Shivam Ojha
UNI: so2639

Homework 0

EECS E6893 Big Data Analytics
September 24, 2021

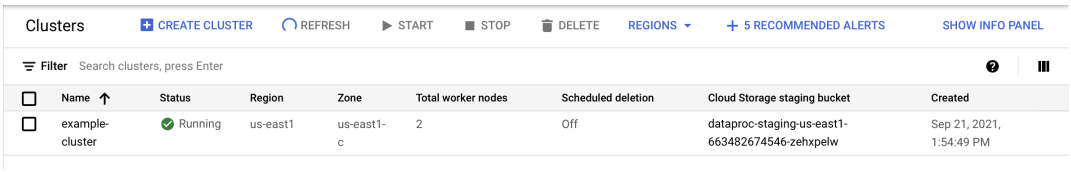
Problem 1: Warm-up exercises

1. Screenshots of exercises.

   Initialising the cluster("example-cluster") using command line:

```
shivamojha@Shivams-MacBook-Air:~$ gcloud dataproc clusters create example-cluster --region=us-east1
Waiting on operation [projects/big-data-6893-326522/regions/us-east1/operations/bbc80142-4484-35b7-9402-37abbb8ee7ee].
Waiting for cluster creation operation...
WARNING: No image specified. Using the default image version. It is recommended to select a specific image version in production, as the default image version may change at any time.
Waiting for cluster creation operation...done.
Created [https://dataproc.googleapis.com/v1/projects/big-data-6893-326522/regions/us-east1/clusters/example-cluster] Cluster placed in zone [us-east1-c].
```

   Example-cluster created in GUI:

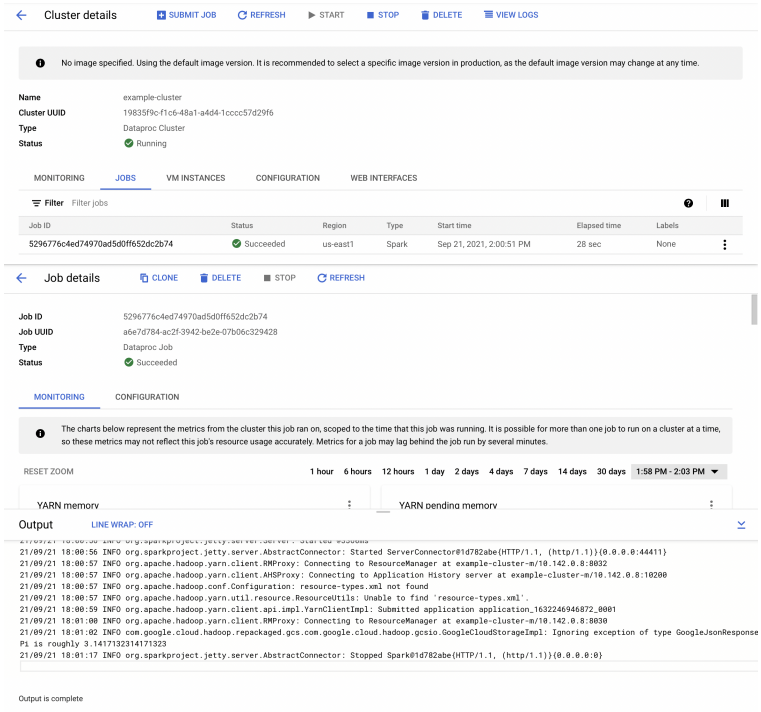


   (a) Pi Calculation:

