UNIVERSITETI /UNIVERSITY
"ISA BOLETINI"
MITROVICË

Big Data Processing

...

RDD

What is RDD?

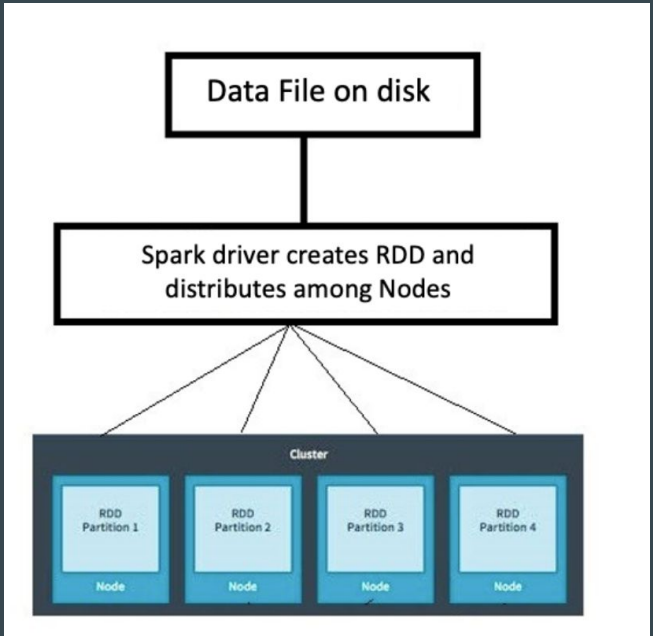
Resilient -> Ability to withstand failures ; **Distributed** -> Spanned across different machines ; **Dataset** -> collection of data

The **main abstraction Spark provides** is a resilient distributed dataset (**RDD**), which is the **fundamental** and **backbone data type** of this **engine**.

Is a fault-tolerant, immutable, distributed collection of objects

Same as list in python but with benefits as fault-tolerant, scalability

Difference is that RDD is computed on several processes scattered across multiple physical servers



How to create RDD?

- **Parallelizing** existing objects:

```
numb = range(5,10)
spark_data = sc.parallelize(numb)
```

To do so we need to have spark initialized and we do that by:

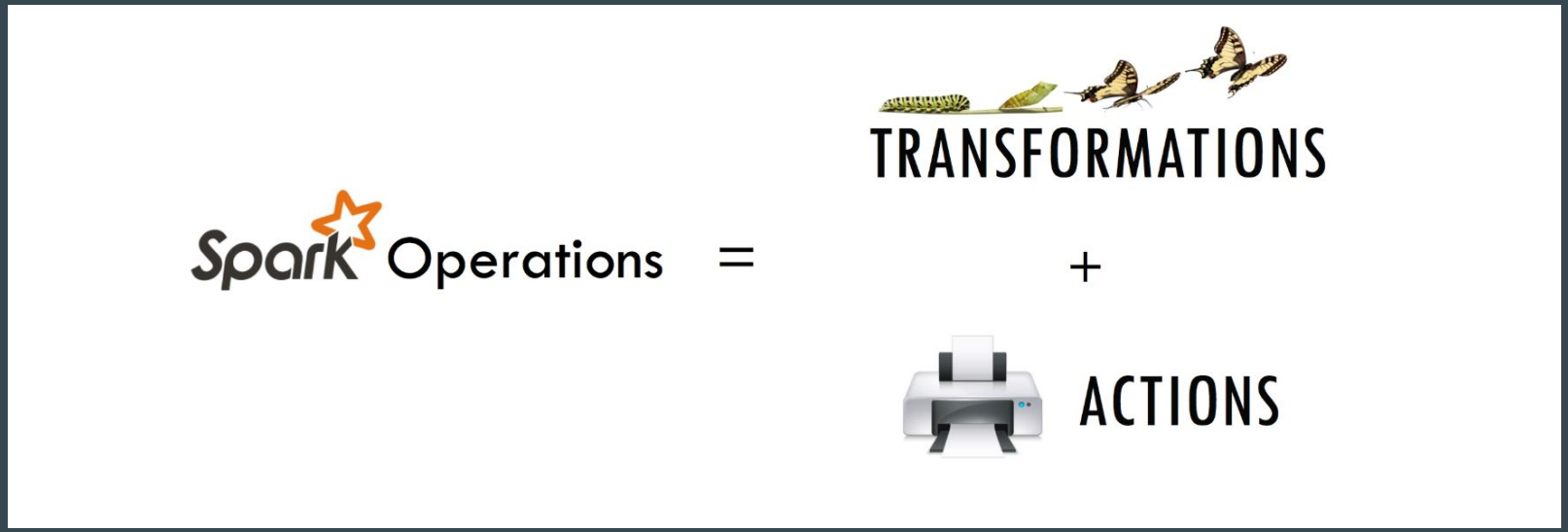
```
# Imports
from pyspark.sql import SparkSession

# Create SparkSession
spark = SparkSession.builder
      .master("local[1]")
      .appName("SparkByExamples.com")
      .getOrCreate()
```

