PageRank Implementation using PySpark and Scala on GCP

A comprehensive overview and comparison

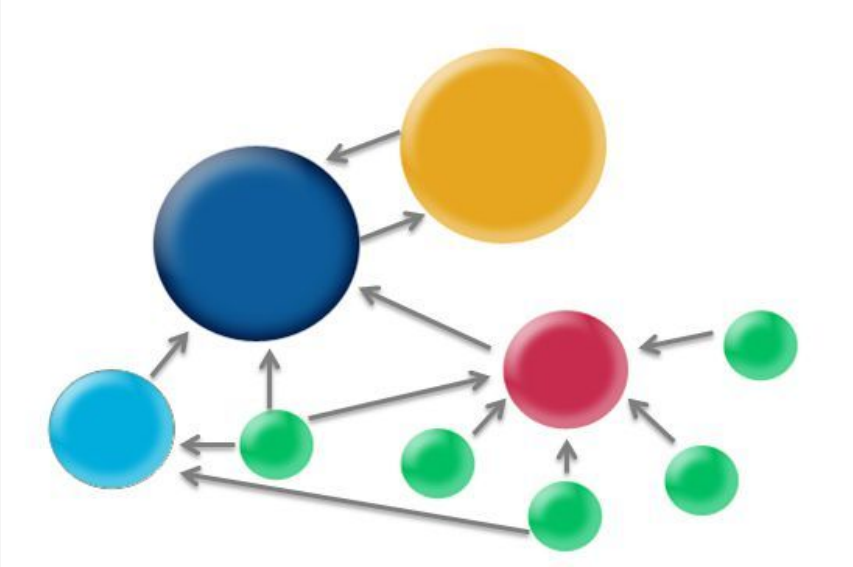
CS570 Big Data Processing Project
By Feven Araya
Instructor: Dr. Chang, Henry

Table of contents

01

Introduction

Explanation of PageRank algorithm
Manual Implementation
Objective of the project

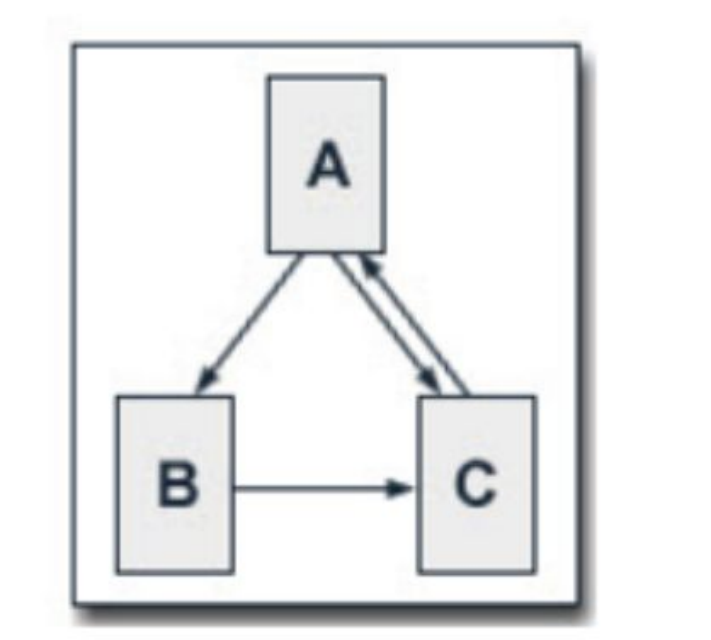


Explanation of PageRank algorithm

PageRank is a link analysis algorithm primarily used to determine the importance of web pages based on link structures. Here are two key points about PageRank:

1. **Authority Based on Inbound Links**: PageRank evaluates the quality and quantity of links to a webpage to determine a rough estimate of the website's importance. The underlying assumption is that more important websites are likely to receive more links from other websites.
2. **Recursive Calculation**: It operates on a recursive principle where a page's rank is derived from the sum of the ranks of all pages that link to it, each contributing a portion of its rank based on the number of outbound links it has. This calculation incorporates a damping factor to handle the scenario where pages do not link out to any other page, ensuring the model remains stable and converges over time.

