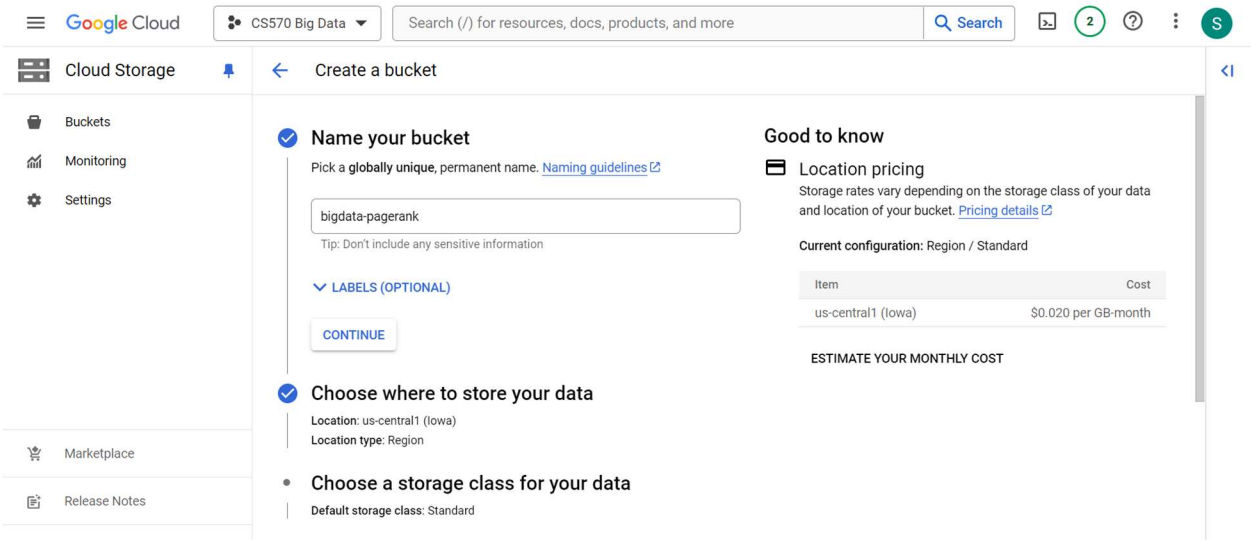
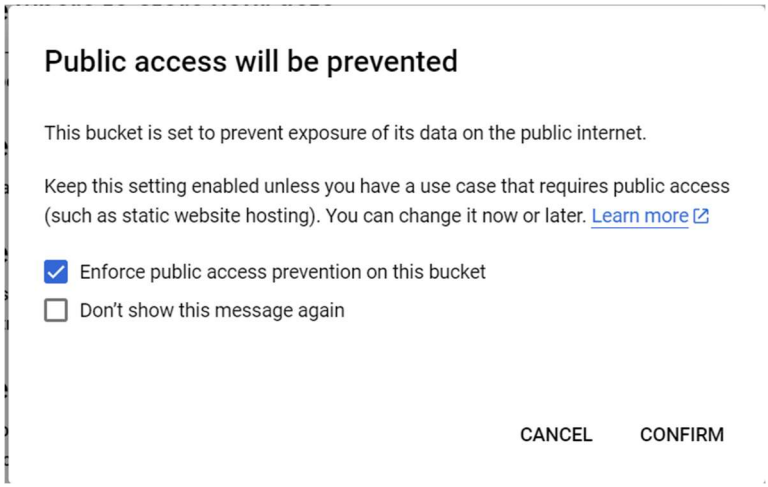
Week 6: Homework 1: Project: PageRank on GCP

