International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 3, March 2016

Network Data Collection and Analysis

[1]Er. Zainab Mirza, [2] Pragya Mishra, [3] Taher Shabbiri, [4] Asmabee Khan
[1]Head Of Department, [2][3][4] B.E Student,
Department of Information technology,
M.H. Saboo Siddik College of Engineering, Mumbai
[1]mirza_zainab@yahoo.com,[2]pragya.mishra1925@gmail.com,[3]tahershabbiri@gmail.com,
[4]asmabee07khan@gmail.com

***Abstract*— In this paper, we try to implement Hadoop for storing and retrieving different parameters of network log data for analysis. This paper also consists of technologies such as shell script, Hive, PHP in accordance with other web technologies to get a better performance with Hadoop for managing Network traffic.**

***Keywords*—Analysis, Big Data,Google File System (GFS), Hadoop, Hadoop Distributed File System (HDFS), Internet Service Provider (ISP), Java Development Kit (jdk), Java Runtime Environment (jre), Java Server Pages (JSP), Network Data, Network Traffic, Relational Data Base Management System (RDBMS), Shell Scripting.**

I. INTRODUCTION

In this paper, we introduce a different algorithm to use technologies. Using Hadoop Hive and Web technologies such as HTML5, PHP and Java Servlet Pages (JSP), we have proposed a system that dynamically captures logs of network traffic at four minutes interval and stores in the Hive database. The purpose of this project is to perform two major objectives Real-time network data analysis and Statistical data analysis. The former will be used by the operations to detect any network anomalies and the latter will be used by the management to enhance the network performance and for bandwidth shaping for different blocks. This should also act as a forensic tool to capture and trace any historic events related to network performance. The snippets are shown wherever needed for better understanding the logic.

II. BACKGROUND STUDY

This paper is outcome of our academic project in which we have to develop a system to manage the big data of collecting network logs and preparing it for mining stages using Hadoop File System (HDFS), HDFS is the basic core file system of Hadoop just likes for windows it is NTFS. To implement basic idea of our project, the prerequisite knowledge of java, Hadoop and its ecosystems is required. Thus we studied the required technologies which are explained in brief below,

A. Java

We need the latest Java Development Kit (jdk), here we have used jdk 1.8. Also we need Java Runtime Environment (jre), here we have used jre 7.

B. Hadoop

The traditional relational database is now facing the problem of huge amount of data generated in today's world. "Big Data" is defined as "Represents the progress of the human cognitive processes, usually includes data method and theory to capture, manage, and process the data within a tolerable elapsed time"[1]. "Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization"[2]. "The Big Data is subjective for every organization and can be commonly defined as so large data that cannot be handled or it becomes difficult to process it using traditional systems." The Big data can be treated as a problem of today's world and it can be solved by one and only one solution known as Hadoop. Hadoop is the outcome of three research papers published by Google on their used systems to manage their large data Google File System (GFS), Map Reduce and Big Table [3]. The basic principle of Hadoop works of two basic keywords distributed and parallel. Hadoop uses concept of distributed systems to store all the data in HDFS and uses the parallel computing power to build the power of main frame using multiple commodity systems. Hadoop environment consists of two core components:

1) Distributed computation (Map Reduce)

Map Reduce was proposed by Dean and Ghemawat as a programming model for processing large amount of Data[4]. It allows you to parallelize work over a large amount of raw data, such as combining web logs with relational data from an OLTP database to model how users interact with your website. The Map Reduce model simplifies parallel processing by abstracting away the

complexities involved in working with distributed systems. Map Reduce decomposes work submitted by a client into Small parallelized map and reduce workers. MapReducelike tools are still new in the market and there is much room for improvement. In particular, from the performance point of view, Map Reduce has been criticized for its inefficiency [5].

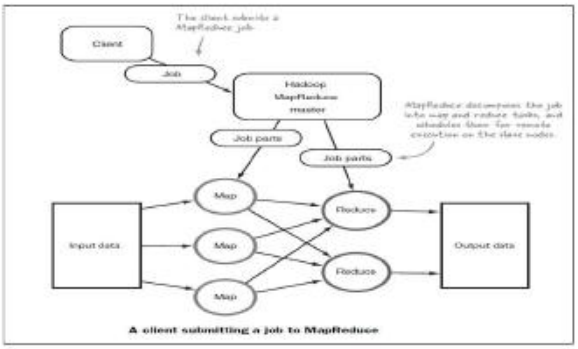

Fig. 1: Map Reduce

2) Distributer Storage (HDFS)

H DFS is the Storage component of Hanoop. It's a distributed five system that modeled after the google file System (GFS) paper. HDFS is optimized for high throughput and works best when reading and writing large files (gigabytes and larger). HDFS replicates files for a configured number of times, is tolerant of both software and hardware failure, and automatically re-replicates data blocks on nodes that have failed.

.