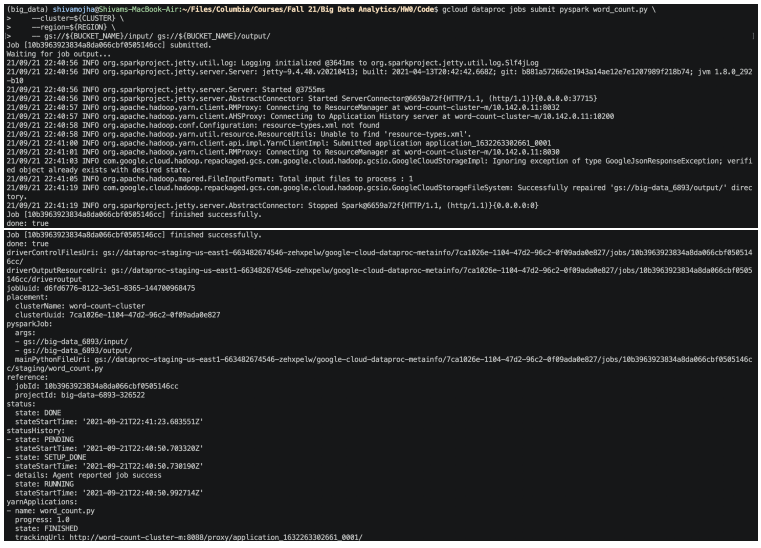
   Successful Job Execution for Pi Calculation on CLI:

```
shivamojha@Shivams-MacBook-Air:~$ gcloud dataproc jobs submit spark --cluster example-cluster \
>     --region=us-east1 \
>     --class org.apache.spark.examples.SparkPi \
>     --jars file:///usr/lib/spark/examples/jars/spark-examples.jar -- 1000
Job [5296776c4ed74970ad5d0ff652dc2b74] submitted.
Waiting for job output...
21/09/21 18:00:56 INFO org.sparkproject.jetty.util.log: Logging initialized @3212ms to org.sparkproject.jetty.util.log.Slf4jLog
21/09/21 18:00:56 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.668Z; git: b881a572662e1943a14ae12e7e1207989f218b74; jvm 1.8.0_292-b10
21/09/21 18:00:56 INFO org.sparkproject.jetty.server.Server: Started @3316ms
21/09/21 18:00:56 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@1d782abe{HTTP/1.1, [http/1.1]}{0.0.0.0:44411}
21/09/21 18:00:57 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at example-cluster-m/10.142.0.8:8032
21/09/21 18:00:57 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at example-cluster-m/10.142.0.8:10200
21/09/21 18:00:57 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
21/09/21 18:00:57 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
21/09/21 18:00:59 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1632246946872_0001
21/09/21 18:01:00 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at example-cluster-m/10.142.0.8:8030
21/09/21 18:01:02 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired state.
Pi is roughly 3.1417132314171323
21/09/21 18:01:17 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@1d782abe{HTTP/1.1, [http/1.1]}{0.0.0.0:0}
Job [5296776c4ed74970ad5d0ff652dc2b74] finished successfully.
done: true
Pi is roughly 3.1417132314171323
21/09/21 18:01:17 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@1d782abe{HTTP/1.1, [http/1.1]}{0.0.0.0:0}
Job [5296776c4ed74970ad5d0ff652dc2b74] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-east1-663482674546-zehxpelw/google-cloud-dataproc-metainfo/19835f9c-f1c6-48a1-a4d4-1cccc57d29f6/jobs/5296776c4ed74970ad5d0ff652dc2b74/
driverOutputResourceUri: gs://dataproc-staging-us-east1-663482674546-zehxpelw/google-cloud-dataproc-metainfo/19835f9c-f1c6-48a1-a4d4-1cccc57d29f6/jobs/5296776c4ed74970ad5d0ff652dc2b74/driveroutput
jobUuid: a6e7d784-ac2f-3942-be2e-07b06c329428
placement:
  clusterName: example-cluster
  clusterUuid: 19835f9c-f1c6-48a1-a4d4-1cccc57d29f6
reference:
  jobId: 5296776c4ed74970ad5d0ff652dc2b74
  projectId: big-data-6893-326522
sparkJob:
  args:
  - '1000'
  jarFileUris:
  - file:///usr/lib/spark/examples/jars/spark-examples.jar
  mainClass: org.apache.spark.examples.SparkPi
status:
  state: DONE
  stateStartTime: '2021-09-21T18:01:19.600962Z'
statusHistory:
- state: PENDING
  stateStartTime: '2021-09-21T18:00:51.787654Z'
- state: SETUP_DONE
  stateStartTime: '2021-09-21T18:00:51.822642Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2021-09-21T18:00:52.088156Z'
yarnApplications:
- name: Spark Pi
  progress: 1.0
  state: FINISHED
  trackingUrl: http://example-cluster-m:8088/proxy/application_1632246946872_0001/
```

