

Principles of Big Data Management

Project Phase - 1



Team Members:

Ankitha Wankhede - 16233344

Rajeswari Devi Namana -18135299

Sravya Reddy Bokka - 16240386

Goal:

- To Collect Tweets using Twitter's Streaming APIs (e.g., 100K Tweets) (<https://dev.twitter.com/docs/streaming-apis>)
- Extract all the hashtags and URLs in the tweets
- Run the WordCount example in Apache Hadoop and Apache Spark on the extracted hashtags/URLs and collect the output and log files from Hadoop. Add a README file.

Step 1: Collection of tweets from Twitter Streaming API:

Executed code in python using tweepy library.

Code Used:

```
#Import the necessary methods from tweepy library
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream

#Variables that contains the user credentials to access Twitter API
access_token = "888958069098635264-AsvKqThYHQL02oVnfa148kEU9ooPks2"
access_token_secret = "rgWc4aerpJruAwdZHpjAZYHn55qD92TJnUUpaNd8G1j7I"
consumer_key = "zKhTdLsdjIG8H1fRa0Fhxoofo"
consumer_secret = "vorTGnycUlmTr6RhjocPmYJ0A3yttUTpXNBiyjB7bNw2YOSGvC"

#This is a basic listener that just prints received tweets to stdout.
class StdOutListener(StreamListener):

    def on_data(self, data):
        print(data)
        return True

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

if __name__ == '__main__':

    #This handles Twitter authentication and the connection to Twitter Streaming API
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, l)

    #This line filter Twitter Streams to capture data by the keywords: 'mom'
    #stream.filter(track=['hashtags', "(?P<url>https?:[/^[^s]+)"])
    stream.filter(track=["mom"])
```

Code Screenshots :

bigdata project phase 1 - [C:\Users\rajin\Desktop\Masters subjects\Principles of big data\bigdata project phase 1] - ...\twitter_streaming.py - PyCharm Com

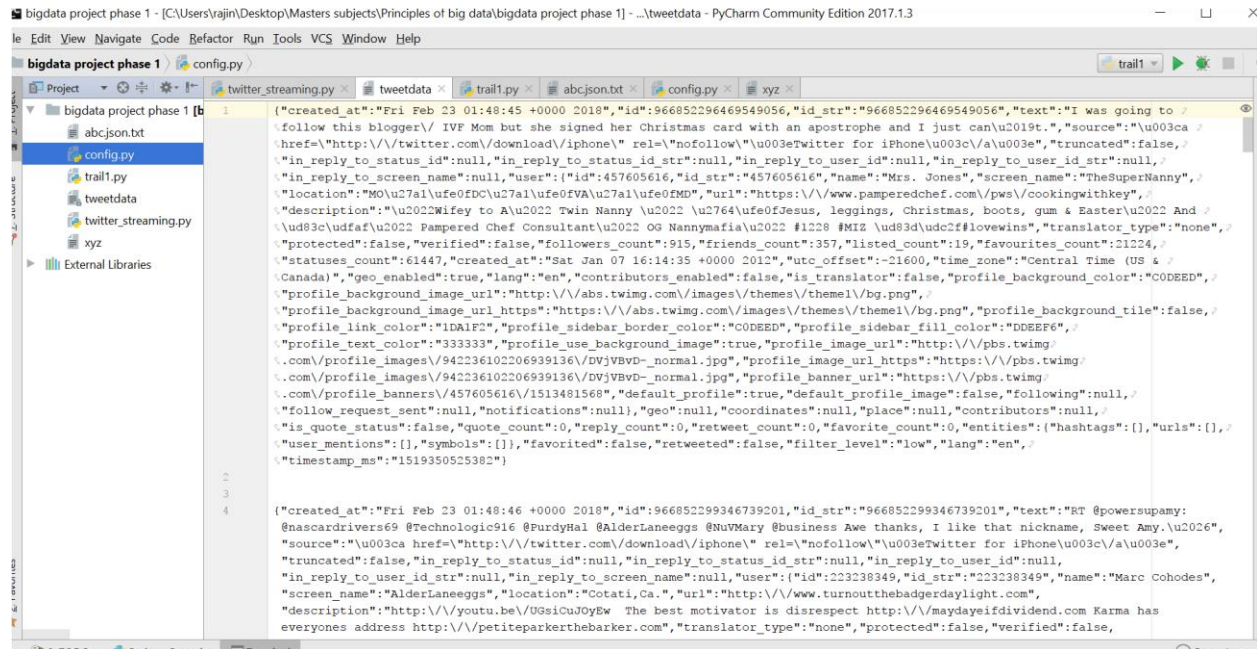
File Edit View Navigate Code Refactor Run Tools VCS Window Help

