ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 5, September 2014

Parallel Processing of Internet Traffic Measurement and Analysis Using Hadoop

Dipti J. Suryawanshi, U. A. Mande

Abstract*— Today the internet users are growing rapidly; hence the result is internet traffic data also increases. And for analyzing these traffic data number of tools are available. Most of these analyzing tools are run on a single node with limited computing and storage resources. The traffic measurement and analysis tool is important for observing the network usage and user behaviors, but at the same time scalability problem arises under the explosive growth of internet traffic and high speed access. To avoid the problem of scalability we use Hadoop. Hadoop is an open source computing platform for Map Reduce and distributed file system has popular infrastructure for large amount of data analytics because it facilitated scalable data processing and storage services on a distributed computing system with commodity hardware. In this paper we develop traffic monitoring system based on Hadoop that performs IP, TCP, HTTP, and Net Flow analysis of multi-terabytes of internet traffic in scalable manner.

***Index Terms*—Hadoop, Hive, Map Reduce, Net Flow, pcap, packet, traffic analysis, traffic measurement etc.**

I. INTRODUCTION

Nowadays internet traffic measurements and analysis are mostly used to characterize and analysis of network usage and user behaviors, but there are problem of scalability under the tremendous growth of Internet traffic and high-speed access. Scalable Internet traffic measurement and analysis is difficult because a large data set requires matching computing and storage resources. It has become a popular infrastructure for large amount of data analytics because it facilitates scalable data processing and storage services on a distributed computing system consisting of commodity hardware. It includes a Hadoop-based traffic monitoring system. This performs IP, TCP, HTTP, and Net Flow analysis of multi-terabytes of Internet traffic in a scalable manner [1]. Scalable Internet traffic measurement and analysis is difficult because a large data set requires matching computing and storage resources. It is becoming increasingly common to have data sets that are too large to be handled by traditional databases, or by any technique that runs on a single computer or even a small cluster of computers. In the age of Big-Data, Hadoop has evolved as the library of choice for handling it.

Generally it not easy task to do the network management or business intelligence analytics on huge data such as traffic classification of packet and Net Flow files; investigation of anomalies such as global Internet worm or DDos attacks; long term network trends or user behaviors. For instance, the volume of traffic data captured at a 10 Gbps directional link of 50% utilization becomes 2.3 TB per hour. However, there is no any analysis tool that can afford this much amount of data at once. It is then expected that the major advantage of using Hadoop to measure Internet traffic is the scale-out feature, which improves the analysis performance and storage capacity in proportion to the computing and storage resources with commodity hardware. Hadoop for its scalability in storage and computing power is a suitable platform for Internet traffic measurement and analysis but brings about several research issues. In this paper, we develop a Hadoop based scalable internet traffic measurement and analysis system that can manage the packets and Net Flow data on HDFS. By applying Hadoop to an Internet traffic measurement and analysis, we need to face some challenges that are: 1) to parallelize Map Reduce I/O of packet dumps and Net Flow 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 Map Reduce analysis algorithm for Net Flow, IP, TCP, HTTP, traffic. After that we prove that how to analyze efficiently the TCP performance metrics in Map Reduce 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 Net Flow data through scalable Map Reduce based analysis algorithms for large IP, TCP, and HTTP data. [7] It also show that the data warehousing tool Hive is useful for providing an agile and elastic traffic analysis framework.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 5, September 2014

