Project : PageRank using Scala and PySpark

Student :

Presented by MANICKAM RAVISEKAR ,
Master of Science in Computer Science, 19599 , Fall Semester 2022

Professor : Dr Henry Chung
TA : Liang

SAN FRANCISCO BAY
UNIVERSITY
47671 WestingHouse Dr.,
Fremont, CA 94539

Contents

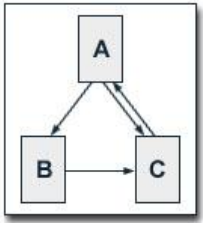
Abstract

In this project to learn basic graph in Pyspark and Scala used in Big Data to find the PageRank of a given graph.
The primary learning goal of the project is to gain familiarity with the syntax, data structures to learn scala , pyspark.
Also learning the computation involved in finding the pagerank of a given graph.

Thank full to professor Dr Henry who encouraged us to work on this assignment on google cloud platform.

Adjacency Matrix of the Graph

Row-Column	A	B	C	No of Links
A	-	1	1	2
B	-	-	1	1
C	1	-	-	1



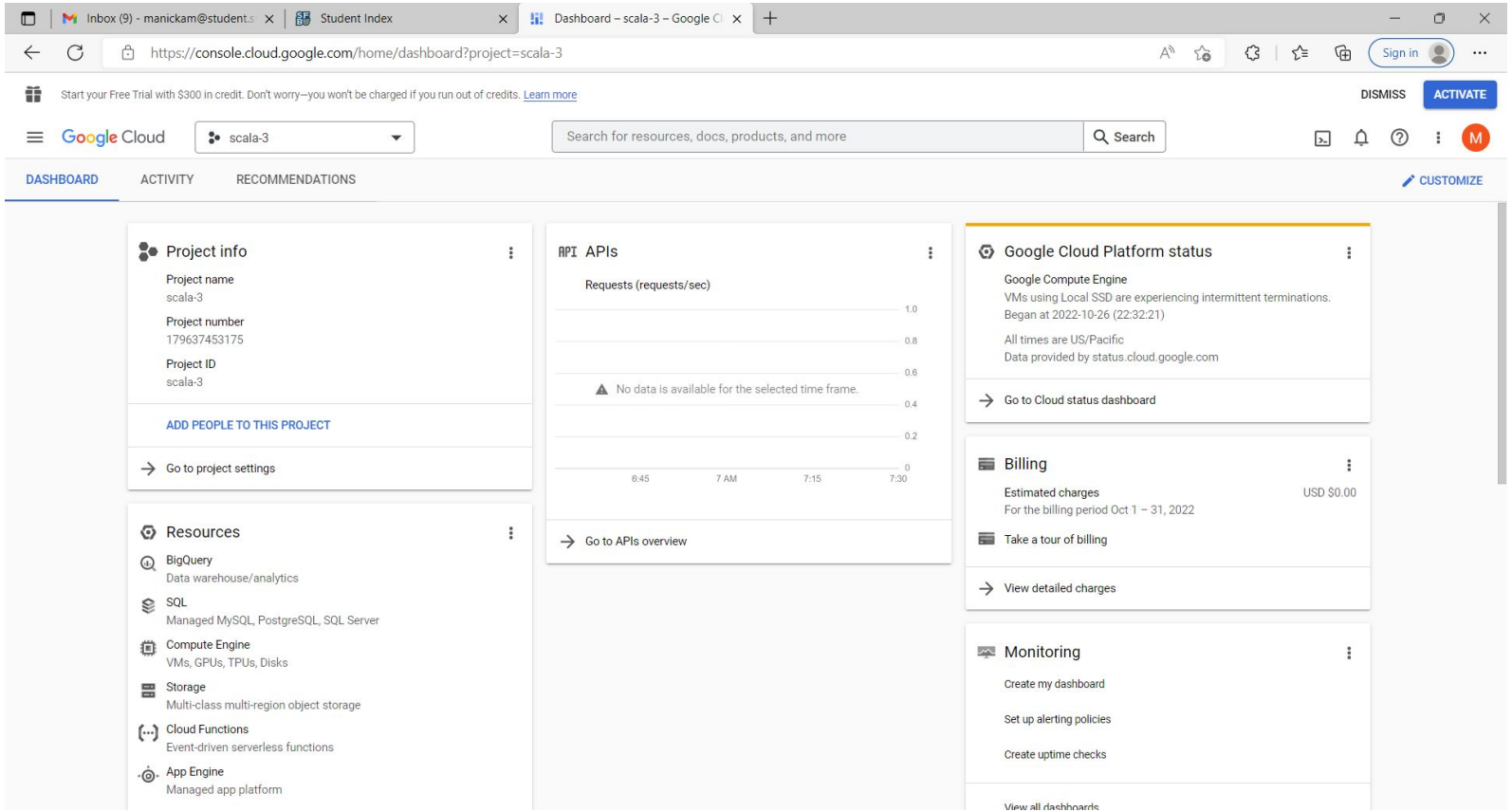
Page Rank Iterations Values

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.575	1.425
2	1.36125	0.575	1.06375
3	1.0541875	0.72853125	1.21728125

Process of Calculating PageRank : Initialize each page's rank to 1.0
On each iteration, have page p send a contribution of rank(p) / numNeighbors(p) to its neighbors (the pages it has links to). Set each page's rank to 0.15 + 0.85 * contributionsReceived.
Note: 0.85 is the damping factor

PageRank overview
If The initial PageRank value for each webpage is 1.
PR(A) = 1 PR(B) = 1 PR(C) = 1
Page B has a link to pages C and A ,Page C has a link to page A ,Page D has links to all three pages
And A's PageRank is
PR(A) = (1-d) + d * (PR(B) / 2 + PR(C) / 1 + PR(D) / 3)
