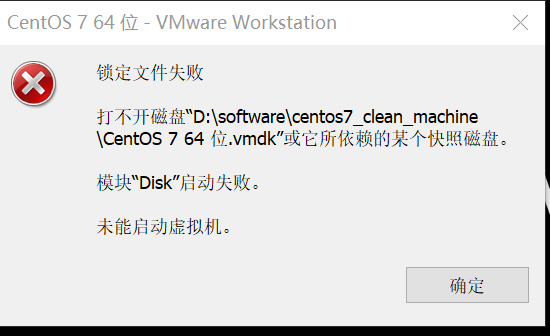
# 虚拟机

问：开机时，出现如下弹窗：



答：之前虚拟机有过非正常关机；虚拟机在开机使用是会生成lck文件，表示虚拟机正在使用；如果非正常关闭虚拟机，来不及将lck文件，编程解锁状态；下次再次开机，因为lck文件的存在，所以就开不了机；

所以，解决办法就是将lck文件删除

进入到自己的虚拟机所在目录，找到文件，删除即可

参考：https://blog.csdn.net/asdfzjs/article/details/48465487

# Mac

如果使用的是mac本，系统自带的jdk可能版本比较高，如jdk11，需要自己安装jdk8才可以

# HDFS

问：这里是直接在hdfs里面对文件追加内容，如果是客户端那边对文件追加内容，再重新上传文件，那么整个文件内容都会存储一遍吧

答：不会的；

你想如果一个文件有1000个block，我如果要追加1kb的内容的话，还需要将1000个block重新传输一遍，hadoop作为一款成熟的框架，肯定不会这么做的

client先获得这个文件最后一个块blk的信息，返回这个blk存储在哪三个dn上，然后建立管道pipeline，接下来，客户端将追加的内容沿着管道依次追加到这三个副本中

至于文件的其他block，不会受到影响

问：@老游～讲师 他是怎么知道是追加的还是修改的修改的话也只替换修改部分的block吗

答：我们说，hdfs不支持文件的随机修改，它只支持一种修改，那就是向文件末尾追加内容

问：老师。snn和nn 为什么要保存 旧的 edit和fsimage

答：应该也是出于数据安全、容错考虑；万一集群出了问题，我还能通过过去的edit、fsimage恢复部分的数据，减少损失；要知道企业中的数据，那就是它的最宝贵的资源；

问：最新的fsimage是所有的元数据,还是最新一个小时或1000命令的,它的后缀编号是代表这一段事务的数据吗@老游～讲师

答：最新的fsiamge，其中包含的除了如正在写入日志文件（edits\_inprogress\_0000000000000000332）外的的所有的元数据

fsimage\_0000000000000000331 比如最新的fsimage名字是它，那么表示它里边包含了1到331所有事务下，所产生的元数据

咱们的集群fsimage存储目录是

/kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/namenodeDatas/current

是在hdfs-site.xml中的如下属性决定的



Edits目录是在

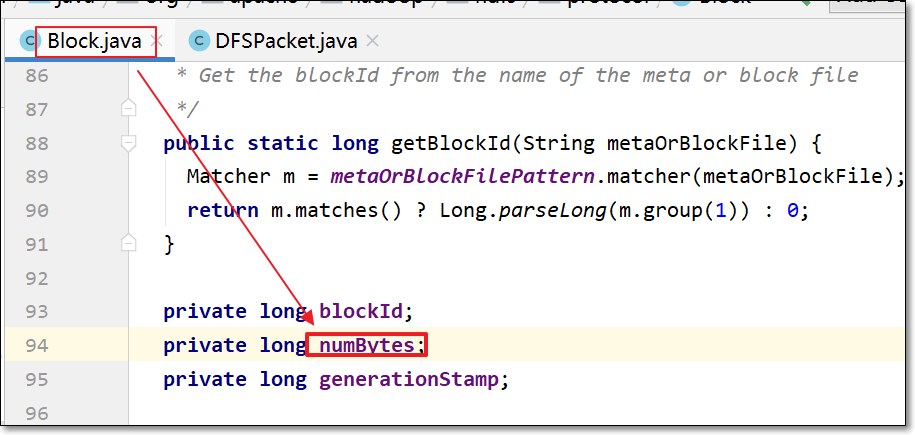
/kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/dfs/nn/edits/current

是由hdfs-site.xml中另外一个属性，如下决定的



问：，DataNode怎么区分最后一个Block块是因为网络问题还是真正写完数据了，需要向NN报备

每个block都有numBytes值



问：三个datanode的hdfs集群，挂了一个dn，会不会出现为了保证三副本，从而一个节点上出现重复的数据块？

答：不会，一个节点上出现重复数据块没有意义

不能保证3个副本，那就两个副本先凑合着呗，等到机器来了，再继续复制三个副本

问：那什么时候一个datanode会出现重复的数据块

比如一个dn1上边存储了许多block，其中一个是blk1，然后dn1宕机

随着各dn向nn发送心跳，nn发现blk1少了一个副本，它会通知dn2（比如它上边存储了blk1的一个副本），让dn2将blk1复制一份到另外一个dn上，复制完成后，blk1副本恢复成3

当dn1又重新上线，发现blk1副本为4；由nn牵头，让其中的一个blk1的副本删除，恢复3个副本

问：老师datanode做校验的时候是自己自动生成校验码与checksum对比吗

答：对

问：老师，datanode节点破坏了，nn生成时间戳的意义是什么呀，为什么不能直接生成一定要版本更新

答：dn1坏了，nn生成新的时间戳，那么当dn1恢复，重启，向nn发送心跳时，nn发现它上边的blk的时间戳是旧的，作为判断的依据，会告知dn1将此blk删除

问：老师，nn 内存什么时候，什么情况下生成第一个fsimge和第一个edit log呀。

答：在初次搭建集群，格式化的时候

问：前面您是说防止数据消失，会在磁盘里生成edit log 下次开机重启时再从edit写入内存，那fsimage是什么时候有的。

答：从初次格式化集群开始，生成空的editslog及空的fsimage；

然后snn每分钟检查一下时候需要checkpoint；如果需要，则将editslog与空的fsimage拉取到snn节点，合并生成新的fsimage，再返回给nn；

然后，等待下次checkpoint。。。

如此循环往复

问：①老师，回传的fsimage是不是包括checkpoint时间点之前的在namenode上出现过的所有数据？②checkpoint后滚动生成的新的edit存着下一次checkpoint前的增量数据？

答：①完全正确；

②完全正确

问：老师，如果是这样的话为什么要保存最新的两个fsimage 呀，最新的那个fsimage不就存着nn以往的所有元数据信息了吗

答：保存最新的fsimage的作用，就是如果nn挂了，重启后，可以直接将最新的fsimage中的元数据读入到内存，然后，再将最新的editslog中的操作回放一遍，从而恢复nn的元数据