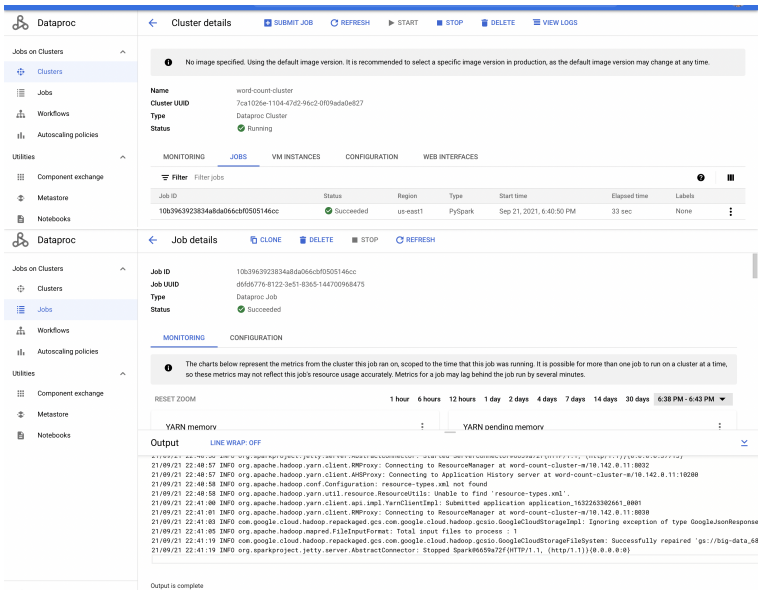
   Successful Job Execution for Pi Calculation on GUI:



(b) Word-Count:
Successful Job Execution for word-count on CLI:

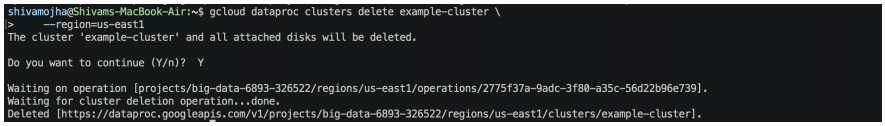


Successful Job Execution for word-count on GUI:



Word Count Result in CLI:

Delete example-cluster after executing the jobs:



2. Spark transformations and actions:

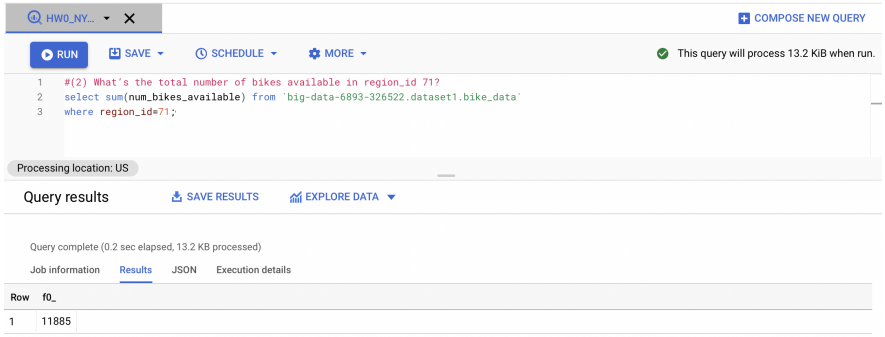
   (a) First, the randomly generated data is parallelized to create a RDD. Then the transformation "filter" is used to filter out the points inside the circle. Action "count" is used to count these points. In the end, we can simulate the the "Pi" value by performing the necessary mathematical calculation.

   (b) A RDD is created initially using the "textFile" function. Then three transformations are applied: "FlatMap" is used to split word from lines, "map" is used to generate key-value pairs, and "reduceByKey" is used to count the frequency of each word. The action "saveAsTextFile" is used afterwards to save the "word-count" result as a text file.

