VELOCITY: A NETFLOW BASED OPTIMIZED GEO-IP LOOKUP TOOL

A Thesis
in
Electrical Engineering

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by
SWATESH VILAS PAKHARE

B.E., University of Mumbai, Maharashtra, India, 2013

Kansas City, Missouri
2016

VELOCITY: A NETFLOW BASED OPTIMIZED GEO-IP LOOKUP TOOL

Swatesh Vilas Pakhare, Candidate for the Master of Science Degree

University of Missouri–Kansas City, 2016

ABSTRACT

It is a challenging task for network administrators to monitor their institution's network against undesirable behavior. While NetFlow is useful to gather flow-level data for any Internet connection, its feature is limited to traditional flow-level information such as source IP address, destination IP address, source port number, destination port number, and the protocol type. Thus, if we are to understand geographic dynamics of any flow connected to hosts at an institution from the outside world, it is not currently possible with NetFlow. To address for geo-location information of such flows, we developed the tool, VELOCITY. This tool allows to correlate IP addresses with geo-location information to visualize geo-location of incoming and outgoing flows. The VELOCITY tool consists of four different methods, with increasing order of efficiency of the methods. We found that Method 3 outperforms Methods 1 and 2 in case of filling database with geographical data for the first time. Method 4, which is an extension of Method 3, finds geographical information for IP addresses that are not present in the currently populated database, thereby

providing a more optimized approach than Method 3 for incremental flow data.

Furthermore, for visualization and near real time experience, we also developed a web application that displays geographical information of IP address of flows on Google maps.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled " Velocity: A NetFlow Based Optimized Geo-IP Lookup Tool ," presented by Swatesh Vilas Pakhare, candidate for the Master of Science degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Deepankar Medhi, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Kenneth Mitchell, Ph.D.
Department of Computer Science & Electrical Engineering

Cory Beard, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

ILLUSTRATIONS

TABLES

ACKNOWLEDGEMENTS

I would like to thank my academic advisor Deepankar Medhi, Ph.D and Zhao Shuai

I would like to thank my family, Narsi Reddy and Sarath Bandaru.

CHAPTER 1

INTRODUCTION

1.1 Overview

Computer networks are complex communication systems that might experience an unusual behavior at any instant of time, thereby making it difficult for network operators to monitor their organization's network. Although SNMP facilitates capacity planning, it does little to characterize traffic applications and patterns. To overcome these limitation, nowadays, network operators are using other means such as NetFlow data for network monitoring purposes. NetFlow, a feature introduced on Cisco routers, provides the ability to collect packet level information as it enters or exits an interface. By analyzing the data provided by NetFlow, network operators can determine source and destination of traffic, class of service, and the causes of congestion.

NetFlow data also allows network operators to understand flow-level behavior of network including application and network usage, network productivity and utilization of network resources, the impact of changes to network, network anomaly and security vulnerabilities, and long term compliance. Thus, NetFlow gives network operators the luxury of knowing who, what, when and where, and how network traffic is flowing. Understanding the network behavior gives an insight about network utilization, reduces vulnerability of network as related to failure and allows efficient operations of the network. Thus, improvements in network operation lowers cost and drives higher business revenues by

better utilization of the network infrastructure.

1.1.1 NetFlow Based Network Awareness

The ability to characterize IP traffic and understand how and where it flows is critical for network availability, performance and troubleshooting. Monitoring IP traffic flows facilitates more accurate capacity planning and ensures that resources are used appropriately in support of organizational goals. It helps network operators to determine where to apply Quality of Service (QoS), optimize resource usage and it plays a vital role in network security to detect Denial-of-Service (DoS) attacks, network-propagated worms, and other undesirable network events.

NetFlow provides solutions to following problems:

- Analyze new applications and their network impact.
- Reduction in peak WAN traffic.
- Troubleshooting and understanding network pain points.
- Detection of unauthorized WAN traffic.
- Security and anomaly detection.

1.1.2 NetFlow Cache

NetFlow provides network information by examining each packet that is forwarded within a router for a set of IP packet attributes. The attributes used by NetFlow

are as follows:

- Source IP address.
- Destination IP address.
- Source port.
- Destination Port.
- Layer 3 protocol type.
- Class of Service.
- Router or switch interface.

After analyzing each packet, packets with the same source/destination IP address, source/destination ports, protocol interface and class of service are grouped into a flow and then packets and bytes are tallied. This methodology of determining a flow is scalable because a large amount of network information is condensed into a database of NetFlow information called the NetFlow cache. The formation of NetFlow cache is shown in Fig. 1. The

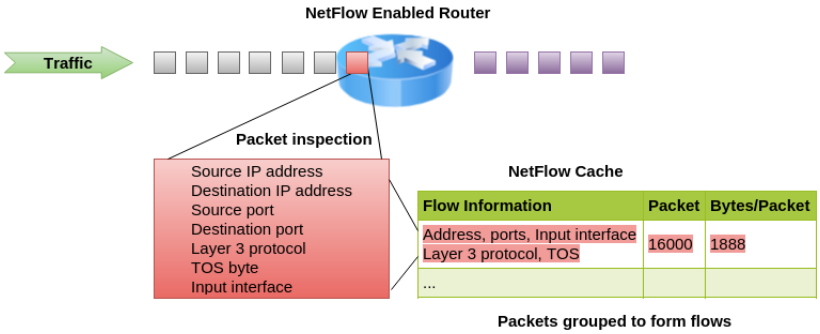

Figure 1: NetFlow cache

flow information allows network operator to understand the following:

- Source address allows the understanding of who is originating the traffic
- Destination address tells who is receiving the traffic
- Ports characterize the application utilizing the traffic
- Class of service examines the priority of the traffic
- The device interface tells how traffic is being utilized by the network device
- Tallied packets and bytes show the amount of traffic

Some additional information added to flow includes:

- Flow timestamps to understand the life of a flow; timestamps are useful for calculating packets and bytes per second.
- Next hop IP addresses including BGP routing Autonomous Systems (AS).
- Subnet mask for the source and destination addresses to calculate prefixes.
- TCP flags to examine TCP handshakes.

The data export format for version 9 of Cisco NetFlow services, for use by implementations on the network elements and/or matching collector programs can be found at [5].

1.1.3 Need for Geolocation

Although network monitoring using NetFlow data enables network operators to troubleshoot network related problems, it does not provide network operators with crucial geographical information about host IP. In addition to this, many other applications share the need for geographical information to provide better services. These services include web pages with regional preferences (e.g. an online user may receive selective and directed advertising according to her location or be subject to automatic language selection when displaying content); control data availability according to user's geographic location (e.g. the system can deny access to some specific set of data based on regional policies as in the case of many Internet TV and radio broadcast services); study and analyze geolocated traffic to understand who talks to whom at the levels of users, regions and countries and help identifying routing anomalies.

Our goal in this work is to find geographical information for all the IP addresses in an optimal amount of time from NetFlow data. The geographical information includes fields like the name of the institution holding that IP, its country, region, street address, city, latitude, longitude, ZIP code, time zone, connection speed, Internet Service Provider (ISP) and domain name, IDD country code, area code and weather station code etc.

As mentioned before, information about all the IP addresses is available from NetFlow data. Based on these IPs, we wish to find geographical information for IP addresses to which requests are sent from UMKC's network. For example, a student in UMKC's network is watching a movie on Netflix, then we look for geographical information of where Netflix's data center might be located. We also find geographical information for

IP addresses from which requests are coming to UMKC. For example, a person in USA is planning to join UMKC and he/she is browsing different websites of UMKC. In this case, we search for geographical data of that person's source IP.

