Twitter Data Analysis Using Streaming APIs:

Team

1. Chakra Pavan Kumar Kota (16283878)
2. S.V.Sai Kumar Reddy (16280780)
3. Ajith Reddy Guduru(16282036)

Project Goal: To develop a system that collects, parses and analyzes and visualizes twitter tweets.

Phase 1:

Objectives:

1. Collect tweets using twitter streaming APIs and store in a text file.
2. Extract Hashtags and Urls from raw text file.
3. Push the extracted file to hdfs
4. Finally, to run word count both in Apache Hadoop and Apache Spark on the extracted Hashtags and Urls.

Prerequisites:

To have the following setup done in the Virtual Machine

1. Apache Hadoop
2. Apache Spark
3. Python
4. Scala
5. Java
6. Tweepy Python Library
7. Twitter developer account
8. A virtual machine with suffient physical memory for processing.

Other softwares used

1. Virtual Machine from Microsoft azure
2. Eclipse

Configuring Streaming API:

In order to get the tweets we need to have access to developer account which can be done by the following steps.

1. Login to the twitter developer platform (https://developer.twitter.com/en/apps)
2. Generate Consumer keys and Access Tokens.

Collecting Tweets:

1. twitter_streaming.py collects the tweets from API using the Access Tokens and Consumer Keys.
   2.Collected tweets are stored in tweets_final.text file.

Hashtags And Urls Extraction

1. hashtags_extraction.py extracts the Hashtags and stores it in to hashtags.txt
2. urls_extraction.py extracts the Urls and stores it in to urls.txt.

Pushing Extracted Files into HDFS:

1.We need to create a directory in HDFS using the following command

```
hdfs dfs -mkdir /path
```

2.We need to push hastags.txt, urls.txt into HDFS using the following command.

```
hdfs dfs -copyFromLocal /localpath /hdfsfolderpath
```

Running the Word Count:

1. Run the following command to find the word count on mapreduce.

```
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount /hdfsinputpath /outputpath
```

2. For the word count on spark, we need to navigate to the bin folder in the archived spark tar folder
   and we need to run the following commands

`./spark-shell` - This opens a spark shell with spark context preset

`val textFile = sc.textFile(“hdfs://localhost:9000/hdfsinputpath”)

val counts = textFile.flatMap(line => line.split(" "))
.map(word => (word,1))
.reducedBykey(*+*)
counts.saveasTextFile(“hdfs://localhost:9000/outputpath”)`

Collecting the result into local file system

Now collect the output from hdfs to local file using

```
hdfs dfs -copyToLocal /hdfs/output/part-r-00000 /localfolder
```

References:

http://docs.tweepy.org/en/v3.5.0/api.html

https://stackoverflow.com/

https://youtu.be/EDcXRPKk7Qk