International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 1 165 - 168

TAMC: Traffic Analysis Measurement and Classification Using Hadoop Map Reduce

Yogesh Vasant Kadam, Student, RMD Sinhgad School of Engg. Pune. India.
Prof. Mrs. Vina Lomte, Assistant Professor, RMD Sinhgad School of Engg. Pune. India.

Abstract- Due to growth in Internet users and bandwidth-hungry applications; the amount of Internet traffic data generated is so huge. It requires scalable tools to analyze, measure, and classify this traffic data. Traditional tools fail to do this task due to their limited computational capacity and storage capacity. Hadoop is a distributed framework which performs this task in very efficient manner. Hadoop mainly runs on commodity hardware with distributed storage and process this huge amount of traffic data with a Map-Reduce programming model. We have implemented Hadoop-based TAMC tool which perform Traffic Analysis, Measurement, and Classification with respect to various parameters at packet and flow level. The results can be used by Network Administrator and ISP's for various usages.

Keywords- *Internet Traffic, HDFS, Map-Reduce.*

*****

1. Introduction

According to Cisco white paper [1], annual global traffic will reach near about 120.6 Exabyte's per month. The reason for this much amount of growth in Internet traffic data is due to the bandwidth-hungry applications like File Transfer applications, Video Streaming, Social Media Network (Facebook, Twitter etc.), Mobile applications, E-commerce websites, Stock Exchange data and much more as shown in Figure 1. As the traffic data increases it is necessary to analyze, measure, and classify it as ISP and Network Administrators need it from various perspectives like network planning, traffic shaping, billing and extract useful information.

Figure 1. Big Data Era

This task needs to be performed with various tools available in market (Ex: Tcpdump, Wireshark etc.)These tools capture the network traffic and store it onto a local server for further processing. As the size of traffic data increases; the storage capacity needs to be expanded which incurs cost as well as to process this data it requires scalable processing framework. The traditional tools are not scalable.

If we distribute this traffic data onto multiple remote machines (server) reduces the storage problem but while processing this data; fault-tolerance policy is not considered i.e. if one machine fails then tool will unable to recover it from failure and will result in inefficient processing. Due to large access to remote machines; the performance of system degrades. Data availability is also another issue while considering traffic analysis. If a machine fails the data present on that machine will be unavailable for further processing. So it is very important to develop a powerful tool which cannot be restricted to storage and processing power capabilities.

Apache Hadoop [2] is a distributed framework written in java to store huge amount of data and process the same. The Hadoop mainly based on two concepts: The Hadoop Distributed File System (HDFS) and MapReduce. HDFS is scalable with respect to data storage on commodity hardware i.e. low cost hardware based on Google File System (GFS) [3]. MapReduce [4] is a parallel processing framework developed by Google to process large amount of data. It mainly works with *map* and *key* function. The Hadoop is mainly used to process semi-structured and unstructured data. It overcomes all the drawbacks of traditional Distributed File System (DFS) such as Grid Computing, Volunteer Computing [5] by providing strong features like fault-tolerance, scalability, and availability.

Today most of the companies are using Hadoop technology like Yahoo, Last.fm, Facebook, IBM etc. Larger clusters are available to users to run their job (Amazon [20], Grid5000 [21]) with Hadoop on Demand (HOD) [6].

The organization of this paper is as follows: Chapter 2 focuses on related work done in traffic analysis with its pros and cons. Chapter 3 gives detailed description of proposed work with implementation. Chapter 4 gives experimental setup and results of proposed system. At last Chapter 5

 ISSN: 2321-8169
165 - 168

concludes the paper and focuses on future direction to our work.

2. Related Work

It is necessary to monitor Internet traffic continuously because of enormous growth in traffic size and changing behavior of user. Various tools are available in market to analyze, classify, and measure the Internet traffic. Traffic measurement can be done at packet level and flow level. Tcpdump [7] is a command-line based tool for capturing and analyzing traffic. A Graphical User Interface (GUI) is integrated with Wireshark [8] which provides a good understanding of traffic analysis by displaying statistics. The CoralReef [9] tool developed by CAIDA is port-based traffic classification tool which achieves high precision and recall for several legacy applications such as Domain Name System (DNS), Chat, Secure Shell (SSH), Mail, and Web Traffic. The method proposed in BLINC [10] also classifies traffic according to behavior where the profile of host is captured to identify applications that host engaged in and then classifies traffic flows. The machine learning algorithm [11] presented towards the IP traffic classification.