1.1.4 Motivation and Contribution

Having knowledge of geographical data, we can find various organizations providing services to students in UMKC; we can also find different ISPs used by these organization for content distribution. In addition to this, if majority of people from France are browsing through UMKC's websites, then we can provide better web service by translating the web content in french language. Furthermore, if people from particular country are sending unwanted UDP traffic towards UMKC, then we can configure firewalls to block traffic coming from that country. Similarly, users can be restricted access to resources in particular region or country.

In order to achieve our goals, we developed a tool that could find geographical information of millions of IP addresses in an optimal amount of time. We used the concept of scraping websites to obtain geographical data for each IP address, xidel, a tool for scraping HTML pages, has been used for this purpose. Xidel takes URL and HTML tags as its arguement and extracts the content between the HTML tags that are passed to it. Another important tool used in our work is GNU's parallel command. Using parallel command, xidel can send multiple HTTP requests to website that is to be scraped. We then wrote a wrapper tool in BASH to process millions of IP addresses. Also, all the IP addresses from NetFlow file are extracted in parallel saving us some time. The

geographical data we are interested in are as follows:

- Continent, Country, Capital, State and City's name
- Organization and ISP's name
- AS number, Host name and Name-servers
- Latitude and Longitude of Country, Continent and City
- IP prefix and Net range

In addition to this, for visual representation, we developed a web application that could display the geographical data of IP addresses on Google maps in the form of markers. We also incorporated a date picker, time slider and drop down menus in this application for near real time experience. These components acts as filters and would help to curb cluttering of markers on Google maps. In the next section, we describe our approach.

1.2 Approach

The basic functionality diagram of our method is shown in Fig. 2. Before presenting the primitive operation, we highlight an important point. In our method, we have considered semantic approach where geographical information belonging to each IP address is retrieved based on DNS queries or WHOIS lookup. Among, various online IP lookup websites, we have used www.ip-tracker.org since it allow us to make unlimited queries to WHOIS database.

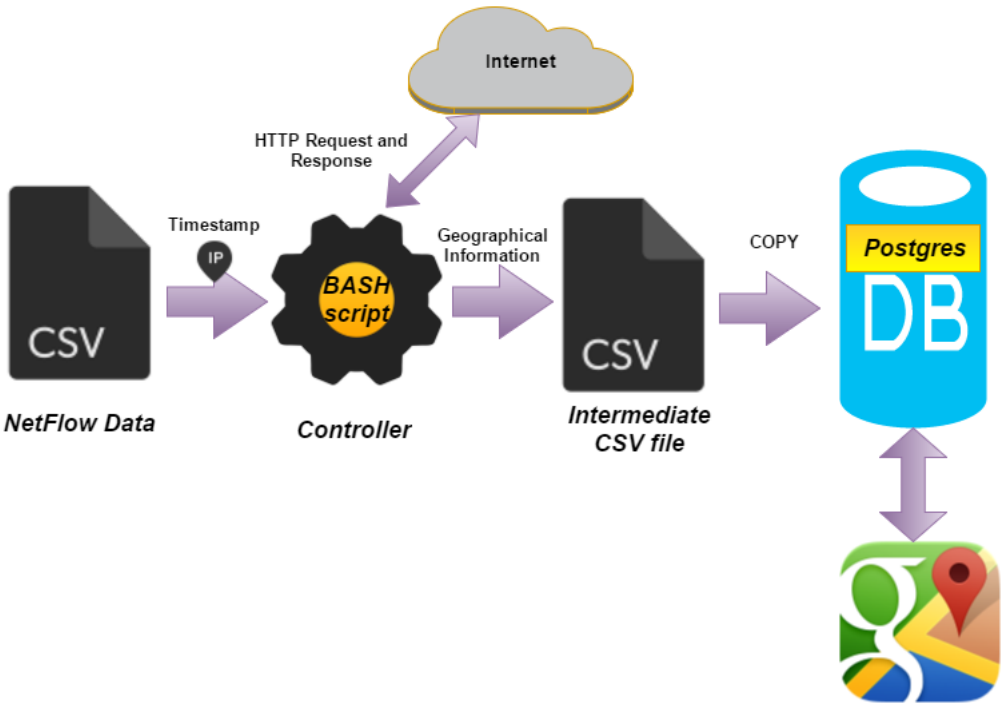

Figure 2: Basic Block Diagram.

We now present the principle working of our method. The NetFlow data provides crucial information like source and destination IP addresses along with timestamp associated with it. Controller, a BASH script, first extracts all the IP addresses from the NetFlow data file. Arrangements are made such that there are no duplicate IP addresses and if present, an IP with latest timestamp is picked. The extracted IP addresses are appended to URL and passed as arguements along with HTML tags to xidel. Xidel on receiving the URL and HTML tags scraps the web content between HTML tags. The usage of xidel can be found at [19]. The controller after receiving geographical data further processes it to produce comma separated geographical data. These comma separated values are first stored in intermediate csv file and then copied into the database using Postgres's COPY

command and we have database filled with geographical information belonging to each IP address present in NetFlow data.

The rest of the thesis is organized as follows. After discussing related work in Chapter 2, we present our four methods in Chapter 3. In Chapter 4, we introduce our web application and then present our results in chapter 5. Finally, we conclude our work in Chapter 6 by stating the limitations of our method.

CHAPTER 2

LITERATURE SURVEY

As our initial input is NetFlow data, we start the literature survey by first briefly discussing usage of NetFlow data to classify internet traffic and to increase security awareness. We aim at finding geolocation of IP addresses and so we briefly discuss about different works on accurately getting the target IP geolocation followed by work that use NetFlow data to obtain geolocation data.

The internet traffic classification based on statistical analysis of NetFlow data has been receiving lot of attention in recent years because of the incompetency of port number based and payload based traffic classification in today's high speed internet.

Several papers has been published demonstrating the use of statistical analysis of NetFlow data and protocol behavior for traffic classification. The work in [10, 16] classifies the internet traffic based on statistical analysis of NetFlow data. The authors in [16] developed a tool to classify network traffic based on simple packet and byte count available in NetFlow data whereas in [10] different settings of NetFlow data were considered for traffic classification including varying the size of NetFlow data and using packet sampling method. In addition to this, the analysis of impact of packet sampling on NetFlow traffic classification is done in [2].The work in [15] focus on selecting the best machine learning classification algorithm for identifying flow-based traffic to their originating applications.

NetFlow based tool, BLINC, developed in [11] introduces a method that classifies traffic to their application types based on connection-level patterns or protocol behavior. This approach is based on observing and identifying patterns of host behavior at the transport layer. VisFlowConnect and NVisionIP developed in [14, 18] respectively relies on NetFlow data to detect intrusions or attack on the network by improving system administrator's situational awareness of current and recent network events. A prototype built in [13] count on NetFlow data for NAT detection.

The research in [9] considers how to scale up existing measurement-based geolocation algorithms like Shortest Ping and Constraint Based geolocation (CBG) to cover the entire Internet by selecting few best vantage points out of many to improve the accuracy of locating target IP. Topology Based Geolocation in [12], inspired by locating nodes in sensor network using radio connectivity, leverages network topology along with measurements of network delay to determine the target IP location. The author in [6] shed light on using CBG to find the coarse-grained area first, then using TBG to find landmark (a known host) followed by again using CBG to find the target location.