B's PageRank is
PR(B) = (1-d) + d * (PR(D) / 3)
C's PageRank is
PR(C) = (1-d) + d * (PR(B) / 2 + PR(D) / 3)
D's PageRank is
PR(D) = 1-d
Damping factor is 0.85
Then after 1st iteration
Output
Page B would transfer half of its existing value, or 0.5, to page A and the other half, or 0.5, to page C.
Page C would transfer all of its existing value, 1, to the only page it links to, A.
Since D had three outbound links, it would transfer one third of its existing value, or approximately 0.33, to A.
Input
PR(A)
= (1-d) + d * (PR(B) / 2 + PR(C) / 1 + PR(D) / 3)
= (1-0.85) + 0.85 * (0.5 + 1 + 0.33)
= 1.71
PR(B)
= (1-d) + d * (PR(D) / 3)
= (1-0.85) + 0.85 * 0.33
= 0.43
PR(C)
= (1-d) + d * (PR(B) / 2 + PR(D) / 3)
= (1-0.85) + 0.85 * (0.5 + 0.33)
= 0.86
PR(D)
= 1-d
= 0.15

Cluster creation of Google Cloud Platform



Snoozed - manickam@student.s | Student Index | APIs & Services – APIs & Service

https://console.cloud.google.com/apis/dashboard?project=scala-3&show=all

Sign in

Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

DISMISS ACTIVATE

Google Cloud | scala-3 | Search for resources, docs, products, and more | Search

APIs & Services

- Enabled APIs & services
- Library
- Credentials
- OAuth consent screen
- Domain verification
- Page usage agreements

APIs & Services

+ ENABLE APIS AND SERVICES

1 hour | 6 hours | 12 hours | ✓ 1 day | 2 days | 4 days | 7 days | 14 days | 30 days

Traffic

1.0/s
0.8/s
0.6/s
0.4/s
0.2/s
0

No data is available for the selected time frame.

12 PM | 6 PM | Mon 31 | 6 AM

Errors

100%
80%
60%
40%
20%
0

No data is available for the selected time frame.

