

PRINCIPLES OF BIG DATA MANAGEMENT

PHASE #1

TEAM MEMBERS:

Gopi Chand Bodepudi (gbhmf@mail.umkc.edu) - 16278459
Vasudeva Madala (vmcpq@mail.umkc.edu) - 16280569
Anusha Palla (apgmc@mail.umkc.edu) - 16280777

Github Link of the Project:

<https://github.com/gbhmf/Principles-of-BigData-Manangement>

Objective:

- The principle point of this stage is to develop a system to store, analyze, and visualize Twitter's tweets

.

• Tasks:

1. Collect Tweets using Twitter's Streaming APIs in any format

preferred JSON.

2. Extract all the hashtags and URLs in the tweets and store the content substance (e.g. tweet's content) from the information into a document in HDFS.
3. Run a Word Count program in Apache Spark and Hadoop on the content document and collect the output and log files from Hadoop

Applications/Software's Used:

Twitter Developer Account, Apache Spark, Python, Java Eclipse, Hadoop.

Collecting tweets from Twitter:

- Firstly, we have made an developer account in Twitter utilizing beneath connect. <https://apps.twitter.com/>
- Below are the factors that contains the client certifications to get to Twitter API

```
ACCESS_TOKEN = "2219941182-  
hJEd5re1y7lbZmVlyZySZvVsJf88fP6um3SsC3r"
```

```
ACCESS_SECRET = "  
BntHym97rzCisKS3BFXqrBgQbgokklZEBcqHXixGJQtX8 "
```

```
CONSUMER_KEY = " 187ztf3hxmT3Nm3YonFzcAvEB "
```

```
CONSUMER_SECRET =
```

```
"hTqPaSjNXw21GXmPCey6CZBCZRoO1EbTkbVO4zMv77kN8lkq0P "
```

We have composed python program that is utilized to bring tweets in JSON design.

([twitter.py](#))

Link: [https://github.com/gbhmh/Principles-of-BigData-Manangement/tree/master/phase1/source/python programs](https://github.com/gbhmh/Principles-of-BigData-Manangement/tree/master/phase1/source/python%20programs)

```
*Python 2.7.14rc1 Shell*
File Edit Shell Debug Options Window Help
_video\/1098468557850066944\/pu\/vid\/640x360\/mN26QT0zLauKPRLm.mp4?tag=6"}, {"bi
trate":2176000,"content_type":"video\/mp4","url":"https:\/\/video.twimg.com\/ext
_tw_video\/1098468557850066944\/pu\/vid\/1280x720\/AYs1IxVRCPCMD2iI.mp4?tag=6"},
{"content_type":"application\/x-mpegURL","url":"https:\/\/video.twimg.com\/ext_t
w_video\/1098468557850066944\/pu\/pl\/8RRe5V3CQQsPXI-Y.m3u8?tag=6"}]], "sizes":{"
thumb":{"w":150,"h":150,"resize":"crop"},"medium":{"w":1200,"h":675,"resize":"fi
t"},"small":{"w":680,"h":383,"resize":"fit"},"large":{"w":1280,"h":720,"resize":
"fit"}}}}, "extended_entities":{"media":[{"id":1098468557850066944,"id_str":"109
8468557850066944","indices":[230,253],"additional_media_info":{"monetizable":fal
se},"media_url":"http:\/\/pbs.twimg.com\/ext_tw_video_thumb\/1098468557850066944
\/pu\/img\/p9rHh1TQ7658IQvH.jpg","media_url_https":"https:\/\/pbs.twimg.com\/ext
_tw_video_thumb\/1098468557850066944\/pu\/img\/p9rHh1TQ7658IQvH.jpg","url":"http
s:\/\/t.co\/ZGH2Xbj5E8","display_url":"pic.twitter.com\/ZGH2Xbj5E8","expanded_ur
l":"https:\/\/twitter.com\/RealALEXAN\/status\/1098468692780859393\/video\/1","t
ype":"video","video_info":{"aspect_ratio":[16,9],"duration_millis":31132,"varian
ts":[{"bitrate":256000,"content_type":"video\/mp4","url":"https:\/\/video.twimg.
com\/ext_tw_video\/1098468557850066944\/pu\/vid\/320x180\/ylucW5fVUXikFCiJ.mp4?t
ag=6"}, {"bitrate":832000,"content_type":"video\/mp4","url":"https:\/\/video.twim
g.com\/ext_tw_video\/1098468557850066944\/pu\/vid\/640x360\/mN26QT0zLauKPRLm.mp4
?tag=6"}, {"bitrate":2176000,"content_type":"video\/mp4","url":"https:\/\/video.t
wimg.com\/ext_tw_video\/1098468557850066944\/pu\/vid\/1280x720\/AYs1IxVRCPCMD2iI
.mp4?tag=6"}, {"content_type":"application\/x-mpegURL","url":"https:\/\/video.twi
mg.com\/ext_tw_video\/1098468557850066944\/pu\/pl\/8RRe5V3CQQsPXI-Y.m3u8?tag=6"}
]], "sizes":{"thumb":{"w":150,"h":150,"resize":"crop"},"medium":{"w":1200,"h":675
,"resize":"fit"},"small":{"w":680,"h":383,"resize":"fit"},"large":{"w":1280,"h":
720,"resize":"fit"}}}}, "quote_count":1738,"reply_count":455,"retweet_count":22
0,"favorite_count":783,"entities":{"hashtags":[],"urls":[{"url":"https:\/\/t.co\/
ORkJmlaIbg","expanded_url":"https:\/\/twitter.com\/i\/web\/status\/109846869278
0859393","display_url":"twitter.com\/i\/web\/status\/1\u2026","indices":[117,140
]}],"user_mentions":[],"symbols":[]},"favorited":false,"retweeted":false,"possib
ly_sensitive":false,"filter_level":"low","lang":"en"},"quoted_status_permalink":
{"url":"https:\/\/t.co\/LvFgNAVF3H","expanded":"https:\/\/twitter.com\/realalexa
n\/status\/1098468692780859393","display":"twitter.com\/realalexa\/sta\u2026"},
"is_quote_status":true,"quote_count":0,"reply_count":0,"retweet_count":0,"favori
te_count":0,"entities":{"hashtags":[],"urls":[],"user_mentions":[{"screen_name":
"Julathomas","name":"Jayla Thomas","id":707634951873699841,"id_str":"70763495187
3699841","indices":[3,14]}],"symbols":[]},"favorited":false,"retweeted":false,"f
```

- The extricated record in JSON arrange contains all the tweet points of interest, for example, id, created at, text, profile_background_image_url and so forth.

- From JSON tweets record just the hashtags, url's content substance is extricated utilizing Python programs.

([twitter1.py](#) and [twitter2.py](#))

Link : <https://github.com/gbhmh/Principles-of-BigData->

Manangement/tree/master/phase1/source/python programs

. The extracted files are at :

<https://github.com/gbhmh/Principles-of-BigData->

Manangement/tree/master/phase1/source/twitter files

Store the extracted files from the data into a file in HDFS.

```
C:\WINDOWS\system32>jps
1924 NameNode
3620 NodeManager
3908 SparkSubmit
4024 DataNode
5992 ResourceManager
9948 Jps
C:\WINDOWS\system32>
```

- The data is moved from local to HDFS.
- First a folder is made in HDFS and the document is moved from local to HDFS utilizing underneath order.

Make directory in local: `hdfs dfs -mkdir -p /hadout`

Move file from local to HDFS: `hdfs dfs -put`

`C:\Python27\FileOutput_hash1.txt / hadout / hadout.txt`

Now we have run Wordcount program using the jar files that are obtained from Madreduce program written in Eclipse IDE using Java

Command : `hadoop jar C:/Users/purnak/Desktop/jab/javjar.jar`

WordCount /hadout /outpath

```
C:\WINDOWS\system32>hadoop jar C:/Users/purnak/Desktop/jab/javjar.jar WordCount

```

```

Total Megabyte Milliseconds taken by all reduce tasks 7733120
Map-Reduce Framework
  Map input records=2
  Map output records=6
  Map output bytes=57
  Map output materialized bytes=29
  Input split bytes=98
  Combine input records=6
  Combine output records=2
  Reduce input groups=2
  Reduce shuffle bytes=29
  Reduce input records=2
  Reduce output records=2
  Spilled Records=4
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=229
  CPU time spent (ms)=3872
  Physical memory (bytes) snapshot=412385280
  Virtual memory (bytes) snapshot=537276416
  Total committed heap usage (bytes)=306184192
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=33
File Output Format Counters
  Bytes Written=15

```

C:\WINDOWS\system32>

once run is successfull u can see the two files in output

command: `hdfs dfs -ls /outputh -->` Here two files will be displayed.

```

C:\WINDOWS\system32>hdfs dfs -ls /outputh
Found 2 items
-rw-r--r--  1 purnak supergroup          0 2019-02-22 09:12 /outputh/_SUCCESS
-rw-r--r--  1 purnak supergroup    238034 2019-02-22 09:12 /outputh/part-r-0000
00

```

Finally the wordcount output will watch by using the following command.

command: `hadoop fs -cat /outputh/part-r-00000`

```
Administrator: Command Prompt
u'https://twitter.com/i/web/status/1098495378188578816' 1
u'https://twitter.com/i/web/status/1098495435080003584' 1
u'https://twitter.com/i/web/status/1098495826077327360' 1
u'https://twitter.com/i/web/status/1098495843877834752' 1
u'https://twitter.com/i/web/status/1098495885690851333' 1
u'https://twitter.com/i/web/status/1098495890753572864' 1
u'https://twitter.com/i/web/status/1098495890858348544' 1
u'https://twitter.com/i/web/status/1098496555890429953' 1
u'https://twitter.com/seanhannity/status/1098330796648263680' 1
u'https://www.cnn.co.jp/fringe/35133092.html' 1
u'https://www.kyoritsu-pub.co.jp/bookdetail/9784320124431' 1
u'https://youtu.be/GNBZ92xDjUw' 1
u'https://youtu.be/_lY0tYqnEfs' 1
u'https://youtu.be/yCHr5cSEWbg' 1
u'iHeartAwards' 5
u'iHeartAwards' 1 91
u'iHeartSweepstakes' 1
u'iaaworldcongress' 1
u'identitypolitics' 2
u'indiedev' 1
u'indiegama' 1
u'influenza' 1
u'infographic' 1
u'infographic' 1 1
u'information' 1
u'infosec' 6
u'infosec' 1
u'innovaci\xf3n' 1
u'internmentcamp' 1
```

Copy output to local space

```
Command: hdfs dfs -copyToLocal /outputh/part-r-00000 C
:/Users/purnak/Desktop/jabout
```

```
C:\WINDOWS\system32>hdfs dfs -copyToLocal /outputh/part-r-00000 C:/Users/purnak/
Desktop/jabout
```


← → ↻ localhost:50070/explorer.html#

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	purnak	supergroup	0 B	22/02/2019, 09:07:59	0	0 B	hadout
drwxr-xr-x	purnak	supergroup	0 B	22/02/2019, 02:27:05	0	0 B	input
drwxr-xr-x	purnak	supergroup	0 B	22/02/2019, 02:28:32	0	0 B	out
drwxr-xr-x	purnak	supergroup	0 B	22/02/2019, 09:12:26	0	0 B	outputpath
drwx-----	purnak	supergroup	0 B	22/02/2019, 02:04:12	0	0 B	tmp

Hadoop, 2015.



Please find the output files link for Hadoop:

[https://github.com/gbhmh/Principles-of-BigData-Manangement/tree/master/phase1/source/wordcount output/Apache Hadoop](https://github.com/gbhmh/Principles-of-BigData-Manangement/tree/master/phase1/source/wordcount%20output/Apache%20Hadoop)

Run a Word Count program in Apache Spark on the extracted file

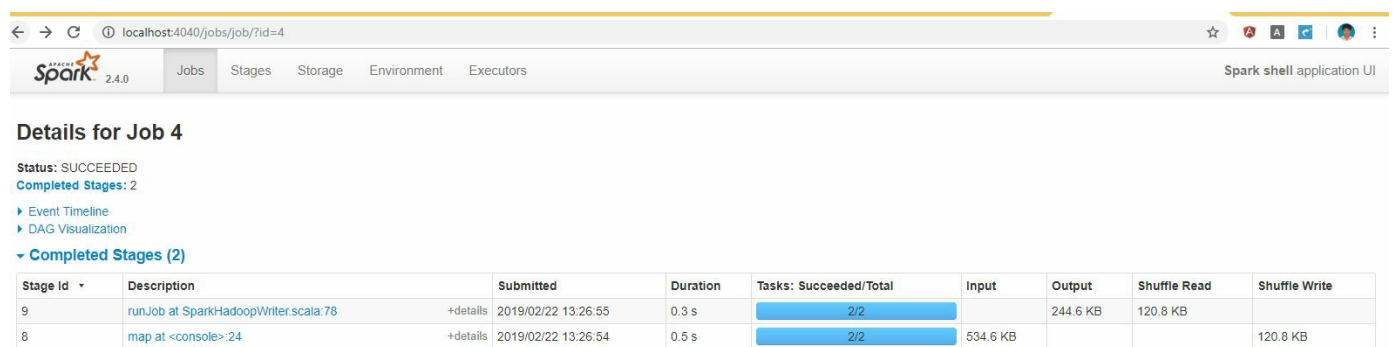
- Then, after running the word count example on Hadoop, now it's time to run the same word count example using Apache Spark.
- The output obtained from the word count running on Apache Hadoop is almost similar to the output obtained from Apache Spark except the minor differences.

The command used for to run wordcount in scala is :

```
scala> val file =sc.textFile("test5/FileOutput_hash1.txt").flatMap(_.split(" "))
map(word => (word,1)).reduceByKey(_+_).saveAsTextFile("C:\\Users\\purnak\\test7
")
[Stage 8:>                                     <0 + 2> / 21
file: Unit = <>
```

Output files are stored in local system after the execution of the above command

Spark directories are



Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
9	runJob at SparkHadoopWriter.scala:78	2019/02/22 13:26:55	0.3 s	2/2		244.6 KB	120.8 KB	
8	map at <console>:24	2019/02/22 13:26:54	0.5 s	2/2	534.6 KB			120.8 KB

[https://github.com/gbhmh/Principles-of-BigData-Manangement/tree/master/phase1/source/wordcount output/Spark](https://github.com/gbhmh/Principles-of-BigData-Manangement/tree/master/phase1/source/wordcount%20output/Spark)