Traffic Measurement and Analysis with Hadoop

Dipti J. Suryawanshi, Prof. Mr. U. A. Mande

Department of Computer Engineering, Sinhgad College of Engineering, University of Pune, India

Department of Computer Engineering, Sinhgad College of Engineering, University of Pune, India

ABSTRACT: *In computer network, network traffic measurement is the process of measuring the amount and type of traffic on a particular network. Nowadays internet traffic measurements and analysis are mostly used to characterize and analysis of network usage and user behaviors, but faces the problem of scalability under the explosive growth of Internet traffic and high-speed access. As the number of network elements, such as routers, switches, and user devices, has increased and their performance has improved rapidly, it has become more and more difficult for Internet Service Providers (ISPs) to collect and analyze efficiently a large data set of raw packet dumps, flow records, activity logs for accounting, management, and security. To satisfy demands for the deep analysis of ever-growing Internet traffic data, ISPs need a traffic measurement and analysis system where the computing and storage resources can be scaled out. Scalable Internet traffic measurement and analysis is difficult because a large data set requires matching computing and storage resources. Hadoop, an open-source computing platform of MapReduce and a distributed file system, has become a popular infrastructure for massive data analytics because it facilitates scalable data processing and storage services on a distributed computing system consisting of commodity hardware. This paper presents a Hadoop-based traffic monitoring system that performs IP, TCP, HTTP, and NetFlow analysis of multi-terabytes of Internet traffic in a scalable manner. And also explain the performance issues related with traffic analysis MapReduce jobs.*

Keywords: *Hadoop, Hive, MapReduce, NetFlow, pcap, packet, traffic analysis, traffic measurement*

I. INTRODUCTION

Traffic measurement and analysis required a large amount of data storage and high-performance computing power to manage a huge amount of traffic data set. Google is one of the search engines that can easily scale out with MapReduce and GFS [10, 11]. MapReduce [10] is a programming model for processing large data sets, generally used for data intensive tasks, by Google. It is typically used to do distribute computing on clusters of computers. MapReduce jobs are executed on Google's clusters every day, processing a total of more than twenty petabytes of data per day. Since Google's MapReduce and Google file system (GFS) [11] are fix, an open-source MapReduce software project, Hadoop [12] an open-source computing platform of MapReduce and a distributed file system, was developed to provide similar capabilities of the Google's MapReduce platform by using thousands of cluster nodes. Hadoop distributed file system (HDFS) is an important component of Hadoop, that corresponds to GFS. Yahoo!, Amazon, Facebook, IBM, Rackspace, Last.fm, Netflix, and Twitter are using Facebook also uses Hadoop to analyze the web log data for its social network service. Hadoop for its scalability in storage and computing power is a suitable platform for Internet traffic measurement and analysis.

In this paper, we develop a Hadoop based scalable internet traffic measurement and analysis system that can manage the packets and NetFlow data on HDFS. By applying Hadoop to an Internet traffic measurement and analysis, we need to face some challenges that are: 1) to parallelize MapReduce I/O of packet dumps and NetFlow records in HDFS-aware manner, 2) to devise traffic analysis algorithms especially for TCP flows dispersed in HDFS, and 3) to design and implementation an integrated Hadoop-based Internet traffic monitoring and analysis system practically useful to operators and researchers. After that we propose a binary input format for reading packet and NetFlow records concurrently in HDFS. Then we present MapReduce analysis algorithm for NetFlow, IP, TCP, HTTP, traffic. After that we prove that how to analyze efficiently the TCP performance metrics in MapReduce in the distributed computing environment. Finally we create web based agile traffic warehousing system using Hive [13] presents a large amount of Internet traffic analysis system with Hadoop that can quickly process IP packets as well as NetFlow data through scalable MapReduce based analysis algorithms for large IP, TCP, and HTTP data. It also show that the data warehousing tool Hive is useful for providing an agile and elastic traffic analysis framework.

The remaining part of this paper is organized as follows. In Section 2, we describe the related work on traffic measurement and analysis as well as work on MapReduce and Hadoop. The architecture of Hadoop based traffic measurement and analysis system and its components are explained in Section 3, And the experimental results are presented in Section 4. Finally Section 5 concludes this paper.

