A picture containing text, screenshot, receipt, algebra

Description automatically generated

B. PageRank + PySpark + GCP

2. Diagram for this Question:

A screenshot of a computer

Description automatically generated with medium confidence

Create a Bucket:

1. Open the google console, go to the Cloud Storage -> Buckets page.
2. Now click on the “create Bucket”. On the bucket page enter your bucket information, for

the Name your bucket, enter a name that meets the bucket name requirements. Choose

the cheaper region. And rest things can be default values.

1. Click create.

A screenshot of a computer

Description automatically generated with medium confidence

1. The bucket is created (my bucket is with the name cs570-pagerank-python).

A screenshot of a computer

Description automatically generated with medium confidence

1. Now click on the bucket you created, click on the create folder and name the folder( I named it as input).

A screenshot of a computer

Description automatically generated with medium confidence

1. Now in the input folder we upload the file by clicking on the upload files.(I uploaded the txt file with name Pagerank.txt to sole the question and my input file consists the data)

A B

A C

B C

C A

A screenshot of a computer

Description automatically generated with medium confidence

Now we create a Dataproc cluster in GCP

1. In the Google Cloud console, go to the Dataproc Clusters page.

2. Click Create Cluster.

A screenshot of a computer

Description automatically generated with medium confidence

1. In the Create Dataproc cluster dialog, click Create in the Cluster on Compute engine row.

A screenshot of a computer

Description automatically generated

1. In the Cluster Name field, enter a name for your cluster.

In the Region and Zone lists, select the same region as the bucket and zone.

Choose the cluster type as a single node. And the rest can be the default. For all the other options, use the default settings.

To create the cluster, click Create.

A screenshot of a computer

Description automatically generated

1. The Cluster is created

A screenshot of a computer

Description automatically generated

To create and save the spark job python file:

1. In the top right corner of the console, click the Activate Cloud Shell button.
2. Once the Cloud Shell is activated, click on the Open Editor button in the top right corner of the Cloud Shell window.
3. Now click on the open editor

A screenshot of a computer

Description automatically generated with medium confidence

1. Click on the new file icon beside your username to create a new file.

A screenshot of a computer

Description automatically generated with medium confidence

1. Here is the python code I used:

import re

import sys

from operator import add

from pyspark.sql import SparkSession

def computeContribs(urls, rank):

"""Calculates URL contributions to the rank of other URLs."""

num\_urls = len(urls)

for url in urls:

yield (url, rank / num\_urls)

def parseNeighbors(urls):

"""Parses a urls pair string into urls pair."""

parts = re.split(r'\s+', urls)

return parts[0], parts[1]

if \_\_name\_\_ == "\_\_main\_\_":

if len(sys.argv) != 3:

print("Usage: pagerank <file> <iterations>", file=sys.stderr)

sys.exit(-1)

print("WARN: This is a naive implementation of PageRank and is given as an example!\n" +

"Please refer to PageRank implementation provided by graphx",

file=sys.stderr)

# Initialize the spark context.

spark = SparkSession\

.builder\

.appName("PythonPageRank")\

.getOrCreate()

####################################################

# Loads in input file. For example,

# argv[1] = data/mllib/pagerank\_data.txt

#

# It should be in format of:

# URL neighbor-URL

# URL neighbor-URL

# URL neighbor-URL

# Thus,

#

# + Physically, the file contains these line format.

# B C

# B A

# C A

# D A

# D B

# D C

# ...

# The lines are stored in the RDD "lines"

# r[0]

# +-------+

# 0 | B C |

# +-------+

# 1 | B A |

# +-------+

# 2 | C A |

# +-------+

# 3 | D A |

# +-------+

# 4 | D B |

# +-------+

# 5 | D C |

# +-------+

# ...

####################################################

lines = spark.read.text(sys.argv[1]).rdd.map(lambda r: r[0])

