SHINDE SHUBHAM SUNIL
smail: me18b183@smail.iitm.ac.in
Course: CS4830 - Big Data Laboratory
Instructor: Prof. Balaraman Ravindran

Lab-3 Assignment
Roll No.: ME18B183
Semester: JAN-MAY 2022
Due Date: 19^{th} February, 2022

Problem 1
Write a Python code to count lines of the file placed in the BDL2022 bucket (gs://bdl2022/lines_big.txt) using Dataflow and provide the screenshot of the file that is generated in your bucket.

Solution:
Python code (*q1.py*) to count the lines of *lines_big.txt* file using Dataflow:

```
import apache_beam as beam
from apache_beam.io import ReadFromText
from apache_beam.io import WriteToText
from apache_beam.options.pipeline_options import PipelineOptions
from apache_beam.options.pipeline_options import GoogleCloudOptions
from apache_beam.options.pipeline_options import StandardOptions

options = PipelineOptions()
google_cloud_options = options.view_as(GoogleCloudOptions)
google_cloud_options.project = 'lab3-341009' # Enter your project ID
google_cloud_options.job_name = 'q1lab3'
google_cloud_options.temp_location = "gs://me18b183_l1/lab3assignment/tmp"
google_cloud_options.region = "us-central1"
options.view_as(StandardOptions).runner = 'DataflowRunner'

output_file = 'gs://me18b183_l1/lab3assignment/lab3_q1.txt'

p = beam.Pipeline(options=options)
lines = p | 'Read' >> beam.io.ReadFromText('gs://bdl2022/lines_big.txt')
          | 'Line Count' >> beam.combiners.Count.Globally()
          | 'Write' >> beam.io.WriteToText(output_file)
result = p.run()
```

Screenshots of the file generated in the bucket:

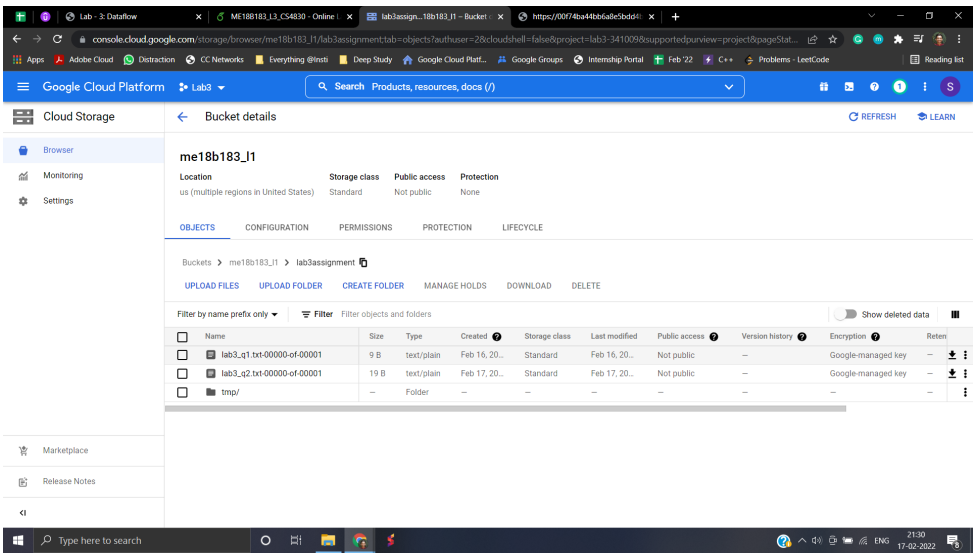

Figure 1: Screenshot of the file *lab3_q1.txt* stored in the bucket.

