Log Analyzer and Integrated Framework using Hadoop and MapReduce for Deploying Anti-Virus

Rahul Pawar[1], Rajkumar Bhosale[2], Bharat Borkar[3], Ajit Borhade[4]
[1, 2, 3, 4] Department of Information Technology
[1, 2, 3, 4] AVCOE Sangamner , India

Abstract- *There are various applications which have a huge amount of database in different format. All databases maintain log files that keep records of database changes in a system formation. This can include tracking various user events and activity. Apache Hadoop can be used for log processing at scale of a system. Log files have become a standard part of large applications and large scale industry , computer networks and distributed systems. Today each and every day a lot of data is generated in increasing order. This is because of today's e-commerce and easy to use technologies. Also, there is increasing number of vulnerabilities in this large data. There are counter measures for these vulnerabilities like anti-viruses or anti-malwares. But, for scanning a large data in less time its difficult. So using Hadoop and MapReduce technology we can scan it parallely in less time. In this project we are scanning malware using Hadoop and MapReduce.*

Keywords- Hadoop, Map-reduce framework, Log files, log analyzer, Heterogeneous database, different Kind of log database.

I. INTRODUCTION

Windows computers Malware is software, a computer program used to perform malicious actions. In fact, the term malware is a combination of the words malicious and software. The end goal of most cyber criminals is to install malware on your computers or mobile devices. Once installed, these attackers can potentially gain total control of them. Many people have the misconception that malware is a problem only for. To scan malwares in large data we can do it with parallel functionality. This can be done with the help of Hadoop [2] framework. The MapReduce [10] developed by the Google works for assigning the job parallely. Let's talk about Hadoop, the main architecture of Apache Hadoop consists of Hadoop Distributed File System which is used for storage and MapReduce for the parallel processing. Hadoop divided the file into the blocks and makes the replication of the blocks in different nodes. To work in parallel we have to submit the code to the Hadoop MapReduce. The nodes take the configuration and work accordingly. Due to this, there is the advantage of parallel working with data which are distributed in different locality. With high end architecture of today's generation and high speed net there is a reliable result with less fault tolerance[13]. Current software applications often produce (or can be configured to produce) some auxiliary text files known as log files or activity log. Such files are used during various stages of software development or software installation, mainly for debugging and profiling purposes of logs.

So, the main purpose of the project is to scan the malware in the large data with the help of the Hadoop and MapReduce technologies. The malware scanning code should be written in the MapReduce. The paper is organized as follows: Section II presents the literature review in the area of malware-detection algorithms. In Section , a description about proposed system of malware detection using Hadoop and MapReduce. Section presents the discussion and conclusion.

II. LITERATURE SURVEY

Mostly in large data set there has been the focus on the intrusion detection system then scanning the malwares on the host machines. This is because widely used World Wide Web. Due to the vast use of the internet all the vulnerability and attacks are done with help of the internet. So, the concentration is done in the intrusion detection system. Many works have done in this area by using different technologies which are as follows.

Ibrahim Aljarah describes an intrusion detection system (IDS) based on a parallel particle swarm optimization clustering algorithm using the MapReduce methodology [3]. This paper presents a parallel intrusion detection system (IDS-MRCPSO) based on the MapReduce framework since it has been confirmed as a good parallelization methodology for many applications [3]. In addition, the proposed system incorporates clustering analysis to build the detection model by formulating the intrusion detection problem as an optimization problem[11].

In, authors focus on the specific problem of Big Data facing in network intrusion traffic. It tells the system challenges presented by the today's Big Data problems associated with network intrusion problems[4]. It describes the management in big data, network topology which gives a specific which used HDFS and public cloud in it [12]. It also defines the communication challenges in case of bandwidth.

In[9] author present Aesop, a scalable algorithm that identifies malicious executable files by applying Aesop's moral that "a man is known by the company he keeps." They use a large dataset voluntarily contributed by the members of Norton Community Watch, consisting of partial lists of the files that exist on their machines, to identify close relationships between files that often appear together on machines. Aesop leverages locality-sensitive hashing to measure the strength of these inter-file relationships to construct a graph, on which it performs large scale inference by propagating information from the labeled files (as benign or malicious) to the preponderance of unlabeled files.

Author[10] proposes a novel behavioral malware detection approach based on a generic system-wide quantitative data flow model. They base their data flow analysis on the incremental construction of aggregated quantitative data flow graphs. These graphs represent communication between different system entities such as processes, sockets, files or system registries. Authors demonstrate the feasibility of our approach through a prototypical instantiation and implementation for the Windows operating system. The experiments yield encouraging results: in our data set of samples from common malware families and popular non-malicious applications.

