Spark函数讲解：combineByKey

使用用户设置好的聚合函数对每个Key中的Value进行组合(combine)。可以将输入类型为RDD[(K, V)]转成成RDD[(K, C)]。

函数原型

```
def combineByKey[C](createCombiner: V => C, mergeValue: (C, V) => C,
   mergeCombiners: (C, C) => C) : RDD[(K, C)]
def combineByKey[C](createCombiner: V => C, mergeValue: (C, V) => C,
   mergeCombiners: (C, C) => C, numPartitions: Int): RDD[(K, C)]
def combineByKey[C](createCombiner: V => C, mergeValue: (C, V) => C,
   mergeCombiners: (C, C) => C, partitioner: Partitioner, mapSideCombine:
   Boolean = true, serializer: Serializer = null): RDD[(K, C)]
```

第一个和第二个函数都是基于第三个函数实现的，使用的是HashPartitioner，Serializer为null。而第三个函数我们可以指定分区，如果需要使用Serializer的话也可以指定。combineByKey函数比较重要，我们熟悉地诸如aggregateByKey、foldByKey、reduceByKey等函数都是基于该函数实现的。默认情况会在Map端进行组合操作。

实例

```
/**
 * User: 过往记忆
 * Date: 15-03-19
 * Time: 上午08:24
 * bolg:
 * 本文地址：/archives/1291
 * 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> val data = sc.parallelize(List((1, "www"), (1, "iteblog"), (1, "com"),
      (2, "bbs"), (2, "iteblog"), (2, "com"), (3, "good")))
data: org.apache.spark.rdd.RDD[(Int, String)] =
      ParallelCollectionRDD[15] at parallelize at <console>:12

scala> val result = data.combineByKey(List(_),
      (x: List [String], y: String) => y :: x, (x: List[String], y: List[String]) => x ::: y)
result: org.apache.spark.rdd.RDD[(Int, List[String])] =
      ShuffledRDD[19] at combineByKey at <console>:14
```

```
scala> result.collect
res20: Array[(Int, List[String])] = Array((1,List(www, iteblog, com)),
      (2,List(bbs, iteblog, com)), (3,List(good)))

scala> val data = sc.parallelize(List(("iteblog", 1), ("bbs", 1), ("iteblog", 3)))
data: org.apache.spark.rdd.RDD[(String, Int)] =
      ParallelCollectionRDD[24] at parallelize at <console>:12

scala> val result = data.combineByKey(x => x,
      (x: Int, y:Int) => x + y, (x:Int, y: Int) => x + y)
result: org.apache.spark.rdd.RDD[(String, Int)] =
      ShuffledRDD[25] at combineByKey at <console>:14

scala> result.collect
res27: Array[(String, Int)] = Array((iteblog,4), (bbs,1))
```

第二个例子其实就是计算单词的个数，事实上，reduceByKey函数就是类似的计算。(x:Int, y: Int) => x + y就是我们传进reduceByKey函数的参数。