Manual PageRank Calculation



Webpage A links to B and C.

Webpage B links to C.

Webpage C links back to A.

Initial Setup:

Each webpage starts with a PageRank value of 1.

Damping factor (d) = 0.85.

First Iteration:

PR(A) = 1−d+dx(PR(C)/1) = 1−0.85+0.85x1 = 1

PR(B) = 1−d+dx(PR(A)/2) = 1−0.85+0.85x1/2 = 0.575

PR(C) = 1−d+dx((PR(A)/2)+PR(B)/1) = 1−0.85+0.85x(0.5+ 1) = 1.425

Second Iteration:

PageRank (A) = 1 – 0.85 + 0.85 * 1.425 = 1.36125

PageRank (B) = 1 – 0.85 + 0.85 * 0.5 = 0.575

PageRank (C) = 1 – 0.85 + 0.85 * 1.075 = 1.06375

Objective

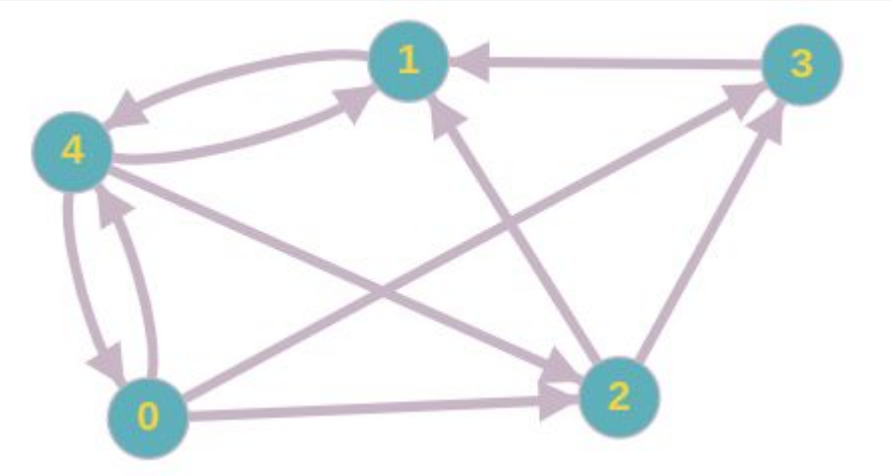
We gonna calculate PageRank on GCP using

1. Scala
2. Pyspark

02

Design

This section will discuss about the process and methods designed to calculate pagerank



Technology used

- Technologies used: GCP, PySpark, Scala

Code structure

- Home
  - PageRank_Pyspark
    - PageRank.py
    - Input.txt
  - PageRank_Scala
    - src/main/scala/SparkPageRank.scala
    - build.sbt
    - input.txt

03

Implementation

Getting ready to implement

1. PageRank + PySpark + GCP

```
faraya85431@cs570ubuntu:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Hit:3 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
0% [Connecting to security.ubuntu.com]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:5 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3388 kB]
Get:6 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [531 kB]
Get:7 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3032 kB]
Get:8 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [424 kB]
Get:9 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1195 kB]
Get:10 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [288 kB]
Get:11 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [27.1 kB]
Get:12 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [7936 B]
Get:13 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3014 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [451 kB]
Get:15 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [2911 kB]
Get:16 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [407 kB]
Get:17 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [976 kB]
Get:18 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [206 kB]
Get:19 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [24.8 kB]
Get:20 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5968 B]
Fetched 17.1 MB in 4s (4455 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
16 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

This command updates the package index files from their sources, ensuring that your local list of available packages is up to date.