In [4], the estimation of target IP is converted into optimization problem by first finding nearest common router and related landmarks to target IP, introducing deviations to landmarks and regarding their locations as areas, calculating relative delay between landmarks and target, and then taking the distance between landmarks and target proportional to the relative delay as constraint conditions to estimate the real deviations of landmarks and the location of the target. The work in [1] demonstrates how the postal address data provided by users to Facebook can be used to locate the target IP. Unlike a

measurement based approach, a tool, Whois Based Geolocation (WBG), developed in [7] takes semantics approach (IP lookup) to find target IP geolocation using WHOIS database and CAIDA database accurately.

In [3], the target IP's geolocation is found using MaxMind GeoLite and added to NetFlow data before sending it to NetFlow collector allowing network administrators to filter, aggregate and generate statistics based on the geolocated data. In our work, we are using online tool such as www.ip-tracker.org because it allows us to make unlimited queries to WHOIS database without any fee. A network monitoring tool, SURFmap, is developed in [8] using Google maps API to provide network traffic information. Our work is related to one in [8] as we both developed a web application using Google maps API to display geographical data on Google maps and both of ours initial input is NetFlow data. But, there is one significant difference, we focus on obtaining the geographical information for huge set of IP addresses in optimal amount of time and then we display it on Google maps in the form of markers.

CHAPTER 3

METHODS

In this chapter, we discuss 4 methods that we developed for our tool. In all the methods, the geographical information belonging to each IP address is obtained by web scraping websites using xidel. The websites that are scrapped to gather all the data are www.ip-tracker.org and www.arin.net. The former is used to get information like Continent, Country, Capital, State, City, Organization, ISP, AS number, Host name and Nameservers and Geolocation of Country, Continent and City and the latter is used to get IP prefix and Net range. Furthermore, we are using PostgreSQL database to store all the downloaded data.

In order to avoid confusion between different IP lookups, we have defined two terms.

1. ARIN lookup: IP lookup performed to get IP prefixes from www.arin.net.

2. Main lookup: IP lookup performed to download geographical data from www.ip-tracker.org.

3.1 Method 1

The flow chart for method 1 is shown in Fig. 3. In this method, we start by

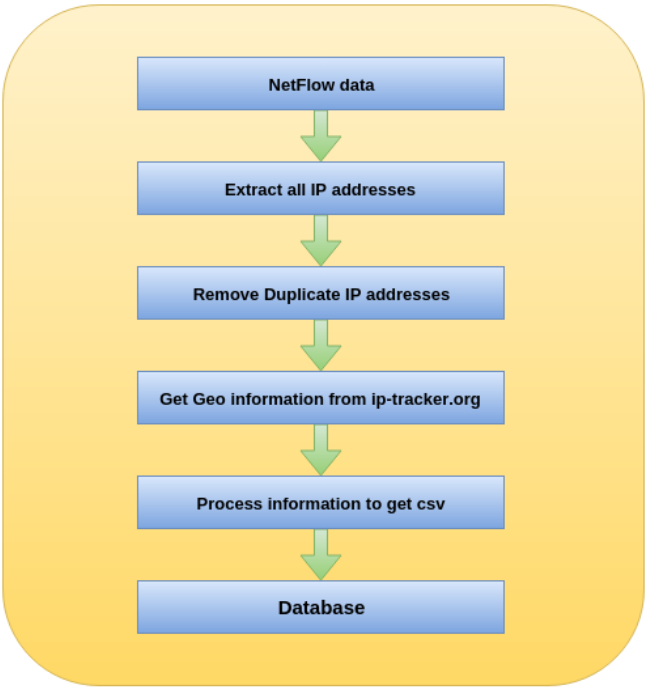

Figure 3: Method 1

extracting IP addresses from NetFlow data. Note that, these IP addresses are external to University of Missouri-Kansas City. Once IP addresses are extracted from NetFlow data, xidel is used to download the required data by web scraping www.ip-tracker.org. The downloaded data is further processed by the controller to obtain comma separated values (csv) and stored in an intermediate csv file. The content of this csv file is then copied into the database by using Postgres's COPY command.

Pros of Method 1:

- Easy to implement.

Cons of Method 1:

- All IP addresses belonging to same subnet are looked up resulting in downloading of redundant information and thereby increasing the processing time.

3.2 Method 2

The flow chart for method 2 is shown in Fig. 4. The framework of method 2 is

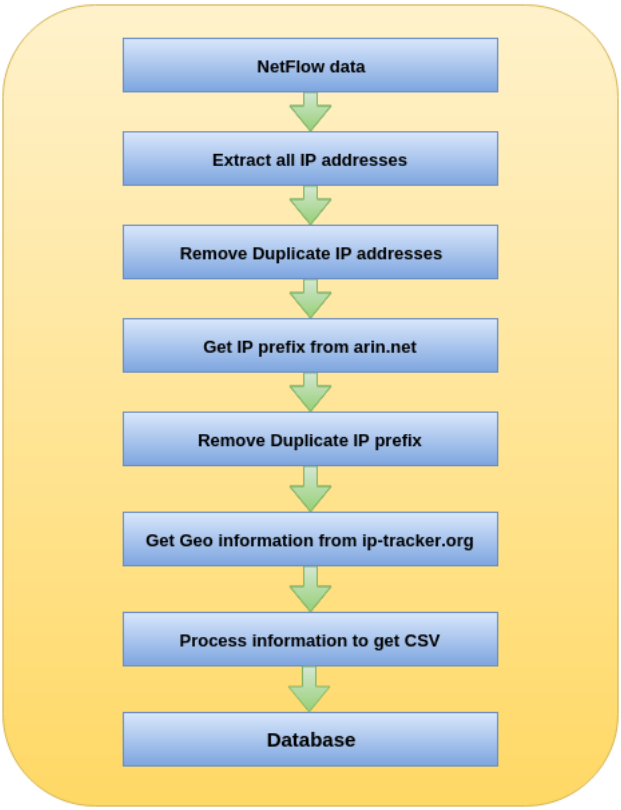

Figure 4: Method 2

similar to method 1 except that main lookup is performed based on IP prefix instead of host's IP address. This resolves the problem of querying for IP addresses belonging to

same subnet multiple times. First, ARIN lookup is performed to get IP prefixes based on IP addresses and then main lookup is performed based on IP prefixes to download all the required geographical information. Using Postgres's COPY command, the geographical information is stored into the database.

Pros of Method 2:

- Easy to implement.
- There is no redundant data as main lookup is performed based on IP prefixes.

Cons of Method 2:

- All HTTP requests to www.arin.net and www.ip-tracker.org are sent sequentially.

3.3 Method 3

The flow chart for method 3 is shown in Fig. 5. Method 3 is different from our previous methods. We start by extracting IP addresses and latest timestamp associated with it. The timestamp field in NetFlow data records time and date when the request was made destined to UMKC network from outside networks and vice-versa. When the controller gets all IP addresses, it performs ARIN lookup first to get all IP prefixes and then main lookup is performed to gather all geographical data. Meanwhile, the timestamps are preserved during both ARIN and main lookup. The timestamps preserved during main lookup are then appended to geographical information downloaded for each IP prefix. In this method, we held an assumption that timestamp associated with each IP address is same for IP prefix as well and in case of repeated IP address/IP prefix, the one with latest

timestamp is picked.