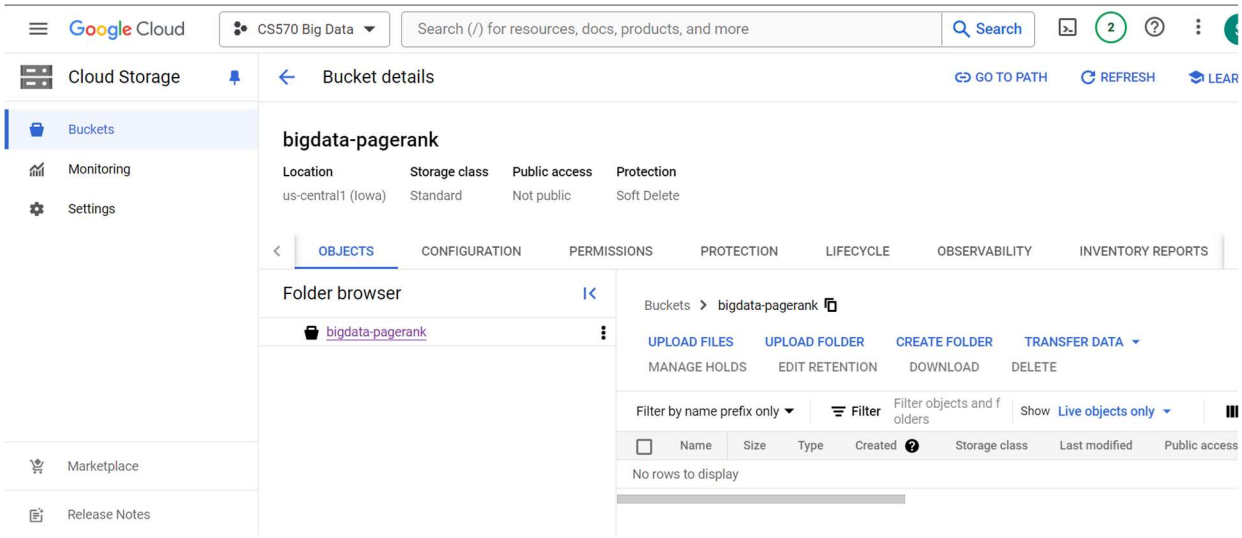
PART ONE – RUN USING PYSPARK

1. **Create a Bucket**



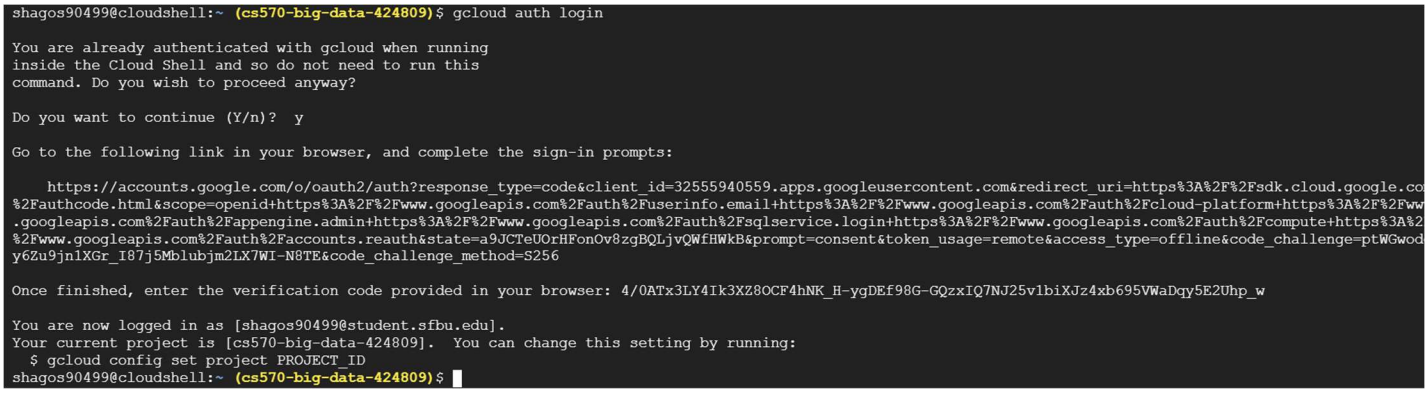


Bucket created successfully!



2. **Create a Cluster:**
   - **Set Up GCP Environment:**
     - Open Cloud Shell from the GCP Console.
     - Authenticate with GCP:

```
gcloud auth login
```



   - **Create a Dataproc Cluster:**

```
gcloud dataproc clusters create pagerank-cluster \
    --region=us-central1 \
    --zone=us-central1-a \
    --single-node \
    --master-machine-type=n1-standard-4 \
    --master-boot-disk-size=50GB \
    --image-version=1.5-debian10
```

