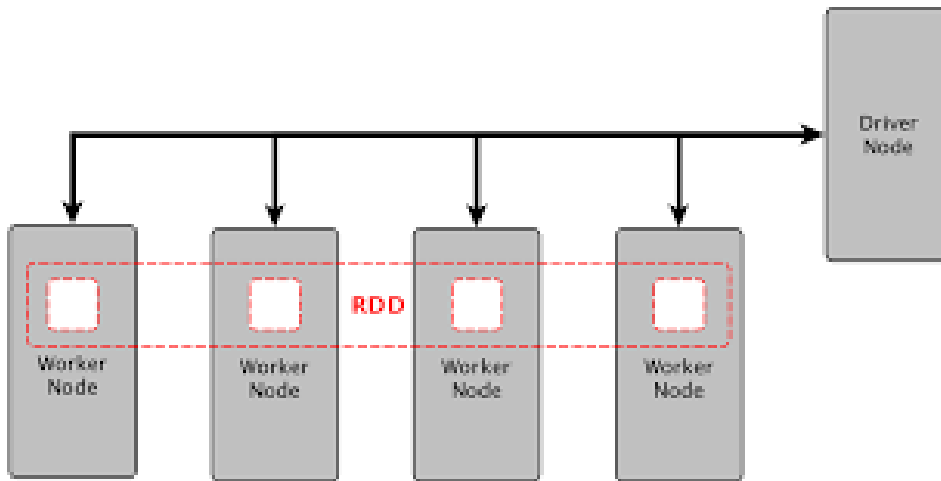


Bu bölümde Apache Spark kütüphanesinde bulunan RDD yapısını inceleyeceğiz

RDD(resilient distributed dataset) genel olarak Spark cluster üzerinde veriler üzerinde hesaplamalar yapmamızı sağlayan bir bileşendir.

Aynı zamanda verileri diğer sistemleri aktarabiliriz.



Örnek verirsek alttaki kod yapısı; okuduğumuz dosyanın kaç satır olduğunu cluster üzerinde hesaplar

```
val spark =
SparkSession.builder.master("local").appName("SparkByExample").getOrCreate()
val
rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/sample3.txt");
println("Count : " + rdd.count())
println("Count : " + rdd.first())
```

RDD Nasıl Olusturulur

Local bilgisayardan dosya ile

```
val  
rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/sample3.txt  
");
```

Hdfs üzerinden

```
val rdd=spark.sparkContext.textFile("hdfs://user/file");
```

Zip dosyalari uzerinden

```
val rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/*.gz");
```

parallelize() metodu ile

```
val spark =  
SparkSession.builder.master("local").appName("SparkByExample").getOrCreate(  
)  
val days = List("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",  
"Friday", "Saturday")  
val rdd=spark.sparkContext.parallelize(days);  
println("Count : " + rdd.count())  
println("Count : " + rdd.first())
```

Bu metod genel olarak Spark öğrenmeye başlarken kullanışlı olabilir. Verileri parametre olarak verebiliriz ve bu veriler cluster üzerinde dağıtılır

RDD Metodlari

RDD operasyonları genel olarak 2 bölüme ayrılır.

Transformation

Bu işlemde mevcut RDD üzerinden yeni bir RDD oluşturulur. Örnek verirse map ve filter metodlarından yeni RDD oluşturabiliriz

Altta ki örnekte filter metodu yeni bir RDD oluşturuyor

```
val spark =
SparkSession.builder.master("local").appName("SparkByExample").getOrCreate(
)
val
rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/error.txt")
;
println("Count : " + rdd.count())
println("Count : " + rdd.first())

val filteredRDD = rdd.filter(line => line.contains("WARN"))

println("----- > Filtered Count : " +
filteredRDD.count())
filteredRDD.foreach(line => println("----->" + line))
```

Action

Bu işlemde RDD üzerinden hesaplama, dış sistemlere verileri kaydetme işlemleri yapılır. Örnek olarak count, first metodları örnek verilebilir

Altta ki örnekte take metodu ile ilk 2 kayıt List yapısında gösteriliyor . Benzer şekilde collect metodu ise tüm verileri cluster üzerinden toplamaya yarar

```
val spark =  
SparkSession.builder.master("local").appName("SparkByExample").getOrCreate()  
val  
rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/error  
.txt");  
  
val twoRecord = rdd.take(2);  
  
for (line <- twoRecord)  
  println("-----> " + line)
```

Lazy Evolution Kavrami

Apache Spark mimarisinde transformasyon işlemlerinde herhangi bir aksiyon alınmaz. Spark count, first, take gibi bir action metodu gördüğü zaman işlemleri başlatır

