



UNIVERSITÉ SIDI MOHAMED BEN ABDLLAH
FACULTÉ DES SCIENCES DHAR EL MAHRAZ DE FÈS



DÉPARTEMENT D'INFORMATIQUE

Département : Informatique

Master : Informatique Décisionnelle et Vision Intelligent

Titre :

TD / TP 2 With SPARK

Présenter par :
ABIBOU SOUKAYNA

Supervisé par :
Pr. Noura

Année universitaire : 2020/2021

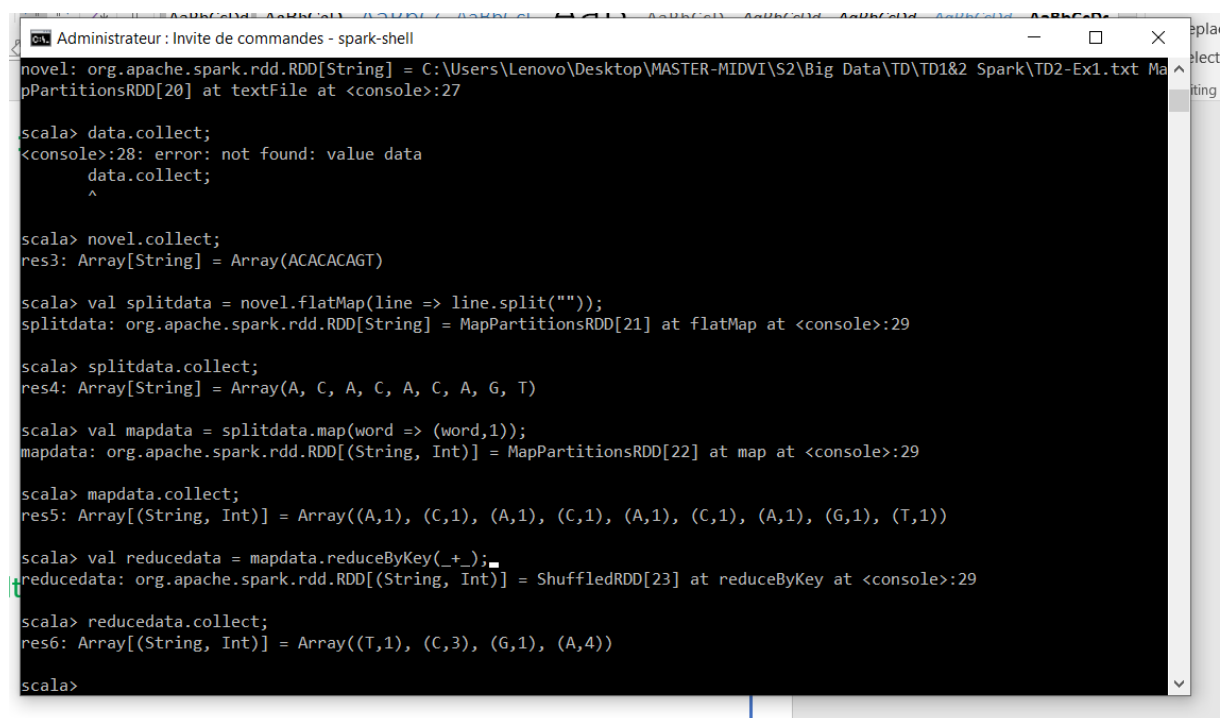
Filière MIDVI

Exercice 1 :

1- Code :

```
val data=sc.textFile("txtFile");
data.collect;
val splitdata = data.flatMap(line => line.split(""));
splitdata.collect;
val mapdata = splitdata.map(word => (word,1));
mapdata.collect;
val reducedata = mapdata.reduceByKey(_+_);
reducedata.collect;
```

2- Résultat d'exécution :



```
Administrateur : Invite de commandes - spark-shell
novel: org.apache.spark.rdd.RDD[String] = C:\Users\Lenovo\Desktop\MASTER-MIDVI\S2\Big Data\TD1&2 Spark\TD2-Ex1.txt Ma
pPartitionsRDD[20] at textFile at <console>:27

scala> data.collect;
<console>:28: error: not found: value data
    data.collect;
    ^

scala> novel.collect;
res3: Array[String] = Array(ACACACAGT)

scala> val splitdata = novel.flatMap(line => line.split(""));
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[21] at flatMap at <console>:29

scala> splitdata.collect;
res4: Array[String] = Array(A, C, A, C, A, C, A, G, T)

scala> val mapdata = splitdata.map(word => (word,1));
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[22] at map at <console>:29

scala> mapdata.collect;
res5: Array[(String, Int)] = Array((A,1), (C,1), (A,1), (C,1), (A,1), (C,1), (A,1), (G,1), (T,1))

scala> val reducedata = mapdata.reduceByKey(_+_);
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[23] at reduceByKey at <console>:29

scala> reducedata.collect;
res6: Array[(String, Int)] = Array((T,1), (C,3), (G,1), (A,4))

scala>
```

Exercice 2 :