These tools are good for analysis but restricted to storage capacity and processing power capability. To overcome these restrictions, traffic sampling [12, 13] can be used where partial observations are made to draw results but it will result in loss of information. The traditional relational database using SQL is also impractical due to sequential nature of query operations.

If we distribute the traffic data among multiple nodes then in distributed environment there are chances of failure of certain machines hence, data availability becomes a critical issue also in traditional tools the fault-tolerance issue is not handled. Therefore it is necessary to develop tool which overcomes all the problems faced by older tools.

3. Proposed System

By considering all the problems of older tools, we have proposed TAMC: Traffic Analysis, Measurement, and Classification tool with Hadoop MapReduce. It considers various aspects of traffic data such as IP address wise traffic count, total Size of traffic data, date-wise traffic count along with port based classification where total traffic and total size per port is calculated. The system architecture (inspired from [14]) is as shown in figure 2.

We have captured Internet traffic (TCP and UDP only) from router of our Bharati Vidyapeeth Campus, Palus which is basically stored in pcap or libpcap format as per specification. The Slave Nodes i.e. DataNodes stores this traffic data with replication factor of 2 means one file get stored onto 2 different slaves for data availability.

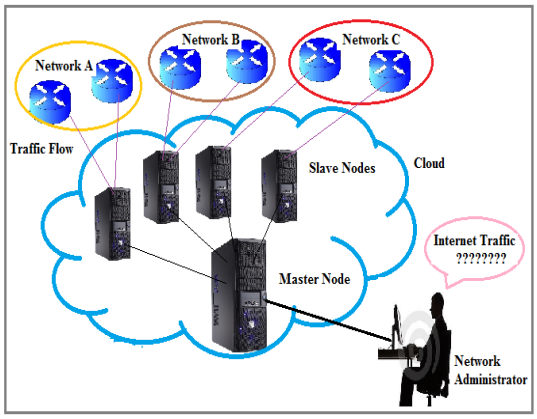

Figure 2. TAMC System Architecture.

The captured file format is libpcap or pcap is then converted to text format because text file contains carriage return character which is helpful while parallelizing input file. We have used TShark [15] (command-line version of Wireshark) which convert pcap or libpcap file into text file. Recently, authors of [16] and [18] have been developed a parallel packet processor which processes the packet in parallel manner using packet timestamp field and sliding window basics i.e. we can directly store and process libpcap files. Then we run our TAMC MapReduce programs to extract useful information from traffic data at packet level as well as at flow level. To convert packets captured into flow we have used CapLoader utility [17] which is useful for flow level analysis.

The proposed method calculates number of packets sent with respect to IP address then it also gives us size of packets sent with respect to IP address and port numbers. It also performs port-based classification of traffic data where we classify our traffic as TCP or UDP. Our tool requires just simple mathematical calculations for the traffic analysis. The algorithm proposed here is as follows:

1: *Algorithm*: *TAMC*
2: *Input*: *traffic captured in libpcap or pcap format*
3: *Output*: *txt file or web-based*
4: *Method*: *start capture*
5: *Convert libpcap file format into txt file format*
6: *Store traffic data onto HDFS*
7: *Calculate*
  i) ip_address_wise_packet_count
  ii) ip_address_wise_packet_size
  iii) port based classification
  iv) Date_wise_traffic_count
  v) Date_wise_traffic_size
8: *Store result back to the local file system*
9: *End*

Figure 3. TAMC Algorithm

 ISSN: 2321-8169
165 - 168

4. Experimental Setup and Results

To carry out the experiment we have installed VMware 8.0.4 on machine with two Ubuntu 12.04 nodes. Both machines having openjdk1.7 installed in it and SSH enabled. Hadoop 1.2.1 have been configured on both nodes to use the HDFS and MapReduce capabilities. The NameNode structure is given in figure 4.

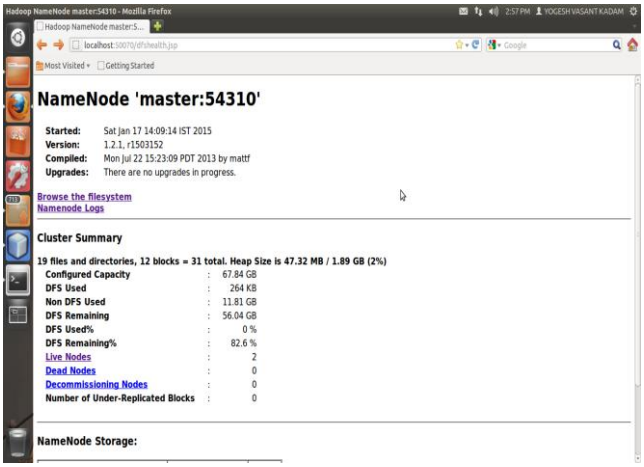

Figure 4. NameNode Structure

The NameNode is core part of Hadoop because it controls all the DataNodes present in cluster. It is a Single-Point-of-Failure but recent versions (0.21+) come with BackupNameNode [19] to make it highly available. The DataNodes contain all the data in cluster on which we will operate our MapReduce programs and view the traffic data from various perspectives. JobTracker controls all the tasks which are running on TaskTrackers shown in following figure 5.