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 Map Reduce 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 lot of tools are available and widely used for Internet traffic monitoring. Tcpdump [8] is the tool used for capturing and analyzing packet traces with libpcap. Wireshark [9] is a popular tool for traffic analyzer that offers user-friendly graphic interfaces and statistics functions. CoralReef [10], was developed by CAIDA, provides flexible traffic capture, analysis, and report functions. Snort [11] is one type of tool that can process open source signature-based intrusion detection tool and designed to support real-time analysis. Bro [12], is the network security monitoring system, has been extended to support the cluster environment [13]. And it provides only independent packet processing at each node for live packet streaming, so it cannot analyze a huge file in the cluster file system. Tstat [14] is one type of passive analysis tool which expand tcptrace, and it provide various analysis capabilities with regard to TCP performance metrics, application classification, and VoIP characteristics. At the same time, we have Cisco NetFlow [15] which is a well-known flow monitoring format for observing traffic through routers or switches. Many open-source or commercial flow analyzing tools exist, including flow-tools [16], flows can [17], argus [18], and Peak flow [19]. Yet, in general, most of the Internet traffic measurement and analysis tools run on a single server and they are not capable of coping with a large amount of traffic captured at high-speed links of routers in a scalable manner.

Mostly Map Reduce applications on Hadoop are developed to analyze large text, web, or log files. In our work [20], we developed the first packet processing method for Hadoop that analyzes packet trace files in a parallel manner by reading packets across multiple HDFS blocks. Recently, RIPE [21] has also announced a similar packet library for Hadoop, but it cannot provide parallel processing capability of reading packet records from HDFS blocks of a file so that its performance is not scalable and its recovery capability against task failures is not efficient. In this paper, we present a comprehensive Internet traffic analysis system with Hadoop that can quickly process IP packets as well as Net Flow data through scalable Map Reduce-based analysis algorithms for large IP, TCP, and HTTP data. We also show that the data warehousing tool Hive is useful for providing an agile and elastic traffic analysis framework.

III. TRAFFIC MEASUREMENT AND ANALYSIS SYSTEM COMPONENT

In this section, we elaborate the components of the traffic measurement and analysis Map reduces algorithms. From following fig, this system consists of a traffic collector; new packet input formats; Map-reduce analysis algorithms for Net Flow, IP, TCP, and HTTP traffic; and web-based interface with Hive.

A. Traffic collector

The traffic collector receives either IP packet or Net-Flow data from probes or trace files on the disk, and writes them to the HDFS. Traffic collection is carried out by a load balancer and HDFS Data Nodes. In online traffic monitoring, the load balancer uses a high-speed packet capture driver, such as PF RING and TNAPI [22], and it forwards packets to multiple Data Nodes evenly with a flow-level hashing function. After that, HDFS Data Nodes capture forwarded packets and write them to HDFS files concurrently.

Since disk I/O performance may be a bottleneck to a Hadoop cluster, each Data Node uses a parallel disk I/O function, such as RAID0 (data striping mode), to boost the overall performance of HDFS. Because of the distributed traffic collecting architecture, a traffic collector achieves the scale-out feature for storing the increased input traffic volume. However, our focus on the offline traffic collection and analysis system because our system currently does not guarantee the end-to-end performance from online traffic collection to real-time traffic analysis, which requires a job scheduling discipline capable of provisioning cluster resources for the given traffic analysis load.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 5, September 2014