```
faraya85431@cs570ubuntu:~$ sudo apt install openjdk-11-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
openjdk-11-jdk is already the newest version (11.0.23+9-1ubuntu1~20.04.2).
openjdk-11-jdk set to manually installed.
The following packages were automatically installed and are no longer required:
  genders libgenders0
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
```

The command sudo apt install openjdk-11-jdk confirms that the latest version of OpenJDK 11 is already installed on the system

```
faraya85431@cs570ubuntu:~$ sudo apt install scala
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer requir
  genders libgenders0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libhawtjni-runtime-java libjansi-java libjansi-native-java libjline2-java
Suggested packages:
  scala-doc
The following NEW packages will be installed:
  libhawtjni-runtime-java libjansi-java libjansi-native-java libjline2-java
0 upgraded, 8 newly installed, 0 to remove and 16 not upgraded.
Need to get 25.0 MB of archives.
After this operation, 28.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal/universe amd64
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal/universe amd64
```

The command installs scala.

```
faraya85431@cs570ubuntu:~$ echo "deb https://repo.scala-sbt.org/scalasbt/debian all main" | sudo tee /etc/apt/sources.list.d/sbt.list
deb https://repo.scala-sbt.org/scalasbt/debian all main
faraya85431@cs570ubuntu:~$ curl -sL "https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x99E82A75642AC823" | sudo apt-key add
OK
faraya85431@cs570ubuntu:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://scala.jfrog.io/artifactory/debian all InRelease [4410 B]
Get:6 https://scala.jfrog.io/artifactory/debian all/main amd64 Packages [2733 B]
Fetched 7143 B in 1s (6774 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
16 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

The commands add the Scala SBT repository to the system's package sources and import its key, then update the package lists to include the new repository.

```
faraya85431@cs570ubuntu:~$ sudo apt install sbt
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  genders libgenders0
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  sbt
0 upgraded, 1 newly installed, 0 to remove and 16 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 1s (28.4 kB/s)
Selecting previously unselected package sbt.
(Reading database ... 107991 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 (2.9.1-1) ...
```

The command sudo apt install sbt installs the Scala Build Tool (sbt) on the system, fetching and setting up the necessary files and dependencies.

```
faraya85431@cs570ubuntu:~$ cp -r ~/PySparkPiProject/spark-3.1.2-bin-hadoop3.2 ~/PageRank-Scala/
faraya85431@cs570ubuntu:~$ ls ~/PageRank-Scala
build.sbt  input.txt  project  spark-3.1.2-bin-hadoop3.2
```

```
faraya85431@cs570ubuntu:~$ echo 'export SPARK_HOME=~/PageRank-Scala/spark-3.1.2-bin-hadoop3.2' >> ~/.bashrc
faraya85431@cs570ubuntu:~$ echo 'export PATH=$SPARK_HOME/bin:$PATH' >> ~/.bashrc
faraya85431@cs570ubuntu:~$ echo 'export PATH=$SPARK_HOME/sbin:$PATH' >> ~/.bashrc
faraya85431@cs570ubuntu:~$ source ~/.bashrc
```

These commands set environment variables for Spark by appending the necessary paths to the .bashrc file and then reloading the file to apply the changes.

Create input.txt file

input.txt

1	A	B
2	A	C
3	B	C
4	C	A

Line 1: A links to B
Line 2: A links to C
Line 3: B links to C
Line 4: C links to A

Create build.sbt file

build.sbt

```
name := "SparkPageRank"

version := "0.1"

scalaVersion := "2.12.19"

libraryDependencies ++= Seq(
  "org.apache.spark" %% "spark-core" % "3.1.1",
  "org.apache.spark" %% "spark-sql" % "3.1.1",
  "org.apache.spark" %% "spark-graphx" % "3.1.1"
)

mainClass in Compile := Some("SparkPageRank")
```

This build.sbt file configures a Scala project named ”SparkPageRank” with version ”0.1”, using Scala version ”2.12.19”, and includes dependencies for Spark Core, SQL, and GraphX, specifying the main class as ”SparkPageRank”.