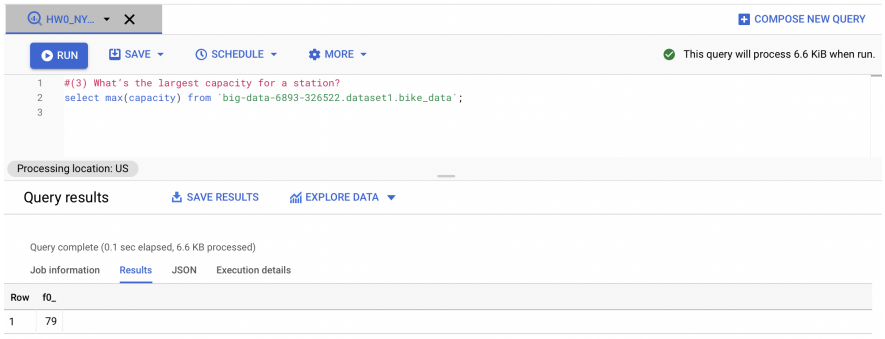
Problem 2: NYC Bike expert

1. How many stations with longitude between -73.94 and -74.04?

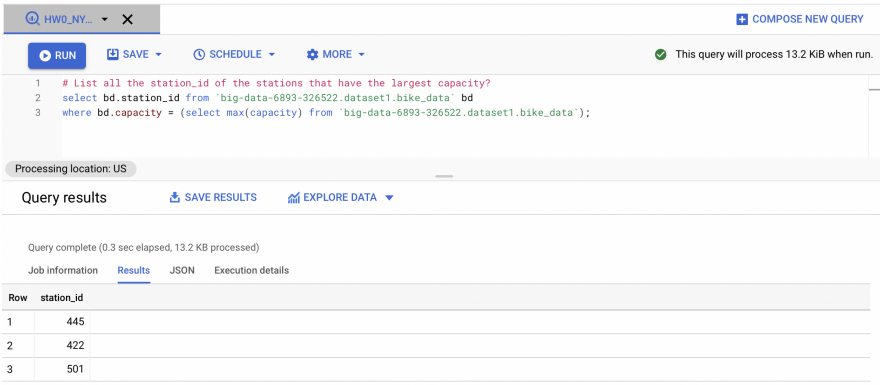
2. What's the total number of bikes available in region_id 71?



3. What's the largest capacity for a station?



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

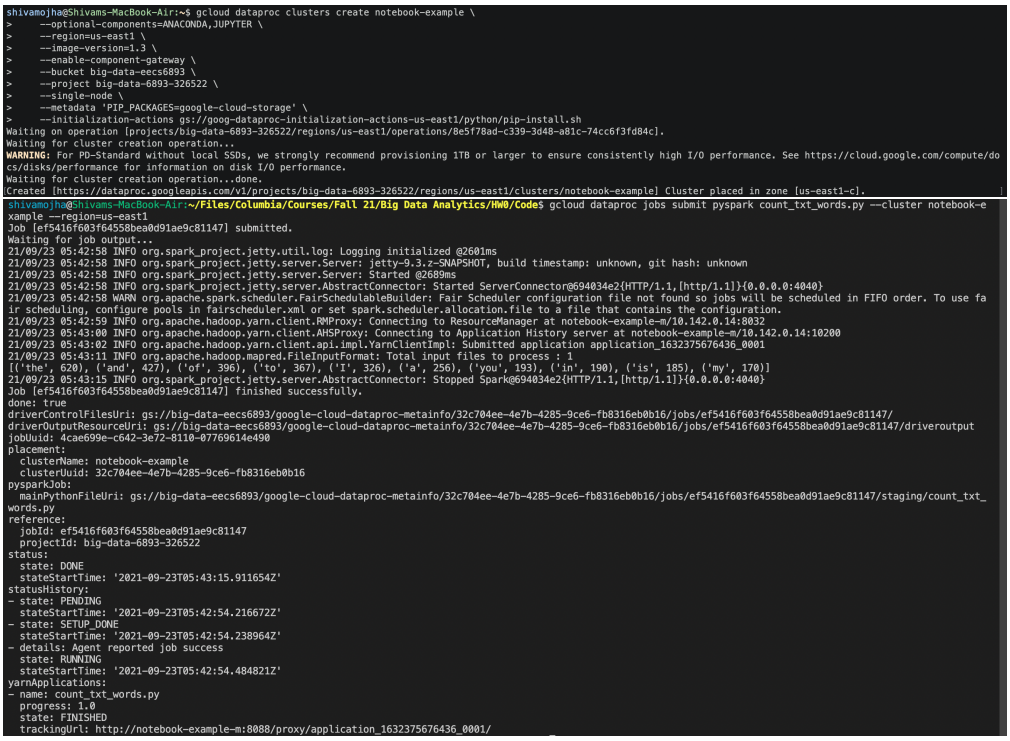


Problem 3: Understanding William Shakespeare

1. Without any text pre-processing, the result for top 10 frequent words:

   [('the', 620), ('and', 427), ('of', 396), ('to', 367), ('I', 326), ('a', 256), ('you', 193), ('in', 190), ('is', 185), ('my', 170)]

   Create cluster and submit job through Cloud shell:



   Code:

```
In [2]: import pyspark
        import sys

        # Cloud storage URI where the text file is stored
        input_uri = "gs://big-data-eecs6893/input/shakes.txt"

        sc = pyspark.SparkContext.getOrCreate()
        txt_file = sc.textFile(input_uri)

        # Split lines to words, and create key value pairs
        words = txt_file.flatMap(lambda line:line.encode("ascii", "ignore").split())
        words_kv = words.map(lambda word: (word, 1))

        # Add values of RDD by key and sort
        counts = words_kv.reduceByKey(lambda v1, v2: v1 + v2).sortBy(lambda x: -x[1])
        print(counts.take(10))

        [('the', 620), ('and', 427), ('of', 396), ('to', 367), ('I', 326), ('a', 256), ('you', 193), ('in', 190), ('is', 185)
        , ('my', 170)]
```