- External datasets: files in HDF, objects in S3 buckets etc.

```
fileRDD = sc.textFile("README.md")
```

- From existing RDDs

RDD Transformations and Actions

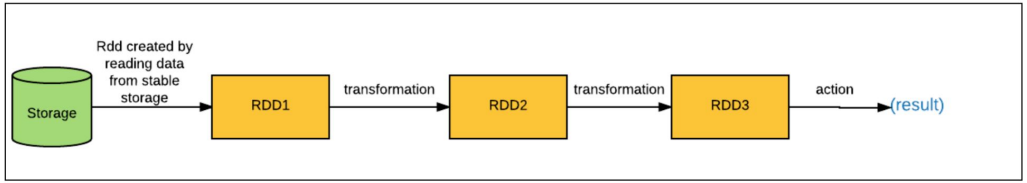


Transformations create new RDDs

Actions perform computation on RDD

RDD Transformations

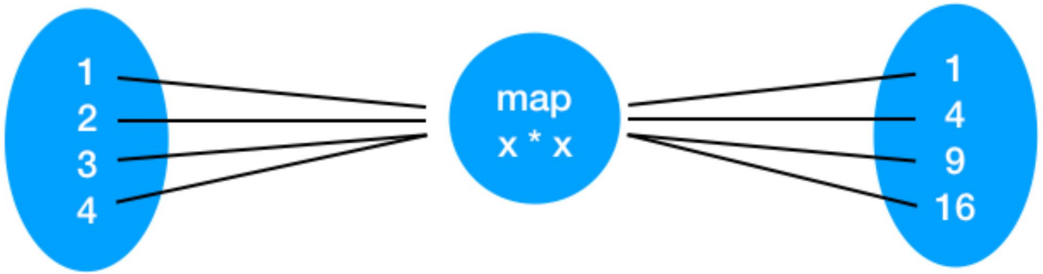
- Transformations follow Lazy evaluation



- Basic RDD Transformations
  - `map()` , `filter()` , `flatMap()` , and `union()`

map() Transformation

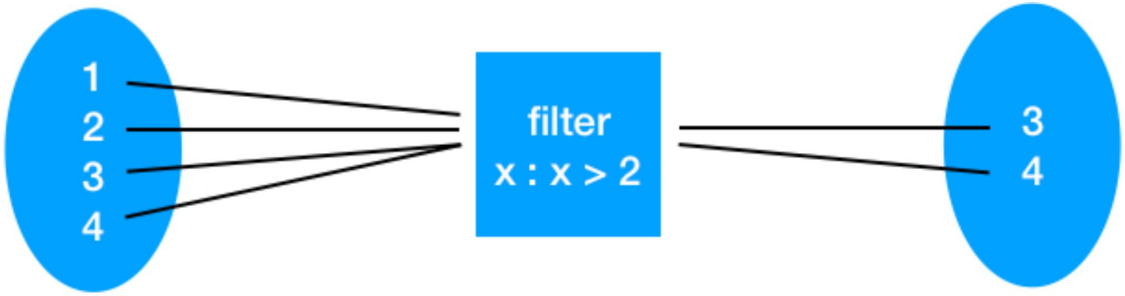
- map() transformation applies a function to all elements in the RDD




```
RDD = sc.parallelize([1,2,3,4])
RDD_map = RDD.map(lambda x: x * x)
```

