Operaciones sobre un RDD

Se pueden realizar dos tipos de operaciones en SPARK:

- **Transformaciones /** map(),flatMap(),filter()
- **Acciones /** count(),first(),collect()

Las **transformaciones** crean nuevos conjuntos de datos, como puede ser la operación **map()** que pasa cada elemento por una determinada función y devuelve un nuevo RDD con el resultado.

Las **transformaciones** son operaciones con RDD que dan como resultado un nuevo RDD. Permiten transformar el conjunto de datos en otro. Las entradas y salidas de las transformaciones son ambos RDDs; Por lo tanto, es posible encadenar múltiples transformaciones, acercándose a un estilo de funcional programación. Además, las transformaciones son perezosas, es decir, no calculan sus resultados inmediatamente.

Las transformaciones permiten aplicar funciones a todos los registros, ex: map, filter, join

Ejemplos de transformaciones:

- **filter():**Un ejemplo de transformación RDD es la de filtrado. El siguiente comando devuelve un nuevo conjunto de datos con únicamente las líneas que contienen la palabra “pyspark” en el RDD original. Filter lo que hace es devolver un nuevo conjunto de datos con aquellos que cumplan una determinada condición.

  - **filtroRDD= lines.filter(lambda line:”pyspark” in line)**

Las **acciones** son operaciones que devuelven resultados. Las acciones devuelven valores de RDDs, como la suma de los elementos,un contador, o simplemente recoger todos los elementos. Las acciones son el disparador necesario para ejecutar la cadena de transformaciones. Un ejemplo de este tipo es la función **reduce()**, que agrega todos los elementos de un RDD mediante una función y devuelve el resultado.

Las acciones permiten aplicar algunos cálculos y devolver los resultados, ex: reduce, count

Ejemplos de acciones:

- **collect()** devuelve todos los elementos de un RDD como una lista de python
- **count()** devuelve el número de elementos de un RDD
- **countByValue()** devuelve un diccionario con el número de apariciones de cada elemento en un RDD
- **take(n)** devuelve una lista con los primeros n elementos del RDD
- **collectAsMap()** devuelve los elementos de un RDD clave/valor como un diccionario de python
- **countByKey()** devuelve un diccionario donde las claves son las diferentes claves que el RDD contiene y los valores son el número de veces que cada clave aparece

Table 3-4. Basic actions on an RDD containing {1, 2, 3, 3}

Function name	Purpose	Example	Result
`collect()`	Return all elements from the RDD.	`rdd.collect()`	`{1, 2, 3, 3}`
`count()`	Number of elements in the RDD.	`rdd.count()`	`4`
`countByValue()`	Number of times each element occurs in the RDD.	`rdd.countByValue()`	`{(1, 1), (2, 1), (3, 2)}`