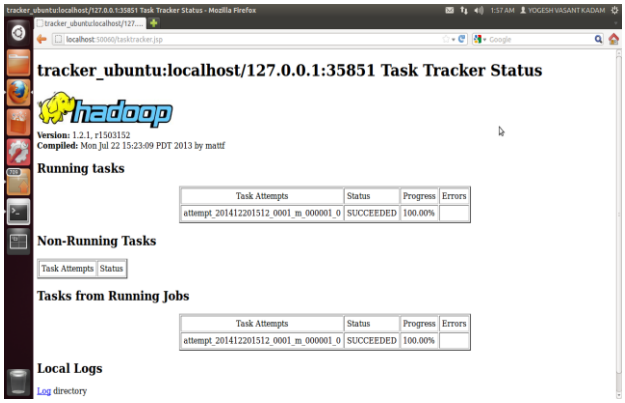

Figure 5. TaskTracker

We have developed different types of MapReduce programs which take the input as a traffic data and calculate the traffic size, port based classification, Date-wise traffic, and IP address-wise traffic. We have captured internet traffic of size 10MB. The results are shown in following figures in terms of graphs:

i) At Flow Level:

The captured packets are converted into flow using CapLoader utility.

a) Protocol-Based Classification: We have classified internet traffic according to protocols such as TCP and UDP. The simulation results are shown in figure 6.

b) Port Based Classification: There are various TCP and UDP ports. Some of which are well-known ports whereas others are classified as ephemeral ports. The port numbers used by server are generally well-known ports and client uses ephemeral ports. For example, port 21 is mainly used for FTP operation; port 80 is used for HTTP communication and like this. The results are shown in figure 7.

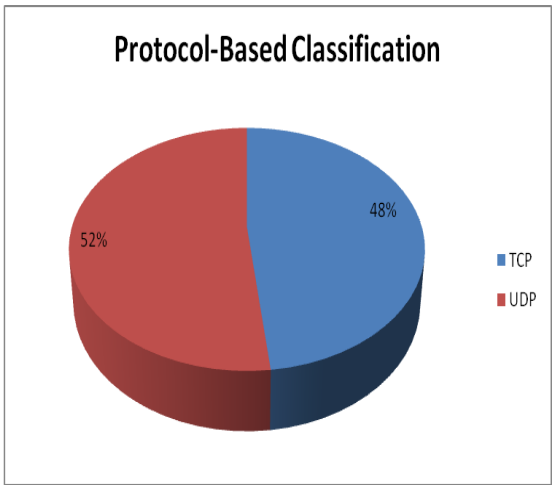

Figure 6. Protocol Based Classification

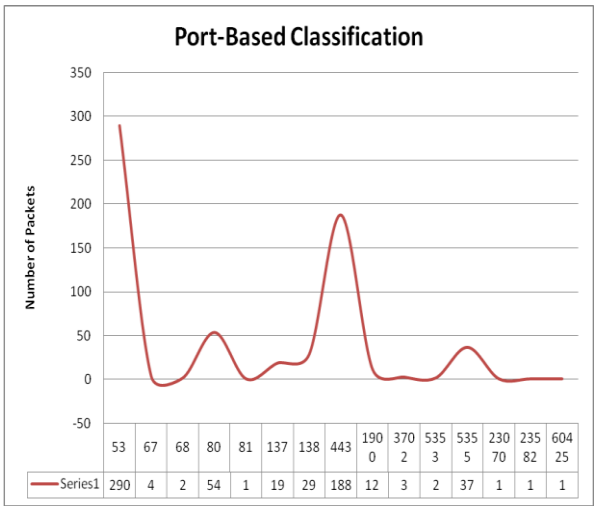

Figure 7. Port Based Classification

ii) At Packet Level:

We will also operate our MapReduce programs at packet level.

a) Top 10 IP Addresses:

ISSN: 2321-8169
165 - 168

We can calculate the top 10 users from the packets captured who generates more traffic as shown in figure 8. This information can be used for predicting who the users which are consuming more bandwidth are.

a) Port-Wise Byte Count:

We have also calculated the total size of packets in port-wise fashion. For simplicity we have shown only top 10 ports. Port 443 (HTTPS) having higher number of byte count. The results are shown in figure 9. We have also calculated number of packets per day and size of packets per day.