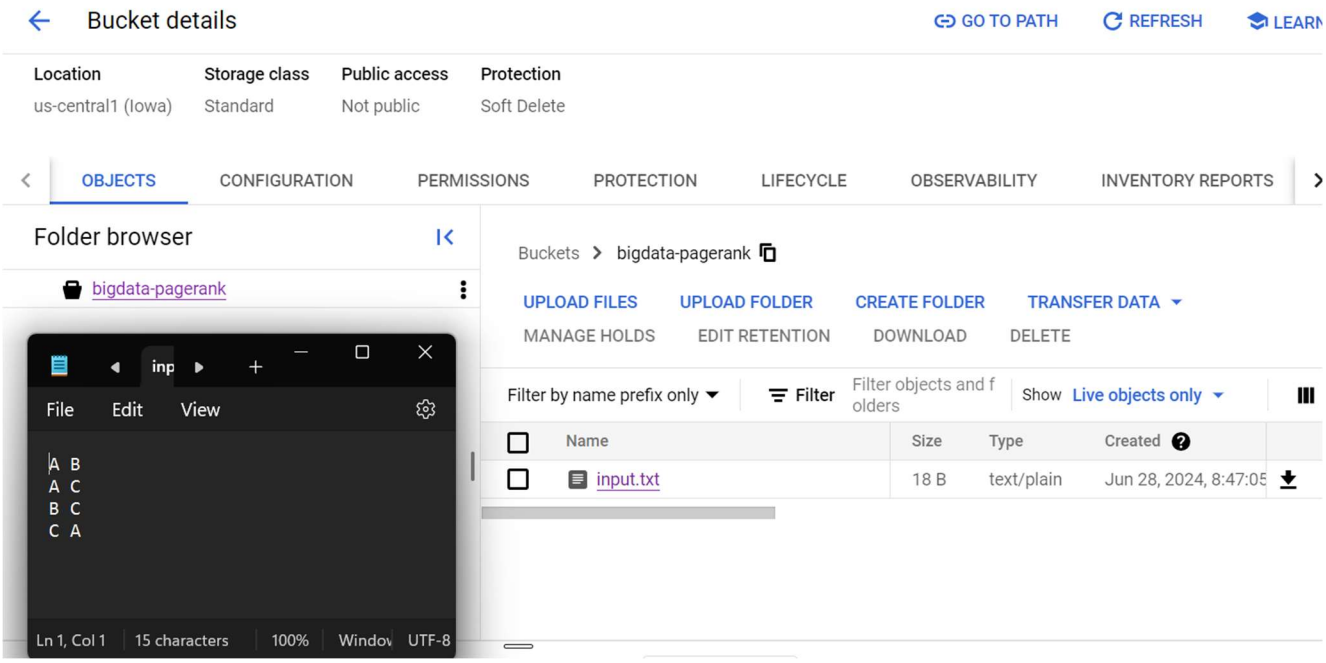
```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gcloud dataproc clusters create pagerank-cluster \
    --region=us-central1 \
    --zone=us-central1-a \
    --single-node \
    --master-machine-type=n1-standard-4 \
    --master-boot-disk-size=50GB \
    --image-version=1.5-debian10
Waiting on operation [projects/cs570-big-data-424809/regions/us-central1/operations/628d8aa0-17e6-30d1-b35f-07f7bf6acd7c].
Waiting for cluster creation operation...
WARNING: Consider using Auto Zone rather than selecting a zone manually. See https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/auto-zone
WARNING: Failed to validate permissions required for default service account: '720083396959-compute@developer.gserviceaccount.com'. Cluster creation could still
be successful if required permissions have been granted to the respective service accounts as mentioned in the document https://cloud.google.com/dataproc/docs/co
ncepts/configuring-clusters/service-accounts#dataproc_service_accounts_2. This could be due to Cloud Resource Manager API hasn't been enabled in your project '72
0083396959' before or it is disabled. Enable it by visiting 'https://console.developers.google.com/apis/api/cloudresourcemanager.googleapis.com/overview?project=
720083396959'.
WARNING: For PD-Standard without local SSDs, we strongly recommend provisioning 1TB or larger to ensure consistently high I/O performance. See https://cloud.goog
le.com/compute/docs/disks/performance for information on disk I/O performance.
WARNING: The firewall rules for specified network or subnetwork would allow ingress traffic from 0.0.0.0/0, which could be a security risk.
WARNING: Unable to validate the staging bucket lifecycle configuration of the bucket 'dataproc-staging-us-central1-720083396959-usvrohuu' due to an internal erro
r, Please make sure that the provided bucket doesn't have any delete rules set.
Waiting for cluster creation operation...done.
Created [https://dataproc.googleapis.com/v1/projects/cs570-big-data-424809/regions/us-central1/clusters/pagerank-cluster] Cluster placed in zone [us-central1-a].
shagos90499@cloudshell:~ (cs570-big-data-424809)$
```

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gcloud dataproc clusters list --region=us-central1
NAME: pagerank-cluster
PLATFORM: GCE
PRIMARY_WORKER_COUNT:
SECONDARY_WORKER_COUNT:
STATUS: RUNNING
ZONE: us-central1-a
SCHEDULED_DELETE:
shagos90499@cloudshell:~ (cs570-big-data-424809)$
```

3. **Prepare the PySpark Script:**
   - Save the input.txt file and upload it to the same bucket.

Input.txt:

```
A B
A C
B C
C A
```



4. **Create the `pagerank.py` Script:**
    - Use `vi pagerank.py` to create the script.
    - **Command-Line Arguments:**
        - `sys.argv[1]`: Path to the input file (e.g., `gs://bigdata_pagerank_pyspark/input.txt`).
        - `sys.argv[2]`: Number of iterations for the PageRank algorithm (e.g., 10).
    - **Spark Session:**
        - The script initializes a Spark session.
    - **Reading the Input File:**
        - `spark.read.text(sys.argv[1]).rdd.map(lambda r: r[0])`: Reads the input file from GCS and converts it to an RDD.
    - **Processing the Data:**
        - `parseNeighbors`: Parses each line to extract the URLs.
        - `links`: RDD containing the neighbors of each page.
        - `ranks`: RDD initializing the rank of each page to 1.0.
    - **PageRank Iterations:**
        - Calculates the contributions of each URL to its neighbors and updates the ranks based on these contributions.
    - **Output:**
        - Prints the final ranks of each URL.