II. RELATED WORK

There are a lot of packet processing tools available for traffic measurement and analysissuch as Cisco NetFlow [1] is the popular flow monitoring format. By using this tool we can easily monitor flows passing through routers or switches without observing every packet, but routers or switches do not support NetFlow, for that we need to use NetFlow-compatible flow generator. Many open-source or commercial flow analysing tools exist, including flow-tools [2], flowscan [3], argus [4], and Peakflow [5]. Tcpdump [6] is mostly used for capturing and analyzing packet traces, and various tools based on the packet capture library, “libpcap” [6], such as wireshark [7], CoralReef [8], and Snort [9] have been deployed to handle packets. However, above traffic tools are not suitable for processing a huge data set of multi-terabytes monitored at high speed linksThe functionality of MapReduce applications on Hadoop is to analyze large text, web, or log files. Observing preliminary work [14], we get result that the first packet processing method for Hadoop that analyses packet trace files in parallel manner by reading packets over multiple HDFS blocks. Recently RIPE [15] provide same library for Hadoop, but it does not have parallel processing capabilities while reading packet records from HDFS blocks of file because of that its performance is not scalable and its recover capability against task failures occurs and it is not efficient. In this paper, we design a global traffic measurement and analysis system using Hadoop that can quickly process IP packets and NetFlow data through scalable MapReduce- based analysis algorithms for large IP, TCP, and HTTP data. We also provide data warehousing tool Hive which is useful for providing an agile and elastic traffic analysis framework.

III. COMPONENTS OF TRAFFIC MEASUREMENT AND ANALYSIS SYSTEM

This section consists of main components of Traffic Measurement and Analysis System, like Traffic Collector/Loader receives packets or NetFlow from various network elements like router stores it in the HDFS, and Traffic Analyzer performs the core operation. And return the analysis results. The overall description and architecture is illustrated below.

1. Traffic Collector/Loader

Traffic collector captures the packet or NetFlow data from various network resources like router and stores them to HDFS simultaneously. The received packet trace file is in the libpcap format. Traffic loader reads it and split them into the fixed-size of chunks, and stores the chunks of files to HDFS. The Traffic Collector uses libpcap and jpcap modules for capturing packets from routers, and the HDFS stream writer use for saving packet data. For stream writer we need to develop a HDFS writing module in java. The Traffic Loader just stores the packet to HDFS by using HDFS stream writer. In HDFS, most of the packet trace file larger than the specified HDFS block size (default 64 MB) is split, and by default three replicas for every block are copied into HDFS for fault tolerance. The trace file if in the binary formed packet data, which has the non-fixed length of packet records.

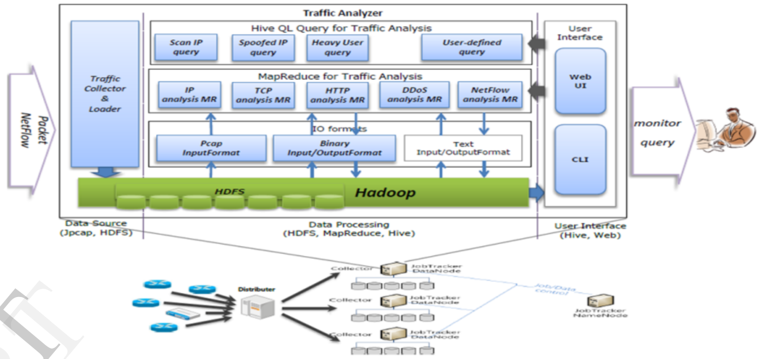


Fig.1: Architecture of Traffic Measurement and Analysis with Hadoop

2. Packet Reader

For reading packet, we need to develop the technique like; new Hadoop APIs that can read or write IP packets and NetFlow v5 data in the native libpcap format on HDFS. The new Hadoop API makes it possible to directly save the libpcap streams or files to HDFS and run MapReduce analysis jobs on the given libpcap files.

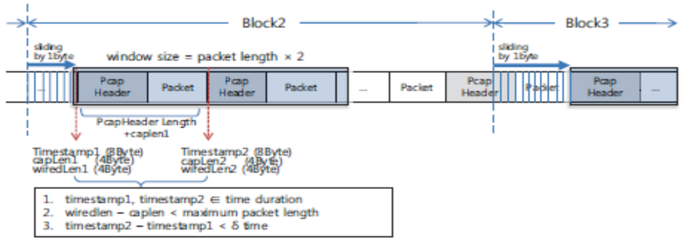


Fig.2: Reading packet records on HDFS blocks for parallel processing.

In order for multiple map tasks to read packet records of HDFS blocks in parallel, for this one algorithm is propose a heuristic algorithm [14] that can process packet records per block by using the timestamp-based bit pattern of a packet header in libpcap. Fig. 2 shows how each map task delineates the boundary of a packet record in HDFS. From assumption the timestamp fields of two continuous packets stored in the libpcap trace file are similar. When a MapReduce job begins, it invokes multiple map tasks to process their allocated HDFS blocks. Each map task equipped with the packet record-search algorithm considers the first 16 bytes as a candidate for a libpcap packet header, reads the related fields of two packets using the caplen field, and verifies each field value of the two packet records. First, the timestamp value should be within the time duration of the captured packet traces. Then, the focus goes to the integrity of the captured length and the wired length values of a packet header, which should be less than the maximum packet length. Third, the timestamp difference between two contiguous packet records should be less than threshold.

