一：到底什么是Shuffle?

Shuffle中文翻译成为“洗牌”，需要Shuffle的关键性原因是某种具有共同特征的数据需要最终汇聚到一个计算节点上进行计算。

二：Shuffle可能面临的问题？**运行Task的时候才会产生Shuffle(Shuffle已经融化在Spark的算子中)**

1. 数据量非常大，网络传输收集数据的时候网络传输就很大；
2. 数据如何分类，即如何Partition，Hash,Sort,钨丝计划；
3. 负载均衡(数据倾斜)；
4. 网络传输效率；需要在压缩和解压缩之间做出权衡，序列化和反序列化也是要考虑的问题。

说明：具体的Task在计算的时候，尽一切最大可能使得数据具备Process Locality的特性。数据在内存中计算最快。如果不能保证所有数据在内存中计算**，退而求其次的做法是增加数据分片，减少每个Task处理数据量，**如果计算计算链条太长，计算非常复杂，实在没办法才将数据缓存到磁盘上，但是数据放到磁盘上，一是数据的读取比较慢，二是读取磁盘I/O本来就很危险。上述是在一个stage里面我们鼓励的做法。

三：Hash Shuffle

1. Key不能是Array
2. Hash Shuffle不需要排序，此时从理论上讲就节省了Hadoop MapReduce中进行Shuffle需要排序时候的时间浪费，因为实际生产环境中有大量的不需要排序的Shuffle类型；

思考：不需要排序的Hash Shuffle是否一定比需要排序的Sorted Shuffle速度更快?

不一定！如果数据规模比较小的情况下，Hash Shuffle会比Sorted Shuffle速度快(很多)！但是如果数据量大，此时Sorted Shuffle 一般都会比Hash Shuffle(很多)

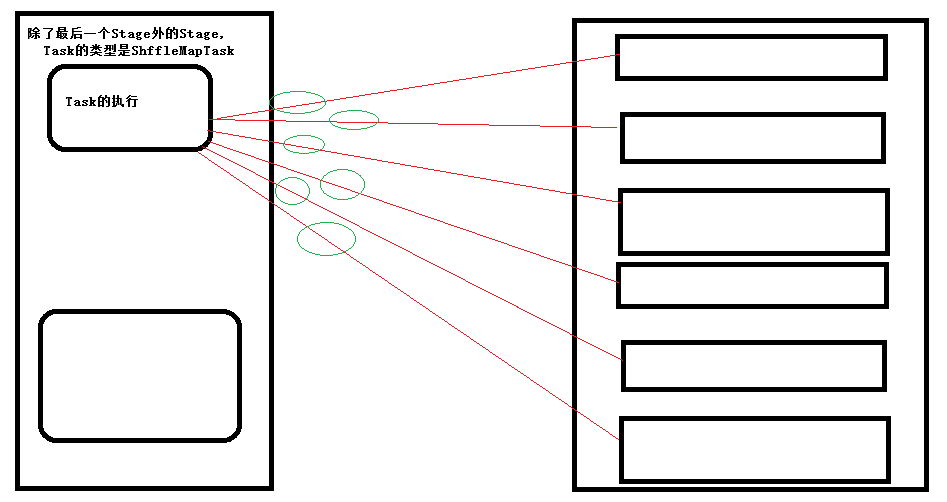
Spark在1.x之间Shuffle使用的是Hash Shuffle，hash Shuffle只能处理中小型数据，不能胜任大规模的数据，但是Sorted Shuffle可以胜任大规模的数据，因此现在Spark不仅可以处理小规模数据，也可以处理大规模数据。

1. 每个ShuffleMapTask会根据Key的哈希值计算出当前Key需要写入的Partition，然后把决定后的结果写入单独的文件。此时会导致每个Task产生R(指下一个Stage的并行度)个文件，如果当前的Stage中有M个ShuffleMapTask，则会M\*R个文件！！！

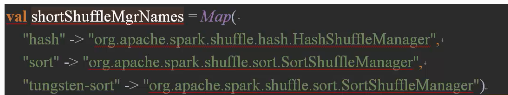
注意：Shuffle操作绝大数情况下都要通过网络，如果Mapper和Reducer在同一台机器上，此时只需要读取本地磁盘即可。

Hash Shuffle的两大死穴：第一：Shuffle前会产生海量的小文件于磁盘之上，此时会产生大量耗时低效的IO操作；第二：内存不够用！！！由于内存中需要保持海量的文件操作句柄和临时缓存信息，如果数据处理规模比较庞大的话，内存不可承受，出现OOM等问题！！

为了改善上述的问题(同时打开过多文件导致Writer Handler内存使用过大以及产生过多文件导致大量的随机读写带来的效率极为低下的磁盘IO操作)，Spark后推出了Consalidate机制，来把小文件合并。此时Shuffle时文件产生的数量为cores\*R，对于ShuffleMapTask的数量明显多于同时可用的并行cores的数量的情况下，Shuffle产生的文件会大幅度减少。会极大的降低OOM的可能：



为此Spark推出了Shuffle Pluggable开发框架，方便系统升级的时候定制Shuffle功能模块，也方便第三方系统改造人员根据实际的业务场景来开放具体最佳的Shuffle模块；核心接口ShuffleManager，具体默认实现有HashShuffleManager，SortShuffleManager等；Spark1.6.0中具体的配置如下：



默认配置是sort的模式，修改Shuffle默认配置是spark.defaults.conf



首先每个ShuffleMapTask不会为每个reduce单独生成一个文件，它会将所有的结果写入到一个文件里，同时会生成一个索引，每个reduce可以根据这个index索引，取得它所要处理的数据，这样就避免了产生大量的文件，节省了内存。这样就不要产生大量的文件，也就不需要大量文件的句柄，同时磁盘上面的文件也变小了，而且我们也有索引，这样就是顺序读取，带来了低延时，节省了内存，另外一方面，可以减少gc分享的频率，节省文件数量，可以避免同时写多个文件给系统带来的压力。

具体在实现的时候，ShuffleMapTask会按照Key相应的partitionId进行sort，如果属于同一个partition上面的Key，不会进行sort，如果内存不足，对于那些需要sort的操作，会将数据写入到外部磁盘，在最后结束的时候进行归并排序，index文件记录不同的partition位置，