PageRank using GCP

—

Shruti Kavishwar

San Francisco Bay University

Guided By: Prof. Henry Chang

Agenda

- Introduction
- Setup PySpark on GCP
- Enable DataProc API and Create GCS bucket
- Create DataProc Cluster
- Upload Data and Execute PySpark Code
- Scala Implemenation of PageRank
- Develop and Package Scala Code
- Upload JAR file and Submit Spark job
- Conclusion

3 key steps involved in the Project

1 Setting up the environment on GCP

- Install PySpark on GCP
- Enable DataProc API
- Create GCS Bucket
- Create DataProc Cluster

2 Running PySpark to perform PageRank

- Create ‘pagerank_data.txt’ file in Notepad
- Upload the file to the created GCS bucket
- Execute the Python code for PageRank

3 Implementing the same algo using Scala

- Install Scala
- Install SDKMAN!
- Install sbt
- Create Scala file for PageRank
- Package JAR file
- Upload the JAR file to GCS
- Submit the Scala job

What is PageRank ?

- PageRank is a link analysis algorithm developed by Larry Page and Sergey Brin at Stanford University.
- It is used by Google Search to rank web pages in search engine results.
- **How it Works:**
  - PageRank assigns a numerical weighting to each element of a hyperlinked set of documents (such as the World Wide Web).
  - The rank value indicates the importance of a particular page.
  - A page is considered important if it is linked to by other important pages.
  - The algorithm is based on the principle that more significant websites are more likely to be linked to by other websites.

Mathematical Representation of PageRank

- The algorithm uses a damping factor, typically set at 0.85, to account for the probability that a user will continue clicking on links.

$$PR(i) = (1 - d) + d \sum_{j \in M(i)} \frac{PR(j)}{L(j)}$$

Where,

- d the damping factor
- M(i) is the set of pages linking to page i
- L(j) is the number of outbound links on page j

Setup PySpark on GCP

- Steps:
  - Open Google Cloud Console.
  - Install PySpark
    - $ sudo apt-get install -y python-pip
    - $ sudo pip3 install pyspark
  - Verify the pyspark version

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ sudo apt-get update
*******************************************************************************
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

Visit https://cloud.google.com/shell/help for more information.
*******************************************************************************
Get:1 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:2 https://cli.github.com/packages stable InRelease [3,917 B]
Get:3 https://packages.cloud.google.com/apt gcsfuse-jammy InRelease [1,225 B]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 https://packages.microsoft.com/ubuntu/22.04/prod jammy InRelease [3,632 B]
Get:6 https://cli.github.com/packages stable/main amd64 Packages [346 B]
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ sudo apt-get install -y python3-pip
*******************************************************************************
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

Visit https://cloud.google.com/shell/help for more information.
*******************************************************************************
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 65 not upgraded.
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ sudo pip3 install pyspark
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.0/317.0 MB 3.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting py4j==0.10.9.7
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 200.5/200.5 KB 18.3 MB/s eta 0:00:00
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488492 sha256=3dc85c166257c9b28b2f3cd5c42e9a0bb8e5e7fe96b0fab9fba783566c55f563
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fbd8d74fb0b7f3a6
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.7 pyspark-3.5.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a v
irtual environment instead: https://pip.pypa.io/warnings/venv
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ pyspark --version
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.1
      /_/

Using Scala version 2.12.18, OpenJDK 64-Bit Server VM, 17.0.11
Branch HEAD
Compiled by user heartsavior on 2024-02-15T11:24:58Z
Revision fd86f85e181fc2dc0f50a096855acf83a6cc5d9c
Url https://github.com/apache/spark
Type --help for more information.
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

Enable DataProc API and Create GCS Bucket

- Steps:
  - Search for DataProc in the GCP Dashboard
  - Enable DataProc API
  - Create a Google Cloud Storage (GCS) bucket
    - Ensure the bucket and the DataProc cluster are in the same region (eg: us-central1)

Google Cloud | MapReduce-Week2-HW1-CS570 | bucke | Sear

Cloud Storage

Buckets

Monitoring

Settings

Marketplace

Release Notes

Create a bucket

Name your bucket

Pick a **globally unique**, permanent name. Naming guidelines

pagerank-bucket1

Tip: Don't include any sensitive information

LABELS (OPTIONAL)