Create plugins.sbt

plugins.sbt

```
addSbtPlugin("ch.epfl.scala" % "sbt-bloop" % "1.4.8")
```

Create build.sbt file

build.sbt

```
name := "SparkPageRank"

version := "0.1"

scalaVersion := "2.12.19"

libraryDependencies ++= Seq(
  "org.apache.spark" %% "spark-core" % "3.1.1",
  "org.apache.spark" %% "spark-sql" % "3.1.1",
  "org.apache.spark" %% "spark-graphx" % "3.1.1"
)

mainClass in Compile := Some("SparkPageRank")
```

This build.sbt file configures a Scala project named ”SparkPageRank” with version ”0.1”, using Scala version ”2.12.19”, and includes dependencies for Spark Core, SQL, and GraphX, specifying the main class as ”SparkPageRank”.

SparkPageRank.scala

```scala
import org.apache.spark._
import org.apache.spark.graphx._
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf
run | debug
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").setMaster("local")
    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.flatMap { s =>
      val parts = s.split("\\s+")
      if (parts.length >= 2) {
        Some((parts(0), parts(1)))
      } else {
        println(s"Invalid line: $s")
        None
      }
    }.distinct().groupByKey().cache()

    // Display the result of links
    var ranks = links.mapValues(v => 1.0)
    // Display the result of ranks
    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))
      }
      // Display the result of contribs
      ranks = contribs.reduceByKey(_ + _).mapValues(0.15 + 0.85 * _)
      // Display the result of ranks
    }

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

    ctx.stop()
  }
}
```

This Scala code defines a Spark application that computes the PageRank of web pages using the GraphX library in Apache Spark, and includes a warning that it is a simplified implementation meant for educational purposes.

Create Directory- PageRank-Scala

```
faraya85431@cs570ubuntu:~/PageRank-Scala$ mkdir -p ~/PageRank-Scala/project
```

```
faraya85431@cs570ubuntu:~/PageRank-Scala$ mv ~/build.sbt .
faraya85431@cs570ubuntu:~/PageRank-Scala$ mv ~/input.txt .
faraya85431@cs570ubuntu:~/PageRank-Scala$ mv ~/build.properties project/
faraya85431@cs570ubuntu:~/PageRank-Scala$ mv ~/plugins.sbt project/
faraya85431@cs570ubuntu:~/PageRank-Scala$
```

Move the files into the correct directories

2. PageRank + Scala + GCP

```
faraya85431@cs570ubuntu:~$ mkdir PageRank_Pyspark
faraya85431@cs570ubuntu:~$ ls
PageRank-Python  PageRank-Scala  PageRank_Pyspark
```

Create a new directory PageRank_Pyspark

```
faraya85431@cs570ubuntu:~$ sudo apt update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-central1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 https://scala.jfrog.io/artifactory/debian all InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
16 packages can be upgraded. Run 'apt list --upgradable' to see them.
faraya85431@cs570ubuntu:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  genders libgenders0
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
```

Perform a package update using sudo apt update and then installs the Python package installer python3-pip using sudo apt install python3-pip.

Install pyspark

```
faraya85431@cs570ubuntu:~$ pip3 install pyspark
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     |                                        | 317.0 MB 20 kB/s
Collecting py4j==0.10.9.7
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)
     |                                        | 200 kB 48.7 MB/s
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
  Stored in directory: /home/faraya85431/.cache/pip/wheels/da/78/6d/
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.7 pyspark-3.5.1
```

