Practice 8

PageRank

What is Scala

➢ Scala

- Modern multi-paradigm programming language that is a combination of object-oriented and functional programming. It is highly scalable which is why it is called Scala**.**

➢ Scala Features

- **Object- Oriented:** Every value in Scala is an object so it is a **purely object-oriented programming language**. The behavior and type of objects are depicted by the classes and traits in Scala.
- **Functional:** It is also **a functional programming language** as every function is a value and every value is an object. It provides the **support for the high-order functions, nested functions, anonymous functions** etc.
- **Statically Typed:** The process of verifying and enforcing the constraints of types is **done at compile time** in Scala. Unlike other statically typed programming languages like C++, C etc., Scala doesn't expect the redundant type of information from the user. In most cases, **the user has no need to specify a type**.

https://spark.apache.org/docs/latest/quick-start.html

What is Scala

➢ Scala example

```
scala> textFile.map(line => line.split(" ").size).reduce((a, b) => if (a > b) a else b)
res4: Long = 15
```

```
scala> import java.lang.Math
import java.lang.Math

scala> textFile.map(line => line.split(" ").size).reduce((a, b) => Math.max(a, b))
res5: Int = 15
```

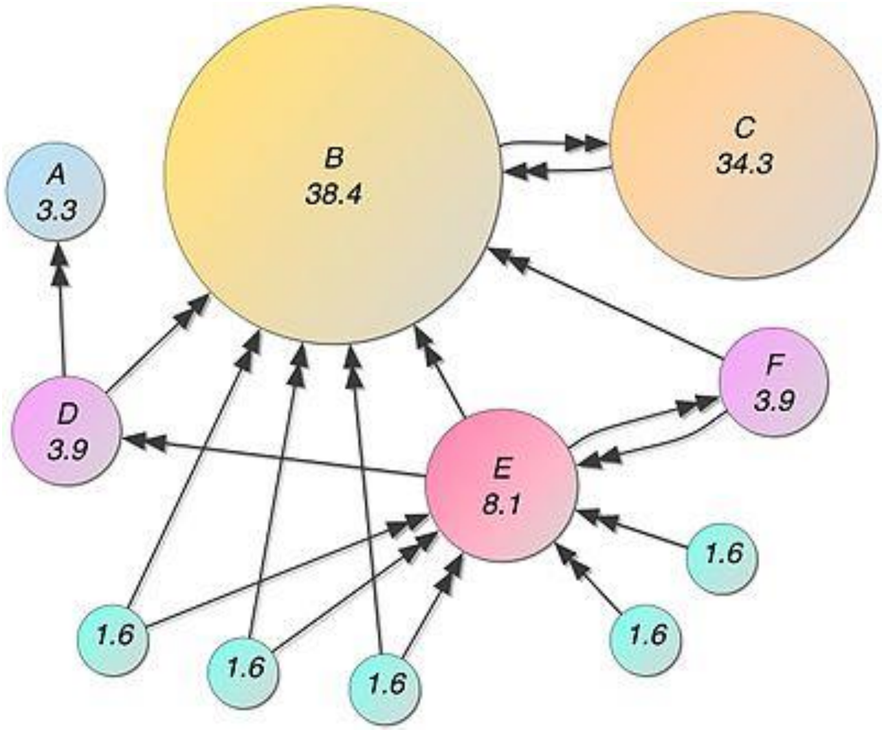
```
scala> val wordCounts = textFile.flatMap(line => line.split(" ")).groupByKey(identity).count()
wordCounts: org.apache.spark.sql.Dataset[(String, Long)] = [value: string, count(1): bigint]
```

```
scala> wordCounts.collect()
res6: Array[(String, Int)] = Array((means,1), (under,2), (this,3), (Because,1), (Python,2), (agree,1), (cluste
r.,1), ...)
```

https://spark.apache.org/docs/latest/quick-start.html

Problem

- **Calculate the pagerank of all nodes in network dataset and report the ID and PageRank score of the top 10 pages.**
  - Use predefined function in org.apache.spark.**graphx.GraphLoader**



https://spark.apache.org/docs/latest/graphx-programming-guide.html#graph-builders

Dataset for PageRank

➢ Citation Network

- Node = paper, Edge = citation (directed)

➢ About Dataset

- We used ***CiteSeer*** network dataset and use edge list only.
- Then, we eliminated some nodes which are not in connected component and made the nodes have unique integer number(ID).