至于为什么保留两个fsimage，是出于安全的考虑，如果最新的fsimage不能用，我还能用第二新的，恢复绝大多数的元数据

问: 老师datanode之间可以通信吗

答：肯定可以；比如向HDFS上传文件时，要建立pipeline时，三个datanode间就要通信；

问：有几个datanode怎么知道maptask在哪个节点上运行呢

答：可以通过查看yarn的weib ui查看

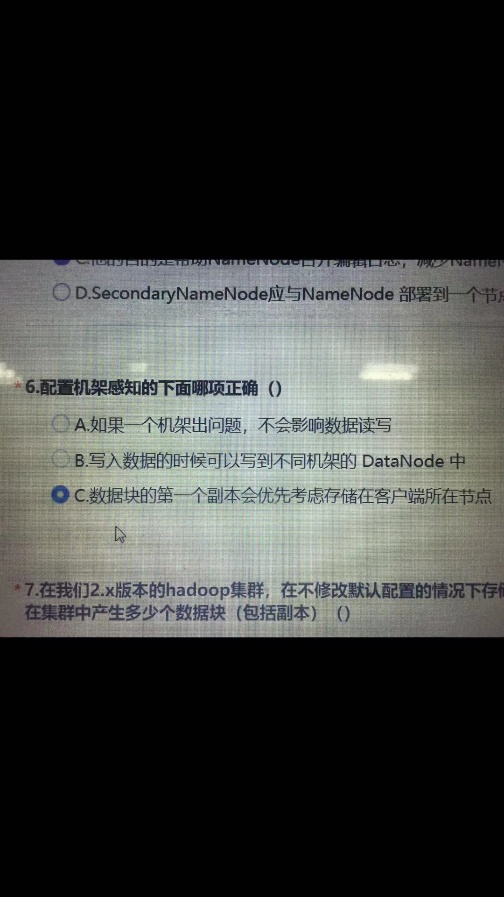
访问<http://node01:8088/>

打开application id对应的连接，然后看出来每个task运行在哪个节点上

问：一个datanode是一个节点吗老师

答：是的；节点是指定服务器；那么一个datanode肯定要运行在一个服务器上；但是，一个服务器不一定是datanode，有可能是namenode或其他角色

问：

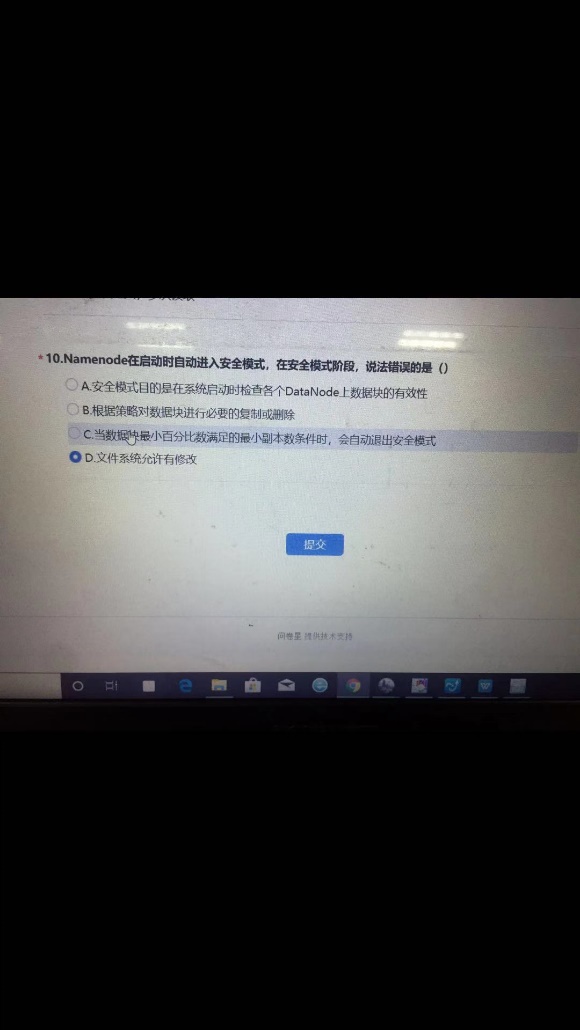


