Session: RDD'S IN SPARK

Assignment

Student Name: Subham Vishal

Course: Big Data Hadoop & Spark Training

Assignment – basic RDD operations.

Contents

Introduction

In this assignment, we are going to perform some basic Spark RDD operation functions with the given problem statement.

Problem Statement

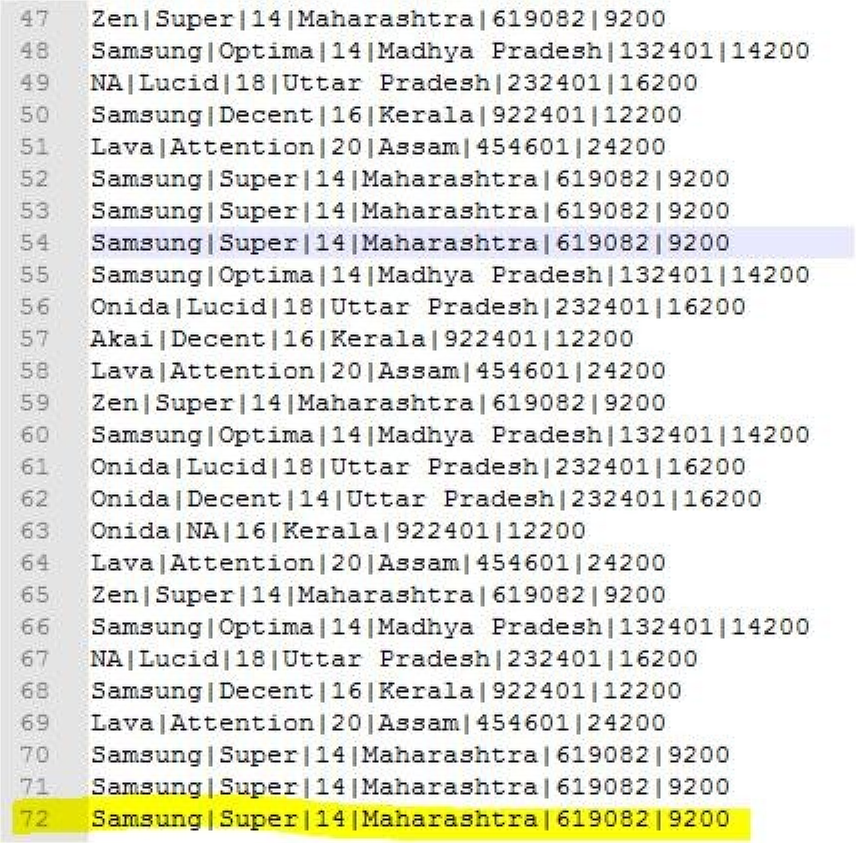
1. Write a program to read a text file and print the number of rows of data in the document.
2. Write a program to read a text file and print the number of words in the document.
3. We have a document where the word separator is -, instead of space. Write a spark code, to obtain the count of the total number of words present in the document.

Sample document:

This-is-my-first-assignment.It-will-count-the-number-of-lines-in-this-document.The-total-number-of-lines-is-3

Task1 - Write a program to read a text file and print the number of rows of data in the document.