12 PM | 6 PM | Mon 31 | 6 AM

Median latency

1.0
0.8
0.6
0.4
0.2
0

No data is available for the selected time frame.

12 PM | 6 PM | Mon 31 | 6 AM

Filter | Filter

Name	Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
BigQuery API				
BigQuery Migration API				
BigQuery Storage API				
Cloud Datastore API				
Cloud Debugger API				
Cloud Logging API				
Cloud Monitoring API				
Cloud SQL				

Snoozed - manickam@student.s | Student Index | APIs & Services – APIs & Service

https://console.cloud.google.com/apis/dashboard?project=scala-3&show=all

Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

DISMISS ACTIVATE

Google Cloud | scala-3 | Search for resources, docs, products, and more | Search

API APIs & Services

- Enabled APIs & services
- Library
- Credentials
- OAuth consent screen
- Domain verification
- Page usage agreements

APIs & Services + ENABLE APIS AND SERVICES

1 hour 6 hours 12 hours ✓ 1 day 2 days 4 days 7 days 14 days 30 days

Traffic

1.0/s 0.8/s 0.6/s 0.4/s 0.2/s 0

No data is available for the selected time frame.

12 PM 6 PM Mon 31 6 AM

Errors

100% 80% 60% 40% 20% 0

No data is available for the selected time frame.

12 PM 6 PM Mon 31 6 AM

Median latency

1.0 0.8 0.6 0.4 0.2 0

No data is available for the selected time frame.

12 PM 6 PM Mon 31 6 AM

Filter Filter

Name	Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
BigQuery API				
BigQuery Migration API				
BigQuery Storage API				
Cloud Datastore API				
Cloud Debugger API				
Cloud Logging API				
Cloud Monitoring API				
Cloud SQL				

Snoozed - manickam@student.s | Student Index | Cloud Dataproc API – Marketpla

https://console.cloud.google.com/marketplace/product/google/dataproc.googleapis.com?returnUrl=%2Fdataproc%2Fclusters%3Fproject%3Dscala-3&project=scal...

Sign in

Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

DISMISS ACTIVATE

Google Cloud scala-3

Cloud Dataproc API

Google Enterprise API

Manages Hadoop-based clusters and jobs on Google Cloud Platform.

TRY THIS API

OVERVIEW DOCUMENTATION

Overview

Manages Hadoop-based clusters and jobs on Google Cloud Platform.

Additional details

Type: SaaS & APIs
Last updated: 7/21/22
Category: Google Enterprise APIs
Service name: dataproc.googleapis.com

Tutorials and documentation

Learn more

Terms of Service

Snoozed - manickam@student.s | Student Index | Clusters – Dataproc – scala-3 – C

https://console.cloud.google.com/dataproc/clusters?project=scala-3

Sign in

Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

DISMISS ACTIVATE

Google Cloud | scala-3 | Search for resources, docs, products, and more | Search

Dataproc

Clusters | CREATE CLUSTER | REFRESH | START | STOP | DELETE | REGIONS | + 5 RECOMMENDED ALERTS

Jobs on Clusters
- Clusters
- Jobs
- Workflows
- Autoscaling policies

Serverless
- Batches

Metastore Services
- Metastore
- Federation

Utilities
- Component exchange
- Workbench

Release Notes

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

Snoozed - manickam@student.s | Student Index | Create a Dataproc cluster on Co

https://console.cloud.google.com/dataproc/clustersAdd?computeInfra=gce&project=scala-3

Sign in

Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

DISMISS ACTIVATE

Google Cloud | scala-3 | Search for resources, docs, products, and more | Search

Dataproc

Jobs on Clusters
- Clusters
- Jobs
- Workflows
- Autoscaling policies

Serverless
- Batches

Metastore Services
- Metastore
- Federation

Utilities
- Component exchange
- Workbench

Release Notes

Create a Dataproc cluster on Compute Engine

- **Set up cluster**
  Begin by providing basic information.
- **Configure nodes** (optional)
  Change node compute and storage capabilities.
- **Customize cluster** (optional)
  Add cluster properties, features, and actions.
- **Manage security** (optional)
  Change access, encryption, and security settings.