CONTINUE

Choose where to store your data

Location: us (multiple regions in United States)
Location type: Multi-region

Choose a storage class for your data

Default storage class: Standard

Choose how to control access to objects

Public access prevention: On
Access control: Uniform

Good to know

Location pricing

Storage rates vary depen
and location of your buc

Current configuration: M

Item
us (multiple regions i States)
With default replicatio

ESTIMATE YOUR MON

Choose where to store your data

This choice defines the geographic placement of your data and affects cost, performance, and availability. Cannot be changed later. Learn more

Location type

- Multi-region
  Highest availability across largest area
- Dual-region
  High availability and low latency across 2 regions
- Region
  Lowest latency within a single region

us-central1 (Iowa)

CONTINUE

Public access will be prevented

This bucket is set to prevent exposure of its data on the public internet.

Keep this setting enabled unless you have a use case that requires public access (such as static website hosting). You can change it now or later. Learn more

- [x] Enforce public access prevention on this bucket
- [ ] Don't show this message again

CANCEL CONFIRM

pagerank-bucket1

Location	Storage class	Public access	Protection
us-central1 (Iowa)	Standard	Not public	Soft Delete

OBJECTS | CONFIGURATION | PERMISSIONS | PROTECTION | LIFECYCLE | OBSERVABILITY | INVENTORY REPORTS

Folder browser

pagerank-bucket1

Buckets > pagerank-bucket1

UPLOAD FILES | UPLOAD FOLDER | CREATE FOLDER | TRANSFER DATA

MANAGE HOLDS | EDIT RETENTION | DOWNLOAD | DELETE

Filter by name prefix only | Filter | Filter objects and folders | Show Live objects only

Name	Size	Type	Created	Storage class	Last modified	Public access
No rows to display						

Created bucket pagerank-bucket1

Create DataProc Cluster

Clusters | CREATE CLUSTER | REFRESH | START | STOP | DELETE | REGIONS

Cluster

Cloud Dataproc

Google Cloud Dataproc lets you provision Apache Hadoop clusters and connect to underlying analytic data stores.

There are no clusters in the currently selected Cloud Dataproc region(s). Create a cluster to get started.

CREATE CLUSTER

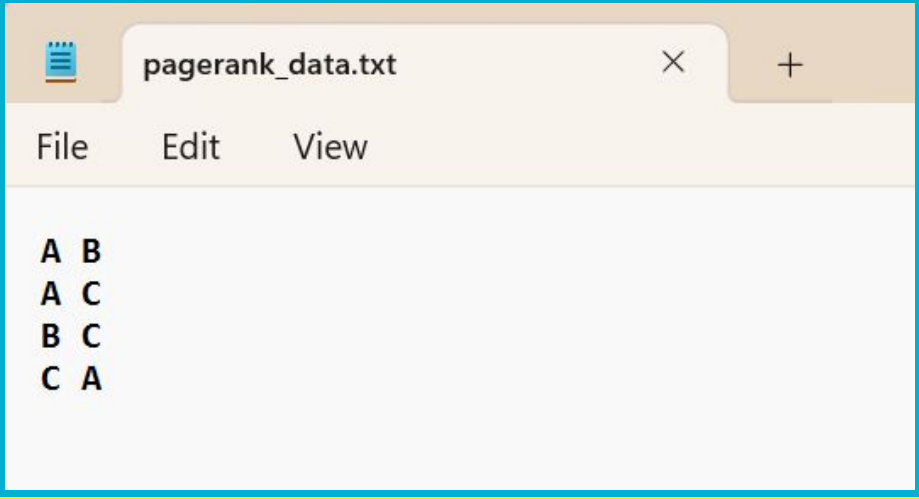
Clusters | CREATE CLUSTER | REFRESH | START | STOP | DELETE | SHOW INFO PANEL

