1. hadoop1.x中namenode保存的元数据有什么？

文件名、权限、大小、被分成哪些块，注意磁盘上保存的并没有块分布在哪些node上，而是在启动集群的时候datanode上报的，在内存中。

1. hadoop1.x中secondarynamenode是如何合并fsimage和edit log的？

①新建一个edits文件用来记录此时刻之后的用户所有操作日志记录

②把此时刻之前的edits文件和对用的fsimage文件拷贝到snn上

③snn进行合并

④把心的fsimage文件传给nn

⑤替换旧的fsimage

注意：内存中的元素局与fsimage中的同步问题，不要认为内存中的慢，因为在操作的时候记录edits的时候内存中就已经改变了，而fsimage就是一个持久化的机制而已，用于再次启动的时候加载。

合并时机：

①fs.checkpoint.period默认3600秒

②fs.checkpoint.size，默认edits文件64M

1. HDFS读写流程？

读流程：

①调用DistributedFileSystemAPI的open方法

②从NN中获取元数据——数据块的位置信息

③调用FSDataInputStreamAPI的read方法

④具体read读取数据，根据第二步返回的block信息，比如block1有三个副本，会找一个空闲的副本所在datanode所在机器读取，其他的Block可以同时进行。

⑤调用FSDataInputStreamAPI的close方法

写流程：

①调用DsitributedFileSystemAPI的create方法

②访问NN，告诉NN我要上传文件，并把文件名、权限、大小、用户给NN，此时，NN会根据文件的大小分blocks，以及blocks的第一个副本需要放在哪些机器上

③调用FSDataOutputStreamAPI的write方法

④根据第二步NN反馈的信息把block的第一个副本写入指定的DN上，写完第一个副本就结束

⑤调用FSDataOutputStreamAPI的close方法

⑥反馈给NN说success

注意：第四步client往DN上写完第一个副本就结束了，那么副本机制怎么实现？这里的第二个第三个副本的复制以及位置选择都由第一个副本所在的DN决定并完成！

1. 说一说你对MR的shuffle过程的理解？

从整理上来说shuffle过程主要分为三个机制的合作：分区、排序、合并、合并

①partition分区

把map的输出进行分区，默认分区是根据key的hash模 partition，可以自定义，目的是负载均衡

②sort排序

当缓冲区快满的时候对80M（总大小100M）的数据进行sort，默认的排序规则是字典排序，一定要注意字典排序中11在9个前面

③combiner合并（可有可无）

内存中有个环形缓冲区默认100M，在内存中进行partition和sort操作，当达到80%的时候将会发生溢写到磁盘，如果大量的小文件肯定不适合网络传输，所以在溢写之前进行combiner操作，将相同key的value加起来，减少溢写数据量。

④merge合并

小文件合并为大文件

用一个例子说一下比较明了

map读入的记录如下（两个reduce）：c: 2,a: 1,b: 1, b: 1,d: 1

partition操作：a和c分为一个区，b和d分为一个区

sort操作：环形缓冲区快满的时候，a和c位置调换，总的顺序是[a:1 c:1][b:1 b:1 d:1]

merge操作：这里合并多次溢写的临时文件{a,[1,1,1,1,1...],c,[1,1,1,1]}...

如果存在combiner的话：

sort之后溢写之前会进行combiner：[a:1 c:1][b:1 b:1 d:1]变成[a:1 c:1][b:2 d:1]

注意：从map端的merge可以看出为什么reduce接收到的value是一个迭代器了！

注意：combiner和reduce端的merge的区别。conbiner是我们可以操控的，合不合并，怎么合并，产生什么的keyvalue我们说了算，但是reduce端的merge和sort是不可控的，是同一个merge和sort，并且merge只是合并相同key，而不是可控的。’

那么map端的merge和reduce端的merge有什么不同呢？也是有区别的，因为map端的输出和copy是同步进行的，所以map端的merge并不是merge所有map输出结果，但是reduce端的merge必须是所有的map输出copy过来的数据的总merge，不然数据肯定会出错。因为reduce的输入是按照key+iterator的，所有不允许出现相同key的多个结果。

1. MR的split大小？

max(min.split, min(max.split, block))

换成语言就是说，最大只能是max.split，最小只能是min.split，其他时候根据block大小决定，所以一般我们说split的大小是64M，不太准确

注意：可以优化，设置最小split和最大split的大小！

1. 请解释一下为什么hadoop2.x解决了hadoop1.x的NN的单点问题和内存受限问题？

单点问题：HA

内存受限：Federation

HA机制：

①两个NN，一个主一个备，DN要想所有的NN汇报块信息、心跳

②备用NN通过journalNode来同步元数据，所以当主宕机时，元数据还可以获取到

③FailoverController这个是主备切换的关键，用来检查NN的心跳和切换NN，具体操作就是Failovercontroller向zk汇报NN情况，如果主宕机了的话，zk就知道了，此时会通知备NN你们来竞争吧，竞争到锁的NN就切换为active。

注意：之前一直有一个疑问就是client是如何找到正在处于active的NN的IP的？纠结过一段时间，现在可以认为是zk的存在，因为zk可以做统一命名空间嘛，这个命名空间下有多个ip，这样client每次去找zk问我们的cluster现在活着的是哪个ip，所以....

Federation机制：

解决了NN可以水平扩展的问题，但是只有大公司才会用HA+Federation，直观上就是通过多个命名空间namespace将元数据的存储和管理分布在多态机器上。

1. 简单说一说yarn，然后具体说一说任务申请执行的整体流程？