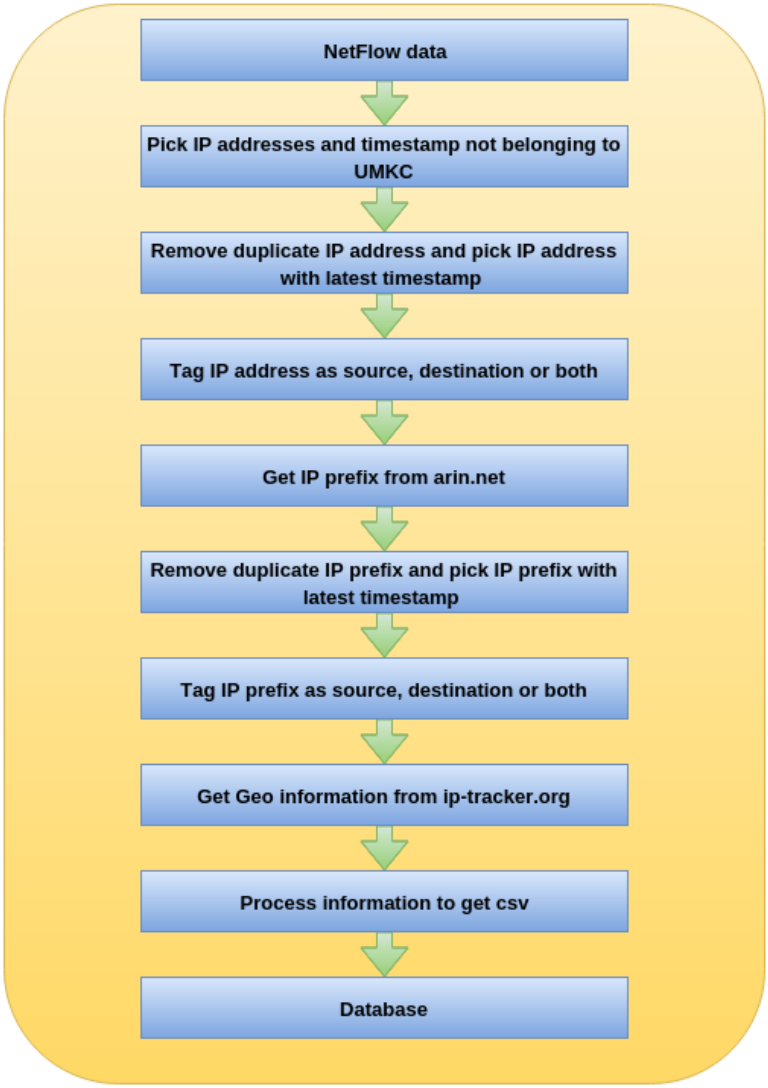

Figure 5: Method 3

We introduced a 'type' parameter in this method. Initially, when the controller receives all IP addresses, it tags each IP address as either source or destination depending upon the field from which it is picked. An IP address may appear as either a source or as a destination for different NetFlow records. In this case, IP address is tagged as 'both'.

Once the tagging is completed, ARIN lookup is performed to get IP prefix. Note that, the previous tags associated with IP are now associated with IP prefix. The controller then finds and replaces the 'type' parameter to 'both' for IP prefixes which were tagged as source and destination. Once this complex process is completed, main lookup is performed to obtain all geographical data. The timestamps are preserved during whole of this process and then finally appended to geographical data.

In our previous methods, the bottleneck was sending http requests to www.ip-tracker.org and www.arin.net in sequence. We overcame this problem by sending multiple request at once using GNU's parallel command. Parallel command works in correspondence to number of threads in computer hardware. The higher the number of threads, higher is the degree of parallelism achieved. Detailed information about parallel command can be found at [17].

For storing data in database, the data is first stored in intermediate csv file and then using Postgres's COPY command the contents of intermediate csv file are moved to database. We will see the use of this csv file in method 4.

Pros of Method 3:

- No redundancy as main lookup is performed based on IP prefixes.
- It is efficient because HTTP requests are sent in parallel.

Cons of Method 3:

- Increased complexity as timestamps and tags belonging to each IP address/IP prefix are preserved through out the ARIN lookup and main lookup.

3.4 Method 4

The flow chart for method 4 is shown in Fig. 6.

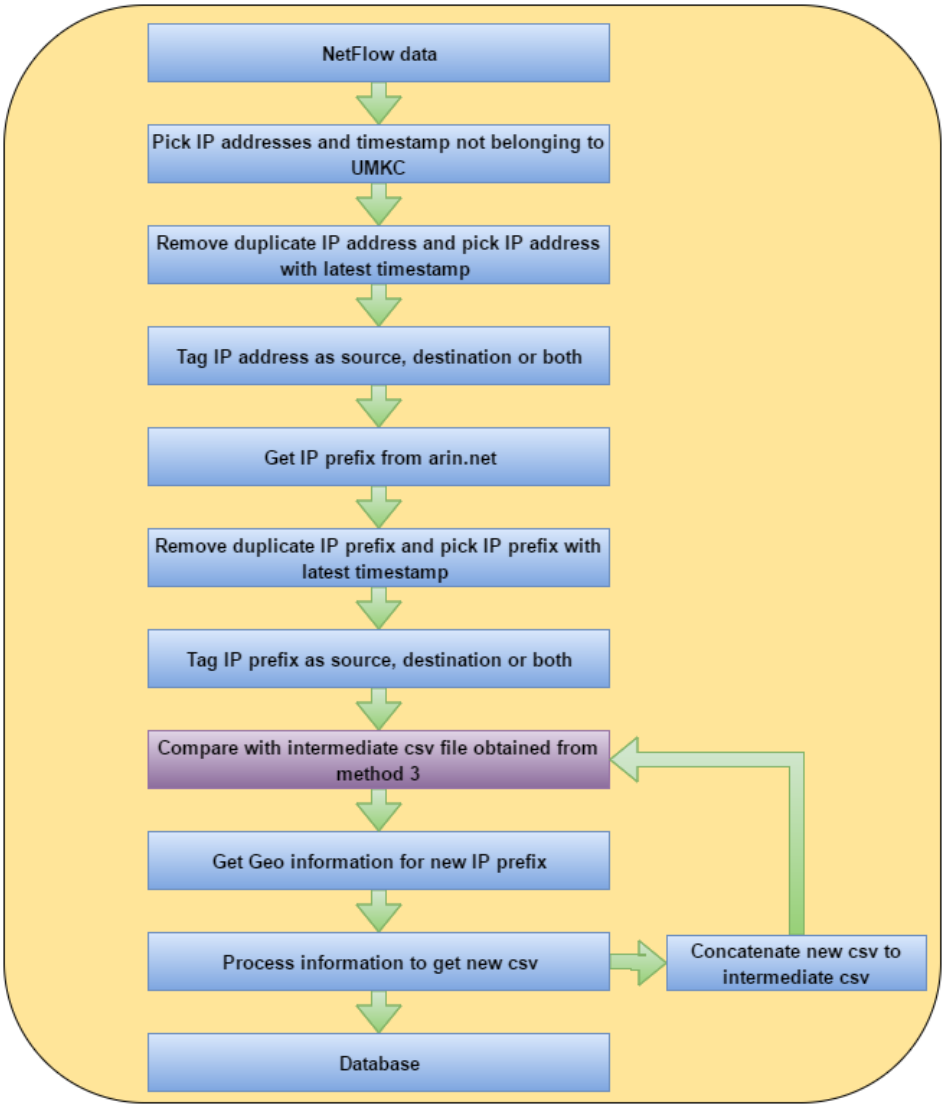

Figure 6: Method 4

Method 4 is similar to method 3 except that it is developed keeping in mind that when the database is filled with geographical data of IP prefixes for the first time and new data is

to be added, there is no need to do main lookup for IP prefixes that already exist in the database. For this, the IP prfixes obtained from ARIN lookup are first compared to IP prefixes in intermediate csv file and then the main lookup is performed only for distinct IP prefixes. The comparison process, thus, avoids addition of same information in the database saving us some time.