In this task, we are using a text file **“television.txt”** which has **72** rows as shown below,



```
47  Zen|Super|14|Maharashtra|619082|9200
48  Samsung|Optima|14|Madhya Pradesh|132401|14200
49  NA|Lucid|18|Uttar Pradesh|232401|16200
50  Samsung|Decent|16|Kerala|922401|12200
51  Lava|Attention|20|Assam|454601|24200
52  Samsung|Super|14|Maharashtra|619082|9200
53  Samsung|Super|14|Maharashtra|619082|9200
54  Samsung|Super|14|Maharashtra|619082|9200
55  Samsung|Optima|14|Madhya Pradesh|132401|14200
56  Onida|Lucid|18|Uttar Pradesh|232401|16200
57  Akai|Decent|16|Kerala|922401|12200
58  Lava|Attention|20|Assam|454601|24200
59  Zen|Super|14|Maharashtra|619082|9200
60  Samsung|Optima|14|Madhya Pradesh|132401|14200
61  Onida|Lucid|18|Uttar Pradesh|232401|16200
62  Onida|Decent|14|Uttar Pradesh|232401|16200
63  Onida|NA|16|Kerala|922401|12200
64  Lava|Attention|20|Assam|454601|24200
65  Zen|Super|14|Maharashtra|619082|9200
66  Samsung|Optima|14|Madhya Pradesh|132401|14200
67  NA|Lucid|18|Uttar Pradesh|232401|16200
68  Samsung|Decent|16|Kerala|922401|12200
69  Lava|Attention|20|Assam|454601|24200
70  Samsung|Super|14|Maharashtra|619082|9200
71  Samsung|Super|14|Maharashtra|619082|9200
72  Samsung|Super|14|Maharashtra|619082|9200
```

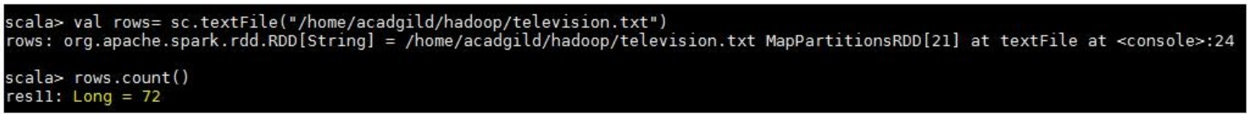
Spark Operation

Read the text file,

scala> val rows= sc.textFile("/home/acadgild/hadoop/television.txt")

scala> rows.count()

res0: Long = 72



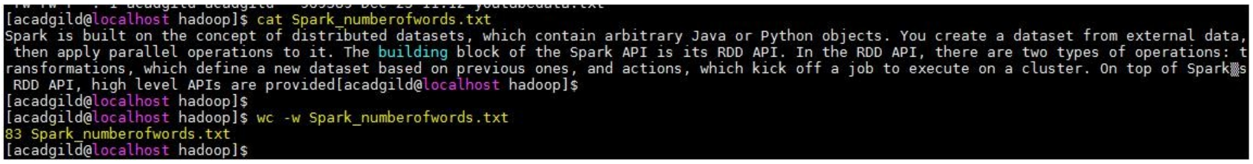
```
scala> val rows= sc.textFile("/home/acadgild/hadoop/television.txt")
rows: org.apache.spark.rdd.RDD[String] = /home/acadgild/hadoop/television.txt MapPartitionsRDD[21] at textFile at <console>:24

scala> rows.count()
res11: Long = 72
```

Task2 - Write a program to read a text file and print the number of words in the document.

In this task, we are using a text file "***Spark_numberofwords.txt***" which we created and it has number of words as 83, please see below,

cat Spark_numberofwords.txt

wc -w Spark_numberofwords.txt



```
[acadgild@localhost hadoop]$ cat Spark_numberofwords.txt
Spark is built on the concept of distributed datasets, which contain arbitrary Java or Python objects. You create a dataset from external data,
 then apply parallel operations to it. The building block of the Spark API is its RDD API. In the RDD API, there are two types of operations: t
ransformations, which define a new dataset based on previous ones, and actions, which kick off a job to execute on a cluster. On top of Spark▒s
 RDD API, high level APIs are provided[acadgild@localhost hadoop]$
[acadgild@localhost hadoop]$
[acadgild@localhost hadoop]$ wc -w Spark_numberofwords.txt
83 Spark_numberofwords.txt
[acadgild@localhost hadoop]$
```

