

EECS E6893 Big Data Analytic HW0

Chong Hu ch3467

September 20, 2019

Problem 1. Warm-up exercises

- ## 1. Screenshots

Initialize the cluster using command line:

```
juchong@juchong:~/juchong/cloudai/EECS_F6993_Big_DataAnalytics$ gcloud beta dataproc clusters create big-data --optional-components=ANACONDA,JUPYTER --image-version=1.3 --enable-component-gateway --bucket big-data --single-node --metadata "PIP_PACKAGES=graphframes==0.6" --initialization-actions gs://dataproc-initialization-actions/python/pip-install.sh
Waiting on operation [projects/hardy-symbol-252200/regions/us-east4/operations/d1d6d9b0-cb99-313b-af6e-e3b81cd68b42].
Waiting for cluster creation operation...
WARNING: For PD-Standard without local SSDs, we strongly recommend provisioning 1TB or larger to ensure consistently high I/O performance. See https://cloud.google.com/compute/docs/disks/performance for information on disk I/O performance.
Waiting for cluster creation operation...done.
Created [https://dataproc.googleapis.com/v1beta2/projects/hardy-symbol-252200/regions/us-east4/clusters/big-data] Cluster placed in zone [us-east4-c].
```

<input type="text" value="Search clusters, press Enter"/>								
<input type="checkbox"/>	Name ^	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket	Created	Status
<input checked="" type="checkbox"/>	big-data	us-east4	us-east4-c	0	Off	big_data_storage	Sep 12, 2019, 2:43:19 PM	Running

(a) "Pi calculation"

<input type="text" value="Search jobs, press Enter"/>							
<input type="checkbox"/>	Job ID	Region	Type	Cluster	Start time	Elapsed time	Status
<input type="checkbox"/>	60659cf855f449fa40c771e12814e92	us-east4	PySpark	big-data	Sep 12, 2019, 2:46:51 PM	30 sec	Succeeded

Result in GUI:

[illegible]

Output in command line interface:

[illegible]

- (b) "word count" job:
Result in command line:

[illegible]

```
huchong@huchong:~/huchong/Columbia/ECS/E6893_Big_Data_Analytics/Homework/hwd$ gsubtl cat gs://big_data_storage/output/part-00000
(u'a', 1)
(u'a', 2)
(u'we', 1)
(u'would', 1)
(u'what's', 1)
(u'sweet', 1)
(u'as', 1)
(u'call', 1)
(u'which', 1)
(u'snell', 1)
```

Result in GUI:

<input type="checkbox"/> Job ID	Region	Type	Cluster	Start time	Elapsed time	Status
<input checked="" type="checkbox"/> c5c6737e324a43aaa2307159475a2266	us-east4	PySpark	big-data	Sep 12, 2019, 10:36:04 PM	28 sec	Succeeded

[illegible]

2. Transformations and actions involved in each exercise:

- (a) "Pi calculation"
First, we parallelize the data that randomly generated to create a RDD. Then we use "map" and "reduce", those two transformations to count how many points are inside the circle. In the end, we can simulate the the "Pi" value.
- (b) "word count"
First, we use "textFile" function to create a RDD from text file. Then, we apply three transformations, "flatMap" to split word from lines, "map" to generate key-value pairs, and "reduceByKey" to count the frequency of each word. In the end, use "saveAsTextFile" (action) to save the "wordcount" result.

Problem 2. NYC Bike expert

1. How many unique *station_ids* are there in the dataset?

1 `SELECT COUNT(DISTINCT station_id) FROM `hardy-symbol-252200.hw0.data citibike stations``

Query results		SAVE RESULTS	EXPLORE WITH DATA STUDIO
Query complete (0.6 sec elapsed, 6.6 KB processed)			
Job information		Results	JSON Execution details
Row	f0_		
1	843		

2. Whats the largest *capacity* for a station?

COUNT DISTINCT station_id		Edited
1	SELECT MAX(capacity) FROM `hardy-symbol-252200.hw0.data_citibike_stations`	

Query complete (0.8 sec elapsed, 6.6 KB processed)	
Job information	Results JSON Execution details
Row	f0_
1	79

List all the *station_id* of the stations that have the largest capacity?

