2018-04

Automated Data Analysis for Network Optimization / Threat Detection in Network Architectures

Kragh, Frank; Miller, Donna L.; Brida, Ben

Monterey, California: Naval Postgraduate School

MONTEREY, CALIFORNIA

AUTOMATED DATA ANALYSIS FOR NETWORK OPTIMIZATION / THREAT DETECTION IN NETWORK ARCHITECTURES

Report Type: Final Report
Period of Performance: 10/01/2017-09/28/2018

Project PI: Dr. Frank Kragh, GSEAS
Additional Author/Authors: Ms. Donna L. Miller, GSEAS
Student Participation: Captain Ben Brida, USMC, Electrical Engineering

Prepared for:
Topic Sponsor: IMEF
Research Sponsor Organization (if different): Marine Corps Network Efficiency Lab (MCNEL)
Research POC Name: Captain Brad Palm, USMC)
Research POC Contact Information: bradley.palm@usmc.mil, 760-725-9319

EXECUTIVE SUMMARY

Project Summary
The Marine Corps Network Efficiency Lab (MCNEL) is tasked with analyzing very large network traffic archives collected from operations in order to improve future network design, operations, and security. Until this time, MCNEL has used conventional single node packet analyzers, which have proven to be very limiting. Conventional single node packet analyzers are unable to monitor network traffic at scale. In this research, elements of the Apache Hadoop ecosystem, including HBase, Spark, and MapReduce were employed to conduct network traffic analysis on a large collection of network traffic thereby establishing a prototype for network analysis at very large scale in computer clusters. The MCNEL clusters could be organic or in the cloud, perhaps using govCloud cloud computing assets. Initially, limited analysis was conducted directly on packet capture next generation (pcapng) files on the Hadoop Distributed File System (HDFS) using MapReduce. To allow for repeated analysis on the same dataset without reading all source files in their entirety for every calculation, network traffic archives were parsed, and relevant meta-data was bulk loaded into HBase, a Not Only Structured Query Language (NoSQL) database employing the HDFS for parallelization on computer clusters. This NoSQL database was then accessed via Apache Spark where pertinent data is loaded into dataframes, and additional analysis on the network traffic takes place. This research demonstrates the viability of custom, modular, automated analytics, employing open-source software to enable parallelization, to conduct traffic analysis at scale.

Keywords: big data, Hadoop, Spark, MapReduce, HBase, packet capture, pcapng, network analysis

Background
In the analysis of large volumes of data, parallelization is a key tenet that solves several problems. The first problem addressed is read/write access limitations. While disk storage capacity has increased one hundred-fold in the past two decades, read speeds on most drives have only increased by a factor of twenty-five [White2015]. Rather than storing all data on a single drive, we can instead partition and store data on hundreds of drives with shared access. By allowing for simultaneous reading of data on multiple drives, parallelization allows for tasks to be completed in a fraction of the time. Second, parallelization not only increases our data reading speed, it also increases our data reliability through redundancy. For example, by storing partitioned data in triplicate on physically separated hard drives, the likelihood of data loss is reduced dramatically. Finally, we can greatly increase our processing power through parallelization. By storing data on two, two hundred, or two thousand servers and allowing each server to individually do the processing of the data it stores, we not only reduce the required network traffic to move and manage data, but we also vastly increase our effective processing power [White2015].

For MCNEL's purposes, it was assessed that using open source software on commodity hardware is most likely to be maximally scalable and financially supportable. Because of those considerations, the Hadoop ecosystem was selected as the large-data framework to be used. For increased functionality and ease of implementation, for this thesis specifically, MapR was employed. MapR is a third-party Hadoop

distribution; however, any properly configured Hadoop cluster would be equally viable. Finally, packet capture next generation (pcapng) was selected as the expected file type for analysis, since network collections conducted by MCNEL use this format.

Findings and Conclusions

The purpose of this research was to provide the Marine Corps Network Efficiency Lab with a prototype capability to conduct automated, large scale packet analysis. This was done employing open-source software designed for use on commodity hardware, specifically, the Apache Hadoop ecosystem. This research took an incremental approach, first seeking to conduct network analytics on Packet Capture Next Generation (pcapng) files using MapReduce. After a successful evaluation on the strengths and limitations of MapReduce for analysis, a more structured storage mechanism than pcapng was sought.

HBase, an open source NoSQL database employing the HDFS, was selected as a preferred storage mechanism, where columns could be dynamically assigned based on packet protocol, and packet data could be separated from metadata. This separation allows for access to data without sequentially reading every file in its entirety. HBase data was then mapped to a Spark DataFrame to conduct multiple types of analysis and more sophisticated analysis than is practical in just MapReduce.