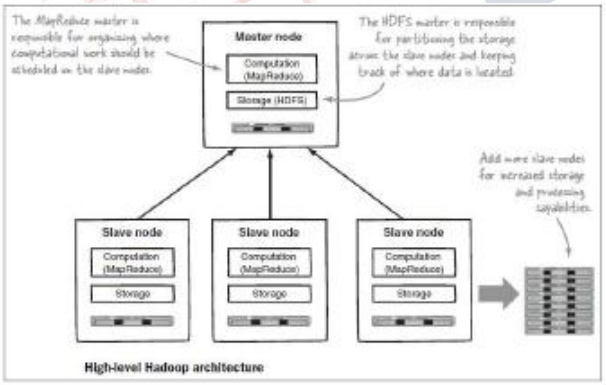

Fig. 2: Hadoo p Architecture

Hadoop framework is a family of technologies that is provided under a single umbrella known as Hadoop ecosystem. The flow of the components of ecosystem makes it possible to use Hadoop to its fullest depending on the type of technology needed in the organization.

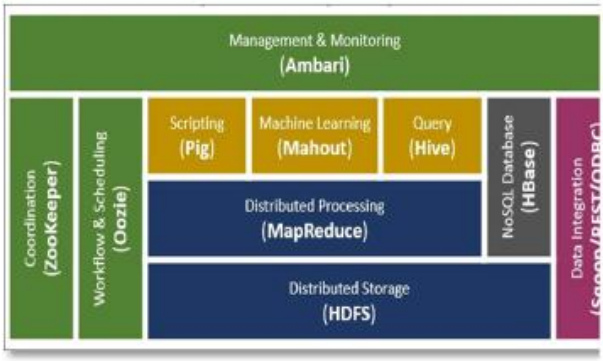

Fig. 3: Hadoo p Ecosystem

a) Hive

Hive is Data warehouse infrastructure for Hadoop that uses SQL-like language called HiveQL for query processing and analysis. There are many other ecosystems as shown and many are not shown. Every ecosystem has its own use and application of system but the core components will always be present that is HDFS and MapReduce, rest can vary, for example, in our project we could use Pig and Hive for implementing basic model of project but we selected Hive over Pig because of one reason that Pig does not support Web Technologies like PHP, HTML and so on. On the other hand, Hive supports web technologies and shell scripting.

III. PROBLEMS IN EXISTING SYSTEM APPROACH

Over the past few decades, a lot of tools have been developed and widely used for Network traffic monitoring such as TCPdump[6], Wireshark[7], CoralReef[8], Cisco NetFlow[9], Peakflow[10], Ntopng[11], etc. Network Traffic Monitoring and Analysis is all about knowing the behavior of your users and being well versed with your network. The above mentioned systems provide partial solution to the institute's requirement. The software's provide limited functionalities and are difficult to learn and use. Typically, it takes months to learn their features and capabilities. Most administrators use only fraction of the features provided, performing only basic tasks such as traffic monitoring and fault monitoring. Further, most administrators change their job. Thus increasing the overhead to retrain the new staff how to use the particular system. Consequently, a tool that is easy to learn and operate in a short period of time is desperately needed. Also, the given tools work on limited amount of network traffic and have counterproductive features that all enterprises do not need. This resulted in the need to develop a robust application for the Network Administrator who shall monitor the user's behavior and receive alerts if any anomaly is detected in the network.

IV. PROPOSED SYSTEM:

The data coming from the ISP (Internet Service Provider) is being port mirrored to the Linux based server which uses ntopng and nprobe to generate binary logs. Since this binary logs are not human readable, these binary logs are converted to .flows file (readable format can be .txt). Now, .flows files are used for two purposes one is to add the new logs generated from network into the Hive database in HDFS of Hadoop Server and on the other hand these .flows files are being used to get the real time network results on webpages.

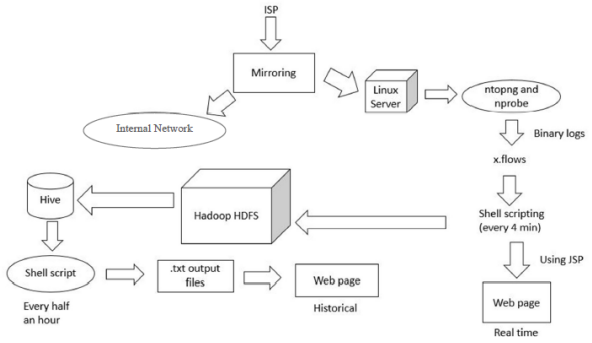

Fig. 4: Flow of project

A. Real Time Data Analysis:

These .flows files are filtered and segregated to extract useful information using JSP or shell script. A web page is used to display the extracted information in real-time using multiple web technologies.