CREATE CANCEL

EQUIVALENT COMMAND LINE

Name

Cluster Name *
cluster-b74f

Location

Region *
us-central1

Zone *
us-central1-a

Cluster type

- (●) Standard (1 master, N workers)
- ( ) Single Node (1 master, 0 workers)
  Provides one node that acts as both master and worker. Good for proof-of-concept or small-scale processing
- ( ) High Availability (3 masters, N workers)
  Hadoop High Availability mode provides uninterrupted YARN and HDFS operations despite single-node failures or reboots

Autoscaling

Automates cluster resource management based on an autoscaling policy.

Policy
None

Enhanced Flexibility Mode

Dataproc Enhanced Flexibility Mode (EFM) manages shuffle data to minimize job progress delays caused by the removal of nodes from a running cluster. EFM offloads shuffle data in one of two user-selectable modes, primary worker shuffle and Hadoop

Snoozed - manickam@student.s | Student Index | Clusters – Dataproc – scala-3 – C

https://console.cloud.google.com/dataproc/clusters?project=scala-3

Sign in

Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

DISMISS ACTIVATE

Google Cloud | scala-3 | Search for resources, docs, products, and more | Search

Dataproc

Jobs on Clusters
- Clusters
- Jobs
- Workflows
- Autoscaling policies

Serverless
- Batches

Metastore Services
- Metastore
- Federation

Utilities
- Component exchange
- Workbench

Release Notes

Clusters | CREATE CLUSTER | REFRESH | START | STOP | DELETE | REGIONS | + 5 RECOMMENDED ALERTS | SHOW INFO PANEL

Filter Search clusters, press Enter

Name	Status	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket	Created
cluster-b74f	Running	us-central1	us-central1-a	2	Off	dataproc-staging-us-central1-179637453175-deik4x2r	Oct 31, 2022, 7:55:15 AM

Snoozed - manickam@student.s | Student Index | cluster-b74f – VM Instances – Cl

https://console.cloud.google.com/dataproc/clusters/cluster-b74f/instances?region=us-central1&project=scala-3

Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

DISMISS ACTIVATE

Google Cloud | scala-3 | Search for resources, docs, products, and more | Search

Dataproc

Jobs on Clusters
- Clusters
- Jobs
- Workflows
- Autoscaling policies

Serverless
- Batches

Metastore Services
- Metastore
- Federation

Utilities
- Component exchange
- Workbench

Release Notes

Cluster details

SUBMIT JOB | REFRESH | START | STOP | DELETE | VIEW LOGS

Consider using Auto Zone rather than selecting a zone manually. See https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/auto-zone MORE

Name	cluster-b74f
Cluster UUID	7ec93b2d-68bd-410d-98fb-bed3fc2c0330
Type	Dataproc Cluster
Status	Running

MONITORING | JOBS | VM INSTANCES | CONFIGURATION | WEB INTERFACES

Filter Filter instances

Name	Role	
cluster-b74f-m	Master	SSH
cluster-b74f-w-0	Worker	
cluster-b74f-w-1	Worker	

EQUIVALENT REST

Ssh the virtual session

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255Enabled=true&sshTrouble...

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255E...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
Linux cluster-b74f-m 5.10.0-0.deb10.16-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.
manickam@cluster-b74f-m:~$
```

Verify Scala Version

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255Enabled=true&sshTrouble...

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255E...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
Linux cluster-b74f-m 5.10.0-0.deb10.16-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.
manickam@cluster-b74f-m:~$ scala -version
Scala code runner version 2.12.14 -- Copyright 2002-2021, LAMP/EPFL and Lightbend, Inc.
manickam@cluster-b74f-m:~$
```

Input file for PageRank : pagerank.txt

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255Enabled=true&sshTrouble...

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255E...