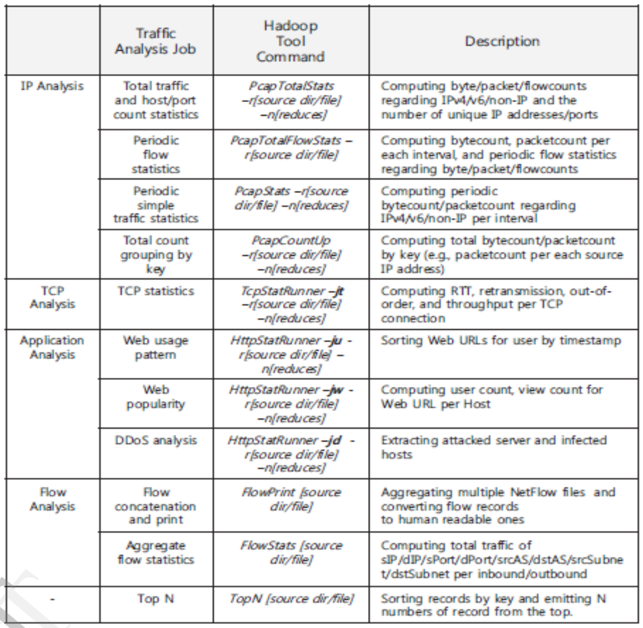
3. Analysis on Network layer in MapReduce

For network-layer analysis, we have develop IP flow statistics tools in MapReduce, as shown in Fig. 3 Computing IP flow statistics certainly fits the MapReduce framework because it is a simple counting job for the given key and it can be independently performed per HDFS block. For the packet trace file, we can tally the IP packet statistics, such as byte/packet/IP flow counting, IP flow statistics, periodic traffic statistics, and top *N*, as shown in Fig. 3, The IP analysis is mostly an aggregation work that summarizes the corresponding count values associated with IP addresses and port numbers. PcapTotalFlowStats periodically calculates flow information, which consists of IP addresses and ports from packet trace files. This analysis requires two MapReduce jobs for flow statistics and aggregated flow information, respectively. PcapStats tallies the total byte/packet and bit/packet rate per time window for the given packet trace files. PcapCountUp summarizes the total byte/packet count for given packet trace files regardless of any time window. TopN finds the most popular statistics, such as top 10 IP or TCP flow information.

4. Analysis on Transport Layer in MapReduce

In contrast to IP analysis, TCP performance metrics such as round-trip time (RTT), retransmission rate, and out-of-order cannot be computed per HDFS block, because the computation of these metrics is not commutative and associative across TCP flow parts on several HDFS blocks. In TCP analysis with MapReduce, in this there are two challenges: constructing a TCP connection by stitching directional TCP flow parts spread out across multiple HDFS blocks; optimizing the TCP performance metric calculation work in MapReduce.

Table1: IP and NetFlow analysis tools



	Traffic Analysis Job	Hadoop Tool Command	Description
IP Analysis	Total traffic and host/port count statistics	*PcapTotalStats –r[source dir/file] –n[reduces]*	Computing byte/packet/flowcounts regarding IPv4/v6/non-IP and the number of unique IP addresses/ports
	Periodic flow statistics	*PcapTotalFlowStats –r[source dir/file]*	Computing bytecount, packetcount per each interval, and periodic flow statistics regarding byte/packet/flowcounts
	Periodic simple traffic statistics	*PcapStats –r[source dir/file] –n[reduces]*	Computing periodic bytecount/packetcount regarding IPv4/v6/non-IP per interval
	Total count grouping by key	*PcapCountUp –r[source dir/file] –n[reduces]*	Computing total bytecount/packetcount by key (e.g., packetcount per each source IP address)
TCP Analysis	TCP statistics	*TcpStatRunner **–jt** –r[source dir/file] –n[reduces]*	Computing RTT, retransmission, out-of-order, and throughput per TCP connection
Application Analysis	Web usage pattern	*HttpStatRunner **–ju** -r[source dir/file] –n[reduces]*	Sorting Web URLs for user by timestamp
	Web popularity	*HttpStatRunner **–jw** -r[source dir/file] –n[reduces]*	Computing user count, view count for Web URL per Host
	DDoS analysis	*HttpStatRunner **–jd** -r[source dir/file] –n[reduces]*	Extracting attacked server and infected hosts
Flow Analysis	Flow concatenation and print	*FlowPrint [source dir/file]*	Aggregating multiple NetFlow files and converting flow records to human readable ones
	Aggregate flow statistics	*FlowStats [source dir/file]*	Computing total traffic of sIP/dIP/sPort/dPort/srcAS/dstAS/srcSubnet/dstSubnet per inbound/outbound
-	Top N	*TopN [source dir/file]*	Sorting records by key and emitting N numbers of record from the top.