答：其实，a、b、c都对

问：什么是机架？

答：在机房中，多台服务器会放到一个柜子里，可以称为机架或机柜

机架概念了解即可

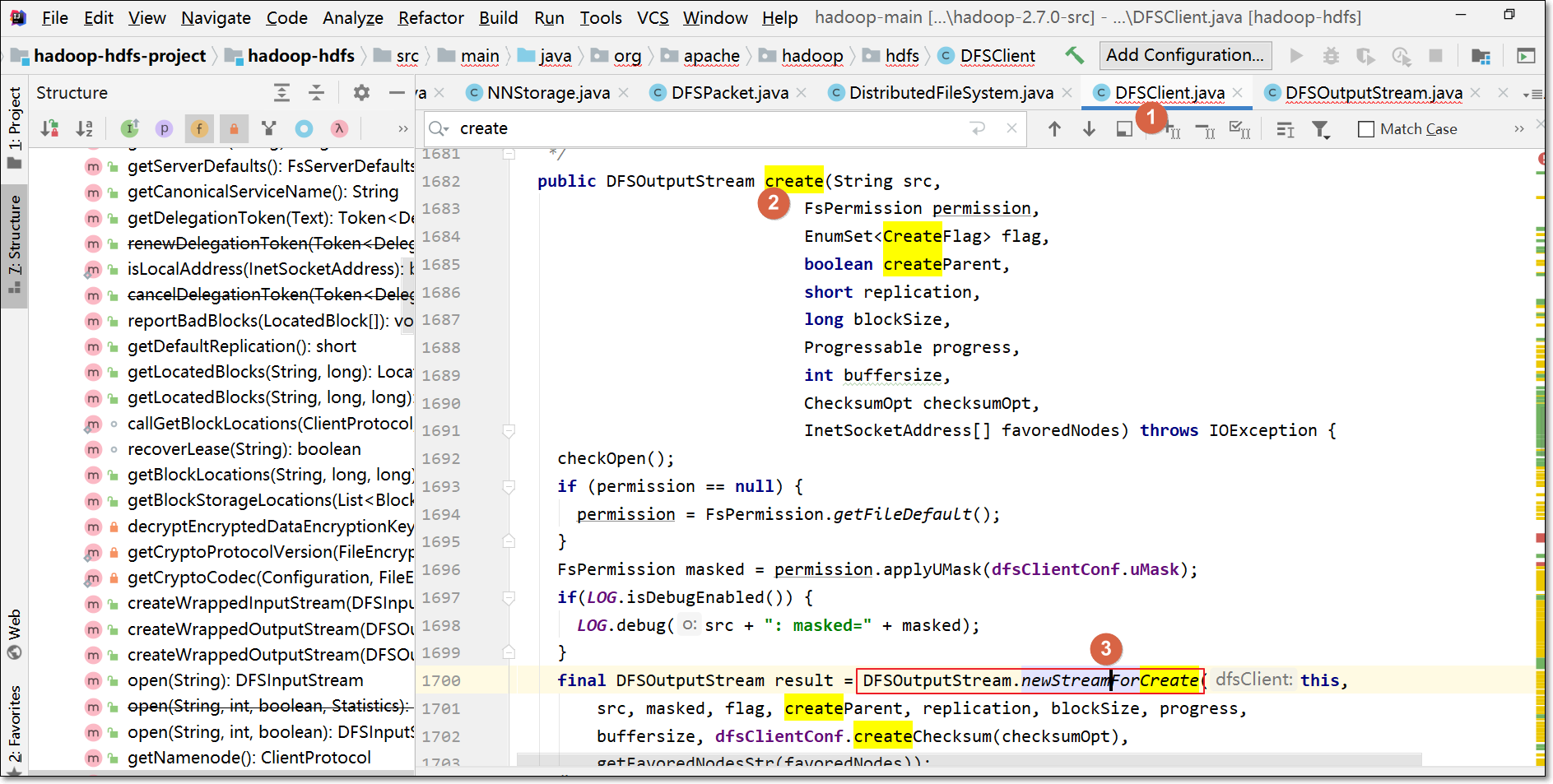
问：

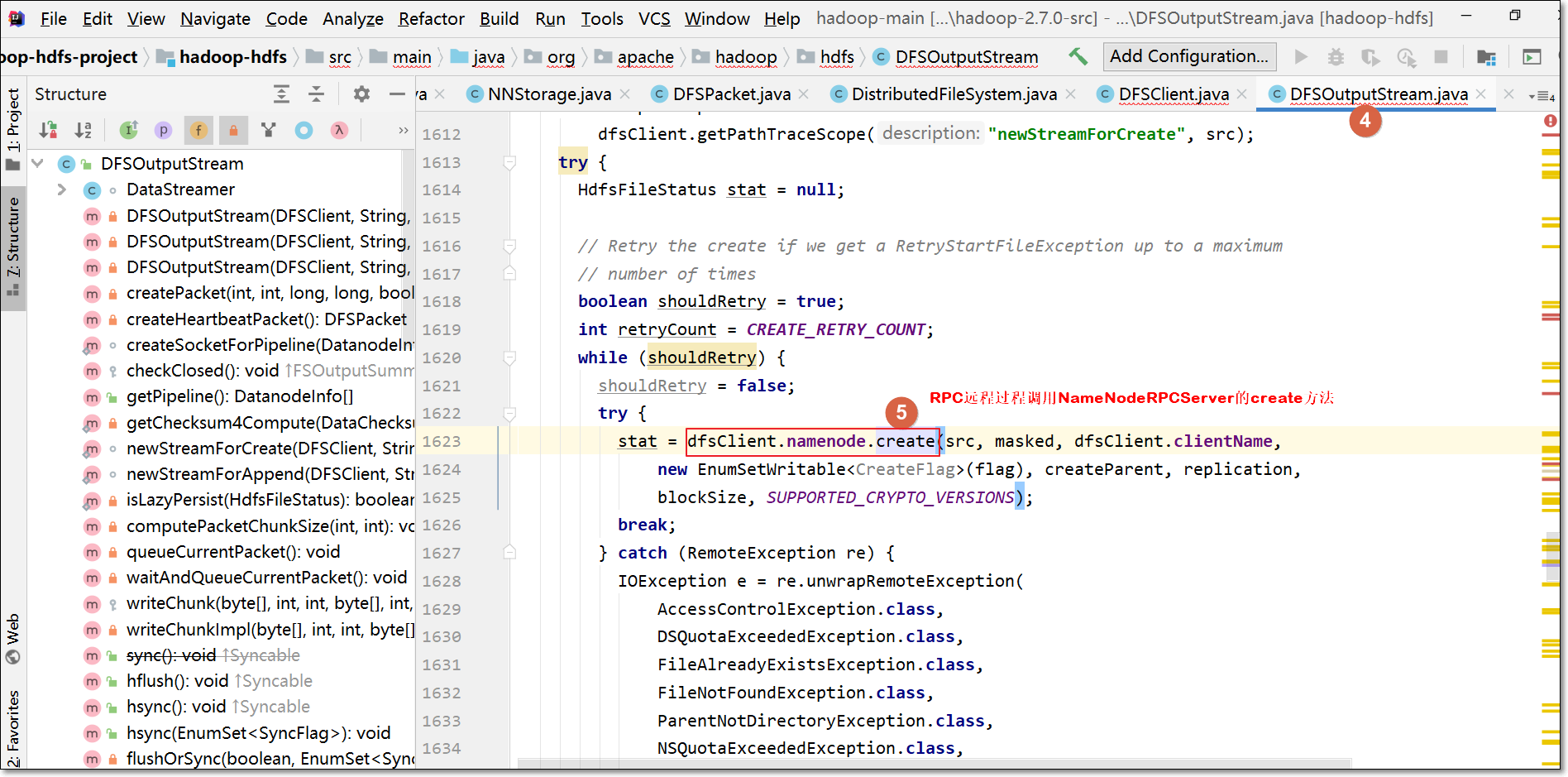
c不是很理解

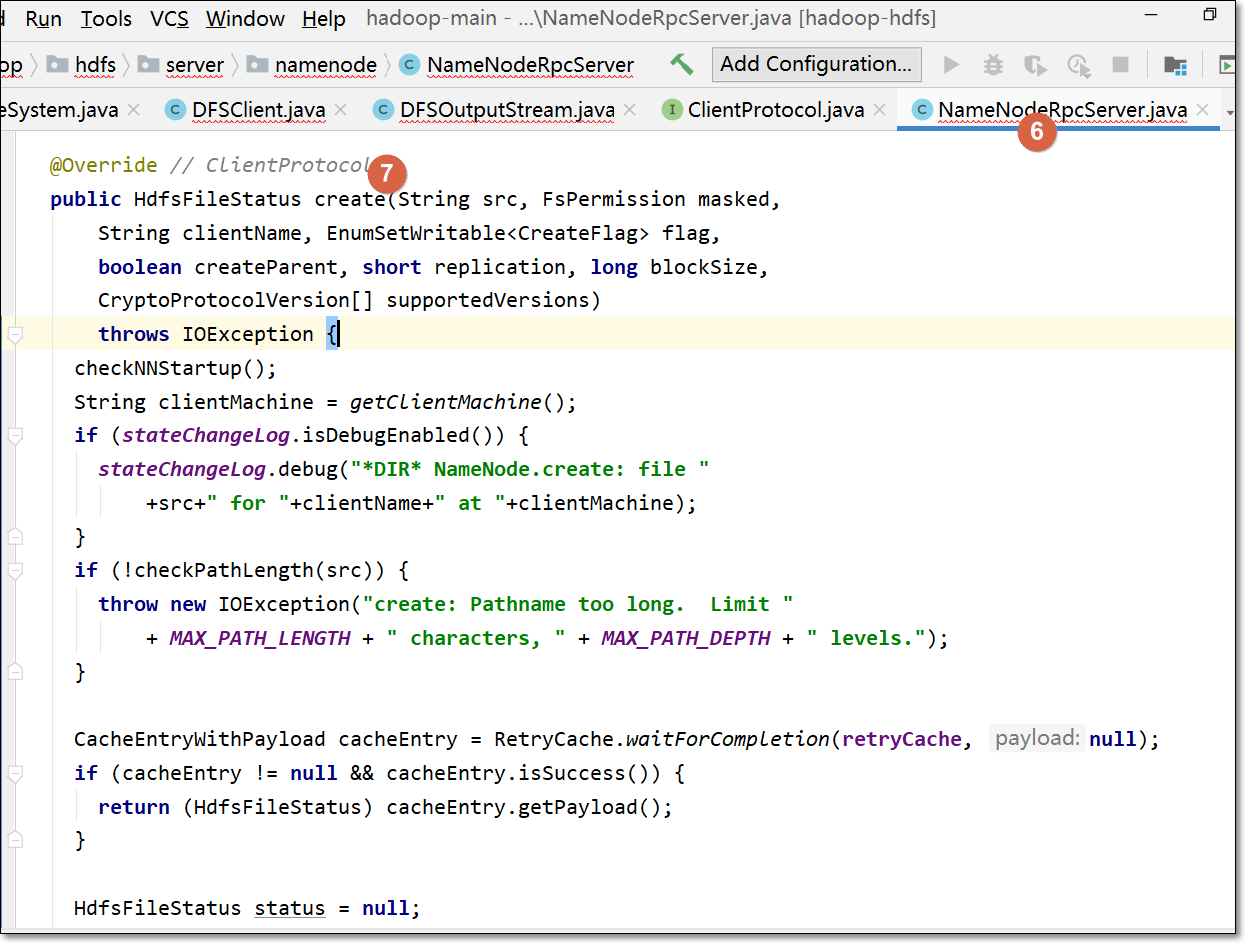
答：意思是当集群中有99.9%的block满足最小副本数时，退出安全模式

# HDFS源码

问：老师，那那个rpc远程调用namenode的create在原码里是写在distributefilesystem的create里吗

答：



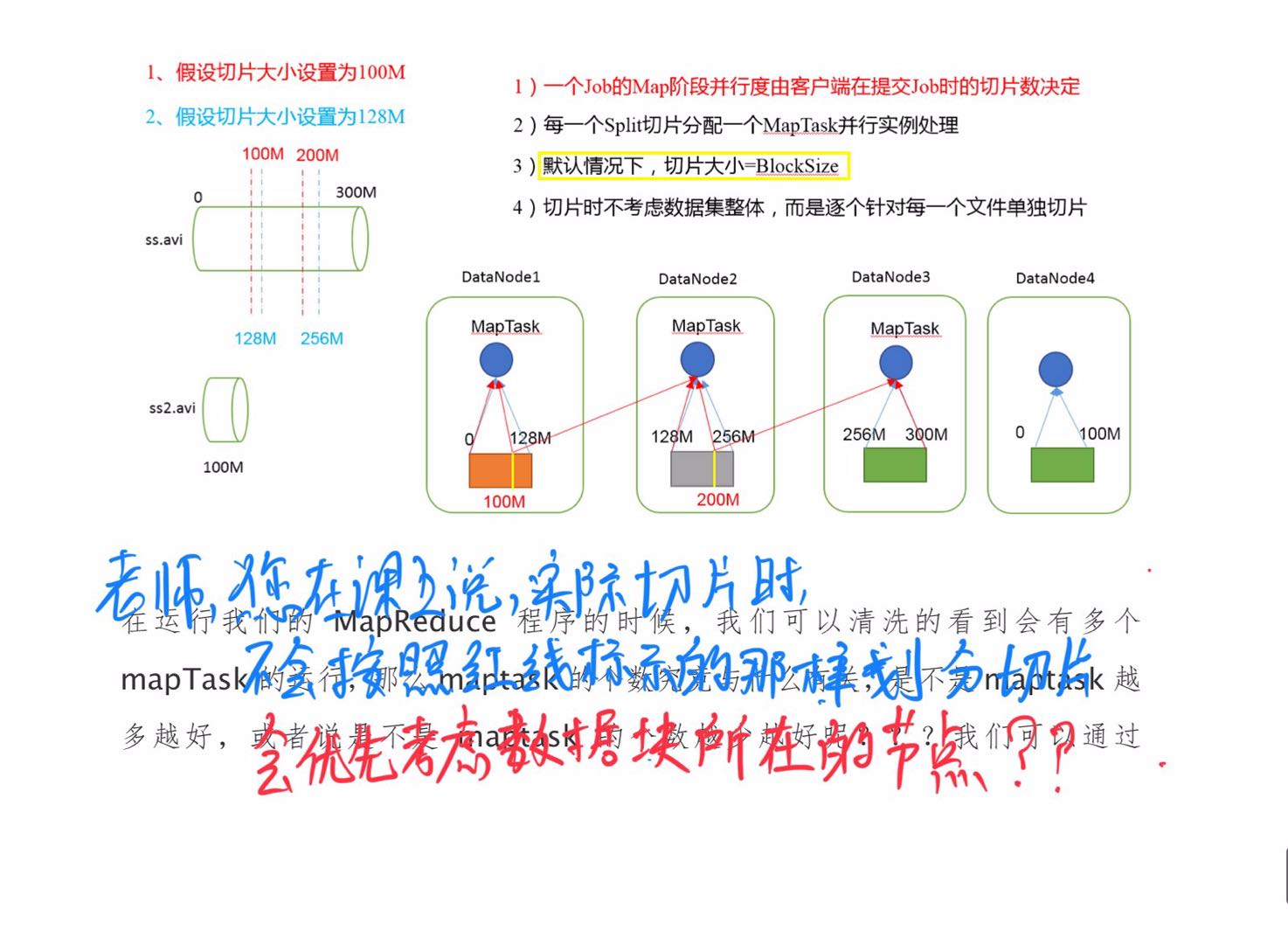


问：老师，这个的原码可以去哪里看呀

答：去把源码包下载下来，解压，然后以maven工程形式，导入idea就行；

或者，在自己的hadoop工程里，关联源码也行

# MapReduce

问：

答：是这样子；

首先考虑如果切片大小默认是128M的情况；

一个block对应一个split，是一一对应的；那么在调度map任务时，会优先考虑split数据在哪个节点，就把对应的map task调度在此节点上；那么，map task运行时，只需要从它运行的节点的本地磁盘读取分片数据即可；思想：移动计算，不移动数据，效率更高

如果split的大小比128M小，比如上图的100M时，那么第一个分片对应一个map task；第二个分片，由从第一个block的100M以后的部分，及第二个block的头72M组成；以此类推；

一第二个分片为例，分片的前部分数据是block1的后28M，分片的后部分数据是block2的头72M数据；72>28，所以，针对这个分片调度map task时，会优先考虑将任务调度在block2所在的节点上；还有很多其他的细节，讲yarn的时候再展开

