Spark函数讲解：coalesce

对RDD中的分区重新进行合并。

函数原型

```
def coalesce(numPartitions: Int, shuffle: Boolean = false)
            (implicit ord: Ordering[T] = null): RDD[T]
```

返回一个新的RDD，且该RDD的分区个数等于numPartitions个数。如果shuffle设置为true，则会进行shuffle。

实例

```
/**
 * User: 过往记忆
 * Date: 15-03-09
 * Time: 上午06:30
 * bolg:
 * 本文地址：/archives/1279
 * 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 * 过往记忆博客微信公共帐号：iteblog_hadoop
 */
scala> var data = sc.parallelize(List(1,2,3,4))
data: org.apache.spark.rdd.RDD[Int] =
            ParallelCollectionRDD[45] at parallelize at <console>:12

scala> data.partitions.length
res68: Int = 30

scala> val result = data.coalesce(2, false)
result: org.apache.spark.rdd.RDD[Int] = CoalescedRDD[57] at coalesce at <console>:14

scala> result.partitions.length
res77: Int = 2

scala> result.toDebugString
res75: String =
(2) CoalescedRDD[57] at coalesce at <console>:14 []
 |  ParallelCollectionRDD[45] at parallelize at <console>:12 []
```

```
scala> val result1 = data.coalesce(2, true)
result1: org.apache.spark.rdd.RDD[Int] = MappedRDD[61] at coalesce at <console>:14

scala> result1.toDebugString
res76: String =
(2) MappedRDD[61] at coalesce at <console>:14 []
 |  CoalescedRDD[60] at coalesce at <console>:14 []
 |  ShuffledRDD[59] at coalesce at <console>:14 []
 +-(30) MapPartitionsRDD[58] at coalesce at <console>:14 []
    |   ParallelCollectionRDD[45] at parallelize at <console>:12 []
```

从上面可以看出shuffle为false的时候并不进行shuffle操作；而为true的时候会进行shuffle操作。RDD.partitions.length可以获取相关RDD的分区数。