```
import re
import sys
from operator import add
from pyspark.sql import SparkSession

def computeContribs(urls, rank):
    num_urls = len(urls)
    for url in urls:
        yield (url, rank / num_urls)

def parseNeighbors(urls):
    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)

    spark = SparkSession.builder.appName("PythonPageRank").getOrCreate()
    lines = spark.read.text(sys.argv[1]).rdd.map(lambda r: r[0])
    links = lines.map(lambda urls:
parseNeighbors(urls)).distinct().groupByKey().cache()
    ranks = links.map(lambda url_neighbors: (url_neighbors[0], 1.0))

    for iteration in range(int(sys.argv[2])):
        contribs = links.join(ranks).flatMap(
            lambda url_urls_rank: computeContribs(url_urls_rank[1][0],
url_urls_rank[1][1]))
        ranks = contribs.reduceByKey(add).mapValues(lambda rank: rank * 0.85 + 0.15)

    for (link, rank) in ranks.collect():
        print("%s has rank: %s." % (link, rank))
    spark.stop()
```

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ vi pagerank.py
shagos90499@cloudshell:~ (cs570-big-data-424809)$
```

```
import re
import sys
from operator import add
from pyspark.sql import SparkSession

def computeContribs(urls, rank):
    num_urls = len(urls)
    for url in urls:
        yield (url, rank / num_urls)

def parseNeighbors(urls):
    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)

    spark = SparkSession.builder.appName("PythonPageRank").getOrCreate()
    lines = spark.read.text(sys.argv[1]).rdd.map(lambda r: r[0])
    links = lines.map(lambda urls: parseNeighbors(urls)).distinct().groupByKey().cache()
    ranks = links.map(lambda url_neighbors: (url_neighbors[0], 1.0))

    for iteration in range(int(sys.argv[2])):
        contribs = links.join(ranks).flatMap(
            lambda url_urls_rank: computeContribs(url_urls_rank[1][0], url_urls_rank[1][1]))
        ranks = contribs.reduceByKey(add).mapValues(lambda rank: rank * 0.85 + 0.15)

    for (link, rank) in ranks.collect():
        print("%s has rank: %s." % (link, rank))
    spark.stop()
```

5. **Upload the Script to the Bucket:**

```
gsutil cp pagerank.py gs://bigdata-pagerank/
```

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gsutil cp pagerank.py gs://bigdata-pagerank/
Copying file://pagerank.py [Content-Type=text/x-python]...
/ [1 files][  1.1 KiB/  1.1 KiB]
Operation completed over 1 objects/1.1 KiB.
shagos90499@cloudshell:~ (cs570-big-data-424809)$
```

Note: the path to the bucket might be different from mine

6. **Submit the PySpark Job:**

```
gcloud dataproc jobs submit pyspark gs://bigdata-pagerank/pagerank.py \
    --cluster=pagerank-cluster \
    --region=us-central1 \
    -- gs://bigdata-pagerank/input.txt 10
```

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gcloud dataproc jobs submit pyspark gs://bigdata-pagerank/pagerank.py \
    --cluster=pagerank-cluster \
    --region=us-central1 \
    -- gs://bigdata-pagerank/input.txt 10
Job [892921bf4e95464c9b98a4716ba78c20] submitted.
Waiting for job output...
24/06/29 03:58:21 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
24/06/29 03:58:21 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
24/06/29 03:58:21 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
24/06/29 03:58:22 INFO org.spark_project.jetty.util.log: Logging initialized @4024ms to org.spark_project.jetty.util.log.Slf4jLog
24/06/29 03:58:22 INFO org.spark_project.jetty.server.Server: jetty-9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/06/29 03:58:22 INFO org.spark_project.jetty.server.Server: Started @4132ms
24/06/29 03:58:22 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@70165716{HTTP/1.1, (http/1.1)}{0.0.0.0:387
24/06/29 03:58:23 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at pagerank-cluster-m/10.128.0.7:8032
24/06/29 03:58:23 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at pagerank-cluster-m/10.128.0.7:10
24/06/29 03:58:23 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
24/06/29 03:58:23 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
```

```
ba78c20/driveroutput
jobUuid: a17553ba-b18d-345b-8ebc-46c2238fe080
placement:
  clusterName: pagerank-cluster
  clusterUuid: 5ea3af5b-34af-44c7-b8df-94a2c61fbf54
pysparkJob:
  args:
  - gs://bigdata-pagerank/input.txt
  - '10'
  mainPythonFileUri: gs://bigdata-pagerank/pagerank.py
reference:
  jobId: 892921bf4e95464c9b98a4716ba78c20
  projectId: cs570-big-data-424809
status:
  state: DONE
  stateStartTime: '2024-06-29T03:58:55.725431Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-06-29T03:58:16.339149Z'
- state: SETUP_DONE
  stateStartTime: '2024-06-29T03:58:16.385588Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-06-29T03:58:16.878709Z'
yarnApplications:
- name: PythonPageRank
  progress: 1.0
  state: FINISHED
```

Confirm by Checking the Output Files:

Go to the url provided in the outpug using gsutil ls {url}, we can see there are two output files

