya Pallavi. An Efficient Online Voting System. International Journal of Modern Engineering Research. 2012 July-Aug; 2(4).
2. Agrawal Sanjay, Pal Amrit, Agrawal Pinki, Jain Kunal. A Performance Analysis of MapReduce Task with Large Number of Files Dataset in Big Data Using Hadoop, IEEE. 2014; DOI: 10.1109/ICICA.2014.16.
3. Gupta Ashutush, Dhyani Praveen, Rishi OP. Cloud based e-voting: one step ahead for good governance in India. International Journal of Computer Application. 2013 April; 7(02).
4. Nadaph Anisara, Katiyar Ashmita, Gosami Durgesh Kumari. An Analysis of secure online voting system. International Journal of Innovative Research in Computer Science and Technology. 2014 Sep; 2(5).
5. Zhouwei, Pierre Guillaume and Chi-Hung Chi. Cloud TPS: Scalable Transactions for web applications in the cloud. IEEE transactions of scalable computing. 2012 Dec; 5(04).
6. Megiba Jasmine R and Nishiba GM. Public Cloud secure group sharing and accessing in cloud computing. Indian Journal of Science and Technology. 2015 July; 8(15).
7. Bhosale Poonam, Vethaka Priyanaka, Thorat Lata, Archana Lomte. Identity Access Management using Multitier Cloud Infrastructure for secure online voting system. IJMRD 2015 March; 2(4).
8. Shymala K, Sunitha Rani T. An analysis on efficient resource allocation mechanism in cloud computing. Indian Journal of Science and Technology. 2015 May; 8(9).
9. Rama Satish KV, Kavya NP. Big Data Processing with harnessing Hadoop-MapReduce for Optimizing Analytical Workloads. IEEE 2014, International Conference on Contemporary Computing and Informatics.
10. Kyoo-Sungnoh and Doo-Sik-Lee. Bigdata platform design and implementation. Indian Journal of Science and Technology. 2015 Aug; 8(9).