PageRank.py

```python
from pyspark import SparkConf, SparkContext

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

def parseNeighbors(urls):
    parts = urls.split()
    if len(parts) >= 2:
        return parts[0], parts[1]
    else:
        return None

if __name__ == "__main__":
    conf = SparkConf().setAppName("PythonPageRank").setMaster("local")
    sc = SparkContext(conf = conf)

    # Load input file
    lines = sc.textFile("input.txt")

    # Parse neighbors
    links = lines.map(parseNeighbors).filter(lambda x: x is not None).distinct().groupByKey().cache()

    # Initialize ranks
    ranks = links.map(lambda url_neighbors: (url_neighbors[0], 1.0))

    # Number of iterations
    iterations = 10

    for iteration in range(iterations):
        # Calculate contributions
        contribs = links.join(ranks).flatMap(lambda url_urls_rank: computeContribs(url_urls_rank[1][0], url_urls_rank[1][1]))

        # Update ranks
        ranks = contribs.reduceByKey(lambda x, y: x + y).mapValues(lambda rank: 0.15 + 0.85 * rank)

        # Collect and print the ranks for the current iteration
        output = ranks.collect()
        print(f"Iteration {iteration + 1}")
        for (link, rank) in output:
            print(f"{link} has rank: {rank}")

    sc.stop()
```

Python code using PySpark to implement the PageRank algorithm, where it reads an input file, parses neighbor relationships, initializes ranks, iterates to compute PageRank contributions, updates ranks, and finally prints the PageRank of each URL.

04

Test

Process to test the project

1. Caculate PageRank Using Scala

```
faraya85431@cs570ubuntu:~/PageRank-Scala$ sbt clean compile
downloading sbt launcher 1.10.0
copying runtime jar...
[info] [launcher] getting org.scala-sbt sbt 1.10.0  (this may take some time)...
[info] [launcher] getting Scala 2.12.19 (for sbt)...
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.jline.terminal.impl.exec.ExecTerminalProvider$ReflectionRedirectPipeCr
sbt/1.10.0/jline-terminal-3.24.1.jar) to constructor java.lang.ProcessBuilder$RedirectPipeImpl()
WARNING: Please consider reporting this to the maintainers of org.jline.terminal.impl.exec.ExecTerminalProvider$
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[info] welcome to sbt 1.10.0 (Ubuntu Java 11.0.23)
[info] loading settings for project pagerank-scala-build from plugins.sbt ...
[info] loading project definition from /home/faraya85431/PageRank-Scala/project
[info] Updating pagerank-scala-build
```

First Iteration

```
faraya85431@cs570ubuntu:~/PageRank-Scala$ sbt "runMain SparkPageRank input.txt 1"
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.jline.terminal.impl.exec.ExecTerminalProvider$ReflectionRedirect
sbt/1.10.0/jline-terminal-3.24.1.jar) to constructor java.lang.ProcessBuilder$RedirectPipeImpl()
WARNING: Please consider reporting this to the maintainers of org.jline.terminal.impl.exec.ExecTerminalPro
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[info] welcome to sbt 1.10.0 (Ubuntu Java 11.0.23)
[info] loading settings for project pagerank-scala-build from plugins.sbt ...
[info] loading project definition from /home/faraya85431/PageRank-Scala/project
[info] loading settings for project pagerank-scala from build.sbt ...
[info] set current project to SparkPageRank (in build file:/home/faraya85431/PageRank-Scala/)
[info] running SparkPageRank input.txt 1

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.

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
24/06/26 21:48:23 INFO SparkContext: Running Spark version 3.1.1
24/06/26 21:48:23 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
24/06/26 21:48:24 INFO ResourceUtils: ==============================================================
24/06/26 21:48:24 INFO ResourceUtils: No custom resources configured for spark.driver.
24/06/26 21:48:24 INFO ResourceUtils: ==============================================================
```

```
24/06/26 21:48:29 INFO Executor: Finished task 0.0 in stage 3.0 (TID
24/06/26 21:48:29 INFO TaskSetManager: Finished task 0.0 in stage 3.0
24/06/26 21:48:29 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose
24/06/26 21:48:29 INFO DAGScheduler: ResultStage 3 (collect at Spark
24/06/26 21:48:29 INFO DAGScheduler: Job 0 is finished. Cancelling po
24/06/26 21:48:29 INFO TaskSchedulerImpl: Killing all running tasks i
24/06/26 21:48:29 INFO DAGScheduler: Job 0 finished: collect at Spark
B has rank: 0.575.
A has rank: 1.0.
C has rank: 1.4249999999999998.
24/06/26 21:48:29 INFO SparkUI: Stopped Spark web UI at http://cs570u
24/06/26 21:48:29 INFO MapOutputTrackerMasterEndpoint: MapOutputTrack
24/06/26 21:48:29 INFO MemoryStore: MemoryStore cleared
24/06/26 21:48:29 INFO BlockManager: BlockManager stopped
24/06/26 21:48:29 INFO BlockManagerMaster: BlockManagerMaster stopped
24/06/26 21:48:29 INFO OutputCommitCoordinator$OutputCommitCoordinato
24/06/26 21:48:29 INFO SparkContext: Successfully stopped SparkContex
[success] Total time: 11 s, completed Jun 26, 2024, 9:48:29 PM
```

