对相互之间不具有计算依赖关系的大数据，实现并行最自然的办法就是采取“分而治之”的策略。

MapReduce执行的全过程概述为以下几个主要阶段：

1）从分布式文件系统读入数据；

2）执行Map任务输出中间结果；

3）通过 Shuffle阶段把中间结果分区排序整理后发送给Reduce任务；

4）执行Reduce任务得到最终结果并写入分布式文件系统。

需要注意的是：

不同的Map任务之间不会进行通信

不同的Reduce任务之间也不会发生任何信息交换

用户不能显式地从一台机器向另一台机器发送消息

所有的数据交换都是通过MapReduce框架自身去实现的

读取HDFS中的文件。每一行解析成一个<k,v>。每一个键值对调用一次map函数

重写map()，对第一步产生的<k,v>进行处理，转换为新的<k,v>输出

对输出的（key，value）进行分区

对不同分区的数据，按照key进行排序、分组。相同key的value放到一个集合中

多个map任务的输出，按照不同的分区，通过网络复制到不同的reduce节点上

对多个map的输出进行合并、排序。

重写reduce函数实现自己的逻辑，对输入的key、value处理，转换成新的key、value输出

把reduce的输出保存到文件中

关于Split（分片），需要指出：HDFS以固定大小的block为基本单位存储数据，而对于MapReduce而言，其处理单位是split。split是一个逻辑概念，它只包含一些元数据信息，比如数据起始位置、数据长度、数据所在节点等。它的划分方法完全由用户自己决定。

Map任务的数量：

Hadoop为每个split创建一个Map任务，split 的多少决定了Map任务的数目。大多数情况下，理想的分片大小是一个HDFS块

Reduce任务的数量：

最优的Reduce任务个数取决于集群中可用的reduce任务槽(slot)的数目

通常设置比reduce任务槽数目稍微小一些的Reduce任务个数（这样可以预留一些系统资源处理可能发生的错误）