SSH-in-browser

UPLOAD FILE DOWNLOAD FILE

```
manickam@cluster-b74f-m:~/PageRank$ cat pagerank.txt
A B
A C
B C
C A
manickam@cluster-b74f-m:~/PageRank$
```

Iteration - 1

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255Enabled=true&sshTrouble...

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255E...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
      /_/

Using Scala version 2.12.14 (OpenJDK 64-Bit Server VM, Java 1.8.0_345)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.SparkSession

scala> import org.apache.spark.HashPartitioner
import org.apache.spark.HashPartitioner

scala> val links = sc.parallelize(List(("A",List("B","C")),("B", List("C")),("C",List("A")))).partitionBy(new HashPartitioner(3)).persist()
links: org.apache.spark.rdd.RDD[(String, List[String])] = ShuffledRDD[1] at partitionBy at <console>:25

scala> var ranks = links.mapValues(v => 1.0) // Initialized
ranks: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[2] at mapValues at <console>:25

scala>

scala> for (i <- 0 to 0) {
     | val contributions = links.join(ranks).flatMap { case (url, (links, rank)) => links.map(dest => (dest, rank / links.size)) }
     | ranks = contributions.reduceByKey((x, y) => x + y).mapValues(v => 0.15 + 0.85*v)
     | ranks.collect
     | }

scala> ranks.collect
res1: Array[(String, Double)] = Array((B,0.575), (C,1.4249999999999998), (A,1.0))

scala>

scala> :quit
manickam@cluster-b74f-m:~/PageRank$
```

Iteration 2

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255Enabled=true&sshTrouble...

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255E...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
      /_/

Using Scala version 2.12.14 (OpenJDK 64-Bit Server VM, Java 1.8.0_345)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.SparkSession

scala> import org.apache.spark.HashPartitioner
import org.apache.spark.HashPartitioner

scala> val links = sc.parallelize(List(("A",List("B","C")),("B", List("C")),("C",List("A")))).partitionBy(new HashPartitioner(3)).persist()
links: org.apache.spark.rdd.RDD[(String, List[String])] = ShuffledRDD[1] at partitionBy at <console>:25

scala> var ranks = links.mapValues(v => 1.0) // Initialized
ranks: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[2] at mapValues at <console>:25

scala>

scala> for (i <- 0 to 1) {
     | val contributions = links.join(ranks).flatMap { case (url, (links, rank)) => links.map(dest => (dest, rank / links.size)) }
     | ranks = contributions.reduceByKey((x, y) => x + y).mapValues(v => 0.15 + 0.85*v)
     | ranks.collect
     | }

scala> ranks.collect
res1: Array[(String, Double)] = Array((B,0.575), (C,1.06375), (A,1.3612499999999996))

scala>

scala> :quit
manickam@cluster-b74f-m:~/PageRank$
```

Iteration 3

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255Enabled=true&sshTrouble...

https://ssh.cloud.google.com/v2/ssh/projects/scala-3/zones/us-central1-a/instances/cluster-b74f-m?authuser=0&hl=en_US&projectNumber=179637453175&useAdminProxy=true&troubleshoot4005Enabled=true&troubleshoot255E...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
      /_/

Using Scala version 2.12.14 (OpenJDK 64-Bit Server VM, Java 1.8.0_345)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.SparkSession

scala> import org.apache.spark.HashPartitioner
import org.apache.spark.HashPartitioner

scala> val links = sc.parallelize(List(("A",List("B","C")),("B", List("C")),("C",List("A")))).partitionBy(new HashPartitioner(3)).persist()
links: org.apache.spark.rdd.RDD[(String, List[String])] = ShuffledRDD[1] at partitionBy at <console>:25

scala> var ranks = links.mapValues(v => 1.0) // Initialized
ranks: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[2] at mapValues at <console>:25

scala>

scala> for (i <- 0 to 2) {
     | val contributions = links.join(ranks).flatMap { case (url, (links, rank)) => links.map(dest => (dest, rank / links.size)) }
     | ranks = contributions.reduceByKey((x, y) => x + y).mapValues(v => 0.15 + 0.85*v)
     | ranks.collect
     | }

scala> ranks.collect
res1: Array[(String, Double)] = Array((B,0.7285312499999999), (C,1.2172812499999996), (A,1.0541874999999998))

scala>

scala> :quit
manickam@cluster-b74f-m:~/PageRank$
```

Apache Pyspark :Sample Page Rank Program