- **CiteSeer for Document Classification**
  - The CiteSeer dataset consists of 3312 scientific publications classified into one of six classes. The citation network consists of 4732 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 3703 unique words. The README file in the dataset provides more details.
  - Download link:
    - https://linqs-data.soe.ucsc.edu/public/lbc/citeseer.tgz
  - Related papers:
    - Qing Lu, and Lise Getoor. "Link-based classification." ICML, 2003.
    - Prithviraj Sen, et al. "Collective classification in network data." AI Magazine, 2008.

https://linqs.soe.ucsc.edu/data

Dataset for PageRank

➢ **Dataset representation**

Start Destination
1 3
2 5
6 3
…
…
n 623

❖ **‘A B’ means ‘A has been cited by B’**

Original dataset

100157 100157
100157 364207
100157 38848
100157 bradshaw97introduction
100157 bylund99coordinating
100157 dix01metaagent
100157 gray99finding
100157 labrou01standardizing
100157 labrou99agent
100157 nodine98overview
100157 nodine99active
100157 wagner97artificial
100598 455651
100598 marquez00machine
100598 punyakanok01use

“*Citerseer.cites*” file

➢ **You can download the pre-processed dataset on iCampus**

Problem

➢ Use PageRank in GraphLoader

- **How to run a scala code**

1. **spark-shell (in terminal)**

```
₩practice8>spark-shell
```

2. **:load ScriptName.scala (in spark-shell)**

```
scala> :load pratice8_pagerank.scala
```

Practice 8

1. **Use predefined classes in *org.apache.spark.graphx : GraphLoader ()***

2. **Find the PageRank of the dataset and report the ID and PageRank score of the top 10 pages.**

https://spark.apache.org/docs/latest/graphx-programming-guide.html#pagerank

Submission

1. You have to submit “**result.txt**” file on iCampus
2. In your **result.txt** file, there must be the pairs of node ID and its PageRank.
3. Deadline: ***June 4th 23:59 P.M.***
4. Your **result.txt** file must like following

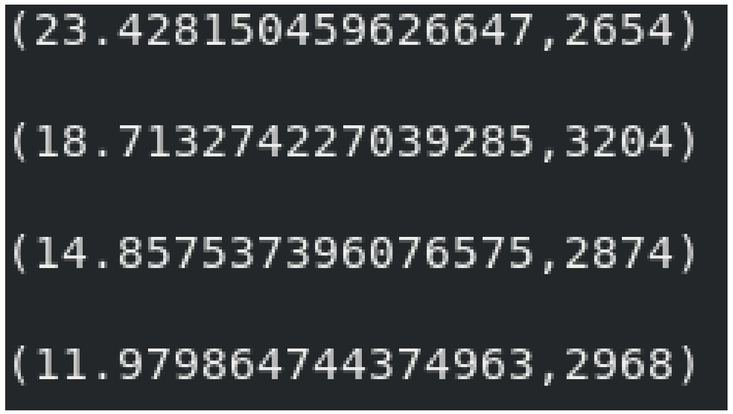
(23.428150459626647,2654)

(18.713274227039285,3204)

(14.857537396076575,2874)

(11.979864744374963,2968)

Windows



Linux

Solution

- Import packages

```
import org.apache.spark.graphx.GraphLoader
import java.io._
```

- Load the edges as a graph

```
val graph = GraphLoader.edgeListFile(sc, "practice8_pagerank.txt")
```

- Run PageRank

```
val ranks = graph.pageRank(0.001).vertices
ranks.foreach(println)
```

Show the pagerank of each vertex in the graph

Solution

- Swap (id, rank) -> (rank, id)

```
val swappedRanks = ranks.map(_.swap)
```

- Sort

```
val sortedRanks = swappedRanks.sortByKey(false)
```

Sort vertex by importance for showing the most important vertex

- Print the result

```
val pw = new PrintWriter(new File("result.txt"))
for(pr_node <- sortedRanks.collect().take(10)){
    pw.write(pr_node.toString)
    pw.println("\n")
    }
pw.close()
sc.stop()
```

Get top 10 nodes and their pagerank

Solution

➢ Result

- We show only top 4 nodes, but you can see there are 10 nodes which have highest pagerank
- Your **result.txt** file must be like following

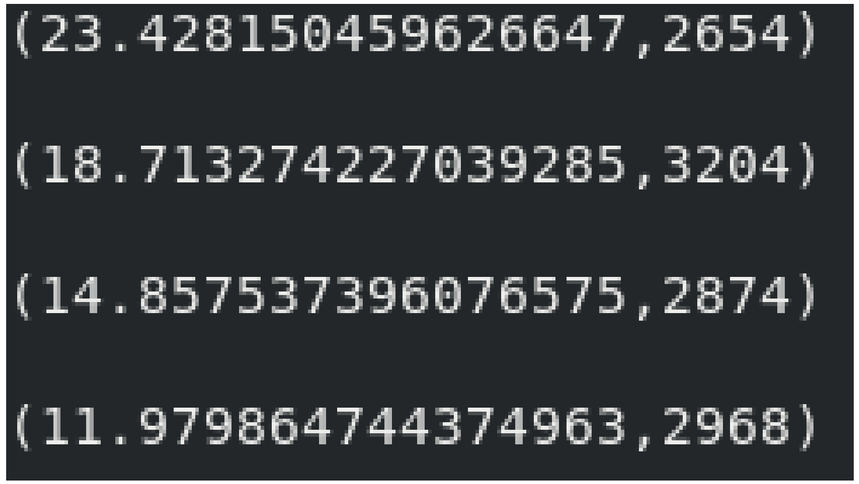
(23.428150459626647,2654)

(18.713274227039285,3204)

(14.857537396076575,2874)

(11.979864744374963,2968)

Windows



Linux