MIS 586 Assignment 4
Social Media Data Analysis using Spark Streaming
Due Date: Mar 30, 2020 by 11:59pm

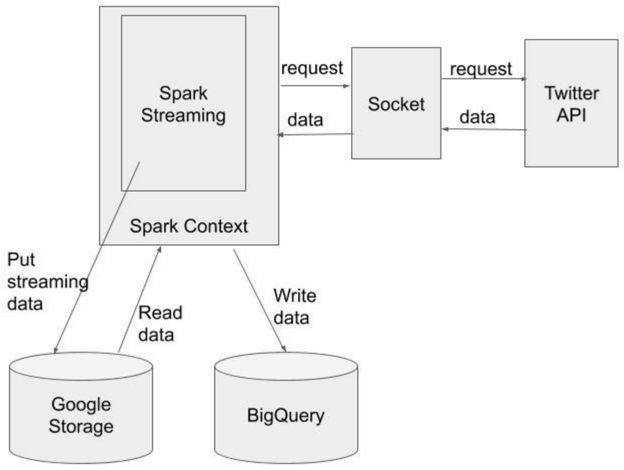
1 Objectives

- Do twitter streaming analysis using Spark Streaming.
- Count hashtags and special words.
- Submit your codes and report to D2L.

2 Overview

In this assignment, you will implement a streaming analysis process. The architecture is as follows. A socket requests data from twitter API and sends data to the spark streaming process. Spark reads real-time data and conducts analysis. It also saves temp streaming results to Google Storage. After the streaming process terminates, it reads the final data from Google Storage and save it to BigQuery, and then clean the data in Storage (see Figure 1).

Figure 1: Twitter streaming architecture



You will be asked to do two analysis tasks based on the tweets you obtain. Also, please figure out how to save temp steaming results to google storage.

3 Tasks

(a) Calculate the accumulated hashtags count sum for 600 seconds and sort it by descending order of the count. Hashtag usually starts with "#" and followed by a series of alphanumeric. **(30 pts)**

(b) Filter the chosen 5 words and calculate the appearance frequency of them every 60 seconds (no overlap). Collect the data for 600 seconds. The words are: 'data', 'spark', 'ai', 'movie', 'coronavirus'. **(30 pts)**

(c) Save results to google BigQuery. You only have to write code to save temp steaming results to google storage. **(20 pts)**

4 Some important notes

(1) Remember to start the socket first (run twitterHTTPClient.py) and then the streaming process (run sparkStreaming.py).
(2) You **don't need to restart the socket process** every time you rerun the streaming process. When a streaming process stops, the connection socket will automatically close. And the listening socket will continue listening to the same IP and Port and wait for the next connection. That is, you can leave twitterHTTPClient.py running and submit sparkStreaming.py multiple times.
(3) Remember to stop the socket program on the cluster when you don't want to stream data from it.
(4) You will get "port already in use" error when you run a socket program multiple times in a short time period, or you try to run multiple socket programs simultaneously.
(5) For task (b), it is fine if you don't see all of the words in your results every time.
(6) For task (c), create a table first before saving results to it.
(7) You can stop your cluster instance in Computer Engine page and keep your cluster. If you stop your instance, you won't pay too much for having that cluster.
(8) You can use Jupyter Notebook, just copy and paste the code. But **remember to create two different notebooks** and paste the code of two files in separate notebooks. The steps to run the code is the same as you submit the file to dataproc.

5 Steps

5.1 Step1: Register on Twitter Apps (Please do this step ASAP)

(1) Go to https://developer.twitter.com/en/apply-for-access.html and apply for a twitter developer account.
(2) Login at https://apps.twitter.com/
(3) Click "Create New App", fill out the form, and click "Create your Twitter application". For the Website URL, you can put any valid URL here. You only need to fill in all the required places.
(4) In the next page, click on "Keys and tokens" tab, and copy your "API key" and "API secret".
(5) Scroll down and click "Generate" Access token & access token secret, copy your "Access token" and "Access token secret".

5.2 Step2: Create a cluster and get streaming data

(1) Use the following command to create a cluster.
```
gcloud beta dataproc clusters create <cluster-name>
--region <region-name>
--project <ProjectID> --bucket <Bucket>
--optional-components=ANACONDA,JUPYTER
--image-version=preview
--enable-component-gateway
--metadata "PIP_PACKAGES=requests_oauthlib google-cloud-bigquery tweepy"
--metadata gcs-connector-version=1.9.16
--metadata bigquery-connector-version=0.13.16
```

```
--initialization-actions=gs://dataproc-initialization-actions/python/
pip-install.sh,gs://dataproc-initialization-actions/connectors/
connectors.sh  --single-node
```

Note: If you are using MacOS, use 'PIP_PACKAGES=...' (i.e., single quotation) instead of "PIP_PACKAGES=..." to avoid an error.

(2) Download start code from D2L.

(3) Open twitterHTTPClient.py Replace the Twitter API credential with your own and run on dataproc. Remember to stop the job after you finish streaming.

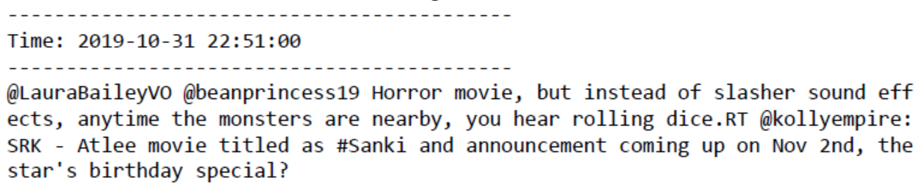
(4) Open sparkStreaming.py. You can first comment code from line 190 (words = dataStream.flatMap...) to line 198 (wordCount.pprint()) and from line 229 (saveToBigQuery...) to line 230 (saveToBigQuery...). And then run it on dataproc.