Pros of Method 4:

- Inherits all the advantages of method 3.
- More efficient as main lookup is performed only for the IP prefixes that are not present in the database.

Cons of Method 4:

- Increased complexity as comparison is carried out to find distinct IP prefixes. Also, preserving timestamps and tags for each IP address/IP prefix throughout the ARIN lookup and main lookup contributes to complex operations.

3.5 Summary

Table 1: Similarities and Differences between methods

Properties	**Method 1**	**Method 2**	**Method 3**	**Method 4**
ARIN Lookup	No	Yes	Yes	Yes
Parallelism	No	No	Yes	Yes
Preserving timestamps and tags	No	No	Yes	Yes
Redundant Geographical information	Yes	No	No	No
Efficient	No	No	Yes	Yes. Better than method 3
Complex	No	No	Yes	Yes. More complex than method 3

As main lookup, in method 3 and 4, is performed based on IP prefixes, we overcome the problem of downloading redundant information for IP addresses belonging to same subnet multiple times. This improves the efficiency of method 3 and 4. In addition to this, method 3 and 4 are more optimized than method 1 and 2 because HTTP requests are sent in parallel and not sequentially. Since, we introduced tagging of IP addresses and IP prefixes in method 3 and 4, they are more complex as compared to method 1 and 2. Moreover, the complexity of method 3 and 4 increases when the tags and timestamps are preserved during the ARIN lookup and main lookup.

CHAPTER 4

WEB APPLICATION

Once the database is filled with geographical information, we display this information on Google maps in the form of markers. The markers, representing geographical data belonging to each IP, has co-ordinates as that of city's latitude and longitude. We incorporated four filters in this application and they are date picker, time slider, drop down menus based on continent's name and ‘type’ field.

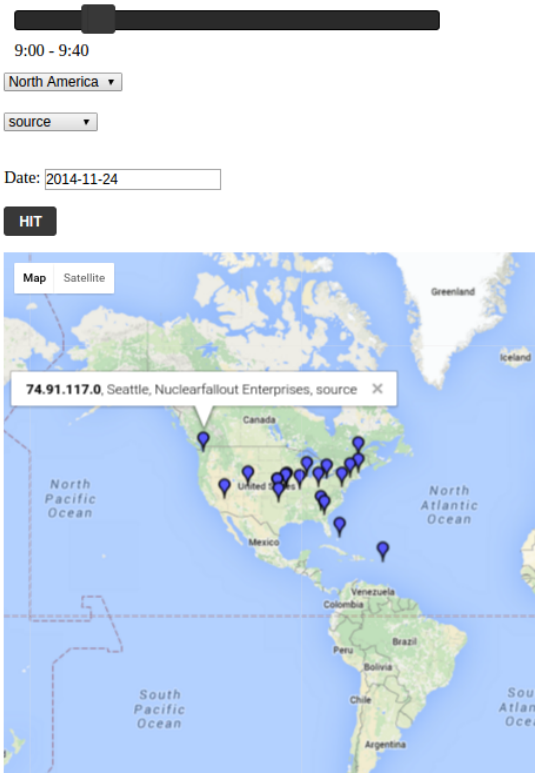

Figure 7: Google map:North America

1. Date picker and time slider are used to filter markers based on date and time. With these filter, we can see IP addresses that were accessed on specific date in particular time range. The range of time slider is 00.00 hours to 23.00 hours.

2. Drop down menu with continents allows to filter markers based on continent's name and drop down menu with ‘type’ field enables to filter markers depending upon whether IP address is either source, destination or both.

The use of filters allows to curb cluttering of markers on Google maps. Also, it helps to visualize data giving near real time experience. In Fig. 7, the information window consists of IP prefix, city location, ISP's name and type field.

Fig. 8 represents the traffic observed between UMKC and Asia on 2014/11/24 between 09:00 hours and 22:40 hours. Fig. 9 shows the traffic observed between UMKC and Europe on 2014/11/24 between 09:00 hours and 16:00 hours.