III. VIRTUAL DATABASE SYSTEM

The scalability and parallel processing should be possible with average computer hardware and which can be made possible with the Hadoop platform. And by the help of Linux OS it becomes more secure and reliable.

The main concern is writing the MapReduce code for scanning the malwares in the large data. After the code is written the MapReduce will split the process in Mappers and Reducers. MapReduce in Hadoop comes with a choice of schedulers. The default is the original FIFO queue-based scheduler, and there are also multiuser schedulers called the Fair Scheduler and the Capacity Scheduler.

We will run our code to the job driver. The job driver will copy the job configuration to the name node as it has information about all data nodes. Now the job driver will submit the code to the job tracker. The job tracker will distribute task configuration to the task tracker. The task configuration will have the malware signatures which I have to match with the data which are stores in the HDFS. Through the task tracker the configuration is given to different Mappers through which the processing is distributed and the malwares will be scanned parallely the data which resides in the datanodes.

After scanning all the data either while mapping is done the same malwares can be found. Then the reducer will sort the repeated detected malwares and will minimize the result. In this way the proposed system will work for finding the malwares in the large data set.

Now with the reference of the figure 1 the client will give the input of the malware scanning to the HDFS through the job driver. Then the process of mapper and reducer will split the process and work in parallel then the output will be saved in the HDFS output and will be given back to the client. The process may be fast depending on the MapReduce code and the language which is suitable to use.

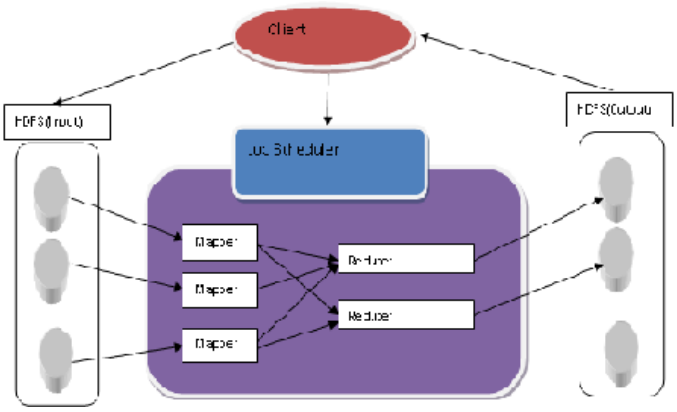

Fig. 1. Malware Detection Flow

Above Fig.1. Shows that the Malware Detection Flow. Which consist of one client and wich consist of HDFS(Hadoop Distributed File System) wich is used in storage of an Hadoop system. Wich are used in Mapping and reducing that log file ie large amount of log file converted in to the small amount.Wich can be used to detect the malware present in that log file.

IV. PRAPOSED SYSTEM

A. Problem Statement

To build a system for generic log analysis using Hadoop Map Reduce Framework by providing user to analyze different type of large scale of log file and malware present that log files.

B. Feature

- Increased efficiency do to use of Hadoop-Map Reduce framework.
- Ability to analyze the different kinds of log files.

C. Scope

Generic log analyzer can be used to analyze various kind of logs such as:

- Email logs.
- Web logs.
- Firewall logs.
- Serever logs.

This system build Generic Log Analyzer for different type of large scale log files.By taking advantage of Hadoop Map Reduce framework and polymorphism for log analysis and will increase efficiency and reliability of log analysis.