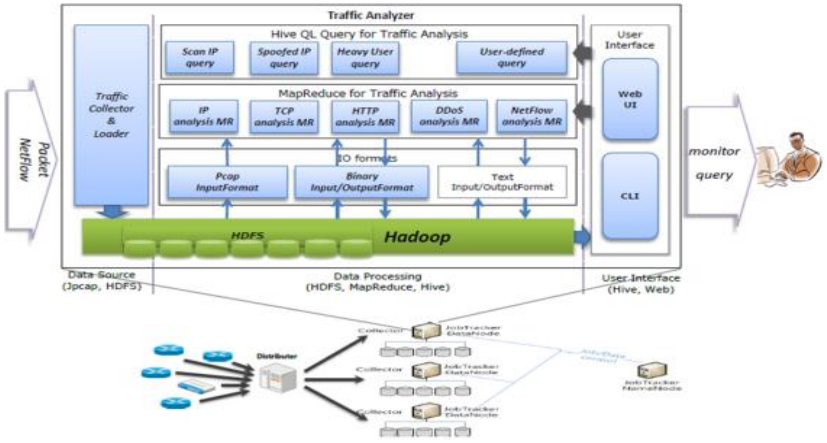

Fig. 1: Architecture of traffic measurement and analysis system with Hadoop

B. IP packet and Net Flow reader in Hadoop

Text files are common input format for the Hadoop because data mining of text file is popular. IP packets and Net Flow data are generally stored in the binary format of libpcap. Hadoop supports a built-in sequence file format for binary input/output. If we want to upload the packet trace files captured by existing probes to HDFS, for that we have convert them into HDFS-specific sequence files. This procedure will result in the computation overhead of reading every packet record sequentially from a packet trace file and saving each one in the format of the sequence file. Though the sequence file format can be used for online packet collection, it will incur the additional space requirements for the header, sync, record length, and key length fields. In this work, we developed new Hadoop APIs that can perform read or write operation on IP packets and Net Flow v5 data in the native libpcap format. The new Hadoop API makes it possible to directly save the libpcap streams or files to HDFS and run Map Reduce analysis jobs on the given libpcap files.

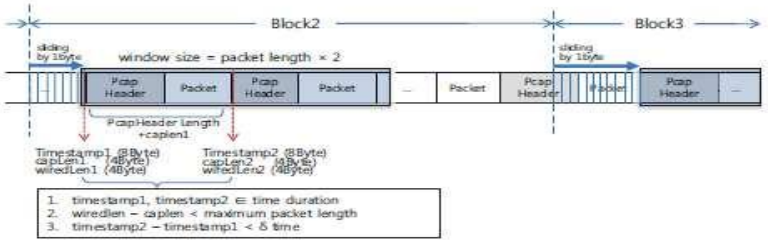

Fig.2: Reading packet records in libpcap on HDFS blocks for parallel processing.

When a Map Reduce job runs on libpcap files in HDFS, each map task reads its assigned HDFS block to parse packet records independently, which is imperative for parallel processing.

Since an HDFS block is chunked in a fixed size (e.g., 64MB), a variable-sized packet record is usually located across two consecutive HDFS blocks. In addition, in contrast to the text file including a carriage-return character at the end of each line, there is no distinct mark between two packet records in a libpcap file. Therefore, it is difficult for a map task to parse packet records from its HDFS block because of the variable packet size and no explicit packet separator. For this case, a single map task can process a whole libpcap file consisting of multiple HDFS blocks in a sequential way, but this file-based processing method, used in RIPE pcap [21], is not appropriate for parallel computing. With RIPE pcap, each trace should be saved in a sufficiently fine-grained size to fully utilize the map task slots of all cluster nodes. Otherwise, the overall performance will be degraded due to the large file size. Moreover, if a map or reduce task fails, the file-based Map Reduce job will roll back to the beginning of a file. In order for multiple map tasks to read packet records of HDFS blocks in parallel, we propose a heuristic algorithm [20] that can process packet records per block by using the timestamp-based bit pattern of a packet header in libpcap. Figure 2 shows how each map task delineates the boundary of a packet record in HDFS.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 5, September 2014

C. Network-layer analysis in Map Reduce

For network-layer analysis, we have developed IP flow statistics tools in Map Reduce, as shown in Fig. 3. Computing IP flow statistics certainly fits the Map Reduce framework because it is a simple counting job for the given key and it can be independently performed per HDFS block. With this tool, we can retrieve IP packet and flow records and determine IP flow statistics that are similarly provided by well-known tools [10, 16]. For the packet trace file, we can tally the IP packet statistics, such as byte/packet/IP flow counting, IP flow statistics and Top N, as shown in the fig. 3.

	Traffic Analysis Job	Hadoop Tool Command	Description
IP Analysis	Total traffic and host/port count statistics	*PcapTotaStats -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	*PcapCount.jo -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.