```
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#    http://www.apache.org/licenses/LICENSE-2.0
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
"""
This is an example implementation of PageRank. For more conventional use,
Please refer to PageRank implementation provided by graphx
Example Usage:
bin/spark-submit examples/src/main/python/pagerank.py data/mllib/pagerank_data.txt 10
"""
import re
import sys
from operator import add
from typing import Iterable, Tuple

from pyspark.resultiterable import ResultIterable
from pyspark.sql import SparkSession

def computeContribs(urls: ResultIterable[str], rank: float) -> Iterable[Tuple[str, float]]:
    """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: str) -> Tuple[str, str]:
    """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. It should be in format of:
    #     URL         neighbor URL
    #     URL         neighbor URL
    #     URL         neighbor URL
    #     ...
    lines = spark.read.text(sys.argv[1]).rdd.map(lambda r: r[0])
    # Loads all URLs from input file and initialize their neighbors.
    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.
    ranks = links.map(lambda url_neighbors: (url_neighbors[0], 1.0))
    # Calculates and updates URL ranks continuously using PageRank algorithm.
    for iteration in range(int(sys.argv[2])):
        # Calculates URL contributions to the rank of other URLs.
        contribs = links.join(ranks).flatMap(lambda url_urls_rank: computeContribs(
            url_urls_rank[1][0], url_urls_rank[1][1]  # type: ignore[arg-type]
        ))
        # Re-calculates URL ranks based on neighbor contributions.
        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()#
```

```python
"""
This is an example implementation of PageRank. For more conventional use,
Please refer to PageRank implementation provided by graphx
Example Usage:
bin/spark-submit examples/src/main/python/pagerank.py data/mllib/pagerank_data.txt 10
"""
import re
import sys
from operator import add
from typing import Iterable, Tuple

from pyspark.resultiterable import ResultIterable
from pyspark.sql import SparkSession


def computeContribs(urls: ResultIterable[str], rank: float) -> Iterable[Tuple[str, float]]:
    """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: str) -> Tuple[str, str]:
    """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. It should be in format of:
    #     URL         neighbor URL
    #     URL         neighbor URL
    #     URL         neighbor URL
    #     ...
    lines = spark.read.text(sys.argv[1]).rdd.map(lambda r: r[0])

    # Loads all URLs from input file and initialize their neighbors.
    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.
    ranks = links.map(lambda url_neighbors: (url_neighbors[0], 1.0))

    # Calculates and updates URL ranks continuously using PageRank algorithm.
    for iteration in range(int(sys.argv[2])):
        # Calculates URL contributions to the rank of other URLs.
        contribs = links.join(ranks).flatMap(lambda url_urls_rank: computeContribs(
            url_urls_rank[1][0], url_urls_rank[1][1]  # type: ignore[arg-type]
        ))

        # Re-calculates URL ranks based on neighbor contributions.
        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()
```

Page Rank For First iteration

:

hduser@cs570bigdata: ~/homework