一个帖子也可以参考一下：<https://www.cnblogs.com/dflmg/p/9552307.html>

问：词频统计时，进行切片时，会不会将一行中中间切开？会不会将一个单词从中间切开？

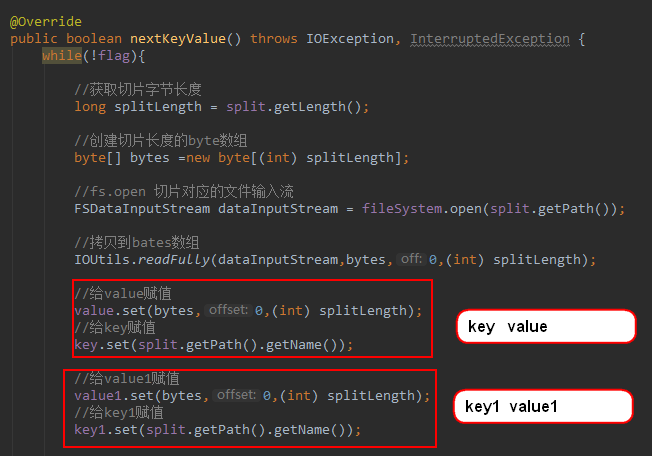
答：

问：老师。请问下，设置split大小后，难保切片不跨block，如您所说的mr优先考虑block所在节点，例如课上的例子 28M + 72M的切片，怎么划分到同一个block@老游～讲师 谢谢

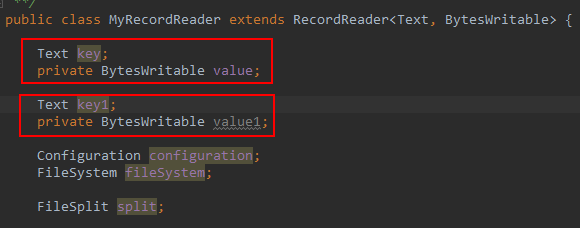
答：参考上边的回答

问：@老游～讲师 老师我想问下，自定义RecordReader ，我设置了多个和key/value 变量类型相同的变量；是怎么确认我最终输出 key value是那个变量的，怎么知道我输出的key 是 key还是key1、value 是 value 还是value1 ？谢谢

图一



图二



答：问一下，为啥定义了两套key, value啊？

Text key;

private BytesWritable value;

Text key1;

private BytesWritable value1;

如果仅仅是为了获得RecordReader读取当前数据，产生的key、value的话，就一套key, value就行了

但是如果就想这么做的话，如何确定RecordReader返回的key，value？比如你要确定返回的分别是key1、value1

可以同过getCurrentKey()方法设置，返回的键

public Text getCurrentKey() {

return key1;

}

通过getCurrentValue()方法设置，返回的值

public BytesWritable getCurrentValue() {

return value1;

}

问：1.CombineTextInputFormat  设置 setMaxInputSplitSize 为4M   测试三个文件  7.5M ，7.5M，7Kb   看运行日志发现map task个数 为 2， 不应该是三个嘛

答：过程一：

生成虚拟文件：3.75M 3.75M 3.75M 3.75M 7KB

切片：有可能是7kb+3.75M<4M，那么7kb+3.75M+3.75M形成一个分片，剩下的两个3.75M形成一个分片；

这个呢，了解生成虚拟文件、及切片过程原理即可，不用过去关注细节

如果你把分片大小设置的更大，比如128M的话，就只有一个map task了

问：

还有 我发现inputFormat 都有setMaxInputSplitSize  setMinInputSplitSize两个属性设置（包括CombineTextInputFormat ），其他inputformat子类的分片规则都是怎么样的呢

答：我看了下，发现只有FileInputFormat的中有setMaxInputSplitSize、setMinInputSplitSize方法；就是说它的子类的这两个方法都是从父类继承过来的

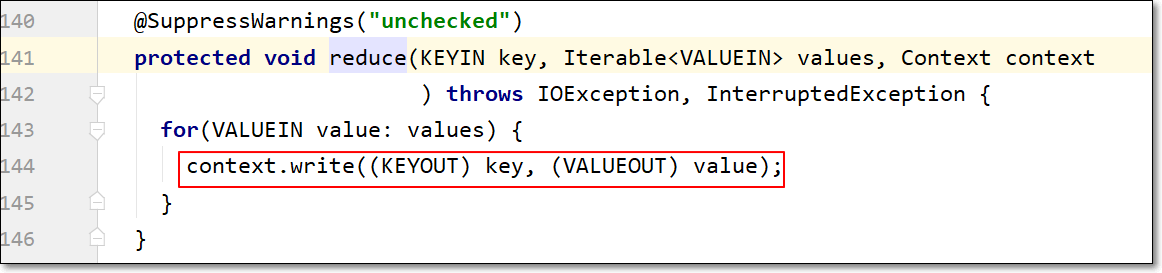
setMaxInputSplitSize   设置最大分片大小

setMinInputSplitSize 设置最小分片大小

问：2.自定义InputFormat 中没有写 自定义reduce类， 为什么，是使用了默认的reduce类嘛

答：如果在main中没有指定job.setNumReduceTasks(0)的话，并且也没有指定自己的reduce逻辑，job.setReducerClass；那么，mr框架默认使用一个reduce task，切使用的是框架的父类Reducer类

这是它的reduce的逻辑，就是将map阶段传输过来的k、v对，原样输出



问：逻辑分片的默认大小和块一样大（128m）。如果这个块存储实际只有100m。那么这个逻辑分片最后是会处理两个block补齐128，还是一个实际只有100m的block数据

答：这个问题得看你所使用的是那种输入格式类，比如使用的是TextFileInputFormat的话，这个块只有100m，说明是它所属文件的最后一个块，啊么，这是此文件的最后一个split，此分片就100m大小，不会与其他文件的内容，平成一个128m的分片；但是如果使用的是CombineFileInputFormat的话，那么，此100m的数据，会与其他文件的数据内容拼成成一个分片，在128M与256M之间，此过程参考CombineTextInputFormat的过程；简言之，前者一个文件的数据不会与其他文件的数据拼成一个分片；后者会

问：在哪里设置map task 和reduce task了呀

答：这个好像没听课吧；课堂上讲了，map task个数，由输入分片的个数决定；reduce task个数自己在程序中指定job.setNumReduceTasks(2)，比如设置成reduce task个数是2

问：分区，分组有点晕。分区是为了把同一分区数据发给同一个reduce。分组，每个分组调用一次reduce。好像没区别

答：以词频统计为例，如果hadoop、hive为键的键值对去往同一个reduce task的话，默认键为hadoop、hive的kv对，分别处于不同的分组中；比如hadoop, List(1,1,1,1)调用一次reduce方法；hive的也是如此；但是，如果想要将hive、hadoop的作为同一组的话，得自定义分组逻辑，实现；

有时并不一定得自定义分组

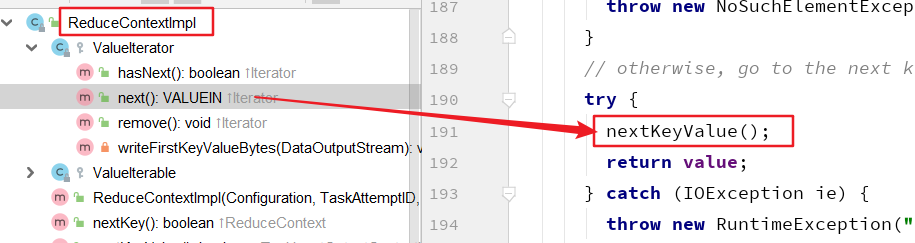
就是如果编程有多重方式能实现，肯定是选择容易的那种

问：没有java背景，有点难理解那段javabean的程序，我的疑问是1. flowbean里的write，readfield和tostring并没有被reducer和mapper调用可依然要override，所以这两个就是javabean里所谓的序列化反序列化接口吗？2. 任何自定义类型的writable class都需要有这三个且tostring那段也需要保持类似的字符格式？