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gsutil ls gs://dataproc-staging-us-central1-720083396959-usvrohuu/google-cloud-dataproc-metainfo/5ea3af5b-34af-44c7-b8df-94a2c61f
bf54/jobs/892921bf4e95464c9b98a4716ba78c20/
gs://dataproc-staging-us-central1-720083396959-usvrohuu/google-cloud-dataproc-metainfo/5ea3af5b-34af-44c7-b8df-94a2c61fbf54/jobs/892921bf4e95464c9b98a4716ba78c20/driveroutput.0000
00000
gs://dataproc-staging-us-central1-720083396959-usvrohuu/google-cloud-dataproc-metainfo/5ea3af5b-34af-44c7-b8df-94a2c61fbf54/jobs/892921bf4e95464c9b98a4716ba78c20/driveroutput.0000
00001
shagos90499@cloudshell:~ (cs570-big-data-424809)$
```

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gsutil cat gs://dataproc-staging-us-central1-720083396959-usvrohuu/google-cloud
fbf54/jobs/892921bf4e95464c9b98a4716ba78c20/driveroutput.000000000
24/06/29 03:58:21 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
24/06/29 03:58:21 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
24/06/29 03:58:21 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
24/06/29 03:58:22 INFO org.spark_project.jetty.util.log: Logging initialized @4024ms to org.spark_project.jetty.util.log.Slf4jLog
24/06/29 03:58:22 INFO org.spark_project.jetty.server.Server: jetty-9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b
24/06/29 03:58:22 INFO org.spark_project.jetty.server.Server: Started @4132ms
24/06/29 03:58:22 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@70165716{HTTP/1.1, (http/1.1)}{0
24/06/29 03:58:23 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at pagerank-cluster-m/10.128.0.7:8032
24/06/29 03:58:23 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at pagerank-cluster-m/10.
24/06/29 03:58:23 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
24/06/29 03:58:23 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
24/06/29 03:58:23 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, t
24/06/29 03:58:23 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = vcores, units = , type =
24/06/29 03:58:26 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1719632375864_000
A has rank: 1.1667391764027368.
B has rank: 0.6432494117885129.
C has rank: 1.1900114118087488.
24/06/29 03:58:53 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@70165716{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
shagos90499@cloudshell:~ (cs570-big-data-424809)$
```

```
A has rank: 1.1667391764027368.
B has rank: 0.6432494117885129.
C has rank: 1.1900114118087488.
```

Running the PageRank Algorithm

Running the PageRank algorithm for multiple iterations helps ensure rank values converge to a stable state. The process typically involves:

1. **Initial Distribution:** Each page is given an equal rank initially, which is distributed based on links between pages.
2. **Propagation of Rank:** In subsequent iterations, ranks propagate through the network of links, adjusting ranks based on the overall web graph structure.
3. **Convergence:** After several iterations, ranks converge to stable values. Typically, 10-20 iterations are sufficient for small to medium-sized graphs.

For our case, we specified 10 iterations:

```
gcloud dataproc jobs submit pyspark gs://bigdata-pagerank/pagerank.py \
    --cluster=pagerank-cluster \
    --region=us-central1 \
    -- gs://bigdata-pagerank/input.txt 10
```

Example Output Check:

To see how the output evolves, start with fewer iterations and gradually increase:

1. **Run with 1 Iteration:**

   ```
   gcloud dataproc jobs submit pyspark gs://bigdata-pagerank/pagerank.py \
       --cluster=pagerank-cluster \
       --region=us-central1 \
       -- gs://bigdata-pagerank/input.txt 1
   ```

   Check the output after 1 iteration.

```
C has rank: 1.4249999999999998.
A has rank: 1.0.
B has rank: 0.575.
```

2. **Increase to 5 Iterations:**

```
gcloud dataproc jobs submit pyspark gs://bigdata-pagerank/pagerank.py \
    --cluster=pagerank-cluster \
    --region=us-central1 \
    -- gs://bigdata-pagerank/input.txt 5
```

Check the output after 5 iterations.

```
C has rank: 1.1618180859374996.
A has rank: 1.1846890624999995.
B has rank: 0.6534928515624998.
```

3. **Run with 10 Iterations:**

```
gcloud dataproc jobs submit pyspark gs://bigdata-pagerank/pagerank.py \
    --cluster=pagerank-cluster \
    --region=us-central1 \
    -- gs://bigdata-pagerank/input.txt 10
```

Check the output after 10 iterations.

```
A has rank: 1.1667391764027368.
B has rank: 0.6432494117885129.
C has rank: 1.1900114118087488.
```

PART TWO – RUN USING SCALA

1. **Create a Bucket and Upload Input Data:**
   - Create a bucket:

     ```
     gsutil mb gs://bigdata_pagerank-scala/
     ```

     ```
     shagos90499@cloudshell:~ (cs570-big-data-424809)$ gsutil mb gs://bigdata_pagerank-scala/
     Creating gs://bigdata_pagerank-scala/...
     shagos90499@cloudshell:~ (cs570-big-data-424809)$
     ```