```
hduser@cs570bigdata:~/homework$ spark-submit PythonPageRank.py pagerank.txt 0
WARN: This is a naive implementation of PageRank and is given as an example!
Please refer to PageRank implementation provided by graphx
22/11/06 22:29:01 INFO SparkContext: Running Spark version 3.3.0
22/11/06 22:29:02 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
22/11/06 22:29:02 INFO ResourceUtils: ==============================================================
22/11/06 22:29:02 INFO ResourceUtils: No custom resources configured for spark.driver.
22/11/06 22:29:02 INFO ResourceUtils: ==============================================================
22/11/06 22:29:02 INFO SparkContext: Submitted application: PythonPageRank
22/11/06 22:29:02 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory
cript: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
22/11/06 22:29:02 INFO ResourceProfile: Limiting resource is cpu
22/11/06 22:29:02 INFO ResourceProfileManager: Added ResourceProfile id: 0
22/11/06 22:29:03 INFO SecurityManager: Changing view acls to: hduser
22/11/06 22:29:03 INFO SecurityManager: Changing modify acls to: hduser
22/11/06 22:29:03 INFO SecurityManager: Changing view acls groups to:
22/11/06 22:29:03 INFO SecurityManager: Changing modify acls groups to:
22/11/06 22:29:03 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(hduser); groups with vi
th modify permissions: Set(hduser); groups with modify permissions: Set()
22/11/06 22:29:04 INFO Utils: Successfully started service 'sparkDriver' on port 33669.
22/11/06 22:29:05 INFO SparkEnv: Registering MapOutputTracker
22/11/06 22:29:05 INFO SparkEnv: Registering BlockManagerMaster
22/11/06 22:29:05 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
22/11/06 22:29:05 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
```

hduser@cs570bigdata: ~/homework

```
22/11/06 22:29:34 INFO DAGScheduler: ResultStage 2 (collect at /home/hduser/homework/PythonPageRank.py:172) finished in 0.186 s
22/11/06 22:29:34 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
22/11/06 22:29:34 INFO TaskSchedulerImpl: Killing all running tasks in stage 2: Stage finished
22/11/06 22:29:34 INFO DAGScheduler: Job 0 finished: collect at /home/hduser/homework/PythonPageRank.py:172, took 1.984248 s
A has rank: 1.0.
B has rank: 1.0.
C has rank: 1.0.
22/11/06 22:29:34 INFO SparkUI: Stopped Spark web UI at http://cs570bigdata:4040
22/11/06 22:29:35 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
22/11/06 22:29:35 INFO MemoryStore: MemoryStore cleared
22/11/06 22:29:35 INFO BlockManager: BlockManager stopped
22/11/06 22:29:35 INFO BlockManagerMaster: BlockManagerMaster stopped
22/11/06 22:29:35 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
22/11/06 22:29:35 INFO SparkContext: Successfully stopped SparkContext
22/11/06 22:29:35 INFO ShutdownHookManager: Shutdown hook called
22/11/06 22:29:35 INFO ShutdownHookManager: Deleting directory /tmp/spark-3805abe3-20d5-4a59-a071-9d2c97ea5508
22/11/06 22:29:35 INFO ShutdownHookManager: Deleting directory /tmp/spark-d6973776-d5f3-4010-8f40-76d86a8f07ef/pyspark-53168603-e352-4
22/11/06 22:29:36 INFO ShutdownHookManager: Deleting directory /tmp/spark-d6973776-d5f3-4010-8f40-76d86a8f07ef/pyspark-f1f18c49-cddd-4
22/11/06 22:29:36 INFO ShutdownHookManager: Deleting directory /tmp/spark-d6973776-d5f3-4010-8f40-76d86a8f07ef
hduser@cs570bigdata:~/homework$
```

Page Rank For Second Iteration

hduser@cs570bigdata: ~/homework

```
hduser@cs570bigdata:~/homework$ spark-submit PythonPageRank.py pagerank.txt 1
WARN: This is a naive implementation of PageRank and is given as an example!
Please refer to PageRank implementation provided by graphx
22/11/06 22:33:58 INFO SparkContext: Running Spark version 3.3.0
22/11/06 22:33:59 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where ap
22/11/06 22:33:59 INFO ResourceUtils: ==============================================================
22/11/06 22:33:59 INFO ResourceUtils: No custom resources configured for spark.driver.
22/11/06 22:33:59 INFO ResourceUtils: ==============================================================
22/11/06 22:33:59 INFO SparkContext: Submitted application: PythonPageRank
22/11/06 22:33:59 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, scri
cript: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
22/11/06 22:33:59 INFO ResourceProfile: Limiting resource is cpu
22/11/06 22:33:59 INFO ResourceProfileManager: Added ResourceProfile id: 0
22/11/06 22:33:59 INFO SecurityManager: Changing view acls to: hduser
```

hduser@cs570bigdata: ~/homework

