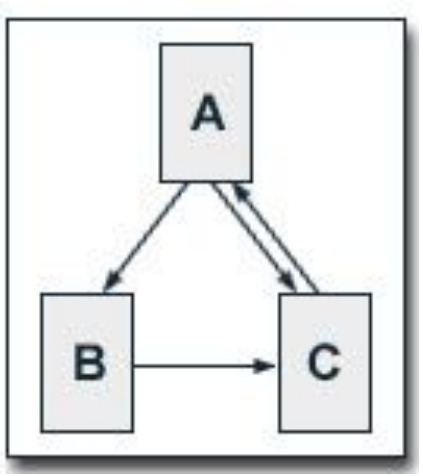
Project: PageRank on GCP

CS570 Big Data Processing & Analytics
Submitted by: Imran Noor Saleh Student ID 19648

Introduction

The objective of PageRank is to assign a numerical importance score to each webpage in a web graph based on its connectivity and the importance of the pages linking to it. The algorithm aims to determine the significance and relevance of web pages within the overall network of interlinked pages.

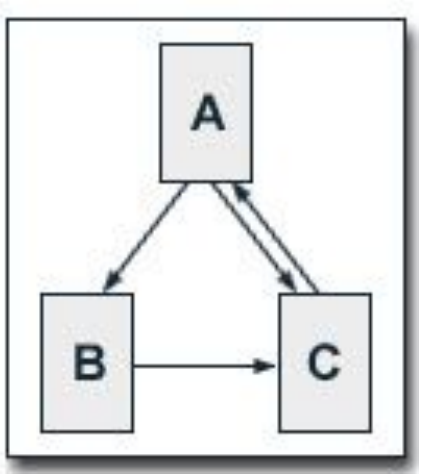
PageRank is designed to address the challenge of organizing and ranking the vast amount of information available on the internet. The algorithm assigns a numerical value, known as the PageRank score, to each web page.



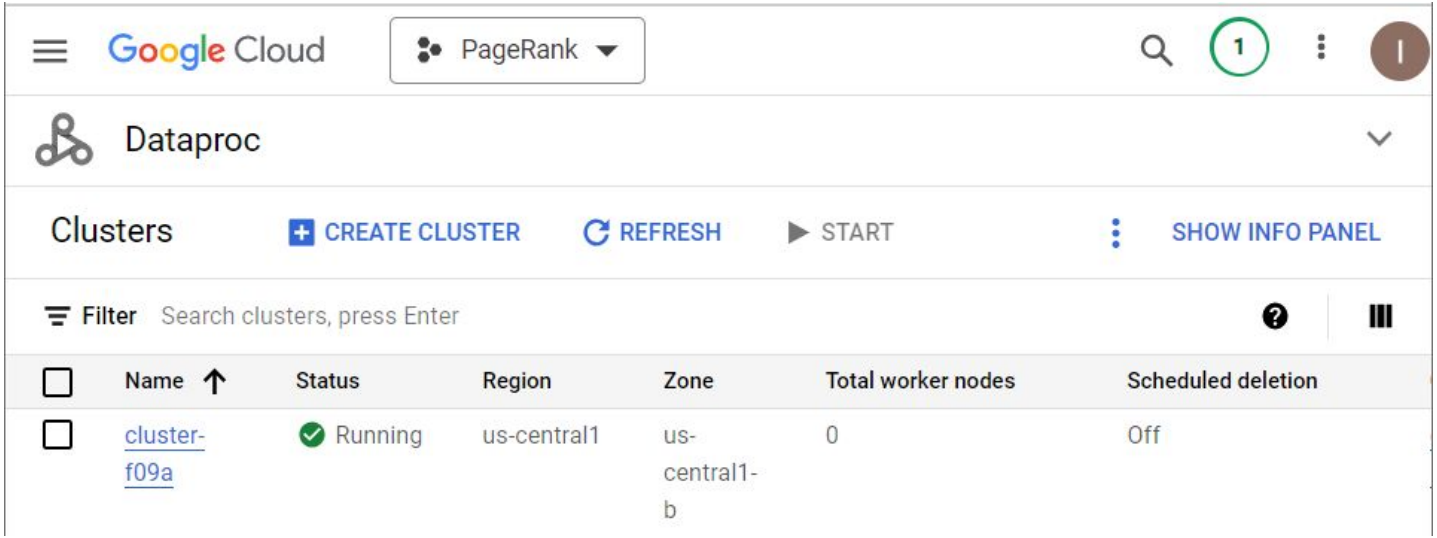
Calculation

1. Initialize each page's rank to 1.0
2. On each iteration, have page p send a contribution of rank(p) / num Neighbors(p) to its neighbors (the pages it has links to).
3. Set each page's rank to 0.15 + 0.85 * contributions Received. Note: 0.85 is the damping factor
4. A has 1 link to C, B has 1 link to C, A has 2 links to B, C

Page Rank of A=> (1-d)+d*(PR(B)/1+(PC)/1)=(0.15+0.85(1+1))=1.85
Page Rank of B=> (1-d)+d*(PR(A)/2+(PC)/1)=(0.15+0.85(0.5+1))=1.425
Page Rank of C=> (1-d)+d*(PR(A)/2+(PB)/1)=(0.15+0.85(0.5+1))=1.425