Spark Operation

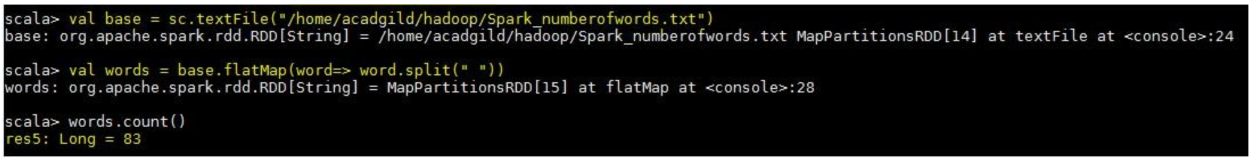
Read the text file,

scala> val base = sc.textFile("/home/acadgild/hadoop/Spark_numberofwords.txt")

scala> val words = base.flatMap(word=> word.split(" "))

scala> words.count()

res5: Long = 83



```
scala> val base = sc.textFile("/home/acadgild/hadoop/Spark_numberofwords.txt")
base: org.apache.spark.rdd.RDD[String] = /home/acadgild/hadoop/Spark_numberofwords.txt MapPartitionsRDD[14] at textFile at <console>:24

scala> val words = base.flatMap(word=> word.split(" "))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[15] at flatMap at <console>:28

scala> words.count()
res5: Long = 83
```

Task3 - We have a document where the word separator is -, instead of space. Write a spark code, to obtain the count of the total number of words present in the document.

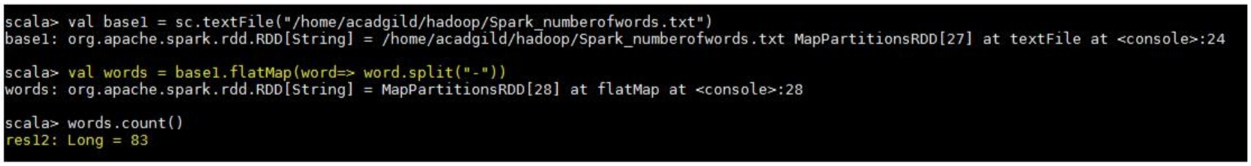
The same file “***Spark_numberofwords.txt”*** has been modified by placing **“-“between** the words, please see below.

scala> val base1 = sc.textFile("/home/acadgild/hadoop/Spark_numberofwords.txt")

scala> val words = base1.flatMap(word=> word.split("-"))

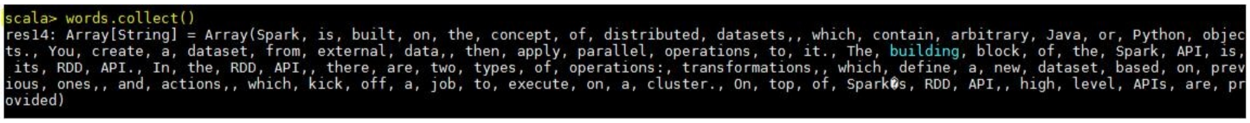
scala> words.count()

res12: Long = 83



```
scala> val base1 = sc.textFile("/home/acadgild/hadoop/Spark_numberofwords.txt")
base1: org.apache.spark.rdd.RDD[String] = /home/acadgild/hadoop/Spark_numberofwords.txt MapPartitionsRDD[27] at textFile at <console>:24

scala> val words = base1.flatMap(word=> word.split("-"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[28] at flatMap at <console>:28

scala> words.count()
res12: Long = 83
```

words.collect()



```
scala> words.collect()
res14: Array[String] = Array(Spark, is, built, on, the, concept, of, distributed, datasets,, which, contain, arbitrary, Java, or, Python, objec
ts., You, create, a, dataset, from, external, data,, then, apply, parallel, operations, to, it., The, building, block, of, the, Spark, API, is,
 its, RDD, API., In, the, RDD, API,, there, are, two, types, of, operations:, transformations,, which, define, a, new, dataset, based, on, prev
ious, ones,, and, actions,, which, kick, off, a, job, to, execute, on, a, cluster., On, top, of, Spark�s, RDD, API,, high, level, APIs, are, pr
ovided)
```