1- Code :

```
import org.apache.spark.SparkContext._
var samplefile = sc.textFile(inputPath)
var ArrayFile = samplefile.collect()

var resultFile = ArrayBuffer[String]()
for ( i<- ArrayFile){
  for (j <- 0 until i.length()-2){
    if ( i.substring(j, j+3).equals("AAA") || i.substring(j, j+3).equals("TTT") )
      resultFile += i.substring(j, j+3)
    }
    for(j <- 0 until i.length()-3){
      if ( i.substring(j, j+4).equals("CCCC") || i.substring(j, j+4).equals("GGGG") )
        resultFile += i.substring(j, j+4)
      }
    }
  }
var rdd = sc.parallelize(resultFile)
  .flatMap(line => line.split(" "))
  .map(word => (word , 1))
  .reduceByKey(_+_)
```

2- Résultat d'exécution :

```
Administrateur: Invite de commandes - spark-shell

scala> val ex2=spark.read.textFile("C:\\Users\\Lenovo\\Desktop\\MASTER-MIDVI\\S2\\Big Data\\TD\\TD1&2 Spark\\TD2-Ex2.txt")
21/05/22 23:08:07 WARN SizeEstimator: Failed to check whether UseCompressedOops is set; assuming yes
ex2: org.apache.spark.sql.Dataset[String] = [value: string]

scala> import org.apache.spark.SparkContext._
import org.apache.spark.SparkContext._

scala> val ex2=spark.read.textFile("C:\\Users\\Lenovo\\Desktop\\MASTER-MIDVI\\S2\\Big Data\\TD\\TD1&2 Spark\\TD2-Ex2.txt")
ex2: org.apache.spark.sql.Dataset[String] = [value: string]

scala> val mots=ex2.flatMap(line => line.splite(" "))
<console>:28: error: value splite is not a member of String
    val mots=ex2.flatMap(line => line.splite(" "))
                                   ^

scala> val mots=ex2.flatMap(line => line.split(" "))
mots: org.apache.spark.sql.Dataset[String] = [value: string]

scala> mots.show()
+-----+
|   value|
+-----+
|ACACACAGT|
+-----+

scala>
```

Exercise 3 :

1- Code :

```

var samplefile = sc.textFile(inputPath)
var ArrayFile = samplefile.collect()
var resultFile = ArrayBuffer[String]()
  for ( i<- ArrayFile){
    for (j <- 0 until i.length()-2){
      if ( i.substring(j, j+3).equals("ATG")){
        for (k <- j+3 until i.length()-2){
          if ( i.substring(k, k+3).equals("TAA") ||
i.substring(k, k+3).equals("TAG") || i.substring(k,
k+3).equals("TGA"))
            resultFile += i.substring(j, k+3) } } }
var rdd = sc.parallelize(resultFile)
  .flatMap(line => line.split(" "))
  .map(word => (word , 1))
  .reduceByKey(_+_ )
  .sortByKey(true, 1)

```

2- Résultat d'exécution :

```

Administrateur : Invite de commandes - spark-shell
|ACACAAACAGGGGTCGTTT|
+-----+

scala> var samplefile = sc.textFile("C:\\Users\\Lenovo\\Desktop\\MASTER-MIDVI\\S2\\Big Data\\TD\\TD1&2 Spark\\TD2-Ex3.txt")
samplefile: org.apache.spark.rdd.RDD[String] = C:\\Users\\Lenovo\\Desktop\\MASTER-MIDVI\\S2\\Big Data\\TD\\TD1&2 Spark\\TD2-Ex3.txt MapPartitionsRDD[15] at textFile at <console>:27

scala> var ArrayFile = samplefile.collect()
ArrayFile: Array[String] = Array(ACACAAACAGGGGTCGTTT)

scala> var resultfile = sc.textFile("C:\\Users\\Lenovo\\Desktop\\MASTER-MIDVI\\S2\\Big Data\\TD\\TD1&2 Spark\\TD2-Ex3.txt")
resultfile: org.apache.spark.rdd.RDD[String] = C:\\Users\\Lenovo\\Desktop\\MASTER-MIDVI\\S2\\Big Data\\TD\\TD1&2 Spark\\TD2-Ex3.txt MapPartitionsRDD[17] at textFile at <console>:27

scala> var resultFile = ArrayBuffer[String]()
<console>:26: error: not found: value ArrayBuffer
      var resultFile = ArrayBuffer[String]()
                        ^

scala> for( i<- ArrayFile){for(j <- 0 until i.length()-2){if(i.substring(j, j+3).equals("ATG")){for (k <- j+3 until i.length()-2){if ( i.substring(k, k+3).equals("TAA") || i.substring(k, k+3).equals("TAG") || i.substring(k, k+3).equals("TGA"))resultFile += i.substring(j, k+3) } } }
| var rdd = sc.parallelize(resultFile).flatMap(line => line.split(" ")).map(word => (word , 1)).reduceByKey(_+_ ).sortByKey(true, 1)
|
|
|

```