答：同学及助教的回答如下

梁江涛：javabean在网络上传输需要实现序列化，write和readFiled在网络传输中需要序列化和反序列，这样传输后你才能看到接收的还是一个javabean，并没有说没用到的情况，tostring一般重写是为了输出的时候的时候好看，否则打印出来的是内存地址值

助教：因为不重写toString(),直接显示的hashCode对程序员来说,没有什么实际意义,反而在许多场合中,需要打印和比较出对象的内容,所以就重写toString()了

问：  答：mapreduce框架会复用key-value对象对，ReduceContextImpl中有成员key，value；在reduce（）时，会复用这两个对象

mapreduce框架会复用key-value对象

The framework calls this method for each <key, (list of values)> pair in the grouped inputs. Output values must be of the same type as input values. Input keys must not be altered. The framework will reuse the key and value objects that are passed into the reduce, therefore the application should clone the objects they want to keep a copy of.

参考：<https://blog.csdn.net/fxdaniel/article/details/50255197>

问：map task、reduce task运行在什么节点上？dn还是nn？还是可以配置运行在哪个节点上？

答：map task、reduce task是计算任务，所以是运行在yarn的从节点nodemanager上

每个任务具体运行在哪个nodemanager上，这是yarn框架在调度job运行时，自动调度的，不需要人工干预

讲yarn时，会展开细节

问：游老师，问您个问题，课上说mr阶段切片是由客户端完成的，并将split信息传给yarn，然后maptask再通过inputformat的RecordReader将信息转换为kv对。切片方法getSplits()和RecordReader处理都是在FileInputFormat中，也就是说客户端先调用FileInputFormat切片然后maptask再调用FileInputFormat完成kv转换吗

答：对的

客户端在提交job到集群运行前，先调用InputFormat的getSplits方法，计算输入共有多少分片

当启动map task后，map task通过InputFormat的RecordReader读取分片数据，每次读取生成kv对，作为map方法的输入

问：还有一个问题，ReduceTask接收完各个MapTask同一个分区的信息之后，会进行一次merge归并排序，然后分组也会进行一次归并排序，这个地方总共进行一次归并还是两次？

答：merge、排序的次数不确定；

merge的原因是为了减少reduce端的数据量，因为reduce task持续不断的从map task中拉取数据，有新的map task执行完成，也得将其上的数据拉取过来；那么在拉取到reduce端后，为了减少内存使用的数据量或者磁盘上的文件数，需要meger，同事也得sort排序

知道所有的map task数据全部拉取过来后，那么内存、磁盘中都会有数据；

进行最终的分组

每组调用一次reduce()

问：@老游～讲师 老师 想麻烦问下mapreduce第二课里的第五个例子，视频时间在1h50min之后一点。

1. 你的做法是把135、136开头的手机号用分区函数提前划分分区。

我想问下，如果不重新定义分区函数，在reduce里进行逻辑计算，有什么区别？

②因为shuffle阶段完成，手机号135的和135的挨着，136的和136的挨着；所以我reduce时，比如计算135开头的流量和，每读入一个样本，都检测一下手机号key是不是还是135开头，如果是就累加到这个组；如果是136了，就切到另一个累加计数组了。

如果把逻辑写到reduce里，相比改partition函数，计算时，是不是按手机号分片，会多用分片，计算慢。

答：①如果不进行自定义分区的话，在reduce端进行处理的话，能实现，但是不好；一个是性能会低，另外需要做一些额外其他的自定义的逻辑

首先咱们的需求是135、136.。。存储在不同的文件中

那么问题来了，如果不自定义分区的话，并且最终想135的保存到一个文件中，那么，需要指定reduce task个数只能为1，因为如果是多个的话，而且使用的是默认的HashPartitioner的话，那么，135xx1与135xx2可能会去往不同的reduce task中

