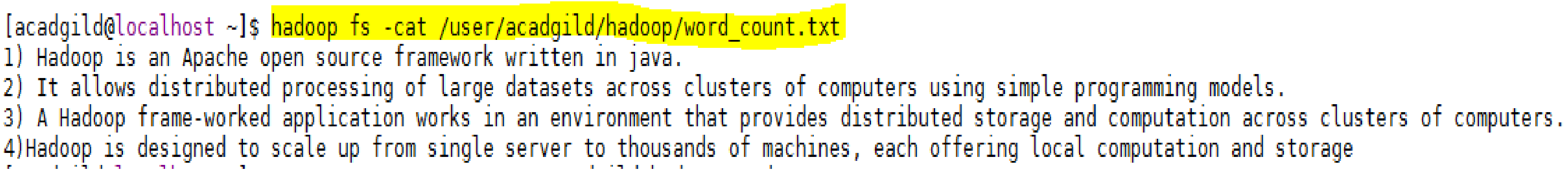
Spark Assignment 17.1

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

Steps Followed:

1) Created a file in HDFS in the path /user/acadgild/hadoop/word_count.txt . The contents of the file is as below.



```
[acadgild@localhost ~]$ hadoop fs -cat /user/acadgild/hadoop/word_count.txt
1) Hadoop is an Apache open source framework written in java.
2) It allows distributed processing of large datasets across clusters of computers using simple programming models.
3) A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers.
4)Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage
```

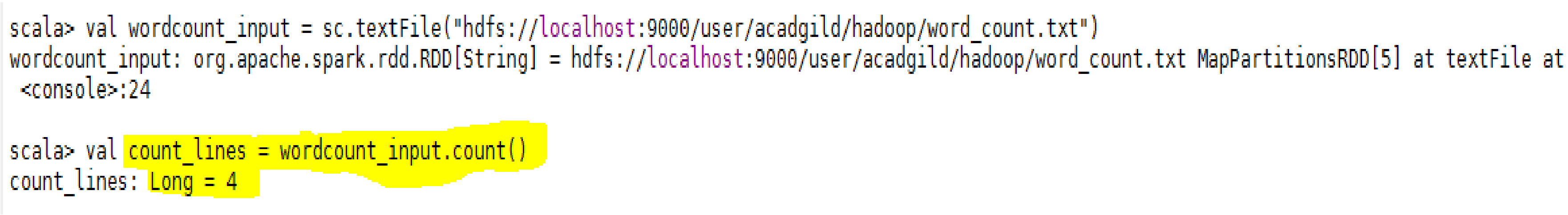
2) From the spark context reading the text file as below.

val wordcount_input = sc.textFile("hdfs://localhost:9000/user/acadgild/hadoop/word_count.txt")

3) To read the number of rows of the text file the code is

val count_lines = wordcount_input.count()

Spark-shell output



```
scala> val wordcount_input = sc.textFile("hdfs://localhost:9000/user/acadgild/hadoop/word_count.txt")
wordcount_input: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/user/acadgild/hadoop/word_count.txt MapPartitionsRDD[5] at textFile at
 <console>:24

scala> val count_lines = wordcount_input.count()
count_lines: Long = 4
```

2) *Write a program to read a text file and print the number of words in the document.*

Steps Followed:

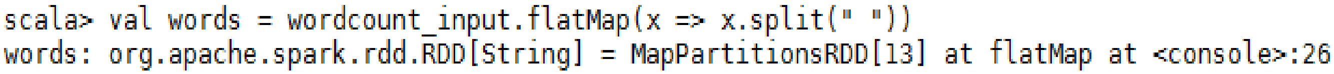
1) From the spark context reading the text file from HDFS as below.

   val wordcount_input = sc.textFile("hdfs://localhost:9000/user/acadgild/hadoop/word_count.txt")

2) To print the number of words in the text file, First we flatten the input file by passing the file delimiter.In this file the delimiter is " ". The flattenMap function will return an array of String.

   val words = wordcount_input.flatMap(x => x.split(" "))

