**共享变量**

spark一个非常重要的特性是共享变量。

如果在一个算子函数中使用了某个外部的变量，那么这个变量的值会被拷贝到每个task中，此时每个task只能操作自己的那份变量副本，如果多个task想要共享某个变量，那么这种方式是做不到的。

spark为此提供了共享变量，一种是broadcast variable（广播变量），一种是Accumulator（累加变量）。广播变量会将使用到的变量仅仅为每个节点拷贝一份，而不是每个task一份，更大的用处是优化性能，减少网络传输以及内存消耗。Accumulator则可以让多个task共同操作一份变量，主要可以进行累加操作。

spark提供的broadcast variable是只读的，并且在每个节点上只有一个副本，而不会为每个task拷贝一份。因此最大的作用是减少变量到各个节点的网络传输消耗和内存消耗。

**二次排序**

1.实现自定义的key，实现scala.math.Ordered和Serializable接口，在key中实现自己对多个列的排序算法。

2.将包含文本的RDD映射成自定义的key，value为二元组。

3.使用sortByKey算子按照自定义的key排序。

**宽依赖和窄依赖**

窄依赖（Narrow Dependency）：子RDD的partition只依赖于一个父RDD的partition。

宽依赖（Shuffle Dependency）：子RDD的每个partition依赖于父RDD的一个以上的partition

**spark2-submit中excutor-cores的含义**

参数说明：该参数用于设置每个Executor进程的CPU core数量。这个参数决定了每个Executor进程并行执行task线程的能力。因为每个CPU core同一时间只能执行一个task线程，因此每个Executor进程的CPU core数量越多，越能够快速地执行完分配给自己的所有task线程。

参数调优建议：Executor的CPU core数量设置为2~4个较为合适。同样得根据不同部门的资源队列来定，可以看看自己的资源队列的最大CPU core限制是多少，再依据设置的Executor数量，来决定每个Executor进程可以分配到几个CPU core。同样建议，如果是跟他人共享这个队列，那么num-executors \* executor-cores不要超过队列总CPU core的1/3~1/2左右比较合适，也是避免影响其他同学的作业运行。