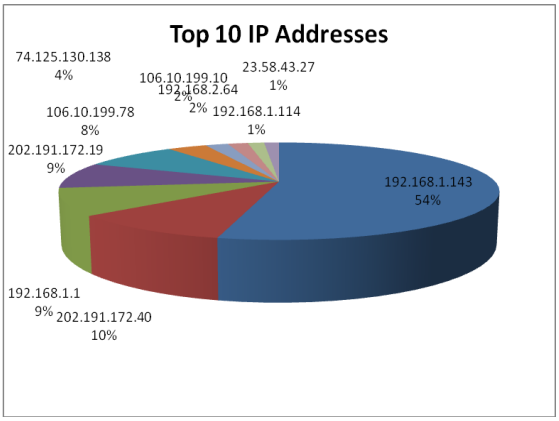

Figure 8. Top 10 IP Addresses

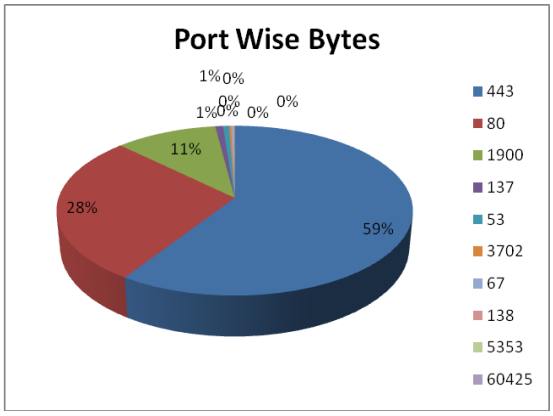

Figure 9. Port-Wise Byte Count

5. Conclusion and Future Work

The traffic analysis, measurement and classification play an important role in day to day life of network engineers as well as ISP's. In this proposed system, we have developed TAMC tool with Hadoop MapReduce which makes the calculations simpler though data size becomes larger upto any extent. The drawback with current system is that it requires manual intervention at certain stages and it uses FIFO scheduler to schedule the MapReduce jobs. The performance can be improved using certain automation techniques and fair scheduler for scheduling.

6. References

[1] Cisco White Paper, "*Cisco Visual Networking Index: Forecast and Methodology, 2012-2017*" May 2013.
[2] Hadoop, http://hadoop.apache.org
[3] S. Ghemawat, H. Gobioff, and S. Leung, The Google File System, ACM SOSP, 2003.
[4] J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on Large Cluster, USENIX OSDI, 2004.
[5] Tom White, "Hadoop: The Definitive Guide", O'Reilly, 3rd ed., 2012
[6] http://hadoop.apache.org/docs/r0.18.3/hod.html
[7] Tcpdump, http://www.tcpdump.org
[8] Wireshark, http://www.wireshark.org
[9] CAIDA CoralReef Software Suite, http://www.caida.org/tools/measurement/coralreef.
[10] Karagiannis, T., Papagiannaki, K. and Faloutsos, M., BLINC: Multilevel Traffic Classification in the Dark, ACM SIGCOMM 2005, August / September 2005.
[11] T. Nguyen, G. Armitage. A Survey of Techniques for Internet Traffic Classification using Machine Learning. IEEE Communications Surveys and Tutorials, vol. 10, no. 4, 2008.
[12] Duffield, N. and Lund, C., Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure, ACM Internet Measurement Conference 2003.
[13] Duffield, N., Lund, C. and Thorup, M., Estimating Flow Distributions from Sampled Flow Statistics, ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.
[14] Y. Lee, W. Kan, and H. Son, An Internet Traffic Analysis Method with MapReduce, 1st IFIP/IEEE Workshop on Cloud Management, April 2010.
[11] TShark,www.wireshark.org/docs/manpages/tshark.html
[12] Y. Lee and Y. Lee, A Hadoop-based Packet Processing Tool, TMA, April 2011.
[13] http://www.netresec.com/?page=CapLoader
[14] Y. Lee and Y. Lee, Towards Scalable Internet Traffic Measurement and Analysis with Hadoop, ACM SIGCOMM Computer Communication Review, 2013.
[15] http://hadoop.apache.org/docs/r1.0.4/hdfs_user_guide.html
[16] http://aws.amazon.com/ec2/
[17] https://www.grid5000.fr/mediawiki/index.php/Grid5000:Home