Fig. 5: Top 10 Downloads

Fig. 6: Top 10 Port usage downloads

B. Statistical Data Analysis:

The same generated .flows files are also stored in HDFS for historical data analysis. Since the real time content is not so huge as compared to historical data stored in HDFS, the traditional approach to extract and display on web page fails because of its poor performance. The user or network admins need to wait for 5 to 10 minutes for a simple query generated and fired. Thus we thought and implemented a different approach to overcome this drawbacks. We used shell scripts program to run hive query every 15 minutes to generate the results of each query and store the output or results into a normal text file say result files as shown below.

```
max_d.txt
158.144.*.*,1410516267
31.216.*.*,221178911
0.0.*.*,165443840
191.234.*.*,162046888
115.254.*.*,140051672
31.13.*.*,105793371
115.254.*.*,104360227
115.254.*.*,99617359
173.194.*.*,95270982
115.254.*.*,94910998
```

```
total 16
-rw-r--r-- 1 root root 239 Jan 20 18:30 max_d.txt
-rw-r--r-- 1 root root 235 Jan 20 18:33 max_u.txt
-rw-r--r-- 1 root root 145 Jan 20 18:35 pd_logs.txt
-rw-r--r-- 1 root root 145 Jan 20 18:36 pu_logs.txt
```

Fig. 7: Results in .txt files

Our logic is based on principle that it is fact that historical analytical results do not have a big change in an hour, which means the difference of result of a query in historical data in hours is almost negligible and doesn't have great impact on our day-to-day decisions. Thus this result files will always be present and will be overwritten every 15 minutes (depends of applications) by using following shell scripts.

```
#!/bin/bash
#top 10 downloads max_d.csv
hive -e 'select src_addr,sum(in_bytes) as downloads from
logs  group by src_addr order by downloads desc limit
10'|sed -e 's/\s/,/g' > /res_files/max_d.txt

#top 10 uploads max_u.txt
hive -e 'select src_addr,sum(out_bytes) as uploads from logs
group by src_addr order by uploads desc limit 10'|sed -e
's/\s/,/g'  > /res_files/max_u.txt

#top 10 protocol wise download
hive -e 'select src_port,sum(out_bytes)as t from logs group
by src_port order by t desc limit 10'|sed -e 's/\s/,/g'  >
/res_files/pd_logs.txt

#top 10 protocol wise upload
hive -e 'select src_port,sum(out_bytes) as t from logs group
by src_port order by t desc limit 10'|sed -e 's/\s/,/g'  >
/res_files/pu_logs.txt
```

Whenever a network admin access the historical analysis web page, the web page actually reads these result files of particular query using PHP and displays to user. This gives amazing speed and in no time result is displayed to network administrations.

The graphs are displayed using Google APIs and JavaScript. More details about these Google APIs can be found on Google developer's console [13]. In our project we have been using JavaScript to build array and send it to Google APIs, in response it returns the required chart to web page. The developer can select any type of graph or chart from many options available provided by Google, following snippet is shown below.

```
<script type="text/javascript"
src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
  google.load("visualization", "1",
{packages:["corechart"]});
  google.setOnLoadCallback(drawChart);
  function drawChart() {
   var dataSet =  <?php echo $che ?>;
    var data =
google.visualization.arrayToDataTable(dataSet);
    var options = {
      title: 'Top 10 Downloads'


  };
  var chart = new
google.visualization.ColumnChart(document.getElementById('ch
art_div'));
  chart.draw(data, options);
 }
</script>
```

V. CONCLUSION

Thus we conclude that this system will provide a better performance than traditional methodology. In this paper we have presented the work on network traffic stored in HDFS and analysis of both real time and historical based data. This is the methodology which can be used by developers the system that consist of processes from collecting the data to managing it for analysis of different network parameters.

REFERENCES

[1] "Big data: science in the petabyte era," Nature 455 (7209): 1, 2008.

[2] Douglas and Laney, "The importance of 'big data': A definition," 2008.

[3] Google Research Papers, static.googleusercontent.com/media/research.google.com/en//archive/. 2004 and 2006

[4] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. In OSDI, pages 137-150{ 2004}.

[5] D. DeWitt and M. Stonebraker, "Mapreduce: A Major Step Backwards," Database Column, 2008.

[6] Tcpdump, http://www.tcpdump.org.

[7] Wireshark, http://www.wireshark.org.

[8] CAIDA CoralReef Software Suite, http://www.caida.org/tools/measurement/coralreef.

[9] Cisco NetFlow, http://www.cisco.com/web/go/netflow.

[10] Arbor Networks, http://www.arbornetworks.com.

[11] Ntopng, http://www.ntop.org/products/trafficanalysis/ntop/

[12] Hadoop in Practise by Alex Holmes https://manningcontent.s3.amazonaws.com/download/0/63e4590-f9ab-4f35-825b-36a3d3b99fc4/HiP_sample_ch1.pdf

[13] Google Developer's Console, https://developers.google.com/chart