Campus Network Monitoring and management: Real-Time Big Data Analytics

Shuai Zhao†, Mayanka Chandra Shekar†, Yugyung Lee†
†University of Missouri-Kansas City, MO, USA. {Shuai.Zhao, mckw9, LeeYu}@mail.umkc.edu

***Abstract*—A campus network can provides a number of services to its users and also generate huge amount of data. Due to the mixture of different type of data, it is often difficult to analyze the data in real-time. The available network monitor and management tools can not provide a comprehensive situation for its network. To fill this void, we propose the a real-time big data analysis framework using apache Hadoop, Storm and Kafka technology. Our approach contains history data batch processing, real-time streaming data analysis and machine learning for automatic traffic issue detection. Data visualization for expressing the different dimension of the network traffic. We explain different components of our proposed framework and show results from a brief study.**

***Index Terms*—Network monitoring and management, Apache Hadoop, Storm, Kafka, and real-time big data analysis**

I. Introduction

A campus network often provides numbers of services to its users. Due to its com- plexity of the network design, it is difficult to monitor and management. Network administrators running a set of networking tools are facing several challenges along with big data traffic. We would like to use the network traffic data generated from UMKC data center as an example to propose a network management and monitor system. The goal is twofold: a) visualize network traffic and let network administrator quickly understand what is going on in its network b) make it easier to pinpoint when issue happens .

II. Proposed Architecture

Figure 1 shows the workflow of proposed system architecture. Both of history and real-time data analytics are included in our domain. Network traffic data generated from different sources are either stored in RDMS or in HDFS based on its schema description. For a history data query, our visualization web server will start a job in our hadoop cluster and result will be sent back to our user-friendly interface. Real-time traffic data can also be processed use Apache storm system for real-time network visualization.

III. Motivation

As discussed in [2], NetFlow [1] data has been used as the main data source. If consider by combing different data source at different data repositories as a whole, such as campus VLAN configuration, it is very difficult to analyze network conditions when processing either history network data and real-time network data. Thus it is difficult to answer questions such as following :

1) What are the top high traffic applications?
   a) Campus internal network traffic or external traffic
   b) What specific application, such as Youtube or Netflix
2) What does the current network condition look like?
   a) What is the network traffic trend?
   b) If there is any abnormal activity going on.

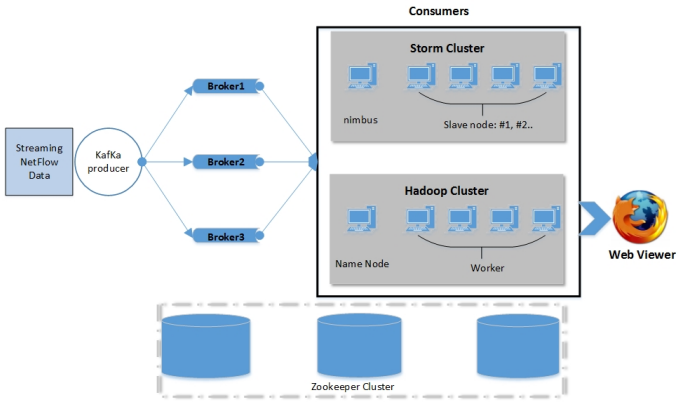

Fig. 1. Proposed Architecture

A. Design

Data model for hadoop batch processing:

TABLE I
HADOOP BATCH PROCESSING DATA MODEL{KEY,VALUE}

Key	Value
(sourceIP, destinationIP)	Rate $(bits/s)$

Data model for real-time data processing:

TABLE II
REAL-TIME PROCESSING DATA MODEL {KEY,VALUE}

Key	Value
(timestamp, sourceIP, destinationIP)	Rate $(bits/s)$

Proposed MapReduce algothrim for history batch process using Hadoop :

Predictive recommendation model and algorithm:

IV. Features Implemented

network data analysis algorithms
Prective algorithm
WebServer User Interface
Mobile User Interface

Algorithm 1: Mapper Algorithm

Input: NetFlow Dataset N_f, Source and destination IP address (key, value) pair $IP_{src,dst}$, Bandwidth for each flow R_{flow}

Output: $(IP_{src,dst}, R_{flow})$ which contains IP address and Bandwidth usage

```
for f ∈ N_f, ip ∈ IP_{src,dst} do
    if ip in IP_{src,dst} then
        print (IP_{src,dst}, R_{flow})
```

Algorithm 2: Reducer Algorithm

Input: Mapper Ootput: $(IP_{src,dst}, R_{flow})$

Output: Flows which have the most bandwidth usage, $IP_{src,dst}, r_{flow}$

```
initialization current_ip = null, current_rate = 0,
previous_rate = 0 ;
for ip ∈ IP_{src,dst} , r_flow ∈ R_flow do
    new_rate = r_flow ;
    if ip == current_ip then
        current_rate = r_flow + previous_rate ;
    else
        current_ip = ip ;
        previous_rate = current_rate
```

V. Result

labelsec:result

History data hadoop Mapreduce

Kafka and storm real-time

Urls

1) Github Url: *https://github.com/CS590RA/Challenge1*
2) Hadoop Cluster Manager: *http://n1.example.com:7180*
3) Django Webserver: *http://n1.exmaple.com*

VI. Conclusion

VII. Future Work and Limitations

References

[1] Cisco NetFlow. http://en.wikipedia.org/wiki/netflow.

[2] Shuai Zhao, Kelsey Leftwich, Matthew Owens, Frank Magrone, James Schonemann, Brian Anderson, and Deep Medhi. I-can-mama: Integrated campus network monitoring and management. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–7. IEEE, 2014.