```
22/11/06 22:34:23 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
22/11/06 22:34:23 INFO TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have all completed, from pool
22/11/06 22:34:23 INFO TaskSchedulerImpl: Killing all running tasks in stage 4: Stage finished
22/11/06 22:34:23 INFO DAGScheduler: Job 0 finished: collect at /home/hduser/homework/PythonPageRank.py:172, took 2.501258 s
C has rank: 1.4249999999999998.
A has rank: 1.0.
B has rank: 0.575.
22/11/06 22:34:23 INFO SparkUI: Stopped Spark web UI at http://cs570bigdata:4040
22/11/06 22:34:23 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
22/11/06 22:34:24 INFO MemoryStore: MemoryStore cleared
22/11/06 22:34:24 INFO BlockManager: BlockManager stopped
22/11/06 22:34:24 INFO BlockManagerMaster: BlockManagerMaster stopped
22/11/06 22:34:24 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
22/11/06 22:34:24 INFO SparkContext: Successfully stopped SparkContext
```

Page Rank For Third Iteration

hduser@cs570bigdata: ~/homework

```
hduser@cs570bigdata:~/homework$ spark-submit PythonPageRank.py pagerank.txt 2
WARN: This is a naive implementation of PageRank and is given as an example!
Please refer to PageRank implementation provided by graphx
22/11/06 22:37:07 INFO SparkContext: Running Spark version 3.3.0
22/11/06 22:37:07 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where ap
22/11/06 22:37:08 INFO ResourceUtils: ==============================================================
22/11/06 22:37:08 INFO ResourceUtils: No custom resources configured for spark.driver.
22/11/06 22:37:08 INFO ResourceUtils: ==============================================================
22/11/06 22:37:08 INFO SparkContext: Submitted application: PythonPageRank
22/11/06 22:37:08 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, scri
cript: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
22/11/06 22:37:08 INFO ResourceProfile: Limiting resource is cpu
22/11/06 22:37:08 INFO ResourceProfileManager: Added ResourceProfile id: 0
22/11/06 22:37:08 INFO SecurityManager: Changing view acls to: hduser
```

hduser@cs570bigdata: ~/homework

```
22/11/06 22:37:37 INFO DAGScheduler: ResultStage 6 (collect at /home/hduser/homework/PythonPageRank.py:172) finished in 0.412 s
22/11/06 22:37:37 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
22/11/06 22:37:37 INFO TaskSchedulerImpl: Killing all running tasks in stage 6: Stage finished
22/11/06 22:37:37 INFO DAGScheduler: Job 0 finished: collect at /home/hduser/homework/PythonPageRank.py:172, took 4.787849 s
C has rank: 1.06375.
B has rank: 0.575.
A has rank: 1.3612499999999996.
22/11/06 22:37:37 INFO SparkUI: Stopped Spark web UI at http://cs570bigdata:4040
22/11/06 22:37:37 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
22/11/06 22:37:37 INFO MemoryStore: MemoryStore cleared
22/11/06 22:37:37 INFO BlockManager: BlockManager stopped
22/11/06 22:37:37 INFO BlockManagerMaster: BlockManagerMaster stopped
22/11/06 22:37:37 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
22/11/06 22:37:37 INFO SparkContext: Successfully stopped SparkContext
```

Conclusion

PageRank is a system developed in 1997 by Google founders Larry Page and Sergey Brin. It was designed to evaluate the quality and quantity of links to a page. Along with other factors, the score determined pages' positions in search engine rankings.
It helps Google to decide the importance of a page and it is the main reason behind which the PageRank for a website is determined in the search results.

References：SFBU course materials