Experiment-2

Q Demo- Play with RDD's

1 In command prompt shell

(a) Python shell

→ invoking pyspark.

```
> cd \
> cd spark\bin
C:\spark\bin> pyspark.
```

→ verifying the shell

```
>>> sc
```

→ create RDD using parallelize method.

```
>>> data = [1,2, 3, 4 ,5]
>>> dist = sc.parallelize (data)
>>> dist
```

→ Applying functions on RDD.

(i) count: returns total no. of elements

```
>>> dist.count()
```

(ii) collect: prints all elements of RDD.

```
>>> dist. collect()
```

(iii) first: returns first element of RDD.

```
>>> dist.first()
```

Teacher's Signature ....

(iv)
```
>>> dist.take(3)
```

(v) lambda function:
```
>>> res = dist.map(lambda x: x+2)
>>> res.collect()
```

~~(vi)~~ create RDD from a file

↳ create file at target dir and name it as "data"

↳ Run mentioned command in python.
```
>>> dist1 = sc.textFile("data.txt")
>>> dist1.
```

(b) spark Scala shell

→ invoking spark shell
```
> cd \
> cd spark\bin
C:\spark\bin>spark-shell
```

→ verifying the shell
```
>sc
```

→ create RDD using parallelize method.
```
scala> val data = Array(1,2,3,4,5)
scala> val dist = sc.parallelize(data)
scala> dist
```

→ Applying functions on RDD.

(i) count()

```
Scala> dist.count()
```

(ii) collect()

```
Scala> dist.collect()
```

(iii) first()

```
Scala> dist.first()
Scala> dist.take(3)
```

(iv) lambda function

```
scala> val dist1 = dist.map (x => x+2)
scala> dist1.collect()
```

→ creating RDD from text file.

↳ create file at target dir and name it as "data"

↳ run mentioned command in shell.

```
Scala> val file = sc.textFile("data.txt")
Scala> ~~val~~ file.collect()
```

2 On Databricks

(a) Python

→ checking sparkContext

```
1 sc
```

→ Creating RDD's

1 data = [1, 2, 3, 4, 5]

2 dist = sc.parallelize (data)

3 dist

→ Applying function on RDD's

1 dist. collect()

2 dist. count()

3 dist. first()

4 dist. take(3)

→ Creating RDD using textFile

1 dist1 = sc.textFile("dbfs:/FileStore/shared_uploads/500068760@stu.upes.ac.in/data.txt")

2 dist1

(b) Scala

→ checking sparkContext

1 Sc

→ creating RDD's

1 val data = Array(1,2,3,4,5)

2 val dist = sc.parallelize (data)

3 dist

→ Applying functions on RDD's

1 dist.collect()

```
2 dist.count()
3 dist.take(3)
```

→ lambda function

```
1 val dist1 = dist.map(x => x+2)
2 dist1.collect()
```

→ creating RDD's from text file

```
1 val file = sc.textFile("dbfs:/FileStore/
                 shared_uploads/500068760@stu.upes.ac.in
                 /data.txt")
2 file.collect()
```

Output: Date - 18/01/2021

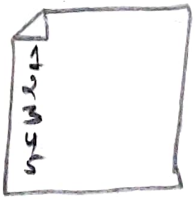
```
C:\spark\bin> pyspark
 spark version 3.0.1

>>> sc
<sparkContext master = local[*] appName = Pysparkshell>

>>> data = [1,2,3,4,5]
>>> dist = sc.parallelize(data)
>>> dist
ParallelCollectionRDD[0] at readRDDFromFile at PythonRDD.scala:262

>>> dist.count()
5
>>> dist.collect()
[1,2,3,4,5]
>>> dist.first()
1
>>> dist.take(3)
[1,2,3]

>>> result = dist.map(lambda x:x+2)
>>> result.collect()
[3,4,5,6,7]
>>> dist1 = sc.textFile("data.txt")
>>> dist.collect()
['1','2','3','4','5']
```



Teacher's Signature ....

Output: Date-18/01/2021

```
C:\ spark\ bin> spark-shell
spark. version 3.0.1.

Scala> sc
res 2: org.apache. spark. SparkContext = org.apache. spark.
       SparkContext@ 2422.

Scala> val data = Array (1,2,3,4,5)
 data: Array [Int] = Array (1,2,3,4,5)

Scala> val dist = sc.parallelize (data)
 dist: org.apache.spark.rdd.RDD[Int] = ParallelCollection
  RDD[0] at parallelize at <console>:26.

Scala> dist. count()
long = 5
scala> dist.collect()
Array[Int] = Array (1,2,3,4,5)
Scala> dist.take (3)
 Array [Int] = Array (1,2,3)

 val dist1 = dist. map (x => x+2)
 dist1: org.apache.spark.rdd.RDD [Int] = MapPartitions
          RDD [1] at map
```

```
scala> dist1.collect()
Array[Int] = Array(3, 4, 5, 6)

scala> val file = sc.textFile("data.txt")
file: org.apache.spark.rdd.RDD[string] = data.txt
        MapPartitionRDD[3]

scala> file.collect()
Array[string] = Array(1,2, 3, 4, 5)
```

Date-18/01/2021

Output:

On Databricks:

Python

```
sc
Out[1]:
Spark Context
Spark UI
Version      V3.0.1
Master        local[8]
AppName       Databricks Shell

→ data = [1,2,3,4,5]
   dist = sc.parallelize(data)
   dist
Out[2]: ParallelCollectionRDD[0] at readRDDFromInputStream
        at PythonRDD.scala:413

→ dist.count()
   dist.collect()
   dist.first()
   dist.take(3)
 Spark Jobs
 Out[3] : 5
 Out[4] : [1,2,3,4,5]
 Out[5] : 1
 Out[6] : [1, 2,3]
```

Date-18/01/2021

Output

```
1 dist1 = sc. textFile ("dbfs : / FileStore / shared_uploads / da
  dist1.
Out[5]: dbfs / FileStore / shared_uploads / 500068760 @stu.upes.ac
        in / data.txt" MapPartitionsRDD[5] at textFile
        at NativeMethodAccessorImpl.java:0.
```

Spark-Scala

```
→ val data = Array(1,2,3,4,5)
  val dist = sc.parallelize(data)
  dist
data: Array[Int] = Array(1, 2, 3, 4, 5)
dist: org.apache.spark.rdd.RDD[Int] = ParallelCollection
      RDD[11] at parallelize at command-1884313242 :2.
```

```
→ val dist1 = dist.map (x => x+2)
  dist1.collect()
dist1: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[
       at map at command -1894132453 9:1
res1 : Array[Int] = Array(3, 4, 5, 6, 7)
```

```
→ val file = sc.textFile(" dbfs : /FileStore / shared_uploads
        / 500068760@stu.upes.ac.in / data.txt")
  file.collect()
file: org.apache.spark.rdd.RDD[String] = dbfs : / FileStore /
    shared_uploads / 500068760 @stu.upes.ac.in / data.txt
     MapPartitions RDD[14] at textFile at command.
res2 : Array [String] = Array (1, 2, 3, 4, 5, 6)
```

Teacher's Signature ....