bigdata project phase 1 > twitter_streaming.py

Project Structure: bigdata project, abcjson.txt, config.py, trail1.py, tweetdata, twitter_streaming.py, xyz, External Libraries

```
1 #Import the necessary methods from tweepy library
2 from tweepy.streaming import StreamListener
3 from tweepy import OAuthHandler
4 from tweepy import Stream
5
6 #Variables that contains the user credentials to access Twitter API
7 access_token = "888958069098635264-AsvKqThYHQL02oVnfa148kEU9ooPks2"
8 access_token_secret = "rgWc4aerpJruAwdZHpjAZYHn55qD92TJnUUpaNd8G1j7I"
9 consumer_key = "zKhTdLsdjIG8HlfRa0Fhxoofo"
10 consumer_secret = "vorTGnycUlmTr6RhjocPmYJ0A3yttUTpXNBiyjB7bNw2YOSGvC"
11
12
13 #This is a basic listener that just prints received tweets to stdout.
14 class StdOutListener(StreamListener):
15
16     def on_data(self, data):
17         print(data)
18         return True
19
20     def on_error(self, status):
21         print(status)
22
23
24 if __name__ == '__main__':
25
26     #This handles Twitter authentication and the connection to Twitter Streaming API
27     l = StdOutListener()
28     auth = OAuthHandler(consumer_key, consumer_secret)
29     auth.set_access_token(access_token, access_token_secret)
30     stream = Stream(auth, l)
31
32
33 #This line filter Twitter Streams to capture data by the keywords: 'mom'
34 #stream.filter(track=['hashtags', "(?P<url>https://[^\s]+)"])
35 stream.filter(track=["mom"])
```

Output Screenshot: Collected Tweets:



Step 2: Extraction of URLs and hashtags from collected tweets:

Code:

```
import codecs
from datetime import datetime
import json
import os
import string
import sys
import time

def parse_json_tweet(line):
    tweet = json.loads(line)
    # print line
    if tweet['lang'] != 'en':
        # print "non-english tweet:", tweet['lang'], tweet
        return ['', '', '', [], [], []]

    date = tweet['created_at']
    id = tweet['id']
    nfollowers = tweet['user']['followers_count']
    nfriends = tweet['user']['friends_count']

    if 'retweeted_status' in tweet:
        text = tweet['retweeted_status']['text']
    else:
        text = tweet['text']

    hashtags = [hashtag['text'] for hashtag in tweet['entities']['hashtags']]
    users = [user_mention['screen_name'] for user_mention in
tweet['entities']['user_mentions']]
```

```

urls = [url['expanded_url'] for url in tweet['entities']['urls']]

media_urls = []
if 'media' in tweet['entities']:
    media_urls = [media['media_url'] for media in tweet['entities']['media']]

return [hashtags, urls]

'''start main'''
if __name__ == "__main__":
    file_timeordered_json_tweets = codecs.open(sys.argv[1], 'r', 'utf-8')
    fout = codecs.open(sys.argv[2], 'w', 'utf-8')

    # efficient line-by-line read of big files
    for line in file_timeordered_json_tweets:
        try:
            [tweet_gmtime, tweet_id, text, hashtags, users, urls, media_urls,
nfollowers, nfriends] = parse_json_tweet(
                line)
            # if not tweet_gmtime: continue
            # fout.write(line)
            # "created_at": "Mon Feb 17 14:14:44 +0000 2014"
            try:
                c = time.strptime(tweet_gmtime.replace("+0000", ''), '%a %b %d
%H:%M:%S %Y')
            except:
                print("pb with tweet_gmtime", tweet_gmtime, line)
                pass
            tweet_unixtime = int(time.mktime(c))
            # fout.write(line)
            fout.write(str(
                [tweet_unixtime, tweet_gmtime, tweet_id, text, hashtags, users, urls,
media_urls, nfollower,
                nfriends]) + "\n")
        except:
            # print "pb with tweet:", line
            # print sys.exc_info()[0], line
            pass
    file_timeordered_json_tweets.close()
    fout.close()

```

Code Screenshots :

bigdata project phase 1 - [C:\Users\rajin\Desktop\Masters subjects\Principles of big data\bigdata project phase 1] - ...newtrail.py - PyCharm Community Edition 2017.1.3

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help

bigdata project phase 1 newtrail.py

1 import codecs
2 from datetime import datetime
3 import json
4 # import requests
5 import os
6 import string
7 import sys
8 import time
9
10 def parse_json_tweet(line):
11     tweet = json.loads(line)
12     # print line
13     if tweet['lang'] != 'en':
14         # print "non-english tweet:", tweet['lang'], tweet
15         return ['', '', '', [], [], []]
16
17     date = tweet['created_at']
18     id = tweet['id']
19     nfollowers = tweet['user']['followers_count']
20     nfriends = tweet['user']['friends_count']
21
22     if 'retweeted_status' in tweet:
23         text = tweet['retweeted_status']['text']
24     else:
25         text = tweet['text']
26
27     hashtags = [hashtag['text'] for hashtag in tweet['entities']['hashtags']]
28     users = [user_mention['screen_name'] for user_mention in tweet['entities']['user_mentions']]
29     urls = [url['expanded_url'] for url in tweet['entities']['urls']]
```

Output Screenshot: Extracted URLs and Hashtags:

```
bigdata project phase 1 [C:\Users\rajin\Desktop\Masters subjects\Principles of big data\bigdata project phase 1] - ...xyz - PyCharm Community Edition 2017.1.3

File Edit View Navigate Code Refactor Run Tools VCS Window Help

bigdata project phase 1 xyz

The file size (6134522 bytes) exceeds configured limit (2560000 bytes). Code insight features are not available.

