Analyzing Apache log files by using Apache Spark

Maruti Sriram

Prasanna Kumar

Sandeep Tammu

I. Introduction

Whenever you visit a website and click a link, the record is stored by the webserver as a log. Imagine what would happen on a moderately busy website on every day. These logs pile up as a stack, and they become so large that it might not be possible to analyze them in a meaningful manner. These log files would possibly contain information about the traffic patterns, geographical information and any mischievous users who are trying compromise the website. When the files become so large, you can't do in-memory processing anymore. This makes it necessary to bring these files into a Hadoop like ecosystems which support both distributed storage and computation. In this project, we would like to analyze Apache log files using one of the big-data processing frameworks, Apache Spark.

Apache Spark is an efficient implementation of original map-reduce paradigm, it can work along with Hadoop and provides a high-level API for doing data analysis and machine learning on top of big data. It uses resilient distributed datasets (RDD's) and has additional inspirations from functional programming. Things like lazy evaluation, and storing the lineage information between RDD's makes it fault-tolerant and scalable for big data needs.

II. Data

The log file data is obtained from the NASA website. It contains the Apache logs of NASA website for the duration of a month. It has 1891715 recorded observations. This figure shows the schema of the parsed NASA log files. NASA did not provide the user identity and client identity which could be useful in extracting some interesting features.

We used pySpark and loaded the data into Spark as a DataFrame. This allowed us to do some nice exploratory data analysis and gain some insights about this data. The accompanying code contains the Python script which parses the data file into a DataFrame.

```
logsDF.show(5)
```

```
+-------------+------------+--------------------+--------------------+--------------------+------+--------+-------------+-------
+
|client_identd|content_size|           date_time|            endpoint|                host|method|protocol|response_code|user_id
|
+-------------+------------+--------------------+--------------------+--------------------+------+--------+-------------+-------
+
|            -|        6245|1995-07-01 00:00:...|    /history/apollo/|        199.72.81.55|   GET|HTTP/1.0|          200|      -
|
|            -|        3985|1995-07-01 00:00:...| /shuttle/countdown/|unicomp6.unicomp.net|   GET|HTTP/1.0|          200|      -
|
|            -|        4085|1995-07-01 00:00:...|/shuttle/missions...|      199.120.110.21|   GET|HTTP/1.0|          200|      -
|
|            -|           0|1995-07-01 00:00:...|/shuttle/countdow...|  burger.letters.com|   GET|HTTP/1.0|          304|      -
|
|            -|        4179|1995-07-01 00:00:...|/shuttle/missions...|      199.120.110.21|   GET|HTTP/1.0|          200|      -
|
+-------------+------------+--------------------+--------------------+--------------------+------+--------+-------------+-------
```

III. Exploratory data analysis

One of the first thing we wanted to see was how the response code distribution would look like. We grouped the data by response code and aggregated counts. We can see a skewed distribution where most of the response codes are 200.

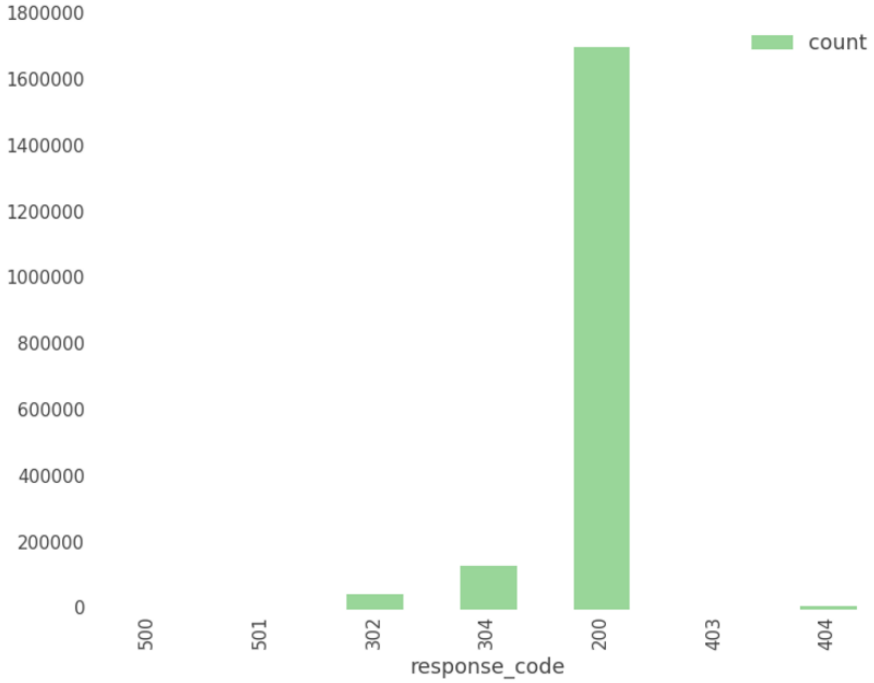

Figure 1: Response code distribution

We then explored what parts of the website are mostly being visited by users and using the same Spark operations allowed us to extract this information easily. Apart from the NASA logo gifs, the most visited pages in the website are the countdown pages.

```
grouped_endpoints.sort('count', ascending= False).head(50)
```

	endpoint	count
16131	/images/NASA-logosmall.gif	111331
14539	/images/KSC-logosmall.gif	89639
12037	/images/MOSAIC-logosmall.gif	60468
10327	/images/USA-logosmall.gif	60014
10294	/images/WORLD-logosmall.gif	59489
544	/images/ksclogo-medium.gif	58802
2950	/images/launch-logo.gif	40871
10694	/shuttle/countdown/	40279
20919	/ksc.html	40226
7844	/images/ksclogosmall.gif	33585
5037	/	32844
21175	/history/apollo/images/apollo-logo1.gif	31072
20315	/shuttle/missions/missions.html	24864
6430	/htbin/cdt_main.pl	22626
18329	/shuttle/countdown/count.gif	22216
2618	/shuttle/countdown/liftoff.html	22000
14242	/shuttle/countdown/count70.gif	20957
17946	/images/launchmedium.gif	20812

Figure 2: Top webpages visited by users

IV. Time based features

The time stamp data is parsed into a time series object in Spark. This allowed us to explore a lot more into the data.

When we plotted number of connection attempts with time of the day, we observed that the traffic load is minimum during early in the morning. Surprisingly, the traffic piqued after the twelfth hour.

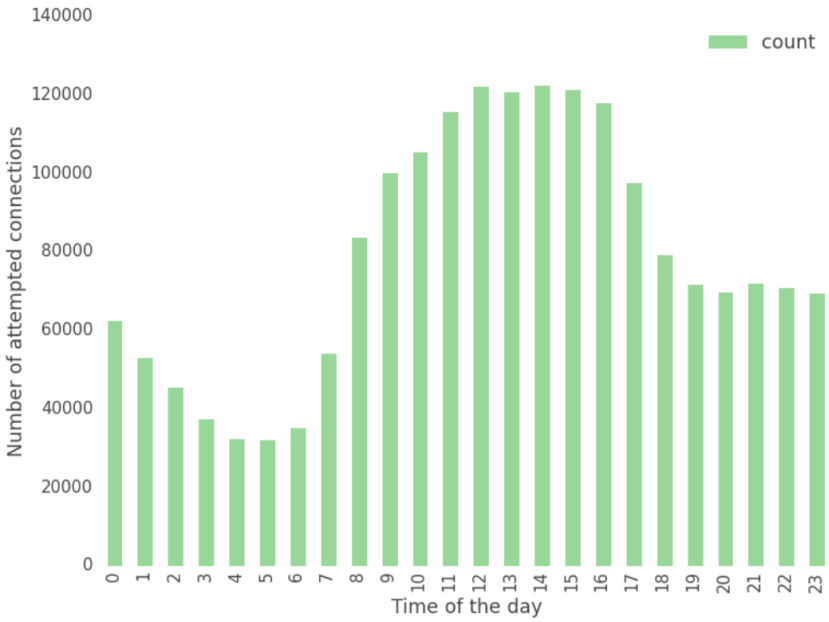

Figure 3: Connections per hour

We also tried to check the patterns in daily data. Since we have only one month of data, we checked connection attempts per day. The maximum number of connections were attempted on 13 July 1995. Checking the date with NASA website revealed that NASA had launched the Discovery space shuttle on that day.

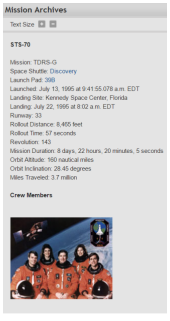

Figure 4: NASA Discovery Space Shuttle: July,13 1995

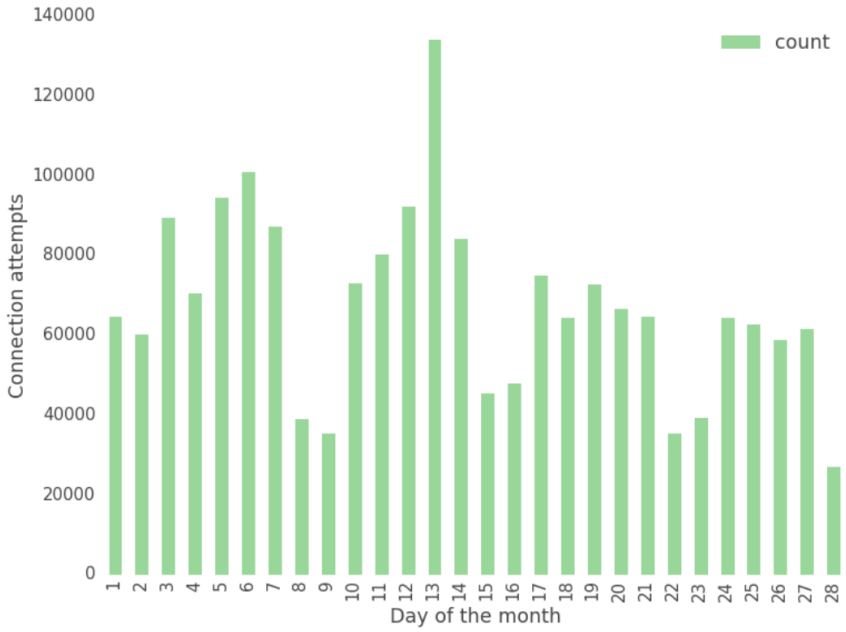

Figure 5: Connection attempts per day of month

We also tried to see patterns in number of connections in a given minute of an hour, and saw an uniform distribution.

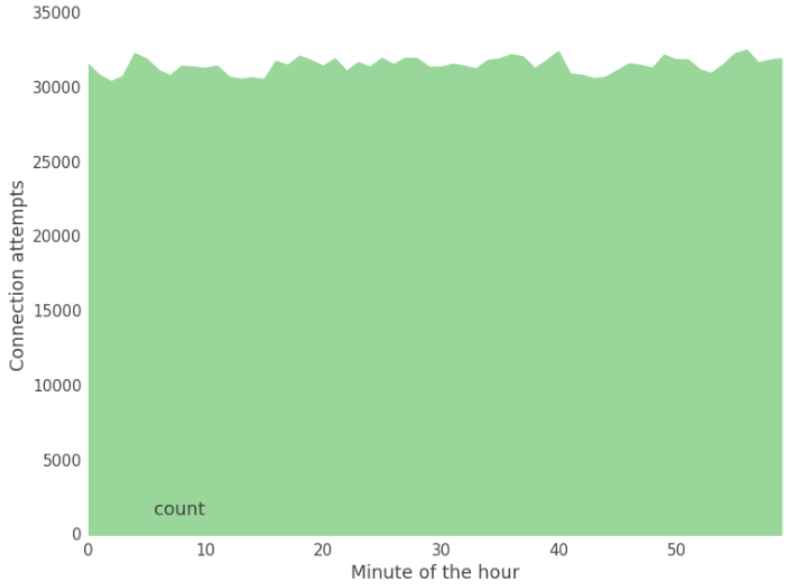

Figure 6: Connection attempts per minute of hour