filter() Transformation

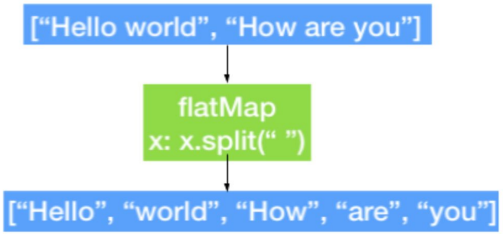
- Filter transformation returns a new RDD with only the elements that pass the condition




```
RDD = sc.parallelize([1,2,3,4])
RDD_filter = RDD.filter(lambda x: x > 2)
```

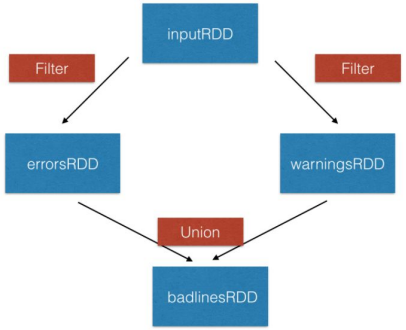
flatMap() Transformation

- flatMap() transformation returns multiple values for each element in the original RDD




```
RDD = sc.parallelize(["hello world", "how are you"])
RDD_flatmap = RDD.flatMap(lambda x: x.split(" "))
```

union() Transformation




```
inputRDD = sc.textFile("logs.txt")
errorRDD = inputRDD.filter(lambda x: "error" in x.split())
warningsRDD = inputRDD.filter(lambda x: "warnings" in x.split())
combinedRDD = errorRDD.union(warningsRDD)
```

What are actions?

They are operations that return a value after running a computation on the RDD

Basic RDD Actions

- collect() -> return all the elements of the dataset as an array
- take(N) -> returns an array with the first N elements of the dataset
- first() -> prints the first element of the RDD
- count() -> return the number of elements in the RDD

```
RDD_map.collect()
```

```
[1, 4, 9, 16]
```

```
RDD_map.take(2)
```

```
[1, 4]
```

```
RDD_map.first()
```

```
[1]
```

```
RDD_flatmap.count()
```

```
5
```

Example 1: Map and Collect

Use map() transformation to cube each number of the “numbRDD” RDD that you can create from a list of numbers. Next, you'll store all the elements in a variable and finally print the output.

Example 2: Filter and Count

Filter out lines containing keyword Spark from “fileRDD” RDD which consists of lines of text from the “examplefile.md” file. Next, you'll count the total number of lines containing the keyword Spark and finally print the first 4 lines of the filtered RDD.

Example 3: ReduceByKey and Collect

First create a pair RDD from a list of tuples, then combine the values with the same key and finally print out the result. Instructions:

- Create a **pair RDD** named Rdd with tuples (1,2),(3,4),(3,6),(4,5).
- Transform the Rdd with reduceByKey() into a pair RDD Rdd_Reduced by adding the values with the same key.
- Collect the contents of pair RDD Rdd_Reduced and iterate to print the output.

Example 4: SortByKey and Collect

Sort the pair RDD “Rdd_Reduced” that you created in the previous exercise into descending order and print the final output.

Instructions:

- Sort the Rdd_Reduced RDD using the key in descending order.
- Collect the contents and iterate to print the output.

Example 5: Count by keys

Create and use a Rdd and count the number of unique keys in that pair RDD.

Instructions:

- countByKey and assign the result to a variable total.
- What is the type of total?
- Iterate over the total and print the keys and their counts.

UNIVERSITETI /UNIVERSITY
"ISA BOLETINI"
MITROVICË

Stay Curious. Questions?

Thank you for your attention.
Lorent Sinani

Scan to connect on LinkedIn