Audrey Long
JHU Web Security Module Big Data Assignment
03/23/2021

Introduction

For this assignment you’re going to be looking at 1 month’s worth of HTTP error logs from NASA and analyzing them using Apache Spark. You will use PySpark, the Python API for Apache Spark, to analyze the server log data to identify any suspicious patterns.

1. Install PySpark Start by spinning up an Ubuntu VM in VirtualBox or VMware. Follow the instructions at the following link for installing Apache Spark and accessing it through a Jupyter Notebook: PySpark Install Guide.
2. Next, download the July set of NASA’s HTTP server logs which you can download here. Uncompress them by using the ’gunzip’ command.

Write a Spark Analytic

Here is a nice, gentle introduction to getting PySpark to work. For example, here’s a simple one-liner for reading one of the files you unzipped into an RDD:

```
http_rdd = sc.textFile(”/home/jkovba/weblog.txt”)
```

You can then print out the first element like so: print http_rdd.first() Or you can print the first 10 elements as follows: print http_rdd.take(10)

Web Log Analytic

Get comfortable with some of the Sparks basics and experiment with some of the simple functions that allow you to modify RDD’s. When you’re ready, write a PySpark analytic, complete with comments, and please answer the following analytic questions:

1. How many distinct hostnames/IP’s are in the file? Please list them. (a) Hint: you can string functions together: rdd.distinct().count()

```
81983
['unicomp6.unicomp.net', '129.94.144.152', 'ppptky391.asahi-net.or.jp', 'slip1.yab.com',
'pm13.j51.com', 'dd14-046.compuserve.com', 'usr7-dialup46.chicago.mci.net', 'teleman.pr.mcs.net',
'isdn6-34.dnai.com', 'ix-ftw-tx1-24.ix.netcom.com']
```

2. Give a count of the number of occurrences of each HTTP reply code. (a) Hint: you can use the map() function to rearrange your tuple and use the HTTP reply code as the key. From there, think about how to do a reduce to get the count.

```
count by key
defaultdict(<class 'int'>, {200: 1701534, 304: 132627, 302: 46573, 404: 10845, 403: 54, 500: 62,
501: 14, 400: 5})
```

3. Sum, and display, the total number of bytes per host/IP.

4. Using the ’request’ field, please tell me the number of GET, PUT, and POST requests per hostname/IP address. (a) Hint: you can construct the key in any way you’d like, by combining fields together, for example.

Deliverables

1. Please submit your commented source code in a file named ’spark http.py.txt’
2. Please submit a file called ’analytic results.txt’ that contains the answers to the analytic questions above.

Suggestion

As an alternate, use this as the parser. it returns the same.count() as the one posted in blackboard, but it allows you to keep the - in the timestamp and the leading / too. Unfortunately, you still end up separating the two pieces of the timestamp, but that doesn't seem to impact the assignment at all, so I'll leave that as an exercise for the reader :)

```
parsed_logs_rdd = http_rdd.map(lambda line: line.encode('utf-8')) \
.map(lambda line: line \
.replace(b' - - [', b',') \
.replace(b'] "', b',') \
.replace(b'" ', b',') \
.replace(b' ', b',') \
.split(b',')) \
.map(lambda line: tuple(line)) \
.filter(lambda line: len(line) == 8)
```