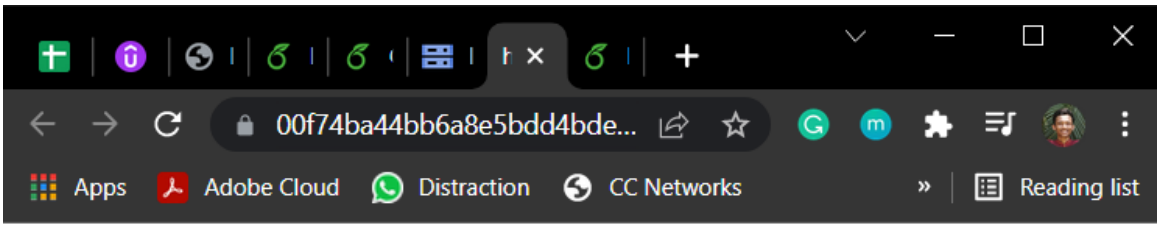

Figure 2: Total number of lines in *lines_big.txt* file, displayed in the output file generated *lab3_q1.txt*.

Problem 2
Write a Python code to get the average number of words in a line of the file placed in the BDL2022 bucket (gs://bdl2022/lines_big.txt) using Dataflow provide the screenshot of the file that is generated in your bucket.

Solution:
Python code (*q2.py*) to count the average no. of words in a line of *lines_big.txt* file using Dataflow:

```
import apache_beam as beam
from apache_beam.io import ReadFromText
from apache_beam.io import WriteToText
from apache_beam.options.pipeline_options import PipelineOptions
from apache_beam.options.pipeline_options import GoogleCloudOptions
from apache_beam.options.pipeline_options import StandardOptions

options = PipelineOptions()

google_cloud_options = options.view_as(GoogleCloudOptions)
google_cloud_options.project = 'lab3-341009' # Enter your project ID
google_cloud_options.job_name = 'q2lab3'
google_cloud_options.temp_location = "gs://me18b183_l1/tmp2"
google_cloud_options.region = "us-central1"
options.view_as(StandardOptions).runner = 'DataflowRunner'

output_file = 'gs://me18b183_l1/lab3assignment/lab3_q2.txt'

p = beam.Pipeline(options=options)

read = p | 'Read' >> beam.io.ReadFromText('gs://bdl2022/lines_big.txt')

class WordsInALine(beam.DoFn):
        def process(self, text):
                yield len(text.split())

#Assuming that the lines in the files_big.txt file are well punctuated.

avg_word = read | 'Words in a line' >> beam.ParDo(WordsInALine())
                | 'Mean' >> beam.combiners.Mean.Globally()
                | 'Write' >> beam.io.WriteToText(output_file)

result = p.run()
```

Screenshots of the file generated in the bucket:

