Ejercicios Pair RDDs:

A

1. var logs=sc.textFile("file:/home/cloudera/BIT/data/weblogs/\*")

val rdd = logs.map(line => (line.split(" ")(2),1))

val reduce = rdd.reduceByKey((v1,v2) => v1 + v2)

1. val swapped = reduce.map(x => x.swap)

val top10 = swapped.sortByKey(false).map(x=>x.swap).take(10)

1. val groupIPs = logs.map(x => x.split(" ")).map(linea => (linea(2),linea(0))).groupByKey()

B

1. val mapAccounts = accounts.map(x => x.split(",")).map(linea => (linea(0), linea))

val joined = mapAccounts.join(reduce)

val prueba = joined.take(5).foreach(x=> println(x.\_1,x.\_2.\_2, x.\_2.\_1(3), x.\_2.\_1(4)))

C

1. var accountsByPCode = accounts.map(\_.split(',')).keyBy(\_(8))

var namesByPCode = accountsByPCode.mapValues(values => values(4) + ',' +values(3)).groupByKey()

namesByPCode.sortByKey().take(10).foreach{x => println ("---" + x.\_1); x.\_2.foreach(println)}

Opcionales:

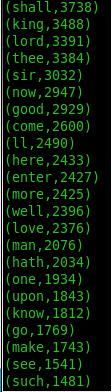
1. Utilizando la terminal, introduce dicho dataset en el HDFS (La ruta por defecto del HDFS es: hdfs://quickstart.cloudera:8020/user/cloudera, copia el dataset en dicha ruta en la carpeta 'shakespeare')
   1. Hadoop fs –put “/home/cloudera/ejercicios/shakespeare” /user/cloudera
2. Ahora vamos a realizar un análisis al estilo de escritura de Shakespeare, para ello nos interesa conocer **las palabras más repetidas en sus obras.** Muestra por pantalla las 100 palabras que más veces aparecen en las obras de Shakespeare, junto con la frecuencia de aparición de cada una, ordenadas descendientemente (de mayor a menor frecuencia de aparición).

val skp = sc.textFile("hdfs://quickstart.cloudera:8020/user/cloudera/shakespeare/\*")

val skpMap = skp.flatMap(x => x.split(" ")).map(word => (word, 1)).reduceByKey(\_+\_)

val countSkp = skpMap.map(x => x.swap).sortByKey(false).take(100).foreach(linea => println(linea))

1. El análisis anterior no es todo lo completo que una situación real nos puede requerir, pues entre las cpalabras más repetidas encontramos: pronombres, artículos, proposiciones...; es decir, palabras que no nos aportan información alguna. A este tipo de palabras se les conoce como [palabras vacías](https://es.wikipedia.org/wiki/Palabra_vac%C3%ADa) (StopWords), y nos interesa eliminarlas de nuestro análisis.
2. Para la realización de este ejercicio vamos a necesitar un dataset que contenga "stop words" en inglés (pues, como es lógico, los libros de Shakespeare están en este idioma). Copia el archivo "stop-word-list.csv" en la máquina virtual y cópialo al HDFS (en la ruta por defecto).
3. También tenemos que tener en cuenta que, para hacer nuestro análisis, no nos interesan ni las líneas que solo contengan espacios en blanco ni las palabras que estén formadas por una única letra. Tampoco haremos distinción entre mayúsculas ni minúsculas, por lo que, por ejemplo, "Hello" y "hello" deberán ser tomadas como la misma palabra en el análisis.
   1. El método para cambiar una línea a minúsculas es: toLowerCase
   2. La expresión regular que toma únicamente caracteres alfabéticos es: "[^a-zA-Z]+"
   3. Un archivo CSV está delimitado por comas.
   4. No nos interesa mostrar las ocurrencias de una única letra. Por lo que deberemos filtrar el RDD final.
4. El resultado de este ejercicio debería ser parecido a:



val stop = sc.textFile("hdfs://quickstart.cloudera:8020/user/cloudera/stop-word-list.csv").flatMap(x=>x.split(",")).map(y=>y.replace(" ", ""))

val skp2 = skp.map(x=> x.toLowerCase).map(y => y.replaceAll("[^a-zA-Z]+", " "))

valskp3=skp2.flatMap(x=>x.split(" ")).subtract(sc.parallelize(Seq(""))).subtract(stop).map(z=>z.replaceAll(" ","")).filter(y=> y.size >

val skp4 = skp3.map(x=>(x,1)).reduceByKey(\_+\_).map(y=>y.swap).sortByKey(false).take(100).foreach(z => println(z))