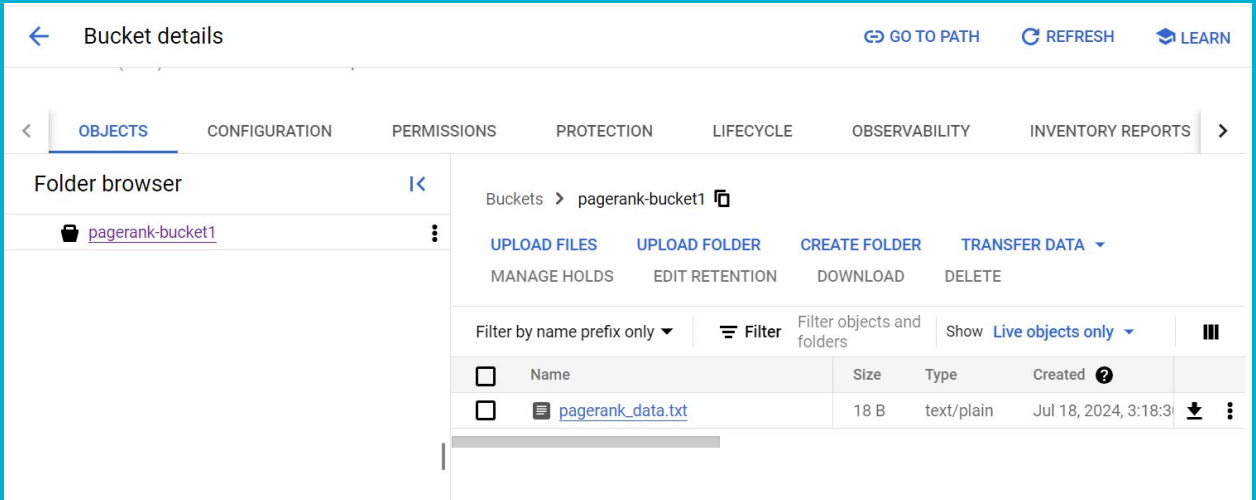
Filter | Search cluster by properties, press Enter

Name	Status	Region	Zone	Total worker nodes	Flexible VMs?	Scheduled deletion	Cloud Storage staging
cluster-3a74	Running	us-central1	us-central1-f	0	No	Off	dataproc-staging-us 174632744699-absc

Upload Data and Execute PySpark Code

- Steps:
  - Create ‘pagerank_data.txt’ file in Notepad.
  - Upload the file to the GCS bucket that was created
  - Execute the Pagerank python code






```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ gcloud dataproc jobs submit pyspark pagerank.py --cluster=cluster-3a74 --region=us-central1 -- gs://
pagerank-bucket1/pagerank_data.txt 10
Job [69d58fcbbc3e47e5a5a5c6f973e4999d] submitted.
Waiting for job output...
24/07/18 23:18:54 INFO SparkEnv: Registering MapOutputTracker
24/07/18 23:18:54 INFO SparkEnv: Registering BlockManagerMaster
```

- $ gcloud dataproc jobs submit pyspark pagerank.py - -cluster=cluster-3a74 – -region=us-central1 - - gs://pagerank-bucket1/pagerank_data.txt 10
- 10 is the number of iterations