The types of analysis conducted were traditional network metrics, most of which are already performed by the Marine Corps Network Efficiency Lab. The emphasis was on evaluating these metrics in a big data framework, allowing for scaling to many nodes simultaneously and completing the work on terabyte and petabyte size data sets. While only a single node virtual machine was used for this research, input data sizes larger than allocated memory were used to validate the concept of scaling, and the code produced is viable for use on an arbitrarily large Hadoop cluster.

The metrics calculated in MapReduce were network usage per Internet Protocol (IP) address, port counts, protocol counts, and network usage by hour of day and day of week per IP address. The network usage by IP address and port count metrics were repeated in Spark. Additionally, IP protocol count and Transmission Control Protocol (TCP) initial round trip time metrics were also calculated in Spark. Even operating in a single node virtual machine, the code executed in this research was able to often outperform a free-ware packet analyzer in terms of speed. In some cases, substantial performance increases were obtained.

Ultimately this research demonstrated the viability of the Apache Hadoop ecosystem for automated bulk data analytics on pcapng network traffic archives, with all prototype software written for execution on a large computer cluster, thereby enabling automated data analysis of very large data sets. In doing so, the whole file input format class and record reader class from [White2015] were used with a custom class developed to directly ingest pcapng files into Hadoop without requiring an intermediary file format. This capability was coupled with custom analytics, developed to determine network metrics in both MapReduce and Spark, using HBase as a long-term storage format.

Recommendations for Further Research

There are four goals which may be pursued as future work on this project.

1. Increasing Functionality and Supported Protocols

The first goal is to increase the capability of the pcapng class used for parsing the files. At a minimum, IPv6 supportability should be added. As IPv6 continues to expand this becomes increasingly relevant. Once this support is added to the pcapng parsing methods, the analytic code will have to be revised to check for protocols and extract metadata as needed, similar to what was done in this thesis by confirming IPv4 and TCP in the HBase bulk loading.

2. Verify capacity with Large Cluster and Large Collect

The second goal is to verify the capability explored in this thesis with a large dataset, either on a local Hadoop cluster or employing a cloud service such as Amazon Web Services (AWS). An additional requirement that accompanies this future work is to obtain a large scale persistent collect from a network. Ideally several weeks or month of data should be gathered, and terabytes of network traffic should be analyzed.

3. Employ Machine Learning to Establish Baselines and Automatically Identify Anomalous Network Behavior

The framework and software provided in this research provide a platform for advanced analysis of huge network traffic data sets including automatically identifying normal and abnormal network traffic. This should be exploited by employing standard machine learning algorithms to analyze the data set for anomaly detection. For example, the provided framework allows easy determination of the average network usage of every IP address, organized by time of day and day of week. This, and similar data, can be used to determine if that IP address is acting abnormally. Similar questions can be asked for latency, given that we can easily determine the initial round trip time (iRTT) distribution from each IP to IP connection. Support vector machines, Gaussian mixture models, or clustering algorithms are standard machine learning algorithms that could be applied to classify abnormal latency. This research effectively established the data framework for the data preparation necessary to answer these questions. A large traffic collection is needed for this work. Months of data would be needed for evaluation and ingestion to have reliable output. Accordingly, once the second goal of verifying capacity with large collect is satisfied, machine learning for network anomaly detection should be explored.

4. Increase Ease of Analyst Use and Improve Metric Output Formatting

The final goal for future work would be to create a user interface for an analyst to easily run the underlying algorithms. By having an interface where queries can be easily submitted, specifically controlling what data is ingested into Spark from HBase, what metrics are calculated, and how they are presented, the utility of this thesis capability development would be drastically increased. Additionally, currently metrics are produced exclusively in table format. By exploring a dashboard, graphical outputs could be easily included and automatically generated, increasing the utility and interpretability of the analytics being conducted.

References & Acronyms

[White2015] T. White. *Hadoop The Definitive Guide*. Beijing: O'Reilly Media, 2015.

HDFS	Hadoop Distributed File System
IP	Internet Protocol
IPv4	Internet Protocol, version 4
IPv6	Internet Protocol, version 6
iRTT	initial round trip time
MCNEL	Marine Corps Network Efficiency Lab
NoSQL	Not Only Structure Query Language
PCAPNG	packet capture next generation
SQL	Structured Query Language
TCP	Transmission Control Protocol