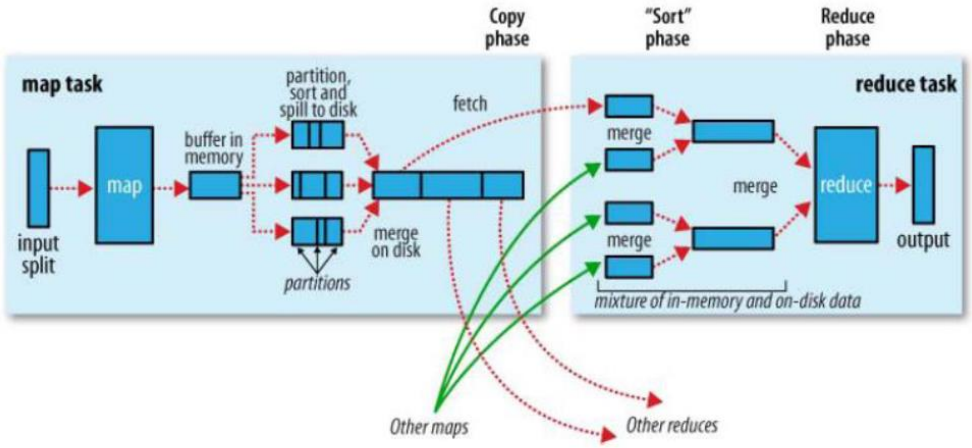
MapReduceshuffle 过程

map阶段处理的数据如何传递给reduce阶段，是MapReduce框架中最关键的一个流程，这个流程就叫shuffle。

shuffle：洗牌、发牌——（核心机制：数据分区，排序，分组，规约，合并等过程）。



shuffle是Mapreduce的核心，它分布在Mapreduce的map阶段和reduce阶段。一般把从Map产生输出开始到Reduce取得数据作为输入之前的过程称作shuffle。

1).Collect阶段：将MapTask的结果输出到默认大小为100M的环形缓冲区，保存的是key/value，Partition分区信息等。

2).Spill阶段：当内存中的数据量达到一定的阀值的时候，就会将数据写入本地磁盘，在将数据写入磁盘之前需要对数据进行一次排序的操作，如果配置了combiner，还会将有相同分区号和key的数据进行排序。

3).Merge阶段：把所有溢出的临时文件进行一次合并操作，以确保一个MapTask最终只产生一个中间数据文件。

4).Copy阶段：ReduceTask启动Fetcher线程到已经完成MapTask的节点上复制一份属于自己的数据，这些数据默认会保存在内存的缓冲区中，当内存的缓冲区达到一定的阀值的时候，就会将数据写到磁盘之上。

5).Merge阶段：在ReduceTask远程复制数据的同时，会在后台开启两个线程对内存到本地的数据文件进行合并操作。

6).Sort 阶段：在对数据进行合并的同时，会进行排序操作，由于 MapTask 阶段已经对数据进行了局部的排序，ReduceTask 只需保证 Copy 的数据的最终整体有效性即可。

Shuffle中的缓冲区大小会影响到mapreduce程序的执行效率，原则上说，缓冲区越大，磁盘io的次数越少，执行速度就越快

缓冲区的大小可以通过参数调整， 参数：mapreduce.task.io.sort.mb 默认100M