34 [], []
35 ['https://twitter.com/hookzilla/status/914140609891520512']]
36 ['https://twitter.com/i/web/status/914148047944261633']]
37 [], []
38 [['trending', 'mom', 'facebook', 'mother'], ['http://ift.tt/2kca2BH']]
39 [], []
40 [], []
41 ['https://www.nytimes.com/2017/09/28/upshot/the-high-price-of-failing-americas-costliest-patients.html']]
42 ['MAGA'], []
43 [], []
44 [['shopping', 'Amazon'], ['https://shop-picks.com/johnny-apple-peeler-by-victorio-vkp1010-cast-iron-suction-base/']]
45 ['https://twitter.com/i/web/status/914148048749678593']]
46 ['day', 'job', 'woman', 'mother', 'Career'], ['http://bit.ly/2wGXCXG']]
47 [], []
48 [], ['https://www.instagram.com/p/BZqzC3BhQ2V/']]
49 [], []
50 [], ['https://www.instagram.com/p/BZqzC3BhQ2V/']]
51 [], []
52 [], ['https://plus.google.com/+KevinGreenRockTheReTweet/posts/5pay41LFZbY']]
53 [['trending', 'india', 'facebook', 'donaldrump'], ['http://ift.tt/2x5buMg']]
54 ['https://www.reuters.com/article/us-usa-audit-army/u-s-army-fudged-its-accounts-by-trillions-of-dollars-auditor']]
55 ['https://twitter.com/i/web/status/91414805227268096011']]
```

Step 3: Running Word Count Program in Hadoop:

Screenshots of commands executed while running word count program:


```
hashtag2 (2).png - Photos
anki@anki-VirtualBox: /usr/local/hadoop
anki@anki-VirtualBox: /usr/local/hadoop$ jps
2866 ResourceManager
2313 NameNode
11561 Jps
2443 DataNode
2668 SecondaryNameNode
3968 NodeManager
anki@anki-VirtualBox: /usr/local/hadoop$ cd
anki@anki-VirtualBox: ~$ cd /home/anki/Desktop/
anki@anki-VirtualBox: ~/Desktop$ ls
bigdata  Bigdata  data  hashtags  hbase-1.2.6-bin.tar.gz  pig-0.16.0.tar.gz  Screenshot
anki@anki-VirtualBox: ~/Desktop$ cd hashtags
anki@anki-VirtualBox: ~/Desktop/hashtags$ ls
Popular hshtags and topics.txt
anki@anki-VirtualBox: ~/Desktop/hashtags$ cd
anki@anki-VirtualBox: ~$ cd /usr/local/hadoop
anki@anki-VirtualBox: /usr/local/hadoop$ bin/h
hadoop  hadoop.cmd  hdfs  hdfs.cmd
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hdfs dfs -mkdir /uri
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hdfs dfs -mkdir /uri/ym1
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hdfs -put /home/anki/Desktop/hashtags /uri/input
Unrecognized option: -put
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hdfs dfs -put /home/anki/Desktop/hashtags /uri/input
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount /uri/input out2
18/02/26 21:45:16 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
18/02/26 21:45:17 INFO Input.FileInputFormat: Total input files to process : 1
18/02/26 21:45:17 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1252)
    at java.lang.Thread.join(Thread.java:1326)
    at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:927)
    at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:578)
    at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:755)
18/02/26 21:45:17 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1252)
    at java.lang.Thread.join(Thread.java:1326)
    at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:927)
    at org.apache.hadoop.hdfs.DataStreamer.closeInternal(DataStreamer.java:792)
    at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:788)
18/02/26 21:45:17 INFO mapreduce.JobSubmitter: number of splits:1
18/02/26 21:45:17 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1519693454677_0004
18/02/26 21:45:17 INFO Impl.YarnClientImpl: Submitted application application_1519693454677_0004
18/02/26 21:45:18 INFO mapreduce.Job: The url to track the job: http://anki-VirtualBox:8088/proxy/application_1519693454677_0004/
18/02/26 21:45:18 INFO mapreduce.Job: Running job: job_1519693454677_0004
18/02/26 21:45:26 INFO mapreduce.Job: Job job_1519693454677_0004 running in uber mode : false
18/02/26 21:45:26 INFO mapreduce.Job: map 0% reduce 0%
```