Implementation:DataPorc



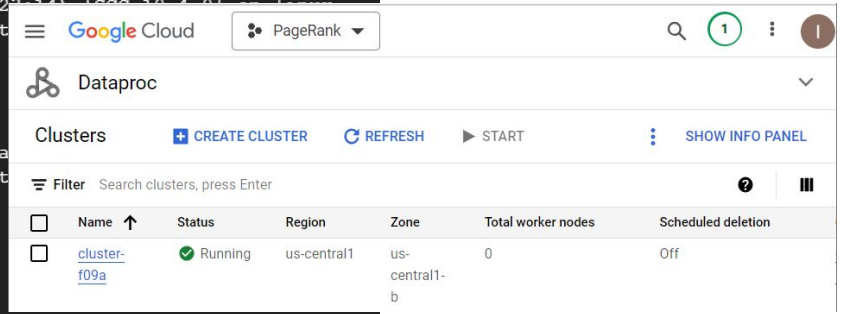
Implementation:DataPorc(PySpark)

```
isaleh@cloudshell:~ (pagerank-396403)$ gcloud dataproc jobs submit pyspark pagerank.py -- clus
ter=cluster-f09a -- region=us-central1 -- pagerank2/Input/input.txt 2
Job [752684906f88453c9c9af33196fcab16] submitted.
Waiting for job output...
23/06/16 16:54:07 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/06/16 16:54:07 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/06/16 16:54:07 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/06/16 16:54:08 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
23/06/16 16:54:08 INFO org.sparkproject.jetty.util.log: Logging initialized @2841ms to org.sparkproject.jetty.util
.log.Slf4jLog
23/06/16 16:54:08 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.66
8Z; git: b881a572662e1943a14ae12e7e1207989f218b74; jvm 1.8.0_372-b07
23/06/16 16:54:08 INFO org.sparkproject.jetty.server.Server: Started @2938ms
23/06/16 16:54:08 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@341915bc{HTTP/1.1,
 (http/1.1)}{0.0.0.0:40315}
23/06/16 16:54:08 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at cluster-1a27-m/10.1
28.0.7:8032
23/06/16 16:54:09 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at cluster
-1a27-m/10.128.0.7:10200
23/06/16 16:54:10 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
23/06/16 16:54:10 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/06/16 16:54:10 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_16
86930633758_0005
23/06/16 16:54:11 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at cluster-1a27-m/10.1
28.0.7:8030
23/06/16 16:54:14 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.
B has rank: 0.575.
C has rank: 1.06375.
A has rank: 1.3612499999999996.
```

Implementation: DataPorc+Scala

```
isaleh@cluster-b444-m:~$  pyspark
Python 3.10.8 | packaged by conda-forge | (main, Nov 22 2022, 08:23:14) [GCC 10.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/08/13 03:29:04 INFO SparkEnv: Registering MapOutputTracker
23/08/13 03:29:04 INFO SparkEnv: Registering BlockManagerMaster
23/08/13 03:29:04 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
23/08/13 03:29:04 INFO SparkEnv: Registering OutputCommitCoordinator
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.0
      /_/

Using Python version 3.10.8 (main, Nov 22 2022 08:23:14)
Spark context Web UI available at http://cluster-b444-m.us-central1-b.c.firstspark-395802.internal:41925
Spark context available as 'sc' (master = yarn, app id = application_1691895175249_0001).
SparkSession available as 'spark'.
>>> # The SparkContext 'sc' is already defined
>>> #set the Logging Level to ERROR only
>>> sc.setLogLevel("ERROR")
>>>
```



Implementation: DataPorc+Scala

```
scala> val lines = sc.textFile("hdfs:///mydata/input.txt")
lines: org.apache.spark.rdd.RDD[String] = hdfs:///mydata/input.txt MapPartitionsRDD[1] at text
File at <console>:23

scala>

scala> val links = lines.map{ s =>
     |     val parts = s.split("\\s+")
     |     (parts(0), parts(1))
     | }.distinct().groupByKey().cache()
links: org.apache.spark.rdd.RDD[(String, Iterabl
<console>:26

scala>

scala> var ranks = links.mapValues(v => 1.0)
ranks: org.apache.spark.rdd.RDD[(String, Double)
le>:23

scala>

scala> for (i <- 1 to 10) {
     |     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 * _)
     | }

scala>

scala> val output = ranks.collect()
output: Array[(String, Double)] = Array((B,0.6432494117885129), (A,1.1667391764027368), (C,1.1
900114118087488))
```

```
scala> val output = ranks.collect()
output: Array[(String, Double)] = Array((B,0.6432494117885129), (A,1.1667391764027368), (C,1.1
900114118087488))

scala> output.foreach(tup => println(tup._1 + " has rank: " + tup._2 + "."))
B has rank: 0.6432494117885129.
A has rank: 1.1667391764027368.
C has rank: 1.1900114118087488.

scala>
```