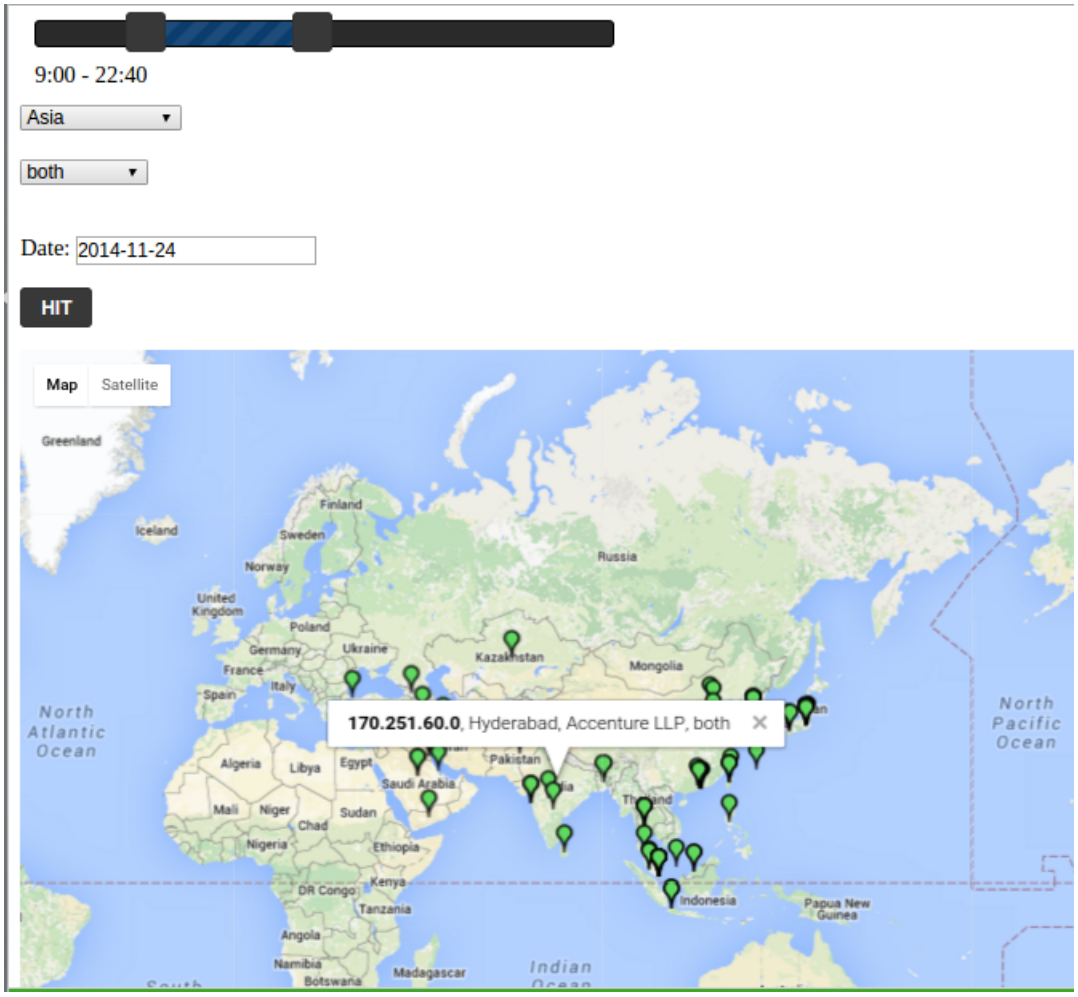

Figure 8: Google map:Asia

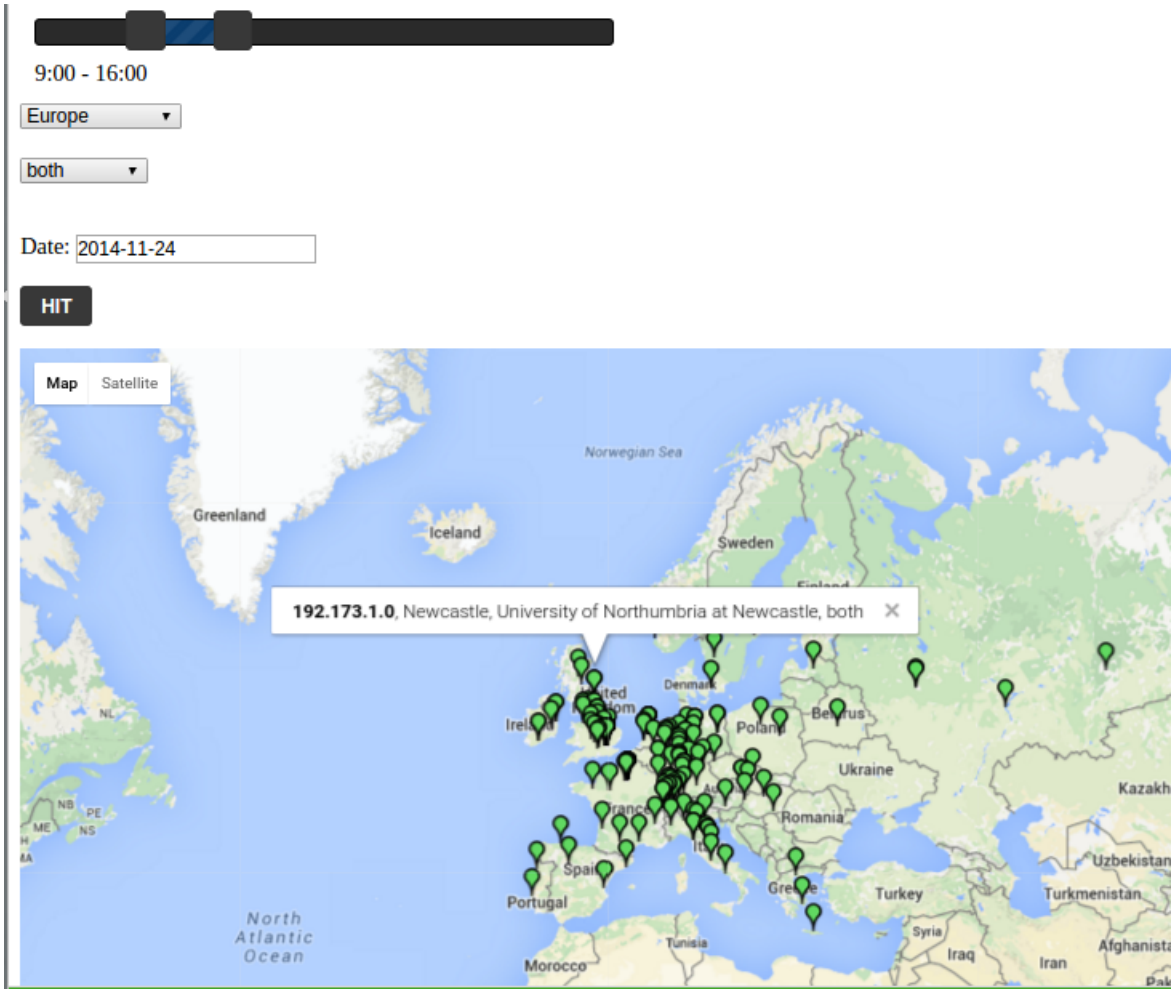

Figure 9: Google map:Europe

CHAPTER 5

RESULTS

This chapter presents results that were observed for the four methods we discussed in Chapter 4. The results were observed in three categories as follows:

1. **Category 1**: Comparison of first three methods on the basis of amount of time taken to process 222 MB of NetFlow data and retrieve geographical information.

2. **Category 2**: The amount of time taken by method 3 to process NetFlow file of various sizes (6 GB, 15 GB, 27 GB and 42 GB).

3. **Category 3**: The amount of time taken by method 4 to process 42 GB of NetFlow file when geographical information obtained from 27 GB of NetFlow data is already stored in the database. We compare method 3 and 4 in this section.

The specifications of 1 core virtual machine (VM) and quad core machine we used to capture results are shown in Table 2. Table 3 represents different sizes of NetFlow data in terms of time duration and number of NetFlow records

Table 2: Hardware Specifications

Specifications	**1 core VM**	**4 core machine**
Processor	Intel(R) Core(TM i5-2410M CPU 2.30 GHz)	8x Intel(R) Core(TM i7-4700HQ CPU 2.40 GHz
Memory	12 GB	8 GB
OS	Ubuntu 14.04.3 LTS	Ubuntu MATE 16.04 LTS

Table 3: NetFlow data in terms of hours and records

Data Set	**NetFlow Data in GB**	**Duration in hours**	**Number of records** $\times 10^6$
1	0.222	0.25	1.51
2	6.00	3.00	44.48
3	15.00	6.00	104.23
4	27.00	12.00	182.83
5	42.00	24.00	288.00

5.1 Category 1

This section consists of comparison of method 1, method 2 and method 3. All the results were observed on 1 core VM. Table 4 and Fig. 10 summarizes the amount of time taken by method 1, 2 and 3 for processing 222 MB of NetFlow data.

Table 4: Method 1 Vs Method 2 Vs Method 3 on Data Set-1 using 1 core VM

Method	**Time in minutes**
Method 1	690.60
Method 2	272.20
Method 3	33.05

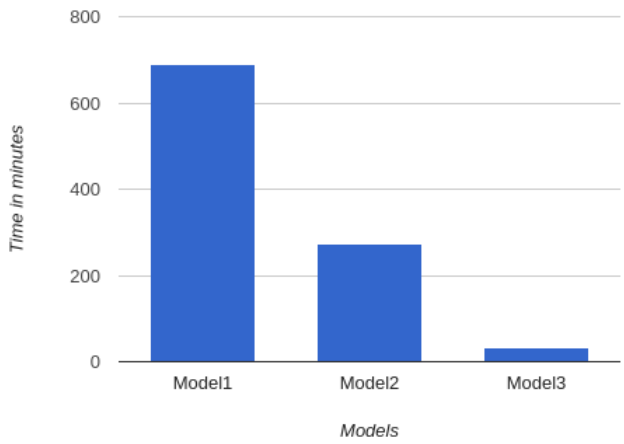

Figure 10: Method 1 Vs Method 2 Vs Method 3 on Data Set-1 using 1 core VM

5.2 Category 2

As method 3 outperformed method 1 and 2, in this section, we present amount of time taken by method 3 to process 6 GB, 15 GB, 27 GB and 42 GB of NetFlow data. All results were observed on 1 core VM and 4 core machine. Table 5 and Fig. 11 represents the amount of time taken to process different sizes of NetFlow data.