所以，设置reduce task个数为1

好，那么如果数据量比较大的话，那么一个reduce task处理所有的结果数据，肯定效率低

另外，虽然135xx1、135xx2都进入了一个reduce task了

但是，我们得想办法让135开头的，是一个分组

所以得自定义分组的逻辑，达到这个效果；135开头的，作为同一个组数据，调用一次reduce方法

还有另外一个问题，我们让不同开头的手机号写入一个单独的文件，而每个reduce task默认对应一个输出文件

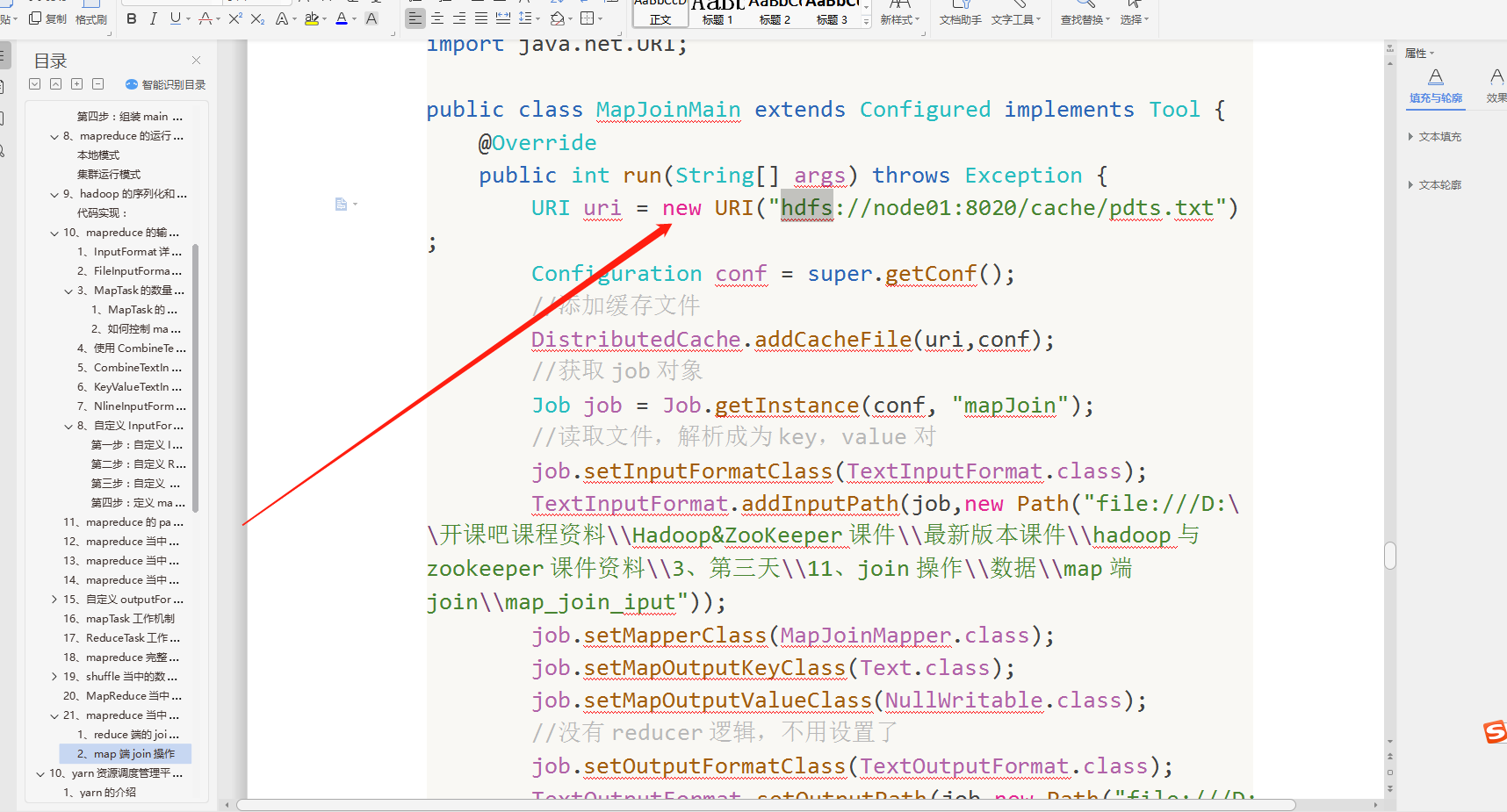
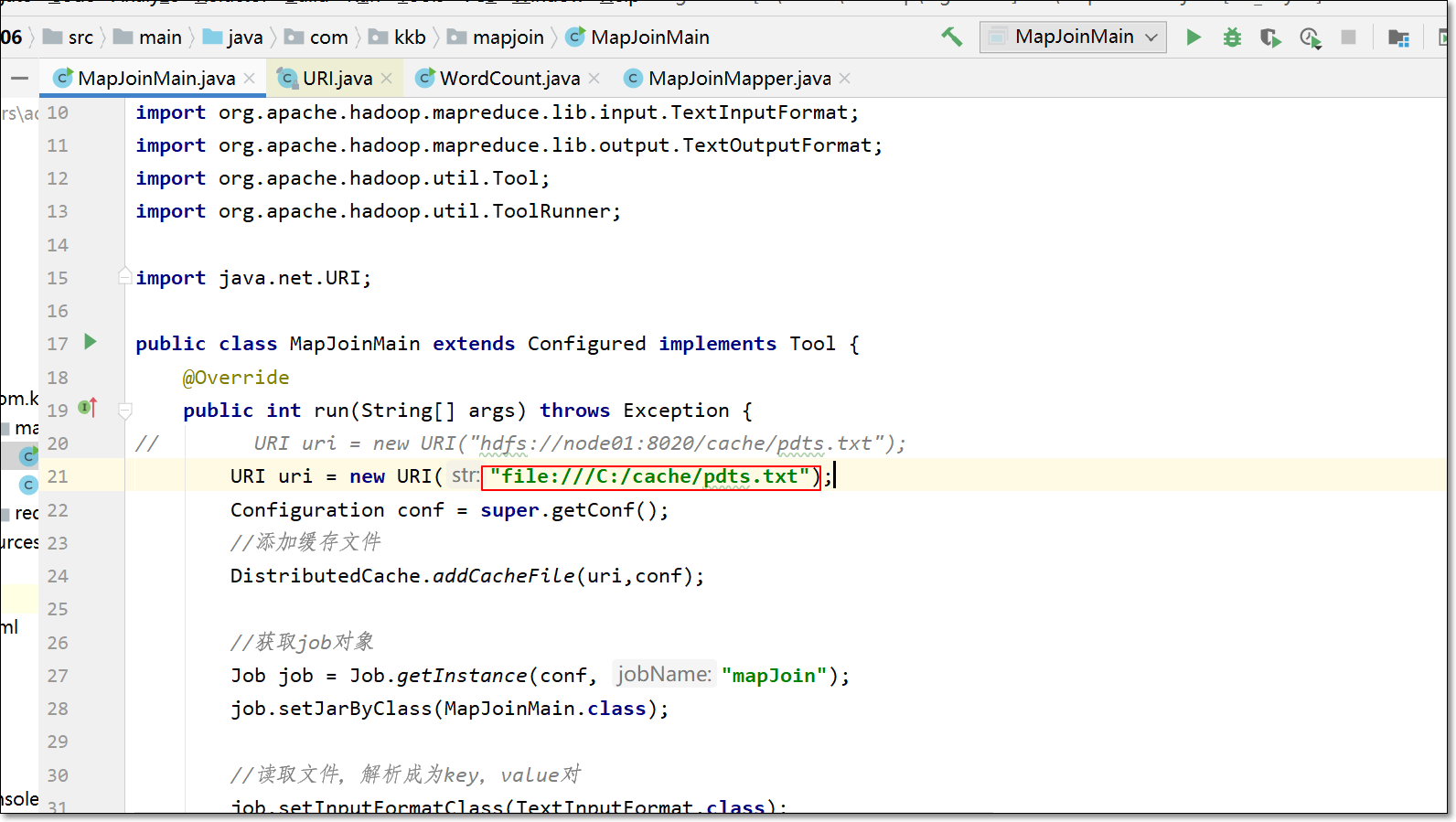
所以，为了达到这个效果，你还得自定义OutputFormat，让135开头的保存到一个文件，136的一个文件…

问：①老师 10个0.1k 的文件 通过combinetextinputformat 统计 ，maxsplit 为4m ，应该有几个maptask？ ②如果第一个虚拟存储块不足4，会和下一个合并，预期应该是5个maptask。可是得到的是1个maptask。确认一下，第一个和第二个合并，如果不足4m，会和第三，第四个合并吗？@老游～讲师

答：①肯定是最起码得凑够4m作为一个分片；但是，如果只有10个0.1k的文件的话，那么只能这十个组成一个分片，对应一个map task

1. 在①的假设上，不成立

问：本地运行这个map端join的话，缓存文件的uri怎么写

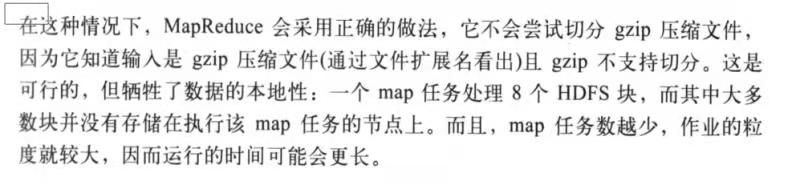
答：

如果<file:///c:\\cache\\pdts.txt>会报错

Exception in thread "main" java.net.URISyntaxException: Illegal character in path at index 10: <file:///C:\cache\pdts.txt>

问：老师 有个问题 请问不可切分的文件传上hdfs之后 mapreduce作业是怎么处理的

为什么不支持切分的文件可以分成block上传到不同的datanode但是却只能用一个map任务处理啊？既然已经把block传到不同的节点为什么不能多开几个map任务？

