**1、hadoop运行的原理?**

hadoop主要由三方面组成:  
1、HDFS  
2、MapReduce  
3、Hbase

Hadoop框架中最核心的设计就是：MapReduce和HDFS。MapReduce的思想是由Google的一篇论文所提及而被广为流传的， 简单的一句话解释MapReduce就是“任务的分解与结果的汇总”。HDFS是Hadoop分布式文件系统（Hadoop Distributed File System）的缩写 ，为分布式计算存储提供了底层支持。

MapReduce从它名字上来看就大致可以看出个缘由，两个动词Map和Reduce，“Map（展开）”就是将一个任务分解成为多个任 务，“Reduce”就是将分解后多任务处理的结果汇总起来，得出最后的分析结果。这不是什么新思想，其实在前面提到的多线程，多任务的设计就可以找到这 种思想的影子。不论是现实社会，还是在程序设计中，一项工作往往可以被拆分成为多个任务，任务之间的关系可以分为两种：一种是不相关的任务，可以并行执 行；另一种是任务之间有相互的依赖，先后顺序不能够颠倒，这类任务是无法并行处理的。回到大学时期，教授上课时让大家去分析关键路径，无非就是找最省时的 任务分解执行方式。在分布式系统中，机器集群就可以看作硬件资源池，将并行的任务拆分，然后交由每一个空闲机器资源去处理，能够极大地提高计算效率，同时 这种资源无关性，对于计算集群的扩展无疑提供了最好的设计保证。（其实我一直认为Hadoop的卡通图标不应该是一个小象，应该是蚂蚁，分布式计算就好比 蚂蚁吃大象，廉价的机器群可以匹敌任何高性能的计算机，纵向扩展的曲线始终敌不过横向扩展的斜线）。任务分解处理以后，那就需要将处理以后的结果再汇总起 来，这就是Reduce要做的工作。

**2、mapreduce的原理?**

Hadoop中的MapReduce是一个使用简易的软件框架，基于它写出来的应用程序能够运行在由上千个商用机器组成的大型集群上，并以一种可靠容错的式并  行处理上T级别的数据集。  
一个MapReduce作业（job）通常会把输入的数据集切分为若干独立的数据块，由map任务（task）以完全并行的方式处理它们。框架会对map的输出先进行排序，然后把结果输入给reduce任务。通常作业的输入和输出都会被存储在文件系统中。整个框架负责任务的调度和监控，以及重新执行已经失败的任务。  
通常，MapReduce框架和分布式文件系统是运行在一组相同的节点上的，也就是说，计算节点和存储节点通常在一起。这种配置允许框架在那些已经存好数据的节点上高效地调度任务，这可以使整个集群的网络带宽被非常高效地利用。  
MapReduce框架由一个单独的master JobTracker和每个集群节点一个slave TaskTracker共同组成。master负责调度构成一个作业的所有任务，这些任务分布在不同的slave上，master监控它们的执行，重新执行已经失败的任务。而slave仅负责执行由master指派的任务

**3、HDFS存储的机制?**

HDFS的三个实体  
数据块  
每个磁盘都有默认的数据块大小,这是磁盘进行读写的基本单位.构建于单个磁盘之上的文件系统通过磁盘块来管理该文件系统中的块.该文件系统中的块一般为磁盘块的整数倍.磁盘块一般为512字节.HDFS也有块的概念,默认为64MB(一个map处理的数据大小).HDFS上的文件也被划分为块大小的多个分块,与其他文件系统不同的是,HDFS中小于一个块大小的文件不会占据整个块的空间.  
HDFS用块存储带来的第一个明显的好处一个文件的大小可以大于网络中任意一个磁盘的容量,数据块可以利用磁盘中任意一个磁盘进行存储.第二个简化了系统的设计,将控制单元设置为块,可简化存储管理,计算单个磁盘能存储多少块就相对容易.同时也消除了对元数据的顾虑,如权限信息,可以由其他系统单独管理.  
DataNode节点  
DataNode是HDFS文件系统的工作节点,它们根据需要存储并检索数据块,受NameNode节点调度.并且定期向NameNode发送它们所存储的块的列表  
NameNode节点  
NameNode管理HDFS文件系统的命名空间,它维护着文件系统树及整棵树的所有的文件及目录.这些文件以两个文件形式永久保存在本地磁盘上(命名空间镜像文件和编辑日志文件).NameNode记录着每个文件中各个块所在的数据节点信息但并不永久保存这些块的位置信息,因为这些信息在系统启动时由数据节点重建.  
没有NameNode,文件系统将无法使用.如提供NameNode服务的机器损坏,文件系统上的所有文件丢失,我们就不能根据DataNode的块来重建文件.因此,对NameNode的容错非常重要.第一种机制,备份那些组成文件系统元数据持久状态的文件.通过配置使NameNode在多个文件系统上保存元数据的持久状态或将数据写入本地磁盘的同时,写入一个远程挂载的网络文件系统.当然这些操作都是原子操作.第二种机制是运行一个辅助的NameNode,它会保存合并后的命名空间镜像的副本,并在Name/Node发生故障时启用.但是辅助NameNode保存.态总是滞后于主力节点,所以在主节点全部失效后难免丢失数据.在这种情况下,一般把存储在远程挂载的网络文件系统的数据复制到辅助NameNode并作为新的主NameNode运行

**4、举一个简单的例子说明mapreduce是怎么来运行的 ?**

**5、面试的人给你出一些问题,让你用mapreduce来实现？**

比如:现在有10个文件夹,每个文件夹都有1000000个url.现在让你找出top1000000url。

**6、hadoop中Combiner的作用?**

1、combiner最基本是实现本地key的聚合，对map输出的key排序，value进行迭代。如下所示：

map: (K1, V1) → list(K2, V2)  
combine: (K2, list(V2)) → list(K2, V2)  
reduce: (K2, list(V2)) → list(K3, V3)

2、combiner还具有类似本地的reduce功能.

例如hadoop自带的wordcount的例子和找出value的最大值的程序，combiner和reduce完全一致。如下所示：  
map: (K1, V1) → list(K2, V2)  
combine: (K2, list(V2)) → list(K3, V3)  
reduce: (K3, list(V3)) → list(K4, V4)

3、如果不用combiner，那么，所有的结果都是reduce完成，效率会相对低下。使用combiner，先完成的map会在本地聚合，提升速度。

4、对于hadoop自带的wordcount的例子，value就是一个叠加的数字，所以map一结束就可以进行reduce的value叠加，而不必要等到所有的map结束再去进行reduce的value叠加。

combiner使用的合适，可以在满足业务的情况下提升job的速度，如果不合适，则将导致输出的结果不正确。