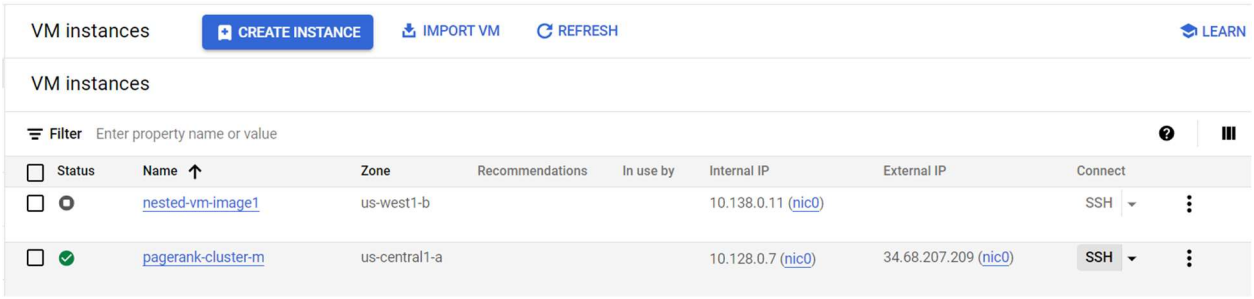
   - Upload input data:

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ vi input.txt
shagos90499@cloudshell:~ (cs570-big-data-424809)$ cat input.txt
A B
A C
B C
C A
```

```
gsutil cp input.txt gs://bigdata_pagerank-scala/
```

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gsutil cp input.txt gs://bigdata_pagerank-scala/
Copying file://input.txt [Content-Type=text/plain]...
/ [1 files][   16.0 B/   16.0 B]
Operation completed over 1 objects/16.0 B.
shagos90499@cloudshell:~ (cs570-big-data-424809)$
```

2. **Pre-Step: SSH to the Cluster from the compute engine and authorize next**




```
Linux pagerank-cluster-m 5.10.0-0.deb10.16-cloud-amd64 #1 SMP Debian 5.10.127-2~bpo10+1 (2022-07-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
shagos90499@pagerank-cluster-m:~$
```

3. **Update System Packages:** make sure that system packages are up-to date

```
sudo apt-get update
```

```
shagos90499@pagerank-cluster-m:~$ sudo apt-get update
Get:1 https://packages.cloud.google.com/apt google-cloud-logging-buster-all InRelease [1123 B]
Get:2 https://storage.googleapis.com/goog-dataproc-bigtop-repo-us-central1/1_5_deb10_20230908_124000-RC01 dataproc InRelease [3708
B]
Get:3 https://download.docker.com/linux/debian buster InRelease [53.9 kB]
Hit:4 http://cloud.r-project.org/bin/linux/debian buster-cran35/ InRelease
Get:5 https://repo.mysql.com/apt/debian buster InRelease [22.1 kB]
Get:6 https://packages.cloud.google.com/apt google-cloud-monitoring-buster-all InRelease [1127 B]
Hit:7 https://storage.googleapis.com/dataproc-bigtop-repo/1_5_deb10_20230908_124000-RC01 dataproc InRelease
Get:8 https://packages.cloud.google.com/apt google-compute-engine-buster-stable InRelease [1311 B]
```

4. **Install Scala:**

```
sudo apt-get install scala
```

```
shagos90499@pagerank-cluster-m:~$ sudo apt-get install scala
Reading package lists... Done
Building dependency tree
Reading state information... Done
scala is already the newest version (2.12.10-400).
0 upgraded, 0 newly installed, 0 to remove and 120 not upgraded.
shagos90499@pagerank-cluster-m:~$
```

5. **Install sbt (Scala Build Tool):**
   - Add sbt repository:

```
echo "deb https://repo.scala-sbt.org/scalasbt/debian all main" |
sudo tee /etc/apt/sources.list.d/sbt.list
```

```
shagos90499@pagerank-cluster-m:~$ echo "deb https://repo.scala-sbt.org/scalasbt/debian all main" | sudo tee /etc/apt/sources.li
/sbt.list
deb https://repo.scala-sbt.org/scalasbt/debian all main
```

   - Add repository key:

```
curl -sL
"https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x642AC823
" | sudo apt-key add
```

```
shagos90499@pagerank-cluster-m:~$ curl -sL "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x642AC823" | sudo apt-key add
OK
shagos90499@pagerank-cluster-m:~$
```

   - Update package list and install sbt:

```
sudo apt-get update
sudo apt-get install sbt
```

```
shagos90499@pagerank-cluster-m:~$ sudo apt-get install sbt
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sbt
0 upgraded, 1 newly installed, 0 to remove and 120 not upgraded.
Need to get 20.0 kB of archives.
After this operation, 50.2 kB of additional disk space will be used.
Get:1 https://scala.jfrog.io/artifactory/debian all/main amd64 sbt all 1.10.0 [20.0 kB]
Fetched 20.0 kB in 0s (42.9 kB/s)
Selecting previously unselected package sbt.
(Reading database ... 167133 files and directories currently installed.)
Preparing to unpack .../archives/sbt_1.10.0_all.deb ...
Unpacking sbt (1.10.0) ...
Setting up sbt (1.10.0) ...
Creating system group: sbt
Creating system user: sbt in sbt with sbt daemon-user and shell /bin/false
Processing triggers for man-db (2.8.5-2) ...
shagos90499@pagerank-cluster-m:~$
```