1	SELECT station_id FROM `hardy-symbol-252200.hw0.data_citibike_stations` WHERE capacity = (SELECT MAX(capacity) FROM `hardy-symbol-252200.hw0.data_citibike_stations`)	
---	---	--

Query complete (0.8 sec elapsed, 13.2 KB processed)	
Job information	Results JSON Execution details
Row	station_id
1	445
2	422
3	501

3. Whats the total number of bikes available in *region_id* 70?

1	SELECT SUM(num_bikes_available) FROM `hardy-symbol-252200.hw0.data_citibike_stations` WHERE region_id = 70	
---	--	--

Job information		Results	JSON	Execution details
Row	f0_			
1	394			

Problem 3. Understanding William Shakespeare

- Without any text preprocessing, here are the result for top 5 frequent words:

```
1      [('the', 620), ('and', 427), ('of', 396), ('to', 367), ('I', 326)]
```

```

[496209a47e2c4c8d15cbf6538288ba] submitted.
Waiting for DIB output...
19/09/17 19:04:08 INFO org.spark_project.jetty.util.log: Logging initialized @2742ms
19/09/17 19:04:08 INFO org.spark_project.jetty.server.Server: jetty 9.3.2-SNAPSHOT, build timestamp: unknown, git hash: unknown
19/09/17 19:04:08 INFO org.spark_project.jetty.server.Server: Started @283ms
19/09/17 19:04:09 WARN org.apache.spark.scheduler.FairSchedulerBuilder: Fair Scheduler configuration file not found so jobs will be scheduled in FIFO order. To use fair scheduling, configure pools in fa
irscheduler.conf or set spark.scheduler.allocation.file to a file that contains the configuration.
19/09/17 19:04:10 INFO org.apache.hadoop.yarn.client.ha: Connecting to Resourcemanager at big-data-n/j0.150.0.6:8032
19/09/17 19:04:13 INFO org.apache.hadoop.yarn.client.ha: VerifiedClientPath: Submitted application application_1508740875534_0001
19/09/17 19:04:22 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
[{"the": 620}, {"and": 427}, {"of": 396}, {"to": 367}, {"I": 326}]
19/09/17 19:04:27 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@704061b5[HTTP/1.1,(http/1.1)](0.0.0.0:4040)
[496209a47e2c4c8d15cbf6538288ba] finished successfully.
driverControlFileUri: gs://jupyter_notebook_storage/google-cloud-dataproc-metainfo/fa85de99-bcd9-4e94-87c6-4f61bc081953/jobs/496209a47e2c4c8d15cbf6538288ba/
driverOutputResourceUri: gs://jupyter_notebook_storage/google-cloud-dataproc-metainfo/fa85de99-bcd9-4e94-87c6-4f61bc081953/jobs/496209a47e2c4c8d15cbf6538288ba/driveroutput
jobId: 5d25867e-204d-361d-8026-902d7729235
placement:
  clusterName: big-data
  clusterId: fa85de99-bcd9-4e94-87c6-4f61bc081953
  pysparkJob:
    args:
      - gs://big_data_storage/shakes.txt
      - main.py
    mainPythonUri: gs://jupyter_notebook_storage/google-cloud-dataproc-metainfo/fa85de99-bcd9-4e94-87c6-4f61bc081953/jobs/496209a47e2c4c8d15cbf6538288ba/staging/count_top.py
    references:
      jobId: 496209a47e2c4c8d15cbf6538288ba
      projectId: hardy-symbol-232260
    status:
      state: DONE
      stateStartTime: '2019-09-17T19:04:31.324Z'
      stateHistory:
        state: PENDING
        stateStartTime: '2019-09-17T19:04:04.400Z'
        state: SETUP_DONE
        stateStartTime: '2019-09-17T19:04:04.440Z'
        details: Agent reported job success
      state: RUNNING
      stateStartTime: '2019-09-17T19:04:04.789Z'
  yarnApplications:
    name: count_top.py
    progress: 1.0
    state: FINISHED
    trackingUri: http://big-data-m:8080/proxy/application_1508740875534_0001/

```

Here is my source code.

```

1  import pyspark
2  import sys
3
4  if __name__ == '__main__':
5      inputUri = sys.argv[1]
6
7      sc = pyspark.SparkContext()
8      lines = sc.textFile(sys.argv[1])
9      words = lines.flatMap(lambda line: line.split())
10     wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda count1, count2: count1 + count2)
11     lambda x: x[1], False)
12     res = wordCounts.take(5)
13     print(res)

```

- With NLTK text preprocessing, here are the result for top 5 frequent words.

