**DSBDA Assignment 12**

**Write a code in SCALA for a simple WordCount application that counts the number of occurrences of each word in each input set using the Apache framework on local-standalone set-up.**

**Word count function**

/\*\* map \*/

var map = sc.textFile("/opt/spark/bin/new.txt").flatMap(line => line.split(" ")).map(word => (word,1));

println(map)

/\*\* reduce \*/

var counts = map.reduceByKey(\_ + \_);

var valu=counts.values

counts.count /\*Gives couts in file\*/

counts.collect /\*gives list of elements in counts\*/

val keysRdd = counts.keys

val sorted = keysRdd.sortBy(x => x, true)

sorted.collect

val m=counts.max

/\*\* save the output to file \*/

counts.saveAsTextFile("/opt/spark/bin/WD\_fri")

sorted.saveAsTextFile("/opt/spark/bin/sort\_fri")

NEW.TXT

keyBy

The keyBy method is similar to the groupBy method. It a higher-order method that takes as

input a function

that returns a key for any given element in the source RDD. The keyBy method applies this

function to all the

elements in the source RDD and returns an RDD of pairs. In each returned pair, the first item

is a key and the

second item is an element that was mapped to that key by the input function to the keyBy

method. The RDD

returned by keyBy will have the same number of elements as the source RDD.

The difference between groupBy and keyBy is that the second item in a returned pair is a

collection of

elements in the first case, while it is a single element in the second case.

sortBy

The higher-order sortBy method returns an RDD with sorted elements from the source RDD.

It takes two

input parameters. The first input is a function that generates a key for each element in the

source RDD. The

second argument allows you to specify ascending or descending order for sort.

pipe

The pipe method allows you to execute an external program in a forked process. It captures

the output of the

external program as a String and returns an RDD of Strings.

randomSplit

The randomSplit method splits the source RDD into an array of RDDs. It takes the weights of

the splits as input.

The coalesce method should be used with caution since reducing the number of partitions

reduces the

parallelism of a Spark application. It is generally useful for consolidating partitions with few

elements. For

example, an RDD may have too many sparse partitions after a filter operation. Reducing the

partitions may

provide performance benefit in such a case.

repartition

The repartition method takes an integer as input and returns an RDD with specified number

of partitions.

It is useful for increasing parallelism. It redistributes data, so it is an expensive operation.

The coalesce and repartition methods look similar, but the first one is used for reducing the

number

of partitions in an RDD, while the second one is used to increase the number of partitions in

an RDD.

wd\_FRI-PART-00000

(this,1)

(case,,1)

(is,12)

(collection,1)

(applies,1)

(takes,4)

(same,1)

(have,2)

(generates,1)

(order,1)

(sort.,1)

(one,2)

(higher-order,2)

(with,4)

(integer,1)

(first,4)

(methods,1)

(increase,1)

(any,1)

(parallelism,1)

(ascending,1)

(parallelism.,1)

(input,5)

(RDD,8)

(from,1)

(pair,,1)

(performance,1)

(too,1)

(allows,2)

(application.,1)

(input.,1)

(two,1)

(after,1)

(few,1)

(second,5)

(parameters.,1)

(specified,1)

(element,4)

(keyBy,6)

(filter,1)

(pair,1)

(example,,1)

(between,1)

(groupBy,2)

(will,1)

(useful,2)

(into,1)

(as,5)

(since,1)

(data,,1)

(case.,2)

(consolidating,1)

(Strings.,1)

(descending,1)

(RDD.,5)

(pipe,2)

(function,4)

(sparse,1)

(RDD,,1)

(so,1)

(single,1)

(program,2)

SORT\_FRI-PART-00000

For

In

It

RDD

RDD,

RDD.

RDDs.

Reducing

Spark

String

Strings.

The

a

after

all

allows

an

and

any

application.

applies

argument

array

as

ascending

be

benefit

between

but

by

captures

case,

case.

caution

coalesce

collection

consolidating

data,

descending

difference

each

element

elements

elements.

example,

execute

expensive

external

few

filter

first

for

forked

from

function

generally

generates

given

groupBy

have

higher-order

in

increase

increasing

input

input.

integer

into

is

it

item

key