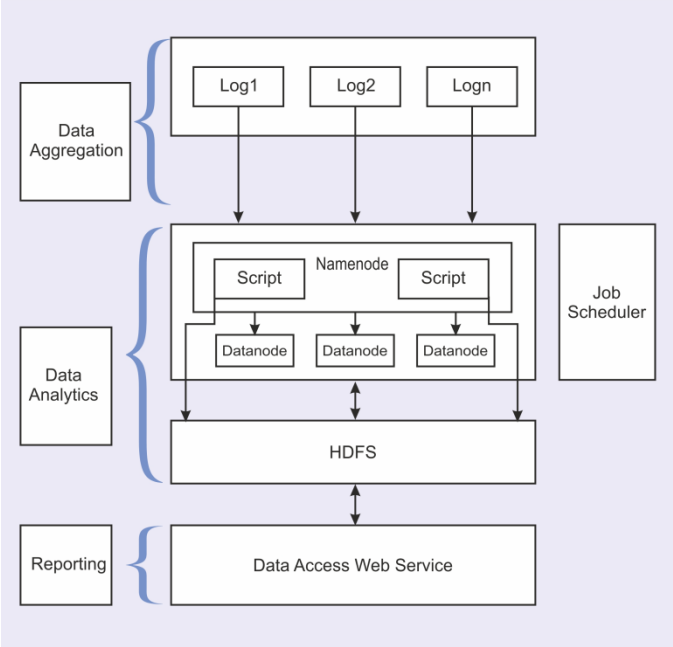

Fig. 2. Architecture of proposed System

The above Fig. 2 shows that the architecture of proposed system. It consists of different type of log files as a input i.e. mail log, Firewall log, Server log. In first process this log files goes to name node and data node where name node is primary and data node is secondary there is writing the script related to that log files. After the name node and data node that log files goes to an HDFS i.e. Hadoop Distributed File System. Finally that file is represented in to the graphical represented format.

V. WORKING METHOD OF MAPREDUCE

Map Reduce a java based distributed programming model consists of two phases: a massively parallel "Map" phase, followed by an aggregating "Reduce" phase. MapReduce is a programming model and an associated implementation for processing and generating large data sets . A map function processes a key/value pair (k1,v1,k2,v2) to generate a set of intermediate key/value pairs, and a reduce function merges all intermediate values [v2] associated with the same intermediate key (k2) (1).

Maps are the individual tasks that transform the input records into intermediate records. A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks. The framework sorts the output of the map, which are then input to the reduce tasks. Both the input and the output of the processed job are stored in a file-system. Typically just zero or one output value is produced by the reducer. In MapReduce, a mapper and reducer is identified by the following signature,

Map (k1, v1) → [(k2, v2)] ` (1.1)
Reduce (k2, [v2]) → [(k3, v3)] (1) (1.2)

Mapreduce suits applications where data is written once and read many times. The data stored in a file system namespace contributes to HDFS (2) which allows master-slave architecture.

VI. ALGORITHM

Input: Raw unstructured Data
Output: Malwares Present in the Database
begin
Collect files from database which we want to analyze;
for each file in the database
    begin
        extract file signatures;
        append signature in signature_file;
end for
load signature_file into Hadoop Distributed File System;
if signature_file is in Apache Pig understandable format then
    Display or Store Result;
end if
for each signature in signature_file
    begin
    if signature has a match in malicious signature database then
        filter out the corresponding file as malicious file;
    end if
end for
end

VII. RESULT

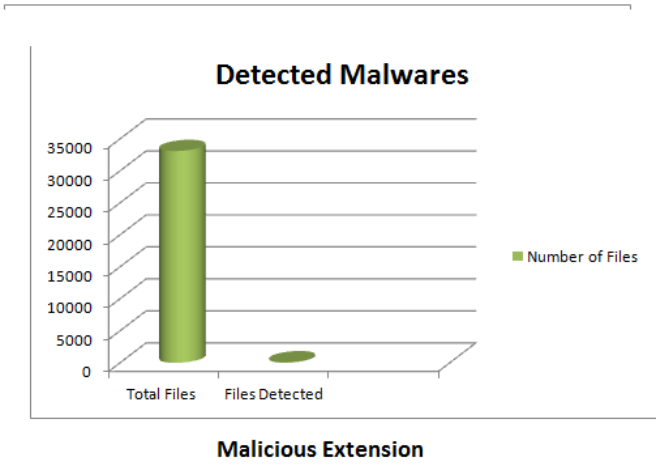

Fig. 3. Detected Malware

From the Fig 3 we found eighty-two malicious files. From the above results it is clear that there are malicious files in the given database.

Compared to the previous techniques which are illustrated in the table no1 this system is efficient for detecting the malicious files. In general the database or the server contains the normal files like text, docs, ppt, jpg etc. The presence of the files with the above extension can be suspect as malicious. Because this server is on the Linux OS and it does not affect the system, but this worm can be attached to it while downloading.