Output of PySpark Pagerank

- We can see that the Pageranks are as follows
- A has rank: 1.0
- B has rank: 0.575
- C has rank: 1.4249999..
- This means that C has the highest importance and receives links from most other pages.
- B has the lowest pagerank indicating low importance and has less incoming links.

```
24/07/18 23:19:03 INFO FileInputFormat: Total input files to process : 1
Iteration 10
B has rank: 0.575
A has rank: 1.0
C has rank: 1.4249999999999998
Job [69d58fcbbc3e47e5a5a5c6f973e4999d] finished successfully.
done: true
```

```
placement:
  clusterName: cluster-3a74
  clusterUuid: 2253cfdf-f3f4-4a0a-8bcf-b9d7e736d326
pysparkJob:
  args:
  - gs://pagerank-bucket1/pagerank_data.txt
  - '10'
  mainPythonFileUri: gs://dataproc-staging-us-central1-174632744699-absg4qwx/google-cloud-dataproc-metainfo/2253cfdf-f3f4-4a0a-8bcf-b9d7e
3e47e5a5a5c6f973e4999d/staging/pagerank.py
reference:
  jobId: 69d58fcbbc3e47e5a5a5c6f973e4999d
  projectId: mapreduce-week2-hw1-cs570
status:
  state: DONE
  stateStartTime: '2024-07-18T23:19:22.217561Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-07-18T23:18:47.734443Z'
- state: SETUP_DONE
  stateStartTime: '2024-07-18T23:18:47.791088Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-07-18T23:18:48.176146Z'
yarnApplications:
- name: PythonPageRank
  progress: 1.0
  state: FINISHED
  trackingUrl: http://cluster-3a74-m.us-central1-f.c.mapreduce-week2-hw1-cs570.internal.:8088/proxy/application_1721329527176_0001/
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

Scala Implementation of PageRank

- Steps:
  - Install Scala
    - $ sudo apt-get install scala
    - VErification: $ scala -version
  - Install SDKMAN!
    - $ curl -s “https://get.sdkman.io” | bash
    - Initialize SDKMAN!: $ source “$HOME/.sdkman/bin/sdkman-init.sh”
  - Install sbt (Scala Build tool)
    - $ sdk install sbt
    - Verify sbt version

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ sudo apt-get install scala
*******************************************************************************
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

Visit https://cloud.google.com/shell/help for more information.
*******************************************************************************
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ scala -version
Scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ curl -s "https://get.sdkman.io" | bash
```

```
Installing script cli archive...
* Downloading...
################################################################################# 100.0%
* Checking archive integrity...
* Extracting archive...
* Copying archive contents...
* Cleaning up...

Set version to 5.18.2 ...
Set native version to 0.4.6 ...
Attempt update of interactive bash profile on regular UNIX...
Added sdkman init snippet to /home/skavishw276/.bashrc
Attempt update of zsh profile...
Updated existing /home/skavishw276/.zshrc



All done!


You are subscribed to the STABLE channel.

Please open a new terminal, or run the following in the existing one:

    source "/home/skavishw276/.sdkman/bin/sdkman-init.sh"

Then issue the following command:

    sdk help

Enjoy!!!
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ source "$HOME/.sdkman/bin/sdkman-init.sh"
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ sdk install sbt

Downloading: sbt 1.10.1

In progress...

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

Installing: sbt 1.10.1
Done installing!


Setting sbt 1.10.1 as default.
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ sbt sbtVersion
copying runtime jar...
[info] [launcher] getting org.scala-sbt sbt 1.10.1  (this may take some time)...
[info] [launcher] getting Scala 2.12.19 (for sbt)...
[info] Updated file /home/skavishw276/project/build.properties: set sbt.version to 1.10.1
[info] welcome to sbt 1.10.1 (Ubuntu Java 17.0.11)
[info] loading project definition from /home/skavishw276/project
[info] Updating skavishw276-build
https://repo1.maven.org/maven2/jline/jline/2.14.6/jline-2.14.6.pom
  100.0% [##########] 19.4 KiB (186.8 KiB / s)
[info] Resolved skavishw276-build dependencies
[info] Fetching artifacts of skavishw276-build
[info] Fetched artifacts of skavishw276-build
[info] set current project to skavishw276 (in build file:/home/skavishw276/)
[info] 1.10.1
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

Develop and Package Scala Code

- Steps:
  - Create Scala file and write PageRank code
  - Create `build.sbt` file
  - Directory setup
    - Create `src/main/scala` directory
    - Copy `pagerank.scala` file to this directory
  - Package JAR file
    - $ sbt package

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ cat PageRank.scala
import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.HashPartitioner
object PageRank {
  def main(args: Array[String]) {
  val sparkConf = new SparkConf().setAppName("PageRank")
  val sc = new SparkContext(sparkConf)
  val lines = sc.textFile(args(0))
  val links = lines.map{ s =>
  val parts = s.split("\\s+")
  (parts(0), parts(1))
  }.distinct().groupByKey().partitionBy(new HashPartitioner(100)).persist()
  var ranks = links.mapValues(_ => 1.0)
  for (i <- 0 until args(1).toInt) {
    val contributions = links.join(ranks).flatMap {
      case (pageId, (links, rank)) =>
        links.map(dest => (dest, rank / links.size))
    }
  ranks = contributions.reduceByKey(_ + _).mapValues(0.15 + 0.85 * _)
  }
  ranks.saveAsTextFile("gs://pagerank-bucket1/ranks")
  sc.stop()
  }
}
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ cat build.sbt
name := "PageRank"
version := "1.0"
scalaVersion := "2.12.15"
libraryDependencies += "org.apache.spark" %% "spark-core" % "3.2.0"
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ sbt package
[info] welcome to sbt 1.10.1 (Ubuntu Java 17.0.11)
[info] loading project definition from /home/skavishw276/project
[info] loading settings for project skavishw276 from build.sbt ...
[info] set current project to PageRank (in build file:/home/skavishw276/)
[info] compiling 1 Scala source to /home/skavishw276/target/scala-2.12/classes ...
[success] Total time: 6 s, completed Jul 19, 2024, 12:41:53 AM
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ ls target/scala-2.12/
classes  pagerank_2.12-1.0.jar  sync  update  zinc
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

Upload JAR file and Submit Spark Job

- Steps:
  - Upload JAR files to the GCS
  - $ gsutil cp target/scala-2.12/pagerank_2.12-1.0.jar gs://pagerank-bucket1/
  - Submit spark job for scala
    - $ gcloud dataproc jobs submit spark –cluster=cluster-3a74 –region=us-central1 –class=PageRank –jars=gs://pagerank-bucket1/pagerank_2.12-1.0.jar –gs://pagerank-bucket1/pagerank_data.txt 1 gs://pagerank-bucket1/ranks
  - Check ranks
    - $ gsutil cat gs://pagerank-bucket1/ranks/*

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$  gsutil cp target/scala-2.12/pagerank_2.12-1.0.jar gs://pagerank-bucket1/
Copying file://target/scala-2.12/pagerank_2.12-1.0.jar [Content-Type=application/java-archive]...
/ [1 files][  4.0 KiB/  4.0 KiB]
Operation completed over 1 objects/4.0 KiB.
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ gcloud dataproc jobs submit spark  --cluster=cluster-3a74 --region=us-central1 --class=PageRank --jars=gs:
//pagerank-bucket1/pagerank_2.12-1.0.jar -- gs://pagerank-bucket1/pagerank_data.txt 1 gs://pagerank-bucket1/ranks
Job [55d4721fb52144d5b1c426df2a971fb3] submitted.
Waiting for job output...
24/07/19 06:50:05 INFO SparkEnv: Registering MapOutputTracker
24/07/19 06:50:05 INFO SparkEnv: Registering BlockManagerMaster
24/07/19 06:50:05 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
24/07/19 06:50:06 INFO SparkEnv: Registering OutputCommitCoordinator
24/07/19 06:50:06 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-3a74-m.us-central1-f.c.mapreduce-week2-hw1-cs570.internal./1
0.128.0.4:8032
24/07/19 06:50:07 INFO AHSProxy: Connecting to Application History server at cluster-3a74-m.us-central1-f.c.mapreduce-week2-hw1-cs570.internal./10.128.0.4:10200
24/07/19 06:50:07 INFO Configuration: resource-types.xml not found
```

```
placement:
  clusterName: cluster-3a74
  clusterUuid: 2253cfdf-f3f4-4a0a-8bcf-b9d7e736d326
reference:
  jobId: 55d4721fb52144d5b1c426df2a971fb3
  projectId: mapreduce-week2-hw1-cs570
sparkJob:
  args:
  - gs://pagerank-bucket1/pagerank_data.txt
  - '1'
  - gs://pagerank-bucket1/ranks
  jarFileUris:
  - gs://pagerank-bucket1/pagerank_2.12-1.0.jar
  mainClass: PageRank
status:
  state: DONE
  stateStartTime: '2024-07-19T06:50:48.426892Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-07-19T06:50:00.207138Z'
- state: SETUP_DONE
  stateStartTime: '2024-07-19T06:50:00.254510Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-07-19T06:50:00.522103Z'
yarnApplications:
- name: PageRank
  progress: 1.0
  state: FINISHED
  trackingUrl: http://cluster-3a74-m.us-central1-f.c.mapreduce-week2-hw1-cs570.internal.:8088/proxy/application_1721329527176_0002/
```

Output of Scala Pagerank

```
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$ gsutil cat gs://pagerank-bucket1/ranks/*
(A,1.0)
(B,0.575)
(C,1.4249999999999998)
skavishw276@cloudshell:~ (mapreduce-week2-hw1-cs570)$
```

Conclusion:

PySpark	Scala
• PySpark is the Python API for Spark, making it more accessible for developers familiar with Python.	• Scala is the native language for Spark and offers more control over Spark's internals.
• Writing PySpark code is generally easier and faster due to Python's simpler syntax and extensive libraries.	• Writing in Scala can be more complex but can lead to more efficient and faster-executing code.

Github Link:
https://github.com/ShrutiK02/Cloud-Computing/tree/8e67d09404926536b8e92ccd71a45fe319cfaa70/MapReduce/PageRank