1 [('macb', 137), ('haue', 122), ('thou', 90), ('enter', 81), ('shall', 68)]

```
huchonghuchong:~/huchong/columbia/ECS_6893_Big_Data_Analytics/Homework/hw5 gcloud dataproc jobs submit pyspark count_top_p.py --cluster big-data --gs://big_data_storage/shakes.txt
Job [d2b6499e3284f1e615bacbd1d137ea] submitted.
waiting for job output...
19/09/20 01:26:12 INFO org.spark_project.jetty.util.log: Logging initialized @5023ms
19/09/20 01:26:12 INFO org.spark_project.jetty.server.Server: jetty-9.3.9-SNAPSHOT build timestamp: unknown, git hash: unknown
19/09/20 01:26:13 INFO org.spark_project.jetty.server.Server: Started @5025ms
19/09/20 01:26:13 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@62175af8(HTTP/1.1,[http/1.1])(0.0.0.0:4040)
19/09/20 01:26:13 WARN org.apache.spark.scheduler.FairSchedulableBuilder: Fair scheduler configuration file not found so jobs will be scheduled in FIFO order. To use fair scheduling, configure pools in fa
ir scheduler conf or set spark.scheduler.allocation.file to a file that contains the configuration.
19/09/20 01:26:14 INFO org.apache.hadoop.yarn.client.impl.Proxy: Connecting to ResourceManager at big-data-n10-150.0.0.8:8032
19/09/20 01:26:14 INFO org.apache.hadoop.yarn.client.impl.Proxy: Connecting to ApplicationHistory server at big-data-n10-150.0.0.8:8020
19/09/20 01:26:17 INFO org.apache.hadoop.yarn.client.impl.YarnClientImpl: Submitted application application_1508941717882_0001
19/09/20 01:26:18 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
[('macb', 137), ('haue', 122), ('thou', 90), ('enter', 81), ('shall', 68)]
19/09/20 01:26:18 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@62175af8(HTTP/1.1,[http/1.1])(0.0.0.0:4040)
Job [d2b6499e3284f1e615bacbd1d137ea] finished successfully.
driverControlURL: gs://jupyter_notebook_storage/google-cloud-dataproc-meta/info/d2b62736-57c5-4f2e-b945-4a7c83fe1f47/jobs/d2b6499e3284f1e615bacbd1d137ea/
driverOutputResourceURL: gs://jupyter_notebook_storage/google-cloud-dataproc-meta/info/d2b62736-57c5-4f2e-b945-4a7c83fe1f47/jobs/d2b6499e3284f1e615bacbd1d137ea/driveroutput
jobId: d2b6499e-c414-3790-ba37-2ae0b1b4d1b3
placement:
  clusterName: big-data
  clusterUuid: d2b62736-57c5-4f2e-b945-4a7c83fe1f47
pysparkJob:
  groupId:
  - gs://big_data_storage/shakes.txt
  mainPythonFileURL: gs://jupyter_notebook_storage/google-cloud-dataproc-meta/info/d2b62736-57c5-4f2e-b945-4a7c83fe1f47/jobs/d2b6499e3284f1e615bacbd1d137ea/staging/count_top_p.py
reference:
  jobId: d2b6499e3284f1e615bacbd1d137ea
  projectId: hardy-symbol-252200
status:
  state: DONE
  stateStartTime: '2019-09-20T01:26:37.157Z'
  statusHistory:
  - state: PENDING
    stateStartTime: '2019-09-20T01:26:08.259Z'
  - state: SETUP_DONE
    stateStartTime: '2019-09-20T01:26:08.298Z'
  - details: Agent reported job success
    state: RUNNING
    stateStartTime: '2019-09-20T01:26:08.589Z'
pythonLocations:
  name: count_top_p.py
  uri: gs://big-data-n10-150.0.0.8
  state: FINISHED
trackingURL: http://big-data-n10-150.0.0.8/proxy/application_1508941717882_0001/
```

Here is my source code.

```
1 import pyspark
2 import sys
3 import nltk
4
5 stop_words = set(nltk.corpus.stopwords.words('english'))
6 tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')
7
8
9 def stop_words_filter(line):
10     text = tokenizer.tokenize(line)
11     words = []
12     for word in text:
13         word = word.lower()
14         if word not in stop_words:
15             words.append(word)
16     return words
17
18
19 if __name__ == '__main__':
20     inputUri = sys.argv[1]
21
22     sc = pyspark.SparkContext()
23     lines = sc.textFile(sys.argv[1])
24     words = lines.flatMap(stop_words_filter)
25     wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda count1, count2: count1 + count2)
26     lambda x: x[1], False)
27     res = wordCounts.take(5)
28     print(res)
```
