Universidad Francisco Marroquín
Facultad de Ciencias Económicas
Computer Science
Big Data
Primer Semestre 2020
Sophia Gamarro: 20170356

Hoja 7: Spark

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.5
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_201)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.log4j.Level
import org.apache.log4j.Level

scala> import org.apache.log4j.Logger
import org.apache.log4j.Logger

scala> import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.{SparkConf, SparkContext}

scala> val conf = new SparkConf().setAppName("count").setMaster("local[*]")
conf: org.apache.spark.SparkConf = org.apache.spark.SparkConf@5345552f

scala> val sc = new SparkContext(conf)
org.apache.spark.SparkException: Only one SparkContext may be running in this JVM (see SPARK-2243). To ignore this error, set spark.driver.allowMultipleContexts = true. Th
e currently running SparkContext was created at:
org.apache.spark.sql.SparkSession$Builder.getOrCreate(SparkSession.scala:926)
org.apache.spark.repl.Main$.createSparkSession(Main.scala:106)
<init>(<console>:15)
<init>(<console>:43)
<init>(<console>:45)
.<init>(<console>:49)
.<clinit>(<console>)
.$print$lzycompute(<console>:7)
.$print(<console>:6)
$print(<console>)
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
java.lang.reflect.Method.invoke(Method.java:498)
scala.tools.nsc.interpreter.IMain$ReadEvalPrint.call(IMain.scala:793)
scala.tools.nsc.interpreter.IMain$Request.loadAndRun(IMain.scala:1054)
scala.tools.nsc.interpreter.IMain$WrappedRequest$$anonfun$loadAndRunReq$1.apply(IMain.scala:645)
```

```
scala> val inputWords = List("spark", "hadoop", "spark", "hive", "pig", "cassandra", "hadoop")
inputWords: List[String] = List(spark, hadoop, spark, hive, pig, cassandra, hadoop)

scala> val wordRdd = sc.parallelize(inputWords)
wordRdd: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[0] at parallelize at <console>:29

scala> println("Count: " + wordRdd.count())
Count: 7

scala> val wordCountByValue = wordRdd.countByValue()
wordCountByValue: scala.collection.Map[String,Long] = Map(cassandra -> 1, hadoop -> 2, spark -> 2, hive -> 1, pig -> 1)

scala> for ((word, count) <- wordCountByValue) println(word + " : " + count)
cassandra : 1
hadoop : 2
spark : 2
hive : 1
pig : 1
```

```
scala> import org.apache.log4j.Level
import org.apache.log4j.Level

scala> import org.apache.log4j.Logger
import org.apache.log4j.Logger

scala> import org.apache.spark.SparkConf
import org.apache.spark.SparkConf
```

```
import org.apache.spark._

scala> val conf = new SparkConf().setAppName("wordCounts").setMaster("local[3]")
conf: org.apache.spark.SparkConf = org.apache.spark.SparkConf@4ee80a94
```

```
scala> val sc = new SparkContext(conf)
org.apache.spark.SparkException: Only one SparkContext may be run
.allowMultipleContexts = true. The currently running SparkContext
org.apache.spark.sql.SparkSession$Builder.getOrCreate(SparkSessio
org.apache.spark.repl.Main$.createSparkSession(Main.scala:106)
<init>(<console>:15)
<init>(<console>:43)
<init>(<console>:45)
.<init>(<console>:49)
.<clinit>(<console>)
.$print$lzycompute(<console>:7)
.$print(<console>:6)
$print(<console>)
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorI
```

```
scala> val lines = sc.textFile("Desktop/word_count.text")
lines: org.apache.spark.rdd.RDD[String] = Desktop/word_count.text MapPartitionsRDD[24] at textFile

scala> val words = lines.flatMap(line => line.split(" "))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[25] at flatMap at <console>:31

scala> val wordCounts = words.countByValue()
wordCounts: scala.collection.Map[String,Long] = Map(Twenties, -> 1, II -> 2, industries. -> 1, econ
 -> 1, for -> 3, eleventh -> 1, ultimately -> 1, support -> 1, channels -> 1, Thereafter, -> 1, subs
, proposed -> 1, any -> 1, 1790, -> 1, city -> 1, war. -> 2, southern -> 2, across -> 1, operations
 21, Park -> 1, expressed -> 1, Civil -> 1, point -> 2, cultural -> 1, 1777, -> 1, claim -> 1, labo
War -> 2, representatives -> 1, patroon -> 1, system -> 1, Iroquoian -> 1, Battery -> 1, nationally
g -> 1, late -> 1, renewed -> 1, City's -> 1, shrank. -> 1, After -> 1, Wall -> 1, In -> 3, state -
```

```
scala> for ((word, count) <- wordCounts) println(word + " : " + count)
Twenties, : 1
II : 2
industries. : 1
economy : 1
 : 7
ties : 2
buildings : 1
for : 3
eleventh : 1
ultimately : 1
support : 1
channels : 1
Thereafter, : 1
subsequent : 1
defense : 1
series : 1
proposed : 1
any : 1
1790, : 1
```

```
scala> import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.{SparkConf, SparkContext}

scala> val conf = new SparkConf().setAppName("sameHosts"
conf: org.apache.spark.SparkConf = org.apache.spark.Spar
```

```
scala> val sc = new SparkContext(conf)
org.apache.spark.SparkException: Only one SparkContext may be running in this JVM (see SPARK-2243). To ignore this
.allowMultipleContexts = true. The currently running SparkContext was created at:
org.apache.spark.sql.SparkSession$Builder.getOrCreate(SparkSession.scala:926)
org.apache.spark.repl.Main$.createSparkSession(Main.scala:106)
<init>(<console>:15)
<init>(<console>:43)
<init>(<console>:45)
.<init>(<console>:49)
.<clinit>(<console>)
```

```
scala> val julyFirstLogs = sc.textFile("Desktop/nasa_01.tsv")
julyFirstLogs: org.apache.spark.rdd.RDD[String] = Desktop/nasa_01.tsv MapPartitionsRDD[30] at textFile at <console>:31

scala> val augustFirstLogs = sc.textFile("Desktop/nasa_02.tsv")
augustFirstLogs: org.apache.spark.rdd.RDD[String] = Desktop/nasa_02.tsv MapPartitionsRDD[32] at textFile at <console>:31

scala> val julyFirstHosts = julyFirstLogs.map(line => line.split("\t")(0))
julyFirstHosts: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[33] at map at <console>:32

scala> val augustFirstHosts = augustFirstLogs.map(line => line.split("\t")(0))
augustFirstHosts: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[34] at map at <console>:32

scala> val intersection = julyFirstHosts.intersection(augustFirstHosts)
intersection: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[40] at intersection at <console>:34

scala> val cleanedHostIntersection = intersection.filter(host => host!= "host")
cleanedHostIntersection: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[41] at filter at <console>:32

scala> cleanedHostIntersection.saveAsTextFile("out/nasa_logs_same_host.csv")
```