2. With NLTK text pre-processing, the result for top 10 frequent words:

   [('macb', 137), ('haue', 122), ('thou', 90), ('enter', 81), ('shall', 68), ('macbeth', 67), ('vpon', 62), ('thee', 61), ('macd', 58), ('th', 57)]

   Submit job through Cloud shell:

```
shivamojha@shivams-MacBook-Air:~/Files/Columbia/Courses/Fall 21/Big Data Analytics/HW0/Code$ gcloud dataproc jobs submit pyspark count_txt_words_nltk.py --cluster noteb
ook-example --region=us-east1
Job [8e02be63c2b14fdcb8494ead25426284] submitted.
Waiting for job output...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
21/09/24 02:54:19 INFO org.spark_project.jetty.util.log: Logging initialized @2678ms
21/09/24 02:54:19 INFO org.spark_project.jetty.server.Server: jetty-9.3.z-SNAPSHOT, build timestamp: unknown, git hash: unknown
21/09/24 02:54:19 INFO org.spark_project.jetty.server.Server: Started @2750ms
21/09/24 02:54:19 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@34cac0a6{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
21/09/24 02:54:19 WARN org.apache.spark.scheduler.FairSchedulableBuilder: Fair Scheduler configuration file not found so jobs will be scheduled in FIFO order. To use fa
ir scheduling, configure pools in fairscheduler.xml or set spark.scheduler.allocation.file to a file that contains the configuration.
21/09/24 02:54:20 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at notebook-example-m/10.142.0.15:8032
21/09/24 02:54:21 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at notebook-example-m/10.142.0.15:10200
21/09/24 02:54:23 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1632442049500_0014
21/09/24 02:54:28 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
[('macb', 137), ('haue', 122), ('thou', 90), ('enter', 81), ('shall', 68), ('macbeth', 67), ('vpon', 62), ('thee', 61), ('macd', 58), ('th', 57)]
21/09/24 02:54:34 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@34cac0a6{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
Job [8e02be63c2b14fdcb8494ead25426284] finished successfully.
done: true
driverControlFilesUri: gs://big-data-eecs6893/google-cloud-dataproc-metainfo/406dc3e8-8b50-4aaa-9444-c6c60570a7d5/jobs/8e02be63c2b14fdcb8494ead25426284/
driverOutputResourceUri: gs://big-data-eecs6893/google-cloud-dataproc-metainfo/406dc3e8-8b50-4aaa-9444-c6c60570a7d5/jobs/8e02be63c2b14fdcb8494ead25426284/driveroutput
jobUuid: 1f272b20-a1eb-3984-8446-cccbc7f85769
placement:
  clusterName: notebook-example
  clusterUuid: 406dc3e8-8b50-4aaa-9444-c6c60570a7d5
pysparkJob:
  mainPythonFileUri: gs://big-data-eecs6893/google-cloud-dataproc-metainfo/406dc3e8-8b50-4aaa-9444-c6c60570a7d5/jobs/8e02be63c2b14fdcb8494ead25426284/staging/count_txt_
words_nltk.py
reference:
  jobId: 8e02be63c2b14fdcb8494ead25426284
  projectId: big-data-6893-326522
status:
  state: DONE
  stateStartTime: '2021-09-24T02:54:39.099256Z'
statusHistory:
- state: PENDING
  stateStartTime: '2021-09-24T02:54:16.219798Z'
- state: SETUP_DONE
  stateStartTime: '2021-09-24T02:54:16.245252Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2021-09-24T02:54:16.499979Z'
yarnApplications:
- name: count_txt_words_nltk.py
  progress: 1.0
  state: FINISHED
  trackingUrl: http://notebook-example-m:8088/proxy/application_1632442049500_0014/
```

   Code:

```python
import pyspark
import sys
import nltk

nltk.download('stopwords')
stop_words = set(nltk.corpus.stopwords.words('english'))


def remove_stop_words(line):
    """
    Function to filter out stop words provided by NLTK package
    """
    words_list = []
    text_list = nltk.tokenize.RegexpTokenizer(r'\w+').tokenize(line)
    for word in text_list:
        word = word.encode("ascii", "ignore").lower()
        if word not in stop_words:
            words_list.append(word)
    return words_list


sc = pyspark.SparkContext()
lines = sc.textFile("gs://big-data-eecs6893/input/shakes.txt")
words_filter = lines.flatMap(remove_stop_words)
words_kv = words_filter.map(lambda x: (x,1))

# Add values of RDD by key and sort
counts = words_kv.reduceByKey(lambda v1, v2: v1 + v2).sortBy(lambda x: -x[1])
print(counts.take(10))
```