In the first Iteration , It gave as the page rank as

PR(A) = 1

PR(B) =0.575

PR(C) = = 1.425

Second Iteration

```
faraya85431@cs570ubuntu:~/PageRank-Scala$ sbt "runMain SparkPageRank input.txt 2"
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.jline.terminal.impl.exec.ExecTerminalPro
sbt/1.10.0/jline-terminal-3.24.1.jar) to constructor java.lang.ProcessBuilder$Redi
WARNING: Please consider reporting this to the maintainers of org.jline.terminal.i
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflectiv
WARNING: All illegal access operations will be denied in a future release
[info] welcome to sbt 1.10.0 (Ubuntu Java 11.0.23)
[info] loading settings for project pagerank-scala-build from plugins.sbt ...
[info] loading project definition from /home/faraya85431/PageRank-Scala/project
[info] loading settings for project pagerank-scala from build.sbt ...
```

```
24/06/26 21:50:08 INFO Executor: Finished task 0.0 in stage 4.0 (TID 4). 1503 byt
24/06/26 21:50:08 INFO TaskSetManager: Finished task 0.0 in stage 4.0 (TID 4) in
24/06/26 21:50:08 INFO TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have a
24/06/26 21:50:08 INFO DAGScheduler: ResultStage 4 (collect at SparkPageRank.scal
24/06/26 21:50:08 INFO DAGScheduler: Job 0 is finished. Cancelling potential spec
24/06/26 21:50:08 INFO TaskSchedulerImpl: Killing all running tasks in stage 4: S
24/06/26 21:50:08 INFO DAGScheduler: Job 0 finished: collect at SparkPageRank.sca
B has rank: 0.575.
A has rank: 1.3612499999999996.
C has rank: 1.06375.
24/06/26 21:50:08 INFO SparkUI: Stopped Spark web UI at http://cs570ubuntu.us-cen
24/06/26 21:50:08 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndp
24/06/26 21:50:08 INFO MemoryStore: MemoryStore cleared
24/06/26 21:50:08 INFO BlockManager: BlockManager stopped
24/06/26 21:50:08 INFO BlockManagerMaster: BlockManagerMaster stopped
24/06/26 21:50:08 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: O
```

In the second Iteration , It gave as the pagerank as :