5. Analysis on Application Layer in MapReduce

libpcap input format in HDFS makes it possible to build up application-specific analysis MapReduce modules for web, multimedia, file sharing, and anomalies. In this work, the main focus on the HTTP-based application analysis in MapReduce, because HTTP is popular in many Internet applications.

5.1 Web Traffic Analysis

For web traffic analysis, it is necessary to develop a MapReduce algorithm that can investigate website popularity and user behaviour by examining HTTP packets. In the MapReduce algorithm, first, the map task extracts several fields out from the header of a HTTP request message, such as a Uniform Resource Identifier (URI), content-related fields, user agent, and host values. Then, the reduce task summarizes the statistics or popularity per website (or domain/host), webpage (or URL), content URI, or user. A website is considered as a unique host name field at the HTTP message. Each web page, specified as a URL, consists of several content URIs that share the same referrer field. For web traffic analysis, next task was implemented two MapReduce jobs: the first job sorts the URL list accessed by each user; the second job summarizes the user and view counts per host and URL. HTTP analysis can be easily extended to other Internet applications, such as SIP for video or VoIP, file-sharing, and anomalies.

5.2 DDoS Traffic Analysis

In order to show the effectiveness of MapReduce-based application-layer traffic analysis, it needs to present a HTTP-based distributed denial-of-service (DDoS) attack detection method implementation in MapReduce. To employ a counter-based DDoS detection algorithm in MapReduce. The map task generates keys to classify the requests and response HTTP messages. Then, the reduce task summarizes the HTTP request messages and marks the abnormal traffic load by comparing it with the threshold. Though the DDoS traffic analysis MapReduce algorithm is used for offline forensic analysis, it can be extended to real-time analysis.

6. Query Interface with Hive

The system implements a NetFlow monitoring system with derived MapReduce flow analysis algorithms and Hive. The traffic collector receives the NetFlow streams from a router and writes them to a single HDFS file every five minutes. Whenever a NetFlow file is closed after that at every five minutes, the job scheduler invokes IP analysis MapReduce jobs to process NetFlow data and uploads flow records and IP statistics to Hive tables. Then, operators can issue user defined queries to the Hive table over the characteristics of IP flow data.

IV. CONCLUSION

In this paper, describes a scalable Internet traffic measurement and analysis scheme with Hadoop that can process multi-terabytes of libpcap files. Based on the distributed computing platform, Hadoop, we have devised IP, TCP, and HTTP traffic analysis MapReduce algorithms with a network data that is packets capable of manipulating libpcap files in parallel. Moreover, for the agile operation of large data, they added the data visualization interface tool with Hive. We believe our approach to the distributed traffic measurement and analysis with Hadoop can provide the scale-out feature for handling the skyrocketing traffic data. In future work, we plan to support real-time traffic monitoring in high-speed networks.

REFERENCES

[1] Cisco White Paper, Cicso Visual Networking Index:Forecast and Methodology, *2011-2016, May 2012.*

[2] M. Fullmer and S. Romig, The OSU Flow-toolsPackage and Cisco NetFlow Logs, *USENIX LISA,2000.*

[3] D. Plonka, FlowScan: a Network Traffic FlowReporting and Visualizing Tool, *USENIX Conferenceon System Administration, 2000*.

[4] QoSient, LLC, argus: network audit record generation and utilization system, http://www.qosient.com/argus/.

[5]ArborNetworks, http://www.arbornetworks.com.

[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] M. Roesch, Snort - Lightweight Intrusion Detection for Networks, *USENIX LISA, 1999.*

[10] J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on Large Cluster, *USENIX OSDI,2004.*

[11] S. Ghemawat, H. Gobioff, and S. Leung, The Google file system, *ACM SOSP, 2003.*

[12] Hadoop, http://hadoop.apache.org/.

[13] A. Thusoo, J. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, Hive: a warehousing solution over a map-reduce framework, *VLDB, August 2009.*

[14] Y. Lee, W. Kang, and Y. Lee, A Hadoop-based PacketTrace Processing Tool, International Workshop onTraffic Monitoring and Analysis *(TMA 2011), April2011.*

[15] RIPE Hadoop PCAP, https://labs.ripe.net/Members/wnagele/large-scalepcap-data-analysis-using- pache-hadoop, *Nov.2011.*