```
anki@anki-VirtualBox: /usr/local/hadoop
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hdfs dfs -mkdir /uri
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hdfs dfs -mkdir /uri/ym1
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hdfs -put /home/anki/Desktop/hashtags /uri/input
Unrecognized option: -put
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hdfs dfs -put /home/anki/Desktop/hashtags /uri/input
anki@anki-VirtualBox: /usr/local/hadoop$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount /uri/input out2
18/02/26 21:45:16 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
18/02/26 21:45:17 INFO Input.FileInputFormat: Total input files to process : 1
18/02/26 21:45:17 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1252)
    at java.lang.Thread.join(Thread.java:1326)
    at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:927)
    at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:578)
    at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:755)
18/02/26 21:45:17 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1252)
    at java.lang.Thread.join(Thread.java:1326)
    at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:927)
    at org.apache.hadoop.hdfs.DataStreamer.closeInternal(DataStreamer.java:792)
    at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:788)
18/02/26 21:45:17 INFO mapreduce.JobSubmitter: number of splits:1
18/02/26 21:45:17 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1519693454677_0004
18/02/26 21:45:17 INFO Impl.YarnClientImpl: Submitted application application_1519693454677_0004
18/02/26 21:45:18 INFO mapreduce.Job: The url to track the job: https://anki-VirtualBox:8088/proxy/application_1519693454677_0004/
18/02/26 21:45:26 INFO mapreduce.Job: Running job: job_1519693454677_0004
18/02/26 21:45:26 INFO mapreduce.Job: Job job_1519693454677_0004 running in uber mode : false
18/02/26 21:45:32 INFO mapreduce.Job: map 100% reduce 0%
18/02/26 21:45:39 INFO mapreduce.Job: map 100% reduce 100%
18/02/26 21:45:39 INFO mapreduce.Job: Job job_1519693454677_0004 completed successfully
18/02/26 21:45:39 INFO mapreduce.Job: Counters: 49
File System Counters
  FILE: Number of bytes read=69135
  FILE: Number of bytes written=411149
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=50975
  HDFS: Number of bytes written=54605
  HDFS: Number of read operations=6
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
```

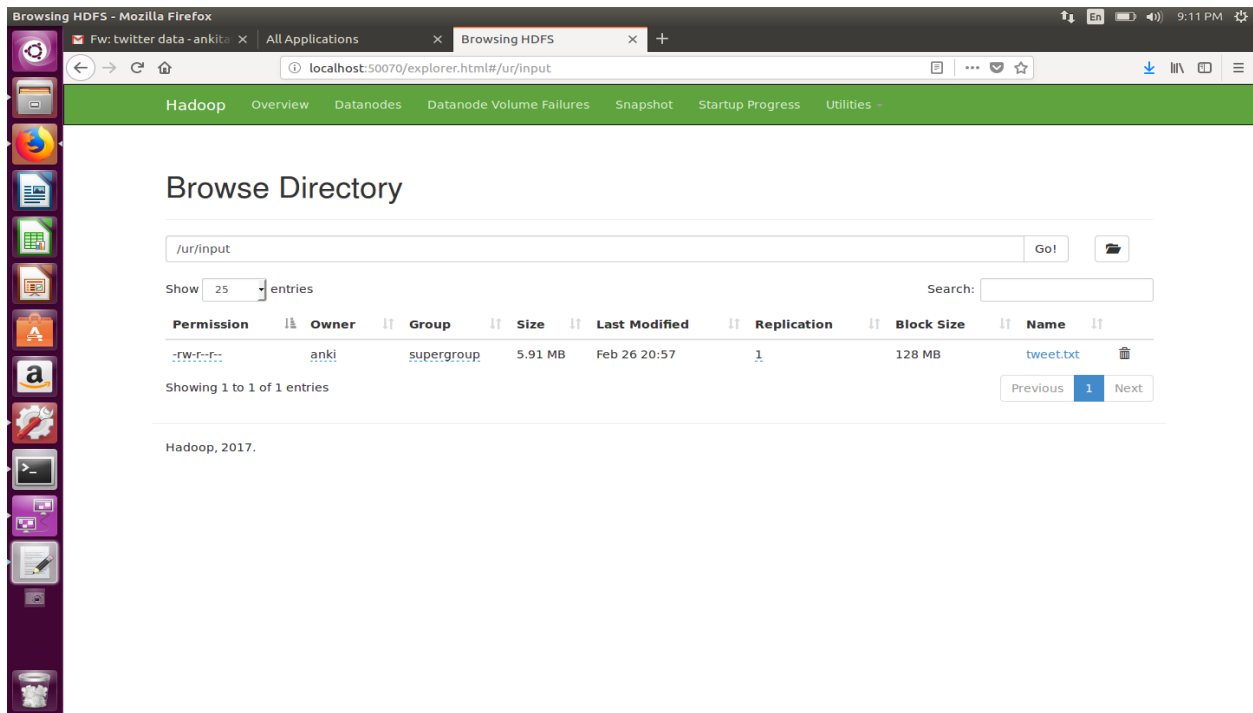
Word Count of URLs:

```
anki@anki-VirtualBox: /usr/local/hadoop
[ 'https://www.google.com/amp/amp.sacbee.com/opinion/california-forum/article175434686.html' ] 1
[ 'https://www.google.com/amp/amp.slate.com/articles/health_and_science/science/2017/09/puerto_rico_needs_long_term_support_not_just_short_term_aid.html' ] 2
[ 'https://www.google.com/amp/amp.thedailybeast.com/the-tine-donald-trump-turned-away-in-disgust-while-a-man-bled-to-death-in-front-of-him' ] 1
[ 'https://www.google.com/amp/amp.tineinc.net/fortune/2014/03/20/bill-clinton-on-leadership' ] 1
[ 'https://www.google.com/amp/amp.tineinc.net/fortune/2017/09/29/collin-kaepernick-nfl-protest' ] 1
[ 'https://www.google.com/amp/amp.tineinc.net/st/tech-media/2017/09/26/nfl-ratings-increase-donald-trump-comments-monday-night-football%3Fsource=dam' ] 1
[ 'https://www.google.com/amp/amp.tineinc.net/time/4666082/jfk-assassination-secrets%3Fsource=dam' ] 1
[ 'https://www.google.com/amp/amp.usatoday.com/story/470245001/' ] 1
[ 'https://www.google.com/amp/amp.usatoday.com/story/706196001/' ] 1
[ 'https://www.google.com/amp/amp.usatoday.com/story/716464001/' ] 1
[ 'https://www.google.com/amp/blog.organizer.com/what-is-deep-canvassing%3Fhs_amp=true' ] 1
[ 'https://www.google.com/amp/deadline.com/2017/09/donald-trump-twitter-puerto-rico-big-decisions-rebuilding-1202179269/amp/' ] 1
[ 'https://www.google.com/amp/denver.cbslocal.com/2017/09/16/rocky-mountain-arsenal-lawsuit/amp/' ] 1
[ 'https://www.google.com/amp/ew.com/gallery/stars-against-trump/amp/' ] 4
[ 'https://www.google.com/amp/fox2now.com/2017/09/29/illinois-high-school-football-team-honors-first-responders-while-taking-field/amp/' ] 1
[ 'https://www.google.com/amp/globalnation.inquirer.net/72279/philippines-a-jewish-refuge-from-the-holocaust/amp/' ] 1
[ 'https://www.google.com/amp/n.huffpost.com/us/entry/us_596cfc31e4b0376db8b659e1/amp/' ] 1
[ 'https://www.google.com/amp/mobile.reuters.com/article/amp/IdUSKNC1C435K' ] 1
[ 'https://www.google.com/amp/mobile.reuters.com/article/amp/IdUSKNC1C301H' ] 4
[ 'https://www.google.com/amp/nypost.com/2014/07/09/nayor-during-katrina-gets-10-years-for-corruption/amp/' ] 1
[ 'https://www.google.com/amp/nypost.com/2017/09/27/puerto-rico-should-have-been-ready-for-maria/amp/#ampshare=http://nypost.com/2017/09/27/puerto-rico-should-have-been-ready-for-maria/' ] 1
[ 'https://www.google.com/amp/nypost.com/2017/09/27/puerto-rico-should-have-been-ready-for-maria/amp/' ] 1
[ 'https://www.google.com/amp/people.com/politics/lin-manuel-miranda-donald-trump-straight-to-hell-puerto-rico/amp/' ] 1
[ 'https://www.google.com/amp/ptx11.com/2017/09/16/instagram-video-of-queens-students-singing-popular-song-with-racial-slur-causes-controversy/amp/' ] 1
[ 'https://www.google.com/amp/profootballtalk.nbcsports.com/2017/09/24/report-raiders-offensive-line-plans-to-kneel-as-a-group/amp/' ] 1
[ 'https://www.google.com/amp/s/amp.businessinsider.com/how-does-novlepass-nake-money-2017-8' ] 1
[ 'https://www.google.com/amp/s/amp.businessinsider.com/salary-after-taxes-us-cities-2017-9' ] 1
[ 'https://www.google.com/amp/s/amp.cnn.com/cnn/2016/04/29/politics/donald-trump-tweets-daniel-scavino/index.html' ] 1
[ 'https://www.google.com/amp/s/amp.cnn.com/cnn/2017/09/26/politics/saudi-arabia-woman-drive/index.html' ] 6
[ 'https://www.google.com/amp/s/amp.cnn.com/cnn/2017/09/27/us/puerto-rico-aid-problem/index.html' ] 3
[ 'https://www.google.com/amp/s/amp.cnn.com/cnn/2017/09/30/politics/trump-tweets-puerto-rico-mayor/index.html#ampshare=http://www.cnn.com/2017/09/30/politics/trump-tweets-puerto-rico-mayor/index.html' ] 1
[ 'https://www.google.com/amp/s/amp.cnn.com/cnn/2017/09/30/politics/trump-tweets-puerto-rico-mayor/index.html' ] 1
[ 'https://www.google.com/amp/s/amp.cnn.com/cnn/2017/09/30/us/puerto-rico-hurricane-recovery/index.html' ] 2
[ 'https://www.google.com/amp/s/amp.freep.com/amp/708496001#ampshare=http://www.freep.com/story/news/nation/2017/09/30/power-puerto-rico-detroit-solar-panel-hurricane-maria/708496001/' ] 1
[ 'https://www.google.com/amp/s/amp.theguardian.com/us-news/2016/sep/22/trump-ohio-campaign-chair-no-racism-before-obama' ] 1
[ 'https://www.google.com/amp/s/amp.theguardian.com/world/2010/sep/14/photographer-ernest-witthers-fbi-informer' ] 6
[ 'https://www.google.com/amp/s/amp.theguardian.com/world/2017/sep/29/puerto-rico-crisis-supply-food-water' ] 1
[ 'https://www.google.com/amp/s/blog.bulletproof.com/womens-health-post-birth-control-syndrome-brain-injuries-dr-jolene-brighten-415/amp' ] 1
[ 'https://www.google.com/amp/s/citizensagainstinequality.org/2017/07/07/congressman-chris-stewart-violates-u-s-c-title-18/amp/' ] 3
[ 'https://www.google.com/amp/s/drjengunter.wordpress.com/2016/04/12/check-your-privilege-and-your-facts-before-discussing-sex-selective-abortion/amp/' ] 3
[ 'https://www.google.com/amp/s/hiddenremote.com/2017/09/28/wednesday-tv-ratings-seal-team-enjoys-solid-premiere-leads-night-total-viewers/amp/' ] 1
[ 'https://www.google.com/amp/s/mobile.nytimes.com/2016/07/27/us/puerto-rico-debt-mayors.amp.html' ] 1
[ 'https://www.google.com/amp/s/mobile.nytimes.com/2017/09/27/nregion/mexico-puerto-rico-disasters-aid-new-york-city.amp.html' ] 1
[ 'https://www.google.com/amp/s/nep.247sports.com/Bolt/Rex-Burkhead-returns-to-New-England-Patriots-practice-on-Friday--108159943/amp' ] 1
```

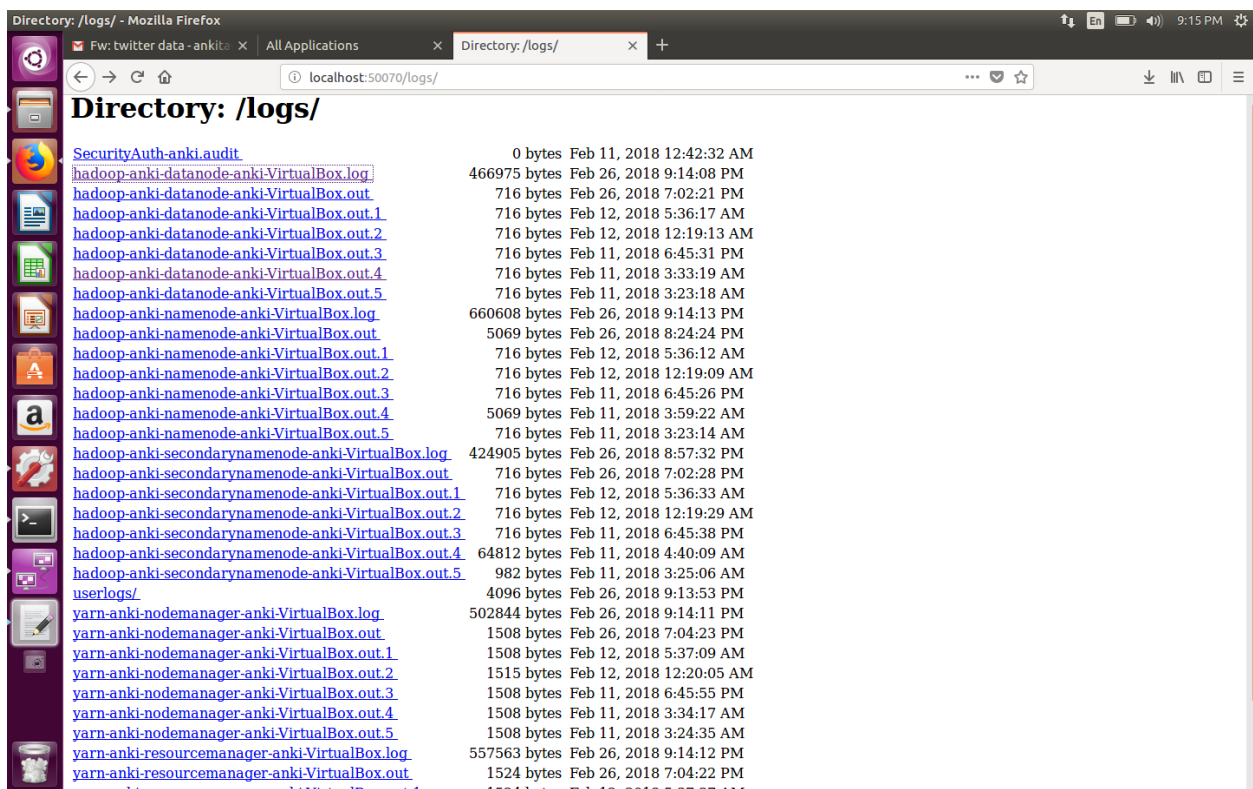
Word Count of HashTags:

```
anki@anki-VirtualBox: /usr/local/hadoop
[ ['RocknRollerCoaster'], 1
[ ['RodmanDisgraceful'], 1
[ ['Rohingya'], 7
[ ['Rohingya'], 3
[ ['RohingyaCrisis'], 1
[ ['RohingyaMuslims'], 5
[ ['RohingyaMuslims'], 4
[ ['RohingyaRefugees'], 9
[ ['RohingyaTerrorExposed'], 1
[ ['Rohingyas'], 8
[ ['Roku'], 1
[ ['Role'], 2
[ ['RollingStone'], 1
[ ['RollyVortex'], 1
[ ['RomanticSuspense'], 1
[ ['Rome'], 1
[ ['RoyMoore'], 1
[ ['RuPaulsDragRace'], 1
[ ['Ruan'], 1
[ ['RuckerFire'], 1
[ ['RussellWestbrook'], 8
[ ['Russia'], 8
[ ['Russia'], 4
[ ['Russian'], 3
[ ['Russian'], 2
[ ['RusstanBot'], 1
[ ['RusstanHacking'], 1
[ ['RyanGosling'], 1
[ ['RyderCup'], 1
[ ['SAIGON'], 2
[ ['SAN'], 1
[ ['SATC'], 1
[ ['SATC'], 1
[ ['SAVEJPMISHTOWN'], 2
[ ['SAVEDUCATION'], 1
[ ['SABAN'], 2
[ ['SBIR'], 1
[ ['SC'], 3
[ ['SCCS'], 1
[ ['SCD2017'], 1
[ ['SCOTUS'], 1
[ ['SCTop10'], 1
[ ['SDLive'], 1
[ ['SEAmayorLive'], 1
[ ['SEATraffic'], 1
[ ['SEC'], 2
[ ['SECHSKIES'], 2
[ ['SEO'], 14
[ ['SEO'], 5
[ ['SERP'], 1
[ ['SHIITE'], 1
```

HDFS Screenshot :



Screenshot of log files created:



Step 4: Running Word Count Program in Apache Spark:

Screenshots of commands executed while running word count program:

1. Starting Spark: using bin/spark-shell command:

```
anki@anki-VirtualBox: /usr/local/hadoop$ jps
4323 ResourceManager
11365 Jps
3799 NameNode
4153 SecondaryNameNode
3929 DataNode
4447 NodeManager
anki@anki-VirtualBox: /usr/local/hadoop$ cd
anki@anki-VirtualBox: ~$ cd /home/anki/spark-2.2.1-bin-hadoop2.7/
anki@anki-VirtualBox: ~/spark-2.2.1-bin-hadoop2.7$ bin/spark-
spark-class spark-shell spark-sql spark-submit
anki@anki-VirtualBox: ~/spark-2.2.1-bin-hadoop2.7$ bin/spark-s
spark-shell spark-sql spark-submit
anki@anki-VirtualBox: ~/spark-2.2.1-bin-hadoop2.7$ bin/spark-shell
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
18/02/27 22:37:59 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/02/27 22:37:59 WARN Utils: Your hostname, anki-VirtualBox resolves to a loopback address: 127.0.1.1; using 192.168.0.18 instead (on interface enp0s3)
18/02/27 22:37:59 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Spark context Web UI available at http://192.168.0.18:4040
Spark context available as 'sc' (master = local[*], app id = local-1519792681089).
Spark session available as 'spark'.
Welcome to

  ____              __
 / ___|  _ \  _ __ | | | |
 \___ \ |_) | | '_ \| |_| |
  ___) |  __/| | |_) |  __/
 |____|_|___|_|_|___|_|___|
                                version 2.2.1

Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0_151)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val textFile = sc.textFile("/home/anki/Desktop/hashtags/tweets")
textFile: org.apache.spark.rdd.RDD[String] = /home/anki/Desktop/hashtags/tweets MapPartitionsRDD[1] at textFile at <console>:24

scala> val textFile = sc.textFile("/home/anki/Desktop/hashtags")
textFile: org.apache.spark.rdd.RDD[String] = /home/anki/Desktop/hashtags MapPartitionsRDD[3] at textFile at <console>:24

scala> val textFile = sc.textFile("/home/anki/Desktop/hashtags/tweets.txt")
textFile: org.apache.spark.rdd.RDD[String] = /home/anki/Desktop/hashtags/tweets.txt MapPartitionsRDD[5] at textFile at <console>:24

scala> val text = sc.textFile("tweets.txt")
text: org.apache.spark.rdd.RDD[String] = tweets.txt MapPartitionsRDD[7] at textFile at <console>:24

scala> pwd
<console>:24: error: not found: value pwd
pwd
^
```

2. Loading the data:

```
anki@anki-VirtualBox: ~/spark-2.2.1-bin-hadoop2.7
scala> val textFile = sc.textFile("/home/anki/tweets")
textFile: org.apache.spark.rdd.RDD[String] = /home/anki/tweets MapPartitionsRDD[16] at textFile at <console>:24

scala> textFile
res2: org.apache.spark.rdd.RDD[String] = /home/anki/tweets MapPartitionsRDD[16] at textFile at <console>:24

scala> textFile.foreach(println)
org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: file:/home/anki/tweets
    at org.apache.hadoop.mapred.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:287)
    at org.apache.hadoop.mapred.FileInputFormat.listStatus(FileInputFormat.java:229)
    at org.apache.hadoop.mapred.FileInputFormat.getSplits(FileInputFormat.java:315)
    at org.apache.spark.rdd.HadoopRDD.getPartitions(HadoopRDD.scala:199)
    at org.apache.spark.rdd.RDD$anonfun$partitions$2.apply(RDD.scala:252)
    at org.apache.spark.rdd.RDD$anonfun$partitions$2.apply(RDD.scala:250)
    at scala.Option.getOrElse(Option.scala:121)
    at org.apache.spark.rdd.RDD.partitions(RDD.scala:250)
    at org.apache.spark.rdd.MapPartitionsRDD.getPartitions(MapPartitionsRDD.scala:35)
    at org.apache.spark.rdd.RDD$anonfun$partitions$2.apply(RDD.scala:252)
    at org.apache.spark.rdd.RDD$anonfun$partitions$2.apply(RDD.scala:250)
    at scala.Option.getOrElse(Option.scala:121)
    at org.apache.spark.rdd.RDD.partitions(RDD.scala:250)
    at org.apache.spark.SparkContext.runJob(SparkContext.scala:2094)
    at org.apache.spark.rdd.RDD$anonfun$foreach$1.apply(RDD.scala:918)
    at org.apache.spark.rdd.RDD$anonfun$foreach$1.apply(RDD.scala:916)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
    at org.apache.spark.rdd.RDD.withScope(RDD.scala:362)
    at org.apache.spark.rdd.RDD.foreach(RDD.scala:916)
    ... 48 elided

scala> val textFile = sc.textFile("/home/anki/tweets")
textFile: org.apache.spark.rdd.RDD[String] = /home/anki/tweets MapPartitionsRDD[18] at textFile at <console>:24

scala> textFile
res4: org.apache.spark.rdd.RDD[String] = /home/anki/tweets MapPartitionsRDD[18] at textFile at <console>:24

scala> textFile.foreach(println)
#100JustinBieber
#11septembre
#11septembre
#11septembre
11 YEARS OF SPN
#14Sep
#15deSeptiembre
#150stopCETA
#15Sep
#15TennuzUNUTULMAZ
#16deSeptiembre
#16HoopClass
#16Sep
```

3. Splitting content in our file as strings and applying map and reduce functions to perform word count, using the below highlighted command:

```
anki@anki-VirtualBox: ~/spark-2.2.1-bin-hadoop2.7
#ZFactorHindi
Z Festival
#ZlmanGoesPink
#ZNation
#ZunaQanda
#ZuritaEnFNOMexico

scala> val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_+_ )
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[21] at reduceByKey at <console>:26

scala> counts
res0: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[21] at reduceByKey at <console>:26

scala> counts.foreach(println)
(Ismail,1)
(#CauchemarEnCuisine,1)
(#ChavezFuerzaInternacional,1)
(Shawty,1)
(Nawaz,1)
(Alexis,1)
(Leverkusen,1)
(#BrusselsDL,1)
(#ALDUB14thMonthsary,1)
(#Benoyplyin,1)
(#TengoGanasDe,1)
(Verratti,1)
(SnapchattenEkle,1)
(#SextaDetremuraSDV,1)
(#SeQueaContigoPero,1)
(Junction,1)
(#CLAPTrabajoYPueblo,1)
(#GalaGemeaux,1)
(#FeedDaBears,1)
(Polaco,1)
(#CycleToWorkDay,1)
(#AmanezcoConGanasDeTirar,1)
(#VampsEspana,1)
(Punjab,1)
(#ARCV,1)
(#DOTSMovieDate,1)
(#ResteadosConSnolansky,1)
(Cara,1)
(#SoleLacreNoKpop,1)
(#OMOL,1)
(IS,3)
(#HappyNamjoonDay,1)
(#SocialMediaGetsMe,1)
(#NoPuedoTengoCrushPero,1)
(#JeuTPMP,1)
(#lastnightoftheproms,1)
(#AskBeckyG,1)
```

4. Finally collecting output which displays the word count in the form of an array of key value pairs:

```
anki@an: Screenshot
(#BurasiBodrum,1)
(#billa,1)
(#PurposeTourParisDans1jour,1)
(#25sEnLaSER,1)
(#15DeSeptiembre,1)
(#Roma2024,1)
(ChInbnha,1)
(#adamlookslike,1)
(#Huskies,1)
(#MisionEnCombate,1)
(#Free_Xios,1)
(#SkittlesWelcome,1)
(#HappyBirthdayTylerHoechlin,1)
(#SABHelNaglin,1)
(#NuevosWitosAlimentos,1)
(#DrunkPopQuiz,1)
(#TeMeAntojasEn,1)
(#LaNocheDeML,1)
(#FutureFest16,1)
(SHOXX,1)
(#dnartedl,1)
(#Lanz,1)
(#GuessingTylersProject,1)
(Plantao,1)
(Stranding,1)
(#PBB3rdEvict,1)
(#PropuestasQueSeAntojan,1)
(Ruben,3)
(#ALDUBHereToStay,1)
(20x,1)
(Branquelas,1)
(#JuanYMarioEnLosKCA,1)
(#PreciosaDecepcion,1)
(Gonzalez,1)
(#DancingKing,1)
(#HuesliPower,1)
(#MARTES13,1)
(#MarDePlastico2,1)
(Makau,1)
(#GetSmartWithARQ,1)
(Sturridge,1)
(#JEP2016,1)

scala> counts.collect()
res0: Array[(String, Int)] = Array((Ismail,1), (#CauchemarEnCuisine,1), (#ChavezFuerzaInternacional,1), (Shawty,1), (Nawaz,1), (Alexis,1), (Leverkusen,1), (#BrusselsDL,1), (#ALDUB14thMonthsary,1), (#Benoyplyin,1), (#TengoGanasDe,1), (Verratti,1), (SnapchattenEkle,1), (#SextaDetremuraSDV,1), (#SeQueaContigoPero,1), (Junction,1), (#CLAPTrabajoYPueblo,1), (#GalaGemeaux,1), (#FeedDaBears,1), (Polaco,1), (#CycleToWorkDay,1), (#AmanezcoConGanasDeTirar,1), (#VampsEspana,1), (Punjab,1), (#ARCV,1), (#DOTSMovieDate,1), (#ResteadosConSnolansky,1), (Cara,1), (#SoleLacreNoKpop,1), (#OMOL,1), (IS,3), (#HappyNamjoonDay,1), (#SocialMediaGetsMe,1), (#NoPuedoTengoCrushPero,1), (#JeuTPMP,1), (#lastnightoftheproms,1), (#AskBeckyG,1), (Alario,1), (#BarcelonaGlobalPreniere,1), (Norte,1), (#JeSuisPasUnGeekMa...

scala>
```

References:

<http://adilmoujahid.com/posts/2014/07/twitter-analytics/>

<https://github.com/heerme/twitter-topics/blob/master/extract-json-to-text-stream.py>
<https://www.youtube.com/watch?v=YZnNb0BTrS4&list=PLJNKKS4iwuamrvNVahopRziurK7XNCc5B&index=3>