PageRank (A) = 1.36125

PageRank (B)= + 0.575

PageRank (C) = 1.06375

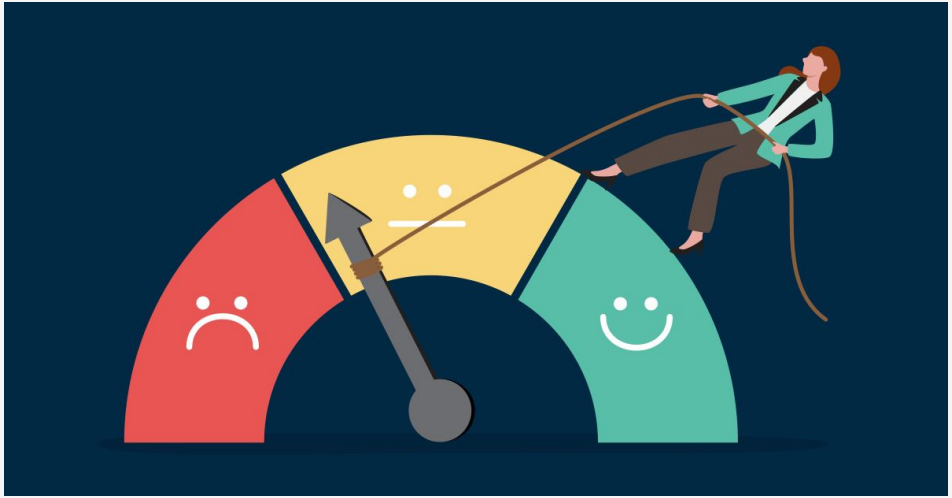
Caculate PageRank Using Pyspark

```
faraya85431@cs570ubuntu:~$ spark-submit pagerank.py
24/06/26 21:57:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
24/06/26 21:57:18 INFO SparkContext: Running Spark version 3.1.2
24/06/26 21:57:18 INFO ResourceUtils: ==============================================================
24/06/26 21:57:18 INFO ResourceUtils: No custom resources configured for spark.driver.
24/06/26 21:57:18 INFO ResourceUtils: ==============================================================
24/06/26 21:57:18 INFO SparkContext: Submitted application: PythonPageRank
```

```
24/06/26 22:03:24 INFO DAGScheduler: Job 0 finished: collect at /home/faraya85431/PageRank_Pyspark/pagerank.py:39, took 3.479157 s
Iteration 1
C has rank: 1.4249999999999998
A has rank: 1.0
B has rank: 0.575
24/06/26 22:03:24 INFO SparkContext: Starting job: collect at /home/faraya85431/PageRank_Pyspark/pagerank.py:39
```

```
24/06/26 22:03:25 INFO DAGScheduler: Job 1 finished: collect at /home/faraya85431/PageRank_Pyspark/pagerank.py:39, took 0.757555 s
Iteration 2
C has rank: 1.06375
B has rank: 0.575
A has rank: 1.3612499999999996
24/06/26 22:03:25 INFO SparkContext: Starting job: collect at /home/faraya85431/PageRank_Pyspark/pagerank.py:39
```

The Page rank for both iterations are calculated using the command spark-submit pagerank.py and gave us same output as what we did using scala.



05 Enhancements

Can we get better result?

- Distributed Computing: Expanding the use of distributed computing frameworks and clusters on GCP to scale the PageRank computation for even larger graphs and datasets.
- Integration with Other Libraries: Integrating with additional data processing and machine learning libraries (e.g., GraphFrames, MLlib) to enhance the functionality and performance of the PageRank algorithm.
- Real-time PageRank Calculation: Implementing real-time PageRank calculations using streaming data frameworks such as Apache Kafka and Spark Streaming.
- Algorithm Enhancements: Exploring and implementing advanced variants of the PageRank algorithm, such as personalized PageRank or topic-sensitive PageRank, to provide more customized and relevant results.

06

Conclusion

- Consistent Results: Both Scala and PySpark implementations of the PageRank algorithm on GCP produced the same results, validating the accuracy of the implementations.
- Verification through Manual Calculation: The manual calculations of PageRank values for each node for the first two iterations matched the outputs from both Scala and PySpark implementations.
- Reliable Technology Stack: Utilizing GCP for deploying and executing the PageRank algorithm with both Scala and PySpark proved to be efficient and reliable, demonstrating the robustness of the platform.

07

References

PySpark Page Rank

Scala Page Rank

ScalaProgramming Language

Thanks!