####################################################

# Loads all URLs from input file and initialize their neighbors.

#

# + "links" contains the list of neighbors of each page,

# (pageID, linkList)

# (B (C,A))

# (C (A))

# (D (A,B,C))

####################################################

links = lines.map(lambda urls: parseNeighbors(urls)).distinct().groupByKey().cache()

####################################################

# Loads all URLs with other URL(s) link to from input file and

# initialize ranks of them to one.

#

# - Initialize each page's rank to 1.0; since we use mapValues,

# the resulting RDD will have the same partitioner as links

# PR(A)=PR(B)=PR(C)=PR(D)=1

#

# - ranks contains the current rank for each page: (pageID, rank):

# (B 1)

# (C 1)

# (D 1)

####################################################

ranks = links.map(lambda url\_neighbors: (url\_neighbors[0], 1.0))

#########################################################################

# Calculates and updates URL ranks continuously using PageRank algorithm.

# argv[2] = 10

#########################################################################

for iteration in range(int(sys.argv[2])):

#####################################################################

# Calculates URL contributions to the rank of other URLs.

#

# Step 1: Obtain the link list and rank for each page ID

# by joining between the current ranks RDD and the

# static links RDD.

#

# links.join(ranks)

#

# Iteration 0

#

# WebPage Links PageRank

# ---------------------------------

# (B (C,A 1))

# (C (A 1))

# (D (A,B,C 1))

#

# Step 2: Apply flatMap to the ouput of join to create “contribution”

# values to send to each of the page’s neighbors.

#

# Dest WebPage Contributed (i.e., received) PageRank

# -----------------------------------------------------------

# C 0.5

# A 0.5

# A 1.0

# A 0.33

# B 0.33

# C 0.33

#

#####################################################################

contribs = links.join(ranks).flatMap(

lambda url\_urls\_rank: computeContribs(url\_urls\_rank[1][0], url\_urls\_rank[1][1]))

#####################################################################

# Step 3: Re-calculates URL ranks based on neighbor contributions.

#

# Add up "contribution" values by dest webpage (i.e. by the page

# receiving the contribution) and set that page’s

# rank to 0.15 + 0.85 \* contributionsReceived.

#

# A = (0.5 + 1.0 + 0.33) \* 0.85 + 0.15

# B = 0.33 \* 0.85 + 0.15

# C = (0.5 + 0.33) \* 0.85 + 0.15

#

#####################################################################

ranks = contribs.reduceByKey(add).mapValues(lambda rank: rank \* 0.85 + 0.15)

# Collects all URL ranks and dump them to console.

for (link, rank) in ranks.collect():

print("%s has rank: %s." % (link, rank))

spark.stop()

1. Save the file. I saved it as pagerank.py and close the shell.
2. Now on the top right corner of the console, click the Activate Cloud Shell button again and then the terminal opens up.
3. To run the code and get the output run the following command on the cloud shell terminal

For iteration 1:

gcloud dataproc jobs submit pyspark pagerank.py --cluster=cluster-d882 --region=us-central1 -- gs://cs570-pagerank-python/input/pagerank.txt 1

A screenshot of a computer

Description automatically generated with medium confidence

For iteration 2:

gcloud dataproc jobs submit pyspark pagerank.py --cluster=cluster-d882 --region=us-central1 -- gs://cs570-pagerank-python/input/pagerank.txt 2

A screenshot of a computer screen

Description automatically generated with medium confidence

The Question can be also done using the SSH in browser. Follow the steps to make it work:

1. Open the cluster through the vm instances and click on the SSH to get the ssh-in browser

A screenshot of a computer

Description automatically generated with medium confidence

1. Now we create a .py file with the above code and run it

$ vi pi.py

$ Python3 pi.py

1. We have the result for 1 iteration :

$ Python3 pi.py gs://cs570-pagerank-python/input/pagerank.txt 1

1. For 2 iterations:

$ Python3 pi.py gs://cs570-pagerank-python/input/pagerank.txt 2

A screenshot of a computer

Description automatically generated

Finally Delete the Dataproc cluster and the bucket:

1. Open the Google Cloud Console. In the console, navigate to the Dataproc Clusters page, select the cluster to delete and click deleted and confirm and the cluster is deleted.

