Apache Spark Scala - Cheat Sheet for Commonly Used Operations

Unary Transformations

Transformation	RDD	Result	Description
map(func) `rdd.map(x => x * x)`	`{1, 2, 3, 3}`	`{1, 4, 9, 9}`	Returns a new RDD by applying a function on each element of the RDD.
flatMap(func) `rdd.flatMap(line => line.split(" "))`	`{"hello awesome world", "hi"}`	`{"hello", "awesome", "world", "hi"}`	Returns a new RDD by applying a function on each element of the RDD, but output is flattened.
filter(func) `rdd.filter(x => x != 1)`	`{1, 2, 3, 3}`	`{2, 3, 3}`	Returns an RDD by selecting those data elements on which *func* returns true.
distinct `rdd.distinct`	`{1, 2, 3, 3}`	`{1, 2, 3}`	Returns a new RDD with all duplicates eliminated.

Binary Transformations

Transformation	RDD1	RDD2	Result	Description
union(RDD) `rdd1.union(rdd2)`	`{1,2,3}`	`{3,4,5}`	`{1,2,3,3,4,5}`	Returns a new RDD containing all elements from source RDD and argument.
intersection(RDD) `rdd1.intersection(rdd2)`	`{1,2,3}`	`{3,4,5}`	`{3}`	Returns a new RDD containing all elements from source RDD and argument.
subtract(RDD) `rdd1.subtract(rdd2)`	`{1,2,3}`	`{3,4,5}`	`{1, 2}`	Returns a new RDD created by removing data elements in source RDD in common with argument
cartesian(RDD) `rdd1.cartesian(rdd2)`	`{1,2,3}`	`{3,4,5}`	`{(1,3),(1,4), … (3,5)}`	Returns a new RDD cross product of all elements from source RDD and argument.

Unary on Pairs Transformations

Transformation	RDD	Result	Description
reduceByKey(func) `rdd.reduceByKey((x, y) => x + y)`	`{(1,2),(3,4),(3,6)}`	`{(1,2),(3,10)}`	Combines values with the same key and applies the function *func* on the values. It operates on two elements.
groupByKey() `rdd.groupByKey()`	`{(1,2),(3,4),(3,6)}`	`{(1,[2]),(3,[4,6])}`	Groups values with the same key.
mapValues(func) `rdd.mapValues(x => x+1)`	`{(1,2),(3,4),(3,6)}`	`{(1,3),(3,5),(3,7)}`	Applies a function to each value of a pair RDD without changing the key.
keys `rdd.keys`	`{(1,2),(3,4),(3,6)}`	`{1,3,3}`	Returns an RDD of just the keys.
values `rdd.values`	`{(1,2),(3,4),(3,6)}`	`{2,4,6}`	Returns an RDD of just the values.

Apache Spark Scala - Cheat Sheet for Commonly Used Operations

sortByKey() rdd.sortByKey()	{(1,2),(3,4),(2,3)}	{(1,2),(2,3),(3,4)}		Returns an RDD sorted by the key.
sortBy(func) rdd.sortBy(x => x._2)	{(1,2),(3,4),(2,1)}	{(2,1),(1,2),(3,4)}		Returns an RDD sorted.
Binary on Pairs Transformations				
Transformation	**RDD1**	**RDD2**	**Result**	**Description**
subtractByKey(RDD) rdd1.subtractByKey(rdd2)	{(1,2),(3,4),(3,6)}	{(3,9)}	{(1,2)}	Removes elements with a key present in the other RDD.
join() rdd1.join(rdd2)	{(1,2),(3,4),(3,6)}	{(3,9)}	{(3,(4,9)), (3,(6,9))}	Performs an inner join between two RDDs.
leftOuterJoin(RDD) rdd1.leftOuterJoin(rdd2)	{(1,2),(3,4),(3,6)}	{(3,9)}	{(1,(2,None)), (3,(4,Some(9))), (3,(6,Some(9)))}	Performs a join between two RDDs where the resulting pair RDD has entries for each key in the first RDD.
rightOuterJoin(RDD) rdd1.rightOuterJoin(rdd2)	{(1,2),(3,4),(3,6)}	{(3,9)}	{(3,(Some(4),9)), (3,(Some(6),9))}	Performs a join between two RDDs where resulting pair RDD has entries for each key in the other RDD.
cogroup(RDD) rdd1.cogroup(rdd2)	{(1,2),(3,4),(3,6)}	{(3,9)}	{(1,([2],[])), (3,([4,6],[9]))}	Groups data from both RDDs sharing the same key.
Actions				
Action	**RDD**	**Result**	**Description**	
count() rdd.count()	{1,2,3,3}	4	Returns the number of elements in the RDD	
countByKey() rdd.countByKey()	{(1,"a"),(1,"b"),(2,"c"), (2,"d"),(2,"e")}	(2,3),(1,2)	Only available on RDDs of type (K, V). Returns a hashmap of (K, Int) pairs with the count of each key.	
collect() rdd.collect()	{1,2,3}	{1,2,3}	Returns an array of all the data elements in an RDD.	
take(n) rdd.take(2)	{1,2,3,4}	{1,3}	Fetch the first n data elements in an RDD. Computed by driver program.	
reduce(func) rdd.reduce(_ + _)	{1,2,3,4}	10	Aggregates the data elements in an RDD using this function which takes two arguments and returns one.	
foreach(func) rdd.take(5).foreach(func)		Executes function for each data element in RDD.		
saveAsTextFile(path) rdd.saveAsTextFile(path)		Writes the content of RDD to a text file or a set of text files to local file system/ HDFS		