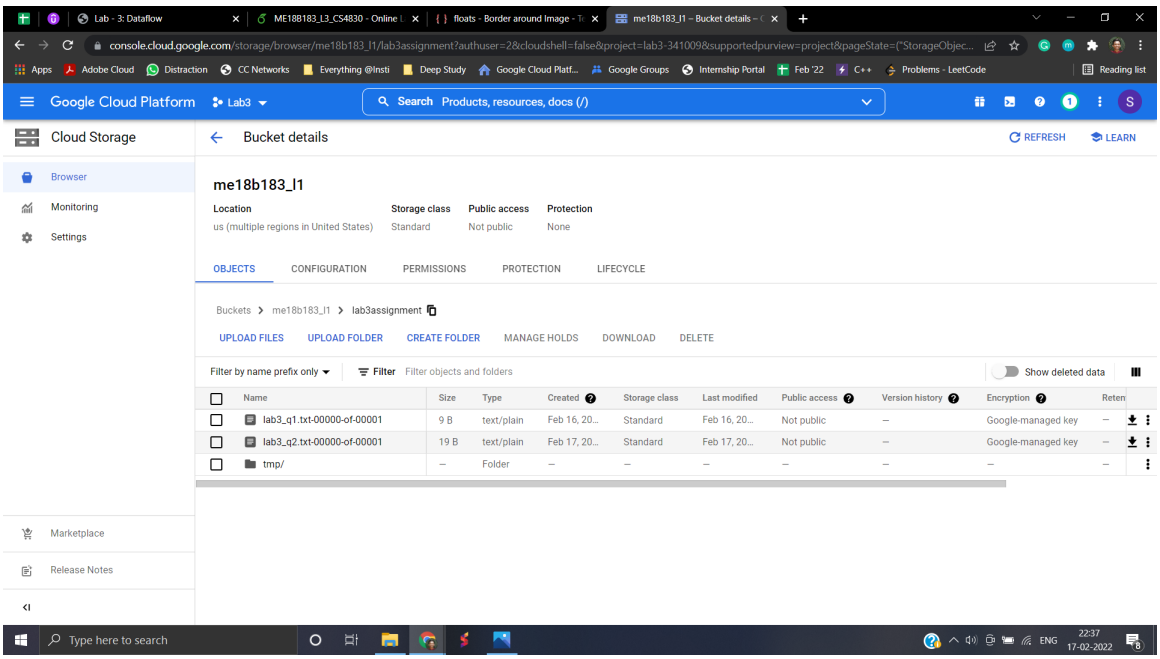

Figure 3: Screenshot of the file *lab3_q2.txt* stored in the bucket.

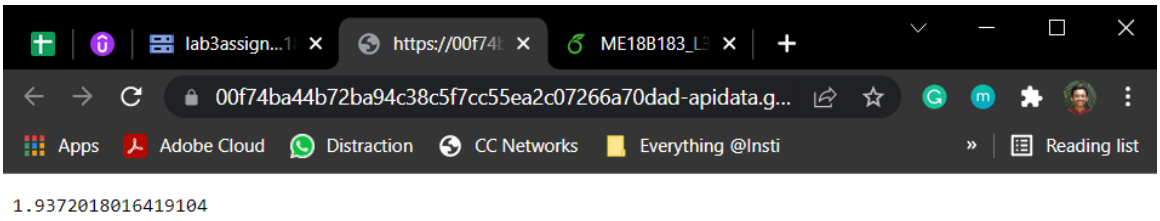

Figure 4: Average no. of words in a line of the *lines_big.txt* file, displayed in *lab3_q2.txt*.

Problem 3
Provide the screenshot for the execution graph created by Dataflow in the background for the pipeline object created for questions 1 and 2.

Solution:
Execution details for the jobs of questions 1 and 2:

Name	Type	End time	Elapsed time	Start time	Status	SDK version	ID	Region
q2lab3	Batch	Feb 17, 2022, 12:06:19 PM	13 min 45 sec	Feb 17, 2022, 11:52:34 AM	Succeeded	2.36.0	2022-02-16_22_22_34-16079740468334125808	us-central1
q1lab3	Batch	Feb 16, 2022, 4:38:01 PM	13 min 56 sec	Feb 16, 2022, 4:24:05 PM	Succeeded	2.36.0	2022-02-16_02_54_04-14297291819084763536	us-central1

Figure 5: Screenshot of successfully executed Dataflow jobs for questions 1 and 2.

Execution graphs created by Dataflow:

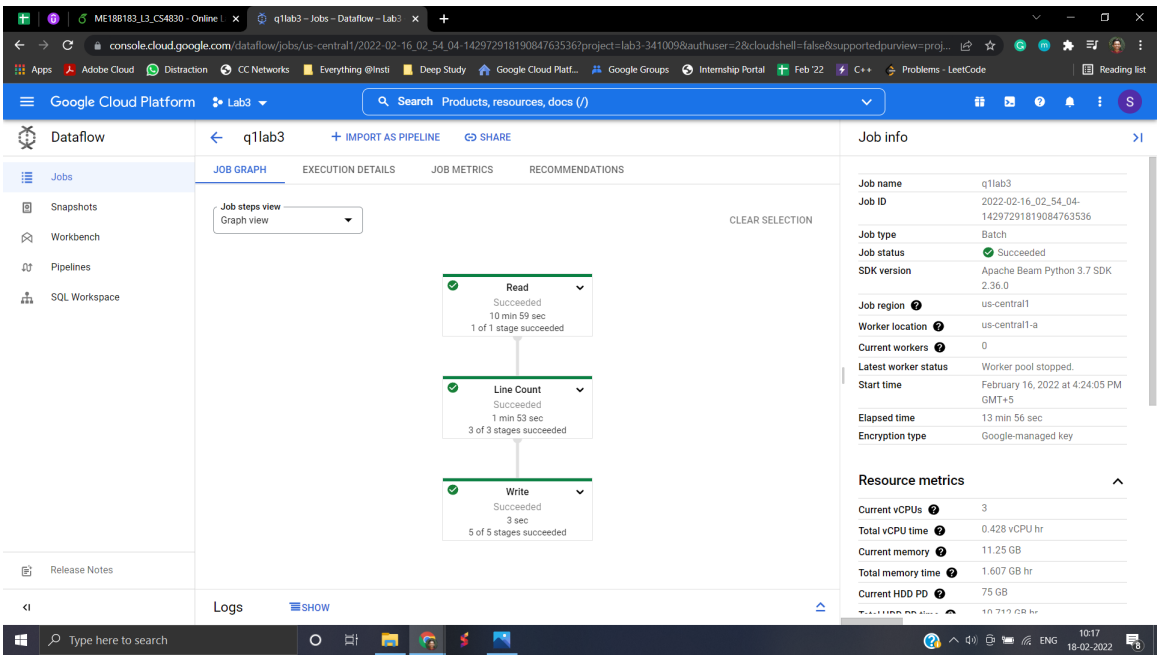

Figure 6: Job- **q1lab3**: Counting lines of *files_big.txt* file.

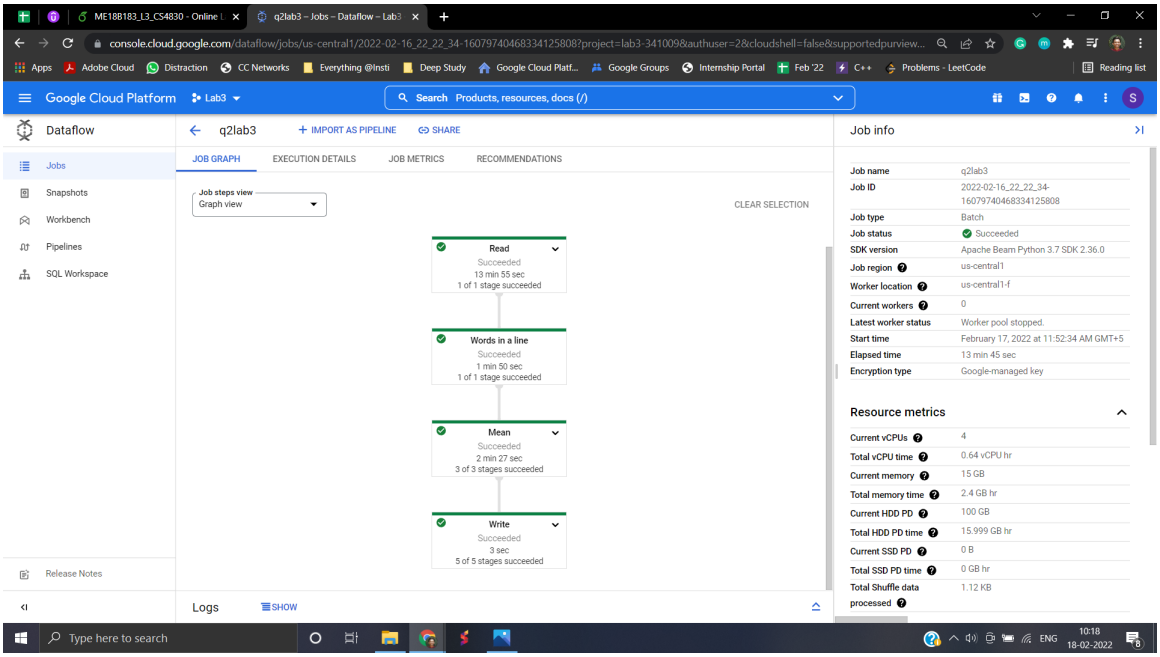

Figure 7: Job- **q2lab3**: Average no. of words in a line of *files_big.txt* file.

Problem 4
Explain the pipeline used in the first two questions. What issues did you face while trying to make the code work for the first two questions, and how did you resolve them?