Table 5: Processing time for different NetFlow files using 1 core VM and 4 core machine

Data Set	Time in minutes on 1 core VM	Time in minutes on 4 core Machine
2	47.16	7.60
3	74.00	12.00
4	175.01	27.42
5	291.10	42.63

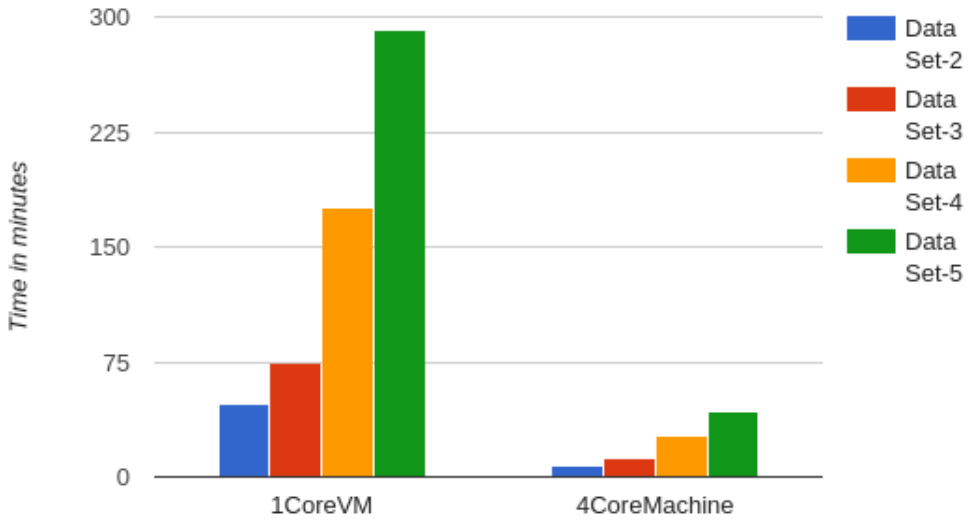

Figure 11: Processing time of Method 3 for different NetFlow data sizes.

5.3 Category 3

This section shows the comparison of method 4 with method 3 on the basis of time required by method 4 to process 42 GB of NetFlow data when geographical information obtained by processing 27 GB of NetFlow data is already present in the database. All the results were observed on 1 core VM and 4 core machine. Table 6 and Fig. 12 summarize this section.

Table 6: Method 3 Vs Method 4 on Data Set-5 using 1 core VM and 4 core machine

Method	Time in minutes on 1 core VM	Time in minutes on 4 core machine
3	291.10	42.63
4	235.80	31.56

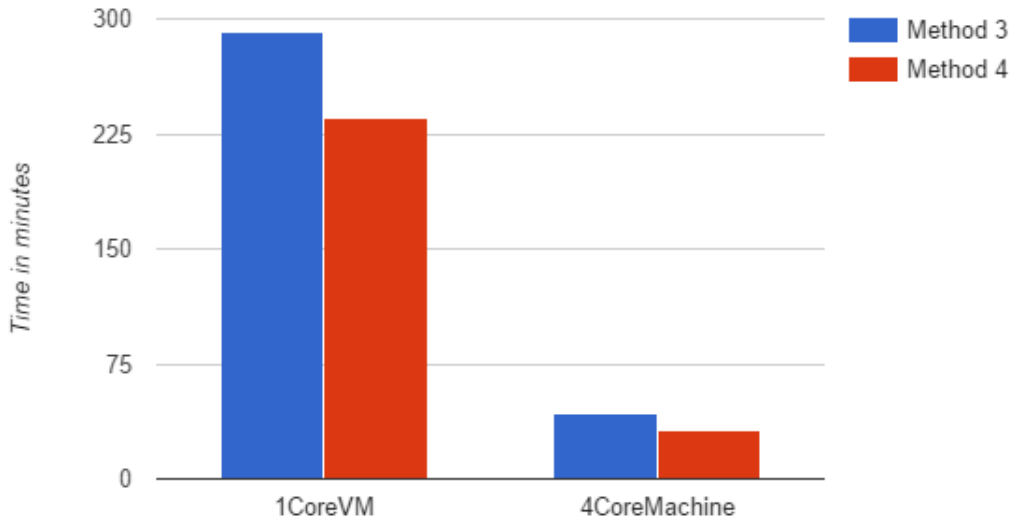

Figure 12: Method 3 Vs Method 4 on Data Set-5 using 1 core VM and 4 core machine

We end this chapter by presenting top ten organizations, ISPs and AS numbers available in our database in Table 7 based on their maximum repetition. We also plot a pie chart in Fig. 13 on the basis of different continents available in our database. The database contains geographical information obtained by processing 42 GB of NetFlow data.

Table 7: Top 10 organization, ISP and AS numbers

Organizations	**ISPs**	**AS numbers**
Comcast Cable	Comcast Cable	7922, Comcast Cable Communications Inc
Optimum Online	Optimum Online	7018, AT&T Services Inc.
Charter Communications	Charter Communications	6128, Cablevision Systems Corp.
AT&T Internet Services	AT&T Internet Services	20115, Charter Communications
Cox Communication	AT&T Services	Unknown
Comcast Business Communication	Cox Communications	22773, Cox Communications Inc.
AT&T Services	Comcast Business Communications	209, Qwest Communications Company LLC
Rogers Cable	CenturyLink	812, Rogers Cable Communications Inc.
Bell Canada	Verizon Business	7029, Windstream Communications Inc
Century Link	Rogers Cable	577, Bell Canada

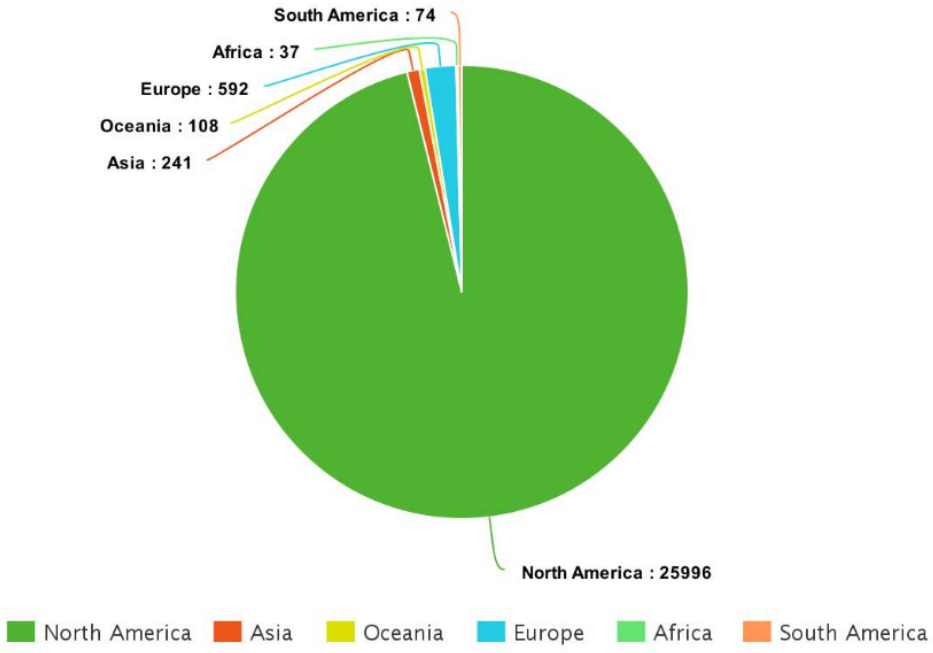

Figure 13: Pie chart representation for continents

CHAPTER 6

CONCLUSION

In this thesis, we started with challenges faced by network operators in monitoring their networks and why it is necessary to have know-how of geographical data. We proposed our idea of scraping websites using xidel based on IP address mapping strategy to optimize the process of Geo-IP Lookup where IP addresses are provided by NetFlow data. From the experimental results, we deduce that method 3 is optimized method as it processes 42 GB of NetFlow data within an hour, that is, 42 minutes on quad core machine. We also saw that GNU's parallel command has played a vital role to achieve optimization. The degree of optimization can further be improved if parallel command is operated on a much better CPU. But, we have to bear the cost of new hardware.

