影响到Spark输出RDD分区的操作函数

下面的操作会影响到Spark输出RDD分区（partitioner）的：
cogroup, groupWith, join, leftOuterJoin, rightOuterJoin, groupByKey, reduceByKey, combineByKey, partitionBy, sort, mapValues
(如果父RDD存在partitioner), flatMapValues(如果父RDD存在partitioner),
和 filter
(如果父RDD存在partitioner)。其他的transform操作不会影响到输出RDD的partitioner，一般来说是None，也就是没有partitioner。下面举个例子进行说明：

```
scala> val pairs = sc.parallelize(List((1, 1), (2, 2), (3, 3)))
pairs: org.apache.spark.rdd.RDD[(Int, Int)] =
ParallelCollectionRDD[4] at parallelize at <console>:12

scala> val a = sc.parallelize(List(2,51,2,7,3))
a: org.apache.spark.rdd.RDD[Int] =
ParallelCollectionRDD[5] at parallelize at <console>:12

scala> val a = sc.parallelize(List(2,51,2))
a: org.apache.spark.rdd.RDD[Int] =
ParallelCollectionRDD[6] at parallelize at <console>:12

scala> val b = sc.parallelize(List(3,1,4))
b: org.apache.spark.rdd.RDD[Int] =
 ParallelCollectionRDD[7] at parallelize at <console>:12

scala> val c = a.zip(b)
c: org.apache.spark.rdd.RDD[(Int, Int)] =
ZippedPartitionsRDD2[8] at zip at <console>:16

scala> val result = pairs.join(c)
result: org.apache.spark.rdd.RDD[(Int, (Int, Int))] =
FlatMappedValuesRDD[11] at join at <console>:20

scala> result.partitioner
res6: Option[org.apache.spark.Partitioner] = Some(org.apache.spark.HashPartitioner@2)
```

大家可以看到输出来的RDD result分区变成了HashPartitioner，因为join中的两个分区都没有设置分区，所以默认用到了HashPartitioner，可以看join的实现：

```
def join[W](other: RDD[(K, W)]): RDD[(K, (V, W))] = {
  join(other, defaultPartitioner(self, other))
}

def defaultPartitioner(rdd: RDD[_], others: RDD[_]*): Partitioner = {
  val bySize = (Seq(rdd) ++ others).sortBy(_.partitions.size).reverse
  for (r <- bySize if r.partitioner.isDefined) {
    return r.partitioner.get
  }
  if (rdd.context.conf.contains("spark.default.parallelism")) {
    new HashPartitioner(rdd.context.defaultParallelism)
  } else {
    new HashPartitioner(bySize.head.partitions.size)
  }
}
```

defaultPartitioner函数就确定了结果RDD的分区。从上面的实现可以看到，

1、join的两个RDD如果都没有partitioner，那么join结果RDD将使用HashPartitioner；

2、如果两个RDD中其中有一个有partitioner，那么join结果RDD将使用那个父RDD的partitioner；

3、如果两个RDD都有partitioner，那么join结果RDD就使用调用join的那个RDD的partitioner。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。
本文链接: 【】（）