Spark-shell screenshot

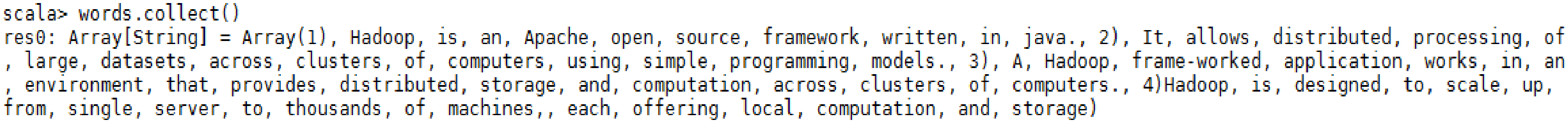


```
scala> val words = wordcount_input.flatMap(x => x.split(" "))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[13] at flatMap at <console>:26
```

3) We can view the contents of words array using collect() function.

   words.collect()

Spark-shell screenshot

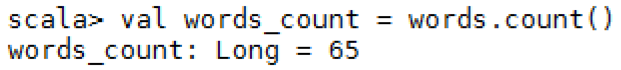


```
scala> words.collect()
res0: Array[String] = Array(1), Hadoop, is, an, Apache, open, source, framework, written, in, java., 2), It, allows, distributed, processing, of
, large, datasets, across, clusters, of, computers, using, simple, programming, models., 3), A, Hadoop, frame-worked, application, works, in, an
, environment, that, provides, distributed, storage, and, computation, across, clusters, of, computers., 4)Hadoop, is, designed, to, scale, up,
from, single, server, to, thousands, of, machines,, each, offering, local, computation, and, storage)
```

4) Then, We can count the size of the words array to find the number of words.

   val words_count = words.count()

Spark-shell output

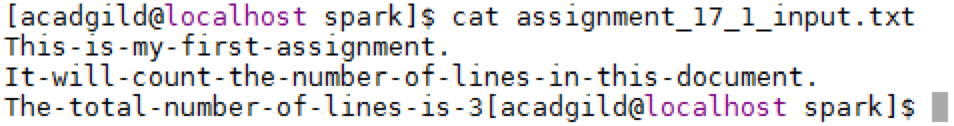


```
scala> val words_count = words.count()
words_count: Long = 65
```

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.*

Steps Followed:

1) Created a file in HDFS in the path /home/acadgild/spark/ assignment_17_1_input.txt. The contents of the file is as below.



```
[acadgild@localhost spark]$ cat assignment_17_1_input.txt
This-is-my-first-assignment.
It-will-count-the-number-of-lines-in-this-document.
The-total-number-of-lines-is-3[acadgild@localhost spark]$
```

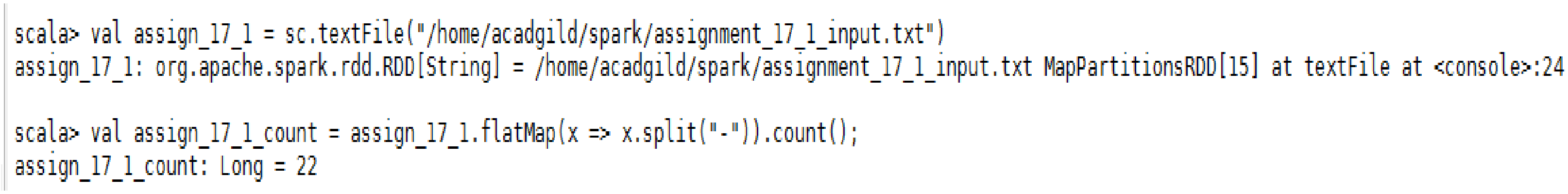
2) From the spark context reading the text file as below.

```
val assign_17_1 = sc.textFile("/home/acadgild/spark/assignment_17_1_input.txt")
```

3) To print the number of words in the text file, First we flatten the input file by passing the file delimiter.In this file the delimiter is “-“. The flattenMap function will return an array of String and then count the items in the array as below.

```
val assign_17_1_count = assign_17_1.flatMap(x => x.split("-")).count();
```

Spark-shell ouput



```
scala> val assign_17_1 = sc.textFile("/home/acadgild/spark/assignment_17_1_input.txt")
assign_17_1: org.apache.spark.rdd.RDD[String] = /home/acadgild/spark/assignment_17_1_input.txt MapPartitionsRDD[15] at textFile at <console>:24

scala> val assign_17_1_count = assign_17_1.flatMap(x => x.split("-")).count();
assign_17_1_count: Long = 22
```