6. **Set Up Project Structure and Compile Code:**
   - Create project directories:

```
mkdir pagerank
cd pagerank
mkdir -p src/main/scala
```

```
shagos90499@pagerank-cluster-m:~$ mkdir pagerank
shagos90499@pagerank-cluster-m:~$ cd pagerank
shagos90499@pagerank-cluster-m:~/pagerank$ mkdir -p src/main/scala
shagos90499@pagerank-cluster-m:~/pagerank$
```

   - Create `build.sbt` file:

```
vi build.sbt
```

```
shagos90499@pagerank-cluster-m:~/pagerank$ vi build.sbt
```

`build.sbt` Content:

```
name := "SparkPageRank"
version := "1.0"
scalaVersion := "2.12.10"
libraryDependencies ++= Seq(
  "org.apache.spark" %% "spark-core" % "2.4.5",
  "org.apache.spark" %% "spark-sql" % "2.4.5"
)
```

```
name := "SparkPageRank"
version := "1.0"
scalaVersion := "2.12.10"
libraryDependencies ++= Seq(
  "org.apache.spark" %% "spark-core" % "2.4.5",
  "org.apache.spark" %% "spark-sql" % "2.4.5"
)
~
~
~
```

- Create `SparkPageRank.scala`:

  ```
  vi src/main/scala/SparkPageRank.scala
  ```

  ```
  shagos90499@pagerank-cluster-m:~/pagerank$ vi src/main/scala/SparkPageRank.scala
  shagos90499@pagerank-cluster-m:~/pagerank$
  ```

`SparkPageRank.scala` Content: This code implements the PageRank algorithm using Scala and Apache Spark. It reads input data, processes the data to compute PageRank, and prints the results.

```scala
package org.apache.spark.examples

import org.apache.spark.SparkContext._
import org.apache.spark.{SparkConf, SparkContext}

object SparkPageRank {

  def showWarning() {
    System.err.println(
      """WARN: This is a naive implementation of PageRank and is given as an example!
        |Please use the PageRank implementation found in org.apache.spark.graphx.lib.PageRank
        |for more conventional use.
      """.stripMargin)
  }

  def main(args: Array[String]) {
    if (args.length < 1) {
      System.err.println("Usage: SparkPageRank <file> <iter>")
      System.exit(1)
    }

    showWarning()

    val sparkConf = new SparkConf().setAppName("PageRank")
    val iters = if (args.length > 1) args(1).toInt else 10
    val ctx = new SparkContext(sparkConf)
    val lines = ctx.textFile(args(0), 1)

    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 iters) {
      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 + "."))

    ctx.stop()
  }
}
```


```
package org.apache.spark.examples

import org.apache.spark.SparkContext._
import org.apache.spark.{SparkConf, SparkContext}

object SparkPageRank {

  def showWarning() {
    System.err.println(
      """WARN: This is a naive implementation of PageRank and is given as an example!
        |Please use the PageRank implementation found in org.apache.spark.graphx.lib.PageRank
        |for more conventional use.
      """.stripMargin)
  }

  def main(args: Array[String]) {
```

```
    if (args.length < 1) {
      System.err.println("Usage: SparkPageRank <file> <iter>")
      System.exit(1)
    }

    showWarning()

    val sparkConf = new SparkConf().setAppName("PageRank")
    val iters = if (args.length > 1) args(1).toInt else 10
    val ctx = new SparkContext(sparkConf)
    val lines = ctx.textFile(args(0), 1)

    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 iters) {
      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 +
"."))

    ctx.stop()
  }
}
```

7. Compile the Project

Compile the project using sbt:

```
sbt package
```

```
shagos90499@pagerank-cluster-m:~/pagerank$ sbt package
downloading sbt launcher 1.10.0
[info] [launcher] getting org.scala-sbt sbt 1.10.0  (this may take some time)...
[info] [launcher] getting Scala 2.12.19 (for sbt)...
[info] Updated file /home/shagos90499/pagerank/project/build.properties: set sbt.version to 1.10.0
[info] welcome to sbt 1.10.0 (Temurin Java 1.8.0_382)
[info] loading project definition from /home/shagos90499/pagerank/project
[info] Updating pagerank-build
https://repo1.maven.org/maven2/jline/jline/2.14.6/jline-2.14.6.pom
  100.0% [##########] 19.4 KiB (192.3 KiB / s)
[info] Resolved pagerank-build dependencies
[info] Fetching artifacts of pagerank-build
[info] Fetched artifacts of pagerank-build
[info] loading settings for project pagerank from build.sbt ...
```

```
  100.0% [##########] 62.3 KiB (1.7 MiB / s)
https://repo1.maven.org/maven2/org/apache/commons/commons-math3/3.4.1/commons-math3-3.4.1.jar
  100.0% [##########] 1.9 MiB (55.5 MiB / s)
https://repo1.maven.org/maven2/org/glassfish/jersey/media/jersey-media-jaxb/2.22.2/jersey-media-jaxb-2.22.2.jar
  100.0% [##########] 71.0 KiB (1.9 MiB / s)
[info] Fetched artifacts of sparkpagerank_2.12
[info] compiling 1 Scala source to /home/shagos90499/pagerank/target/scala-2.12/classes ...
[info] Non-compiled module 'compiler-bridge_2.12' for Scala 2.12.10. Compiling...
[info]   Compilation completed in 12.851s.
[success] Total time: 22 s, completed Jun 29, 2024 5:26:19 AM
shagos90499@pagerank-cluster-m:~/pagerank$
```

These commands set up the project structure, define dependencies, write the Scala code for PageRank, and compile the code into a JAR file.