(5) To stream the data:

   a) submit twitterHTTPClient.py. (Jupyter Notebook: run the related cells in one notebook). Once you see "Waiting for TCP connection..." in the output window, go to the next step.
   b) submit sparkStreaming.py. (Jupyter Notebook: run the related cells in another notebook)
   c) You can test sparkStreaming.py multiple times and leave twitterHTTPClient.py running *Note:* If you face with an error "ValueError: Cannot run multiple SparkContexts at once...", then run `sc.stop()` and try again.
   d) Stop twitterHTTPClient.py. You can go to job page of cluster and cancel it. Or you can use `gcloud dataproc jobs kill JOB`. (Jupyter Notebook: stop the cell)

(6) You are expected to see a tweet stream printed out like Figure 2. Ignore all warnings while running.

Figure 2:



```
-------------------------------------------
Time: 2019-10-31 22:51:00
-------------------------------------------
@LauraBaileyVO @beanprincess19 Horror movie, but instead of slasher sound eff
ects, anytime the monsters are nearby, you hear rolling dice.RT @kollyempire:
SRK - Atlee movie titled as #Sanki and announcement coming up on Nov 2nd, the
star's birthday special?
```

5.3 Step3: Analyse Stream data and store data in BigQuery

(a) Once you successfully get the stream, you can start writing your own code. The analytics tasks are listed before (the bottom of page 1).

(b) You can first change the global variable "STREAMTIME" to a smaller value when testing your algorithm. And then collect the data for 600 seconds.

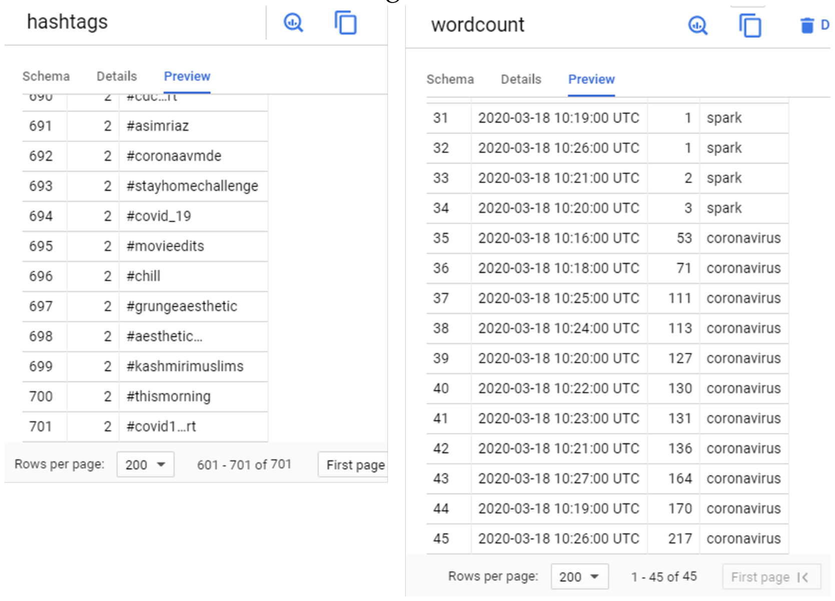
(c) You can first set your streaming window size to a smaller value (less than 60 seconds). After you make sure your code is correct and then you can change it back to 60 seconds and start collecting the data.

(d) To save the data, remember to:

   a) Replace the bucket global variable in the code with your own bucket.
   b) Create a BigQuery dataset first using bq mk <your dataset name> in Google Cloud SDK Shell or any alternative approaches.

(e) You are expected to see the following tables created in your dataset (Figure 3).

Figure 3:



690	2	#cuc…rt
691	2	#asimriaz
692	2	#coronaavmde
693	2	#stayhomechallenge
694	2	#covid_19
695	2	#movieedits
696	2	#chill
697	2	#grungeaesthetic
698	2	#aesthetic…
699	2	#kashmirimuslims
700	2	#thismorning
701	2	#covid1…rt

31	2020-03-18 10:19:00 UTC	1	spark
32	2020-03-18 10:26:00 UTC	1	spark
33	2020-03-18 10:21:00 UTC	2	spark
34	2020-03-18 10:20:00 UTC	3	spark
35	2020-03-18 10:16:00 UTC	53	coronavirus
36	2020-03-18 10:18:00 UTC	71	coronavirus
37	2020-03-18 10:25:00 UTC	111	coronavirus
38	2020-03-18 10:24:00 UTC	113	coronavirus
39	2020-03-18 10:20:00 UTC	127	coronavirus
40	2020-03-18 10:22:00 UTC	130	coronavirus
41	2020-03-18 10:23:00 UTC	131	coronavirus
42	2020-03-18 10:21:00 UTC	136	coronavirus
43	2020-03-18 10:27:00 UTC	164	coronavirus
44	2020-03-18 10:19:00 UTC	170	coronavirus
45	2020-03-18 10:26:00 UTC	217	coronavirus

Assignment Submissions

1) A report includes:

   a) Screenshot of your code to do all the Tasks (see the bottom of page 1).
   b) Screenshot of the preview of your data stored in BigQuery. You have to include two tables: *hashtags* and *wordcount*.

2) Your source code.

Useful links:
https://spark.apache.org/docs/latest/streaming-programming-guide.html
https://docs.python.org/3/library/re.html