In addition to this, we saw how method 4 can be used to optimize the performance of method 3 by not performing main lookup for IP prefixes that already exists in the database. Furthermore, a visualization tool was developed to give near real time experience by displaying geographical data on Google Maps in the form of markers. In order to avoid cluttering of markers, the geographical information was filtered based on date, time, and continent and type field.

Optimized Geo-IP Lookup tool could aid UMKC's IT department to locate IP prefixes who are generating traffic towards UMKC and IP prefixes of various organizations, ISPs providing service to UMKC users. It could also be used to find continent, country,

state, and city specific traffic by running simple SQL queries. In order to further reduce the processing time, we also tried to perform tagging of IP prefixes in parallel but the processing time did not come down. On the basis of results of method 4, we anticipate that the processing time will gradually come down as more and more geographical data for IP prefixes is stored in the database. We believe that the processing time can further be reduced if IP prefixes are known beforehand, allowing us to skip ARIN lookup and directly perform main lookup. We conclude our work by stating the limitations involved in this method.

- We are scraping website to get geographical data. All methods fails if there is no internet connection.
- If HTML elements of webpage changes, similar changes should be made in HTML tags that are passed to xidel.
- The accuracy of geographical data obtained depends upon how often the geolocation database used by www.ip-tracker.org is updated.
- For better performance, appropriate hardware is required.
- If a person is connected to Virtual Private Network (VPN), then the main lookup is performed based on virtual network's IP prefix providing us false information. In addition to this, false information can also be obtained when user uses hosted services. In this case, main lookup is performed based on IP prefixes of servers providing these services.

APPENDIX A

XIDEL

In this chapter, we will we present the usage of xidel to scrape websites. Xidel is a command line tool to download and extract data from HTML/XML pages. We used xidel to extract geographical data from www.ip-tracker.org and IP prefix from www.arin.net. The usage of xidel in our work is as follows:

```
#!/bin/bash
xidel http://www.ip-tracker.org/locator/ip-lookup.php?ip
   =8.8.8.8 -e "//table/tbody/tr[3]/td[2]/table/tbody/tr[2]
   "
```

The arguments passed to xidel are URL (www.ip-tracker.org) and HTML tags. In this case, HTML tags passed to xidel are Xpath expressions for a HTML table available at www.ip-tracker.org. The -e flag tells xidel to extract web content between HTML tags.

APPENDIX B

GNU'S PARALLEL

GNU parallel is a shell tool for executing jobs in parallel using one or more computers. A job can be a single command or a small script that has to be run for each of the lines in the input. The usage of parallel in our work is as follows:

```
#!/bin/bash
iptrac() {
xidel http://www.ip-tracker.org/locator/ip-lookup.php?ip
   =8.8.8.8 -e "//table/tbody/tr[3]/td[2]/table/tbody/tr
   [2]"
}
cat NetFlow_data | parallel iptrac
```

Here, we first define a function iptrac that contains xidel to extract webpages. And, then the NetFlow data is piped into this function which executes xidel in parallel.

REFERENCE LIST

[1] Backstrom, L., Sun, E., and Marlow, C. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 61–70.

[2] Carela-Español, V., Barlet-Ros, P., Cabellos-Aparicio, A., and Solé-Pareta, J. Analysis of the impact of sampling on NetFlow traffic classification. *Computer Networks 55*, 5 (2011), 1083–1099.

[3] Celeda, P., Velan, P., Rabek, M., Hofstede, R., and Pras, A. Large-scale geolocation for NetFlow. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)* (2013), IEEE, pp. 1015–1020.

[4] Chen, J., Liu, F., Luo, X., Zhao, F., and Zhu, G. A landmark calibration-based IP geolocation approach. *EURASIP Journal on Information Security 2016*, 1 (2016), 1–11.

[5] Claise, B. Cisco Systems NetFlow Services Export Version 9. RFC 3954, RFC Editor, October 2004. `http://www.rfc-editor.org/rfc/rfc3954.txt`.

[6] Dahnert, A. HawkEyes: An advanced IP Geolocation approach: IP Geolocation using semantic and measurement based techniques. In *Cybersecurity Summit (WCS), 2011 Second Worldwide* (2011), IEEE, pp. 1–3.

[7] Endo, P. T., and Sadok, D. F. H. Whois based geolocation: A strategy to geolocate internet hosts. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications* (2010), IEEE, pp. 408–413.

[8] Hofstede, R., and Fioreze, T. SURFmap: A network monitoring tool based on the Google Maps API. In *2009 IFIP/IEEE International Symposium on Integrated Network Management* (2009), IEEE, pp. 676–690.

[9] Hu, Z., Heidemann, J., and Pradkin, Y. Towards geolocation of millions of IP addresses. In *Proceedings of the 2012 ACM conference on Internet measurement conference* (2012), ACM, pp. 123–130.

[10] Jiang, H., Moore, A. W., Ge, Z., Jin, S., and Wang, J. Lightweight application classification for network management. In *Proceedings of the 2007 SIGCOMM workshop on Internet network management* (2007), ACM, pp. 299–304.

[11] Karagiannis, T., Papagiannaki, K., and Faloutsos, M. BLINC: multilevel traffic classification in the dark. In *ACM SIGCOMM Computer Communication Review* (2005), vol. 35, ACM, pp. 229–240.

[12] Katz-Bassett, E., John, J. P., Krishnamurthy, A., Wetherall, D., Anderson, T., and Chawathe, Y. Towards IP geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (2006), ACM, pp. 71–84.

[13] Krmicek, V., Vykopal, J., and Krejci, R. Netflow based system for NAT detection. In *Proceedings of the 5th international student workshop on Emerging networking experiments and technologies* (2009), ACM, pp. 23–24.

[14] Lakkaraju, K., Yurcik, W., and Lee, A. J. NVisionIP: netflow visualizations of system state for security situational awareness. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (2004), ACM, pp. 65–72.

[15] Mohd, A. B., and bin Mohd Nor, S. Towards a flow-based internet traffic classification for bandwidth optimization. *International Journal of Computer Science and Security (IJCSS) 3*, 2 (2009), 146–153.

[16] Rossi, D., and Valenti, S. Fine-grained traffic classification with netflow data. In *Proceedings of the 6th international wireless communications and mobile computing conference* (2010), ACM, pp. 479–483.

[17] Tange, O. GNU Parallel - The Command-Line Power Tool. *;login: The USENIX Magazine 36*, 1 (Feb 2011), 42–47.

[18] Yin, X., Yurcik, W., Treaster, M., Li, Y., and Lakkaraju, K. VisFlowConnect: netflow visualizations of link relationships for security situational awareness. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (2004), ACM, pp. 26–34.

[19] Zander, B. Xidel - HTML/XML/JSON data extraction tool.

VITA

Swatesh Vilas Pakhare was born on December 20, 1991 in Mumbai, Maharashtra, India. He graduated from University of Mumbai in 2013 with B.E. degree in Electronics and Telecommunication.