**Hadoop核心思想是MapReduce:**

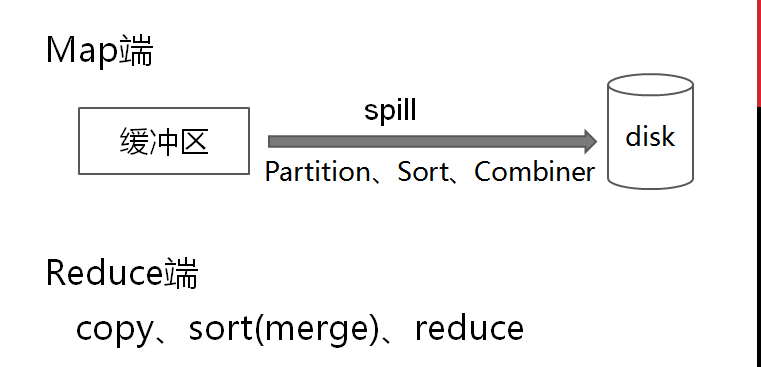
Map的作用是过滤一些原始数据;

shuffle就是map和reduce之间的过程，包含了两端的combine和partition。

Reduce则是处理这些数据，得到我们想要的结果.

参考[hadoop运行原理之shuffle](http://www.cnblogs.com/gwgyk/p/3997849.html)

http://www.cnblogs.com/gwgyk/p/3997849.html



1. **Map:映射**

应用到食谱：制作酱料时，要先把材料切碎

map操作就是将数据分成碎片，其中若遇到不相关或错误数据，map会过滤掉不相关或错误数据，这样在分布式计算中就不会出现较大失误。

Map reduce:key

应用到食谱：想制作各种口味的酱料，在搅拌阶段采用不用的原料

在以key为基础的map操作下，遍历整个作业，过滤有效数据，即分类含相同的key的键值对<key, value>，然后传给reduce函数进行整合，这样就有多种keys。

1. **Shuffle:洗牌**

为了确保每个reducer的输入都按键排序，系统执行排序的过程--Shuffle描述着数据从map task输出到reduce task输入的这段过程。

作用：

1. 完整地从map task端拉取数据到reduce 端。
2. 在跨节点拉取数据时，尽可能地减少对带宽的不必要消耗。
3. 减少磁盘IO对task执行的影响。



(图中的partitions、copy phase、sort phase所代表的就是shuffle的不同阶段)

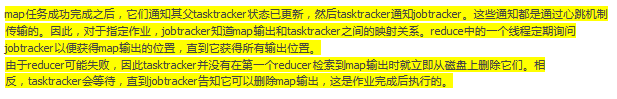
参考**[MapReduce:详解Shuffle过程](http://langyu.iteye.com/blog/992916)**

<http://langyu.iteye.com/blog/992916>

1. **Ruduce:化简**

应用到食谱：你将捣碎的材料放入研磨机研磨，就可以得到酱料

Ruduce就是将Map后的数据碎片聚集在一起。



**实例**：如果想统计下过去10年计算机论文出现最多的几个单词，看看大家都在研究些什么，那收集好论文后，该怎么办呢？

我们可以将程序部署到N台机器上去，然后把论文集分成N份，一台机器跑一个作业。这个方法跑得足够快，但是部署起来很麻烦，我们要人工把程序copy到别的机器，要人工把论文集分开，最痛苦的是还要把N个运行结果进行整合（当然我们也可以再写一个程序）。

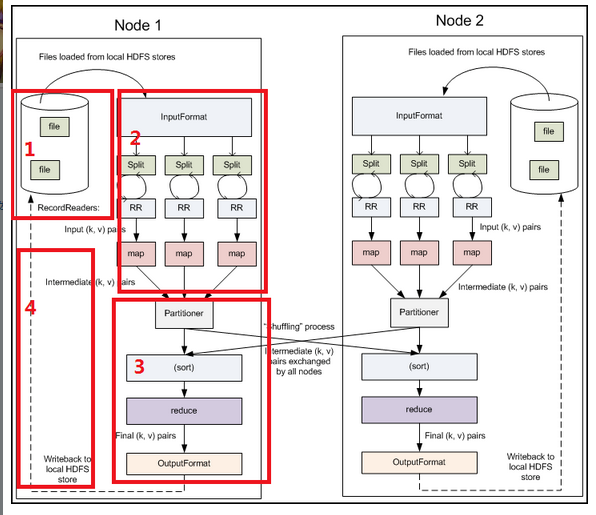
MapReduce本质上属于上述方法，但是如何拆分文件集，如何copy程序，如何整合结果这些都是框架定义好的。我们只要定义好这个任务（用户程序），其它都交给MapReduce。

map函数：接受一个键值对（key-value pair），产生一组中间键值对。MapReduce框架会将map函数产生的中间键值对里键相同的值传递给一个reduce函数。   
　　reduce函数：接受一个键，以及相关的一组值，将这组值进行合并产生一组规模更小的值（通常只有一个或零个值）。

注意Map/Reduce作业和map/reduce函数的区别：

Map作业处理一个输入数据的分片，可能需要调用多次map函数来处理每个输入键值对； Reduce作业处理一个分区的中间键值对，期间要对每个不同的键调用一次reduce函数，Reduce作业最终也对应一个输出文件。

工作机制



图中1：表示待处理数据，比如日志，比如单词计数  
图中2：表示map阶段，对他们split，然后送到不同分区  
图中3：表示reduce阶段，对这些数据整合处理。  
图中4：表示二次mapreduce,这个是mapreduce的链式，详细可以看[让你真正明白什么是MapReduce组合式，迭代式，链式](http://www.aboutyun.com/thread-7435-1-1.html" \t "http://www.aboutyun.com/_blank)