Solution: Pipeline of **q1lab3** job:

1. Launch a virtual machine using Google Cloud Platform.
2. Open SSH to operate the virtual machine.

3. Write a Python script for performing below tasks:
   (a) Import Apache Beam Python SDK.
   (b) **{Pipeline}** Create a Pipeline.
   (c) **{Transform 1}** Create a PCollection by reading the file *lines_big.txt*.
   (d) **{Transform 2}** Invoke the transform *(beam.combiners.Count.Globally())* for counting all the elements of the PCollection created in step 3(c), which basically means counting all the lines from the file *lines_big.txt*.
   (e) **{Transform 3}** Invoke the write transform for writing the output PCollection of step 3(d) to the *lab3_q1.txt* file.
4. Run the Python script in SSH.

Pipeline of **q2lab3** job:

1. Launch a virtual machine using Google Cloud Platform.
2. Open SSH to operate the virtual machine.
3. Write a Python script for performing below tasks:
   (a) Import Apache Beam Python SDK.
   (b) **{Pipeline}** Create a Pipeline.
   (c) **{Transform 1}** Create a PCollection by reading the file *lines_big.txt*.
   (d) Define a class *(WordsInALine())* that can extract words form each element (lines of the file read in step 3(c)) and return their element-wise sum.
   (e) **{Transform 2}** Invoke ParDo element-wise transform to execute *WordsInALine()* class.
   (f) **{Transform 3}** Invoke the transform *(beam.combiners.Mean.Globally())* for computing the mean of output PCollection of step 3(e).
   (g) **{Transform 4}** Invoke the write transform for writing the output PCollection of step 3(f) to the *lab3_q2.txt* file.
4. Run the Python script in SSH.

Issues faced:

1. Scripting in SSH in order to smoothly operate the virtual machine. A quick tutorial was helpful to get used to Shell.
2. Getting used to the concepts of Apache Beam SDK like Pipeline, PCollections, Runners, etc. Going through the documentation helped me grasp these concepts, even though it was a complicated task I was able to understand after spending a good amount of time with it.
3. Understanding the functioning of each transformer and then choosing the correct one to return the expected output was quite a cumbersome task. Again going through the documentation was helpful but in this case I had to even check various stack exchange pages to understand the functioning of transformers properly.
4. In the bonus question, my code couldn't execute the expected output. Adding to that I exhausted my GCP credits as I thought that simply "Stopping" the virtual machines will turn off all the running dataflow jobs. As of now I have realised my mistakes after talking to the TAs and applied for free trial credits using a credit card. However, I have attached my code below for the bonus question.

Problem 5
(Bonus) Trigger a dataflow using GCF for any one of the first two questions.

Solution:
Code of the Python script, main.py.

```
def AVGTRIG(data, context):

        file = data['name']

        from google.cloud import storage
        from apache_beam.io import ReadFromText
        from apache_beam.io import WriteToText
        from apache_beam.options.pipeline_options import PipelineOptions
        from apache_beam.options.pipeline_options import GoogleCloudOptions
        from apache_beam.options.pipeline_options import StandardOptions

        options = PipelineOptions()

        google_cloud_options = options.view_as(GoogleCloudOptions)
        google_cloud_options.project = 'lab3-341009' # Enter your project ID
        google_cloud_options.job_name = 'q2lab3'
        google_cloud_options.temp_location = "gs://me18b183_l1/tmp2"
        google_cloud_options.region = "us-central1"
        options.view_as(StandardOptions).runner = 'DataflowRunner'

        output_file = 'gs://me18b183_l1/lab3assignment/lab3_q2.txt'

        p = beam.Pipeline(options=options)

        read = p | 'Read' >> beam.io.ReadFromText(f'gs://me18b183_l1/{file}')

        class WordsInALine(beam.DoFn):
                def process(self, text):
                        yield len(text.split())

        #Assuming that the lines in the files_big.txt file are well punctuated.

        avg_word = read | 'Words in a line' >> beam.ParDo(WordsInALine())
                        | 'Mean' >> beam.combiners.Mean.Globally()
                        | 'Write' >> beam.io.WriteToText(output_file)

        result = p.run()
```

As mentioned in question 4, I was not able to execute the code properly hence I couldn't attach the log screenshots.

∗∗ End of Assignment ∗∗