A screenshot of a computer

Description automatically generated with medium confidence

1. Now Navigate to the Cloud Storage page. Select the bucket you want to delete. Click on delete and confirm by again typing the DELETE and the bucket is deleted.

A screenshot of a computer

Description automatically generated with medium confidence

c. PageRank + Scala + GCP

2. Diagram for this Question:

A screenshot of a computer

Description automatically generated with medium confidence

1. Create a Cloud Storage bucket( I created it with name **cs570-pagerank-scala**)

A screenshot of a computer

Description automatically generated with medium confidence

1. Create a Dataproc cluster

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Write and compile Scala code locally

using the Scala REPL (Read-Evaluate-Print-Loop or interactive interpreter)

1. From the link <https://www.scala-lang.org/download/install.html> download the binaries

A screenshot of a computer

Description automatically generated with medium confidence

Copy the command and paste it in the SSH terminal

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

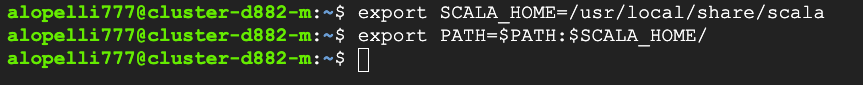
Description automatically generated

1. Set the environment

set the SCALA\_HOME environment variable

export SCALA\_HOME=/usr/local/share/scala

export PATH=$PATH:$SCALA\_HOME/



1. Launch the Scala REPL

$ scala

A black screen with white text

Description automatically generated with medium confidence

1. Insert the manual input data

A screenshot of a computer program

Description automatically generated with medium confidence

The input data has the data as follow:

A screenshot of a computer

Description automatically generated

1. create a directory (folder) to store the data and to verify that the file is indeed located in the mydata folder, run the following command:

hdfs dfs -mkdir hdfs:///mydata

hdfs dfs -put pagerank\_data.txt hdfs:///mydata

hdfs dfs -ls hdfs:///mydata

A screenshot of a computer screen

Description automatically generated with medium confidence

1. Prepare the program and Running the program

$ spark-shell

A screenshot of a computer program

Description automatically generated with medium confidence

1. Run the code for 1 iteration:

val lines = sc.textFile("hdfs:///mydata/pagerank\_data.txt")

val links = lines.map{ s =>

val parts = s.split("\\s+")

(parts(0), parts(1))

}.distinct().groupByKey().cache()

var ranks = links.mapValues(v => 1.0)

for (i <- 1 to 1) {

val contribs = links.join(ranks).values.flatMap{ case (urls, rank) =>

val size = urls.size

urls.map(url => (url, rank / size))

}

ranks = contribs.reduceByKey(\_ + \_).mapValues(0.15 + 0.85 \* \_)

}

val output = ranks.collect()

output.foreach(tup => println(tup.\_1 + " has rank: " + tup.\_2 + "."))

A screenshot of a computer program

Description automatically generated with medium confidence

For 2 iteration:

A picture containing text, screenshot

Description automatically generated

For 10 iteration:

A picture containing text, screenshot, software

Description automatically generated

Finally Delete the Dataproc cluster and the bucket:

1. Open the Google Cloud Console. In the console, navigate to the Dataproc Clusters page, select the cluster to delete and click deleted and confirm and the cluster is deleted.

A screenshot of a computer

Description automatically generated with medium confidence

1. Now Navigate to the Cloud Storage page. Select the bucket you want to delete. Click on delete and confirm by again typing the DELETE and the bucket is deleted.

A screenshot of a computer

Description automatically generated with medium confidence