



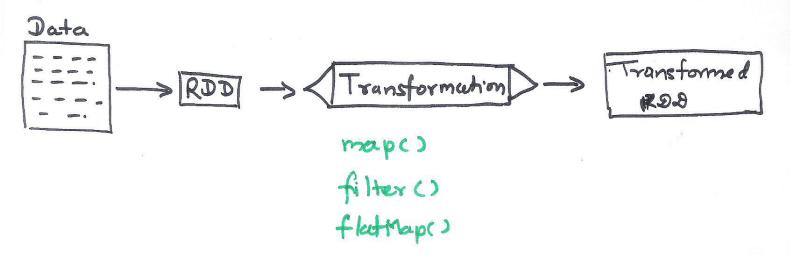
SPARK TRANSFORMATION IN DEPTH

By www.HadoopExam.com

Note: These instructions should be used with the HadoopExam Apache Spark: Professional Trainings.

Where it is executed and you can do hands on with trainer.

Transformation In RDD



Element-wise Tournsformation: -

- Transformation works on each element
- Two most common transformation
 - mapc)
 - filteres

=> map() 1

List [1,2,3,4]. map(?) With each element of list

- 3 I want to double each element
- 3 Replace? with function, which can double an element

List [1,2,3,4].
$$map(x \Rightarrow) \times +2)$$

List [2,4,6,8]

Filter:

What do you want to

List [1,2,3,4,5,6] filter(?) do with each element

of list

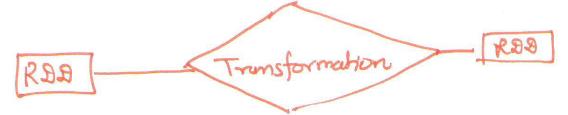
- 1 We want to filter/ select values other than 3
- 3 Pass the function which filter values other than 3

List [1,2,3,4,5,6]. filter (X=>X!=3)

Input ROD

List [1,2,4,5,6] = Output RAD

> mape) and filtered both are the function which idependently applied on each element of input Roo



=> Return type does not have to be the same as its input type.

RAD[String] -> Txons. RAD[Double]

FlatMap: produce multiple output for each element

"Learning Hadoop"

"Learning Scala"

"Learning HBase"

"Learning HBase"

"Learning Data Science"

"Learning", "HBase"

"Learning", "HBase"

"Learning", "Pata",

"Science"

"Do the word count)

flotMap(?) Will help to create multiple elements
(Return iterator) e.g. Split () function

- => The function (?) we provide to flat May () is called individually for each element to input ROD
- => This "?" function will not return individual element, it return an iterator with return values.

fled Map [List [Iterators, Iterators, Iteratoria]

=> Now this flat Map will not return list of iterators

=> It will return all the elements combined from all iterators.

-, Given Simy Sample split all the words from the

"Learning Input Ros 8 plit (" Learning scala") "Learning Spenk" List [Learning, scala] "Learning HBase" Learning Scala" input ROD. flat Map (Split) input Rea. map (Split) ? [Learning, Hadoop], ? Learning, Hadrop, Learning, [Learning, 8 park], Spenk, Learning, MBase, [Learning, HBase], Learning, scale } [Learning, Scala] 9

Output ROD

output ROD

Hands on Exercise Lab

Map function squaring the values

```
val input = sc.parallelize(List(1, 2, 3, 4))
val result = input.map(x => x * x)
println(result.collect().mkString(","))
```

• Splitting lines in multiple words.

```
val lines = sc.parallelize(List("hello HadoopExam.com", "hi"))
val words = lines.flatMap(line => line.split(" "))
words.first() // returns "hello"
```

Return an RDD consisting of only elements that pass the condition passed to filter().

```
val result = input.filter(x => x != 1)
println(result.collect().mkString(","))
```

• Another example on map()

```
val I = sc.parallelize(List(1, 2, 3, 4,5))

val result = l.map( x => x*2 )
println(result.collect().mkString(","))

def f(x: Int) = if (x > 2) Some(x) else None

val result = l.map(x => f(x))
println(result.collect().mkString(","))

def g(v:Int) = List(v-1, v, v+1)
val result = l.map(x => g(x))
println(result.collect().mkString(","))

val result = l.flatMap(x => g(x))
println(result.collect().mkString(","))
```

Set operation on RDD

```
val rdd = sc.parallelize(List(1,2,3))
val other = sc.parallelize(List(3,4,5))
val result = rdd.union(other)
```

println(result.collect().mkString(","))

val result = rdd.intersection(other)
println(result.collect().mkString(","))

val result = rdd.subtract(other)
println(result.collect().mkString(","))

val result = rdd.cartesian(other)
println(result.collect().mkString(","))

HADOOPEXAM ALL PRODUCTS TRAINING'S

- Hadoop BigData Professional Training (3500INR/\$79)
- HBase (NoSQL) Professional Training (3500INR/\$79)
- Apache Spark Professional Training (3500INR/\$79)

CLOUDERA HADOOP AND NOSQL CERTIFICATION

- CCD410 : Hadoop Developer
- CCA50X : Hadoop Administrator
- HBase/NoSQL Developer

DATABRICKSA OREILLY SPARK CERTIFICATION

Apache Spark Developer

AWS: AMAZON WEBSERVICE CERTIFICATION

- AWS Solution Architect : Associate
- AWS Solution Architect: Professional
- AWS Developer : Associate
- AWS Sysops Admin : Associate

MICROSOFT AZURE CERTIFICATION

- Azure 70-532
- Azure 70-533

DATA SCIENCE CERTIFICATION

• EMC E20-007

EMC CERTIFICATIONS

EMC E20-007

SAS ANALYTICS CERTIFICATION

- SAS Base A00-211
- SAS Advanced A00-212
- SAS Analytics: A00-240
- SAS Administrator : A00-250

ORACLE JAVA CERTIFICATION

- <u>J</u>ava 1z0-808
- <u>l</u>ava 1z0-809

ORACLE DATABASE CLOUD CERTIFICATION

- <u>1z0-060 (Oracle 12c)</u>
- <u>1z0-061 (Oracle 12c)</u>