Fig 3: Overview of IP and Net Flow analysis tools in Map Reduce

D. Transport-layer analysis in Map Reduce

It is totally opposite to IP analysis, TCP performance metric 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 associated across TCP flows parts on several HDFS blocks. If we are trying to do TCP analysis with Map Reduce, we face 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 Map Reduce.

E. Application-layer analysis in Mapreduce

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

a. Web traffic analysis

For web traffic analysis, it is necessary to develop a Map Reduce algorithm that can investigate website popularity and user behavior by examining HTTP packets. Then focus on the header information of the first N (e.g., 3) HTTP packets because the main focus only in the header fields, not the user content.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 5, September 2014

b. DDos traffic analysis

In order to show the effectiveness of Map Reduce based application-layer traffic analysis, it need to present a HTTP-based distributed denial-of-service (DDoS) attack detection method implementation in Map Reduce.

F. Interactive query interface with Hive

Map Reduce framework is useful for processing a huge amount of IP packets and Net Flow data in parallel, researchers capable of developing the measurement and analysis function can invent their own MapReduce modules for quick response time. However, it may take long time for analysts to write application-specific analysis programs in Map Reduce, though analysis modules implemented in MapReduce can bring a better throughput. Therefore, a simple query interface is more convenient than MapReduce programming to users interested in agile traffic analysis. Moreover, it is more expressive and extensive for users to ask versatile questions on the traffic data through the query interface. Hive provides the ability to generate MapReduce codes through the SQL-like query interface, Hive Query Language (HiveQL). Therefore, we harness the Hive query interface for easy operational Internet traffic analysis.

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. In future work, it can be integrating the network packet data with other possible data sets like web logs; can lead the analysis in a broader scope.

ACKNOWLEDGMENT

I would like to take this opportunity to thank all those who were directly or indirectly involved in making my research a near perfect one. I would like to extend my deep gratitude towards my project in charge Prof. U. A. Mande for supporting me and giving guidance with understanding me every step. I would also like to thank our Head of Department Prof. P. R. Futane for giving me that opportunity to work on this interesting topic.

REFERENCES

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

[2] K. Cho, K. Fukuda, H. Esaki, and A. Kato, Observing slow crustal movement in residential user traffic, ACM CoNEXT2008.

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

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

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

[6] T. White, Hadoop: the Definitive Guide, O'Reilly, 3rded. 2012.

[7] 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.

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

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

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

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

[12] Bro, http://www.bro-ids.org.

[13] M. Vallentin, R. Sommer, J. Lee, C. Leres, V. Paxson, and B. Tierney, The NIDS Cluster: Scalable, Stateful Network Intrusion Detection on Commodity Hardware, International Conference on Recent Advances in Intrusion Detection (RAID), 2007.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 5, September 2014

[14] A. Finamore, M. Mellia, M. Meo, M. M. Munafo, and D. Rossi, Live traffic monitoring with tstat: Capabilities and experiences, 8th International Conference on Wired/Wireless Internet Communication, 2010.

[15] Cisco Net Flow, http://www.cisco.com/web/go/netflow.

[16] M. Fullmer and S. Romig, The OSU Flow-tools Package and Cisco Net Flow Logs, USENIX LISA, 2000.

[17] D. Plonka, Flow Scan: Network Traffic Flow Reporting and Visualizing Tool, USENIX Conference on System Administration, 2000.

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

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

[20] Y. Lee, W. Kang, and Y. Lee, A Hadoop-based Packet Trace Processing Tool, International Workshop on Traffic Monitoring and Analysis (TMA 2011), April 2011.

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

[22] F. Fusco and L. Deri, High Speed Network Traffic Analysis with Commodity Multi-core Systems, ACM IMC 2010, Nov. 2010.

[23] Y. Lee and Y. Lee, Detecting DDoS Attacks with Hadoop, ACM CoNEXT Student Workshop, Dec.2011.

[24] CNU Project on traffic analysis in Hadoop, https://sites.google.com/a/networks.cnu.ac.kr/dnlab/research/hadoop.