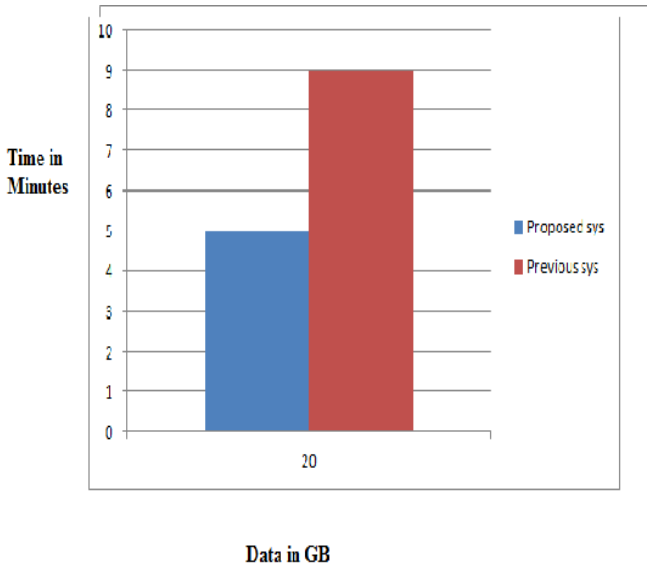

Fig. 4. Comparison of Result

In the malware analysis their proposed systems takes the raw unstructured data and check their malicious signatures. While in our proposed system we first fetch the signature of each file and then compress all the signatures in the text file. By compressing it into individual text file analyzing it will be fast compared to previous system. We tested for 20 GB data i.e. compressed into a single text file. We get a total time of 5 minutes for scanning. While in the previous system we assumed the time of 7 minutes for doing the same task. In our proposed work we are estimating less time because there is benefit of the single text file in which we have signatures with the file name.

VIII. CONCLUSION

We presented a new approach to detect malware through malicious file signature. The previous work done is on intrusion detection system, finding malwares using clusters etc. We have used Hadoop and MapReduce based approach to detect the malware. Writing the code in the MapReduce is quite challenging job in various languages. As we have to specifically give instruction to Mapper and Reducer by using different API's and libraries. To overcome this we utilized Apache Pig which is in the Hadoop Ecosystem and it directly converts its scripts into MapReduce. By using the Apache Pig language we wrote the scripts using different functions of it and performed an optimized MapReduce work for detecting the malicious files. We can detect malicious files with different signatures simultaneously in the given dataset.

REFERENCES

[1] Sayalee Narkhede and Tripti Baraskar, "HMR Log Analyzer: Analyze Web Application Logs over Hadoop MapReduce,"International Journal of UbiComp (IJU) vol.4, No.3, July 2013.

[2] Konstantin Shvachko, et al., "The Hadoop Distributed File System," Mass Storage Systems and Technologies (MSST), IEEE 26th Symposium on IEEE, 2010.

[3] Milind Bhandare, Vikas Nagare et al., "Generic Log Analyzer Using Hadoop Mapreduce Framework,"International Journal of Emerging Technology and Advanced Engineering (IJETAE), vol.3, issue 9, September 2013.

[4] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data" Processing on Large Clusters," Google, Inc.

[5] Savitha K and Vijaya MS "Mining of Web Server Logs in a Distributed Cluster Using Big Data Technologies" International Journal of Advanced Computer Science and Applications (IJACSA). Vol. 5, No. 1, 2014.

[6] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in Proceedings of the OSDI '04, 2004, pp. 137–150.

[7] Aljarah and Simone A. Ludwig. "MapReduce Intrusion Detection System based on a Particle Swarm Optimization Clustering Algorithm," Evolutionary Computation (CEC), 2013 IEEE Congress, June 2013.

[8] Shan Suthaharan. "Big data classification: problems and challenges in network intrusion prediction with machine learning." ACM, March 2014.

[9] Tobias Wüchner, Martín Ochoa and Alexander Pretschner. "Malware Detection with Quantitative Data Flow Graphs." ACM 978-1-4503-2800-5/14/06

[10] Zhiyong Shan and Xin Wang. "Growing Grapes in Your Computer to Defend Against Malware."IEEE, VOL. 9, NO. 2, FEBRUARY 2014.

[11] Acar Tamersoy, Kevin Roundy and Duen Horng Chau , "Guilt by Association: Large Scale Malware Detection by Mining File-relation Graphs," KDD'14, August 24–27, 2014

[12] Tobias Wüchner, Martín Ochoa and Alexander Pretschner. "Malware Detection with Quantitative Data Flow Graphs." ACM 978-1-4503-2800-5/14/06

[13] Ibrahim Aljarah and Simone A. Ludwig. "Towards a Scalable Intrusion Detection System based on Parallel PSO clustering Using MapReduce." ACM 978-1-4503-1964-5/13/07