8. Upload Compiled JAR to Google Cloud Storage

Copy the compiled JAR file to a GCS bucket: *the path could be different*
`gs://bigdata_pagerank-scala/`

```
gsutil cp target/scala-2.12/sparkpagerank_2.12-1.0.jar gs://bigdata_pagerank-scala/
```

```
shagos90499@pagerank-cluster-m:~/pagerank$ gsutil cp target/scala-2.12/sparkpagerank_2.12-1.0.jar gs://bigdata_pagerank-scala/
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023. Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Copying file://target/scala-2.12/sparkpagerank_2.12-1.0.jar [Content-Type=application/java-archive]...
/ [1 files][  5.4 KiB/  5.4 KiB]
Operation completed over 1 objects/5.4 KiB.
shagos90499@pagerank-cluster-m:~/pagerank$
```

9. Submit Spark Job on Dataproc

Use Google Cloud Shell to submit the Spark job to Dataproc:

```
gcloud dataproc jobs submit spark --cluster=pagerank-cluster --region=us-central1 \
    --jars=gs://bigdata_pagerank-scala/sparkpagerank_2.12-1.0.jar \
    --class=org.apache.spark.examples.SparkPageRank \
    -- gs://bigdata_pagerank-scala/input.txt 10
```

```
shagos90499@cloudshell:~ (cs570-big-data-424809)$ gcloud dataproc jobs submit spark --cluster=pagerank-cluster --region=us-central1 \
    --jars=gs://bigdata_pagerank-scala/sparkpagerank_2.12-1.0.jar \
    --class=org.apache.spark.examples.SparkPageRank \
    -- gs://bigdata_pagerank-scala/input.txt 10
Job [ecdf0d2634684c73b793860216a077cb] submitted.
Waiting for job output...
WARN: This is a naive implementation of PageRank and is given as an example!
Please use the PageRank implementation found in org.apache.spark.graphx.lib.PageRank
for more conventional use.

24/06/29 05:34:59 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
24/06/29 05:34:59 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
24/06/29 05:34:59 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
24/06/29 05:34:59 INFO org.spark_project.jetty.util.log: Logging initialized @3139ms to org.spark_project.jetty.util.log.Slf4jLog
24/06/29 05:34:59 INFO org.spark_project.jetty.server.Server: jetty-9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/06/29 05:34:59 INFO org.spark_project.jetty.server.Server: Started @3367ms
24/06/29 05:34:59 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@39109136{HTTP/1.1, (http/1.1)}{0.0.0.0:42523}
24/06/29 05:35:00 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at pagerank-cluster-m/10.128.0.7:8032
24/06/29 05:35:00 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at pagerank-cluster-m/10.128.0.7:10200
24/06/29 05:35:00 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
24/06/29 05:35:00 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
24/06/29 05:35:00 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
24/06/29 05:35:00 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
24/06/29 05:35:02 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1719632375864_0006
24/06/29 05:35:11 INFO org.apache.hadoop.mapred.FileInputFormat: Tot
```

This command submits a Spark job to the Dataproc cluster, specifying the JAR file, main class, input data file, and number of iterations.

```
B has rank: 0.6432494117885129.
A has rank: 1.1667391764027368.
C has rank: 1.1900114118087488.
24/06/29 05:35:18 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark
Job [ecdf0d2634684c73b793860216a077cb] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-central1-720083396959-usvrohuu/google-
driverOutputResourceUri: gs://dataproc-staging-us-central1-720083396959-usvrohuu/google
jobUuid: 4cc02ce4-71e3-367e-9f10-1e17d88bf1a4
placement:
  clusterName: pagerank-cluster
  clusterUuid: 5ea3af5b-34af-44c7-b8df-94a2c61fbf54
reference:
  jobId: ecdf0d2634684c73b793860216a077cb
  projectId: cs570-big-data-424809
sparkJob:
  args:
  - gs://bigdata_pagerank-scala/input.txt
  - '10'
  jarFileUris:
  - gs://bigdata_pagerank-scala/sparkpagerank_2.12-1.0.jar
  mainClass: org.apache.spark.examples.SparkPageRank
status:
  state: DONE
  stateStartTime: '2024-06-29T05:35:23.136973Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-06-29T05:34:54.979307Z'
- state: SETUP_DONE
  stateStartTime: '2024-06-29T05:34:55.015525Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-06-29T05:34:55.294293Z'
yarnApplications:
- name: PageRank
  progress: 1.0
  state: FINISHED
  trackingUrl: http://pagerank-cluster-m:8088/proxy/application_1719632375864_0006/
```

Output: