Flying Squirrels Twitter Project

- **Raji Chavali (TRM35C)**
- **Pranoop Mutha (NMGFG)**
- **Vinay Jaibheem (VJ8GB)**

Objective: "Retrieve tweets from twitter and then extract the hash tags and URL's from the retrieved tweets and run the word count on the extracted hash tags and URL's".

Tools Used: Python, Hadoop, Apache Spark and Hive

Steps:

Step1: Retrieving Tweets from Twitter

Step2: Getting Hash Tags and URL's (expanded_URL's, display URL's and URL's)

Step3: Merging all the files to the path.

Step4: Running Word Count Program on Apache Hadoop

Step5: Running Word Count Program on Apache Spark

Implementation:

Step1: Retrieving Tweets from Twitter and Tweets Link

We achieved the above functionality in Python. The main part is getting consumer key, consumer secret, access token and access secret. Then we will be using the filter command to get the tweets.

Code:

```
import tweepy
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener

consumer_key = 'D31fLYJyQagysqPAc92Q7HmmE'
consumer_secret = 'xDOgZaB0aqJX9tikibq0Mc6ZtHL1UXO3FpB6I5KIzjUfDPvLUv'
access_token = '2628031760-9vEVVEY9uAi6nn3JNSVCsVfB4ieEEZpuarH9Kqu'
access_secret = 'bvaSlQh1AzMYeM2vRZM2uWroAA5qaFDabf5xhBLW8dTE9'

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)
```

```
api = tweepy.API(auth)
print(api)

class MyListener(StreamListener):

    def on_data(self, data):
        try:
            with open('twittertweets.json', 'a') as f:
                f.write(data)
                return True
        except BaseException as e:
            print("Error on_data: %s" % str(e))
        return True

    def on_error(self, status):
        print(status)
        return True

twitter_stream = Stream(auth, MyListener())
twitter_stream.filter(track=['#','@'])
```

Tweets File (JSON):

https://drive.google.com/open?id=0BwgZejXOtTRdRkw1MkRSbUMtVG8

Step2: Getting Hash Tags and URL's (expanded_URL's, display URL's and URL's)

We have loaded twitter json file using hive to twitter_table and retrieved hashtags,urls (url,expanded_url,display_url)

Code Snippet:

- create table twitter_table (json string);
- load data local inpath '/home/raji/Downloads/tweets.json' `INTO TABLE` twitter_table;

- INSERT OVERWRITE DIRECTORY '/test/proj1_hashtags'

```
select explode(split(substr(get_json_object(twitter_table.json,
'$.entities.hashtags.text'),2,length(get_json_object(twitter_table.json,
'$.entities.hashtags.text')) - 2) , ',') ) as hashtags from twitter_table
where get_json_object(twitter_table.json, '$.entities.hashtags.text') is not null ;
```

- INSERT OVERWRITE DIRECTORY '/test/proj1_url'

```
select get_json_object(twitter_table.json, '$.entities.urls.url')
from twitter_table
where get_json_object(twitter_table.json, '$.entities.urls.url')  is not null;
```

- INSERT OVERWRITE DIRECTORY '/test/proj1_expanded_url'

```
select  get_json_object(twitter_table.json, '$.entities.urls.expanded_url')
from twitter_table
where get_json_object(twitter_table.json, '$.entities.urls.expanded_url') is not null;
```

- INSERT OVERWRITE DIRECTORY '/test/proj1_display_url'

```
select  get_json_object(twitter_table.json, '$.entities.urls.display_url')
from twitter_table
where get_json_object(twitter_table.json, '$.entities.urls.display_url') is not null;
```

Step3: Merging all the files to the path (Extracted URL's and Hash Tags)

Output:

https://github.com/PranoopMutha/CS5540_PB_FlyingSquirrels_TwitterProject/tree/master/Documentation

Step4: Running Word Count Program on Apache Hadoop

Code Snippet:

```
hadoop jar hadoop/hadoop-2.8.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount /test/proj1_output/hashtags_urls /test/projwc
```

Apache Hadoop Output:

https://github.com/PranoopMutha/CS5540_PB_FlyingSquirrels_TwitterProject/tree/master/Documentation/HadoopWordCount%20Output

Step5: Running Word Count Program on Apache Spark

Code Snippet:

```
scala > var hturl_count=sc.textFile("/home/raji/Downloads/PROJ1/flying_squarels_hashtags_urls.txt")

scala > val hucounts=hturl_count.flatMap(_.split(" ")).map(word => (word, 1)).reduceByKey(_+_)

hucounts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[13] at reduceByKey at <console>:26

scala> hucounts.collect()

scala> hucounts.saveAsTextFile("/test/sparkwc.txt")
```

Apache Spark Output:

https://github.com/PranoopMutha/CS5540_PB_FlyingSquirrels_TwitterProject/tree/master/Documentation/SparkWordCount%20Output

OUTPUT/HADOOP LOG FILES

https://github.com/PranoopMutha/CS5540_PB_FlyingSquirrels_TwitterProject/tree/master/Documentation/HadoopLogs

REFERNECES

http://thornydev.blogspot.com/2013/07/querying-json-records-via-hive.html