

Log File Analysis Using Pig

Chan-Ching Hsu

Purpose

The goal is to use the Pig platform and the scripting language Pig Latin to analyze large log files and trace files. Scripts written in Pig Latin are automatically converted to MapReduce jobs.

Pig

Here is the link of pig scripts in book “Programming Pig” by Alan Gates

<https://github.com/alanfgates/programmingpig>

Look here for documentation on pig:

<http://pig.apache.org/docs/r0.14.0/>

The Pig Latin language can be learnt from the link:

<http://pig.apache.org/docs/r0.14.0/basic>

Pig utilities:

<http://pig.apache.org/docs/r0.14.0/cmds>

Task 1

For this task we analyze US demographic data. It can be found at:

<http://www.census.gov/geo/maps-data/data/gazetteer2010.html>

We are interested in the filed “**ALAND – Land Area (square meters)**” and want to find out the top 10 states according to the land area. The output file is the following.

```
AK      1.477953211577E12
TX      6.76586997978E11
CA      4.03466310059E11
MT      3.7696187867E11
NM      3.1416074824E11
AZ      2.94207314414E11
NV      2.84331937541E11
CO      2.68431246426E11
WY      2.51470069067E11
OR      2.48607802255E11
```

Task 2

For this task we have a network trace file from a network monitor. The file can be queried for information about network traffic.

An example entry in the file is:

10:20:00.000020 IP 244.131.189.196.22379 > 245.184.172.199.80: tcp 0

The data format is as follows:

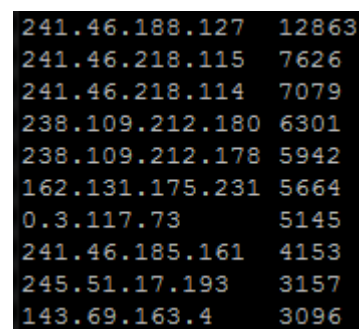
<Time> "IP" <Source IP> ">" <Destination IP> <protocol> <protocol dependent data>

The protocol dependent data will be different for TCP, UDP etc. Usually, IP addresses are of the format A.B.C.D. However, the data presents IP addresses in the format of A.B.C.D.E. The extra information including and after the fourth "." need to be processed and got rid of.

In network monitoring, it is usual to know the identity of those IP sources that connect to a large number of distinct IP destinations. Such sources are often malicious nodes, or compromised by malicious software, and maybe sending spam.

We are interested in finding the top 10 source IP addresses ranked according to the number of unique destination IP addresses that they connect to using the TCP protocol.

The output looks like the snapshot below.



241.46.188.127	12863
241.46.218.115	7626
241.46.218.114	7079
238.109.212.180	6301
238.109.212.178	5942
162.131.175.231	5664
0.3.117.73	5145
241.46.185.161	4153
245.51.17.193	3157
143.69.163.4	3096

Task 3

Suppose that there was a firewall blocked IP addresses that it is believed potentially unsafe. The list of all IP connections that were blocked is stored in memory, and also in a log file, but the firewall log was lost due to a failure. We want to regenerate this log file from the other data sources that we have. It is important to regenerate this information as the IP addresses that are blocked regularly are added to a black list.

The lost firewall log contained details of all blocked connections and looks as follows:

<Time> <Connection ID> <Source IP> <Destination IP> "Blocked"

The task is to regenerate the log file in the above format by combining information

from other logs that are available. In particular, we have the following files: an IP trace file having information about connections received from different source IP addresses, along with connection IDs and time, and a file containing the connection IDs that were blocked.

The IP trace file has the following format:

0:0:0:9 8 215.160.81.159 > 174.83.200.101 UDP 943

The format is similar to the previous task

<Time> <Connection ID> <Source IP> ">" <Destination IP> <protocol> <protocol dependent data>

The other file has lines in the following format: <Connection ID> <Action Taken>

For instance, it could look as follows:

0 Allowed

1 Blocked

2 Allowed

3 Blocked

The task is as follows.

1. Regenerate the firewall log file
2. Generate the list of all unique source IP addresses that were blocked and the number of times that they were blocked. This list should be sorted by the number of times each IP was blocked.

For this task, it is useful to use the "JOIN" operation. Joining data is a key feature of Pig.

Part of the output file of regenerated firewall log file is shown below.

0:0:0:29	28	40.27.250.205	236.51.254.78	Blocked
0:0:0:202	201	101.168.151.198	27.66.248.101	Blocked
0:0:0:321	320	161.249.225.242	238.78.166.252	Blocked
0:0:0:384	383	13.230.226.105	209.192.49.87	Blocked
0:0:0:412	411	151.166.183.111	15.119.109.212	Blocked
0:0:0:461	460	212.37.157.23	140.199.133.63	Blocked
0:0:0:482	481	99.199.147.128	228.37.202.8	Blocked
0:0:0:538	537	20.155.42.26	194.72.117.225	Blocked
0:0:0:559	558	120.147.218.157	179.33.75.87	Blocked
0:0:0:587	586	50.241.201.37	166.65.111.155	Blocked

The top 10 IP addresses that were blocked in terms of the number of times are listed in the following.

108.2.235.200	9798
13.110.126.21	9788
63.101.203.191	9776
107.68.208.3	9772
79.60.169.28	9769
141.2.188.213	9755
2.164.178.163	9751
152.244.120.111	9750
216.91.116.228	9746
104.230.13.142	9743