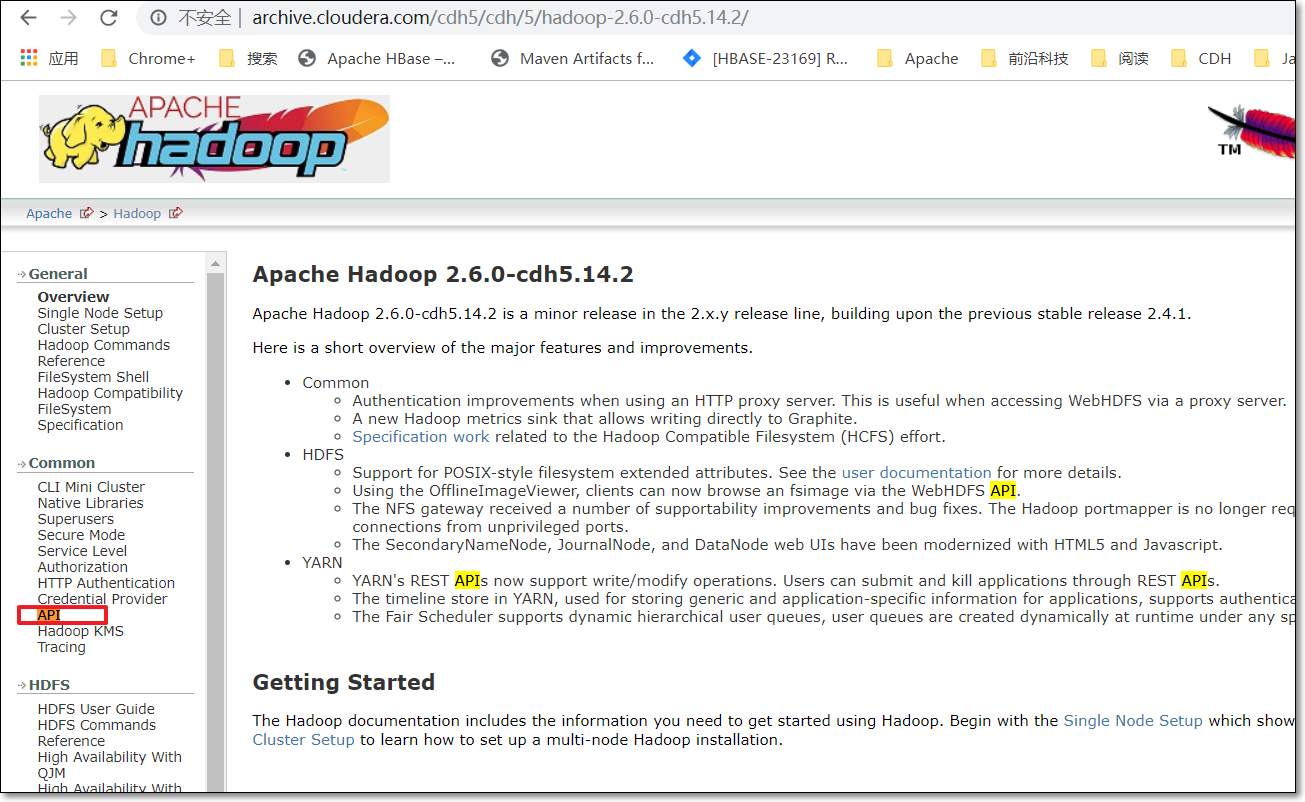


答：不支持切分的压缩算法，压缩出来的文件，存储在HDFS上也是分块存储；但是，如果以此压缩文件作为mr的输入的话，这个文件只能作为一个分片，由一个map task处理；

因为，不支持切分的压缩算法，它无法搜索数据流的任意位置，并往下读取数据；所以只能从文件头开始读取，直至末尾

问：mr的api文档

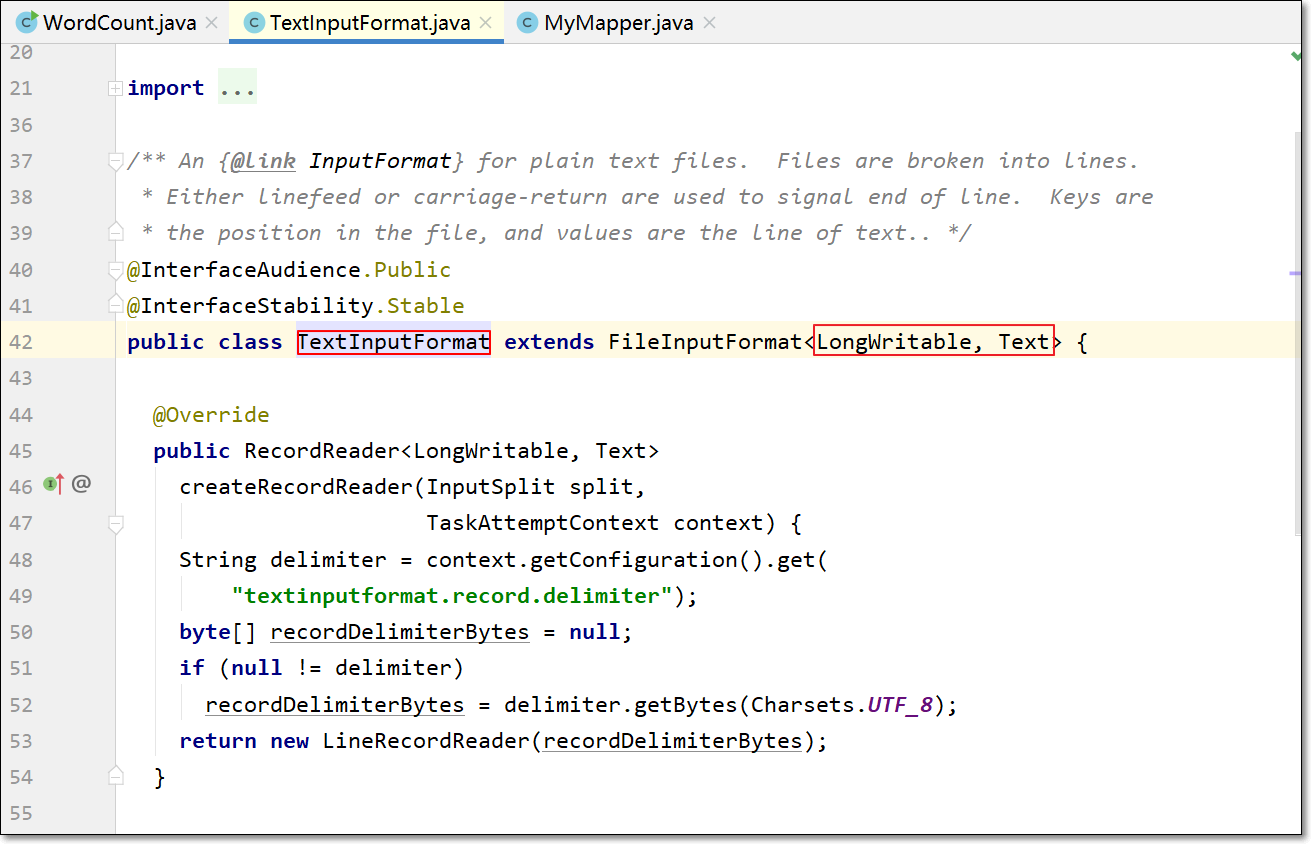
答：<http://archive.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-cdh5.14.2/>



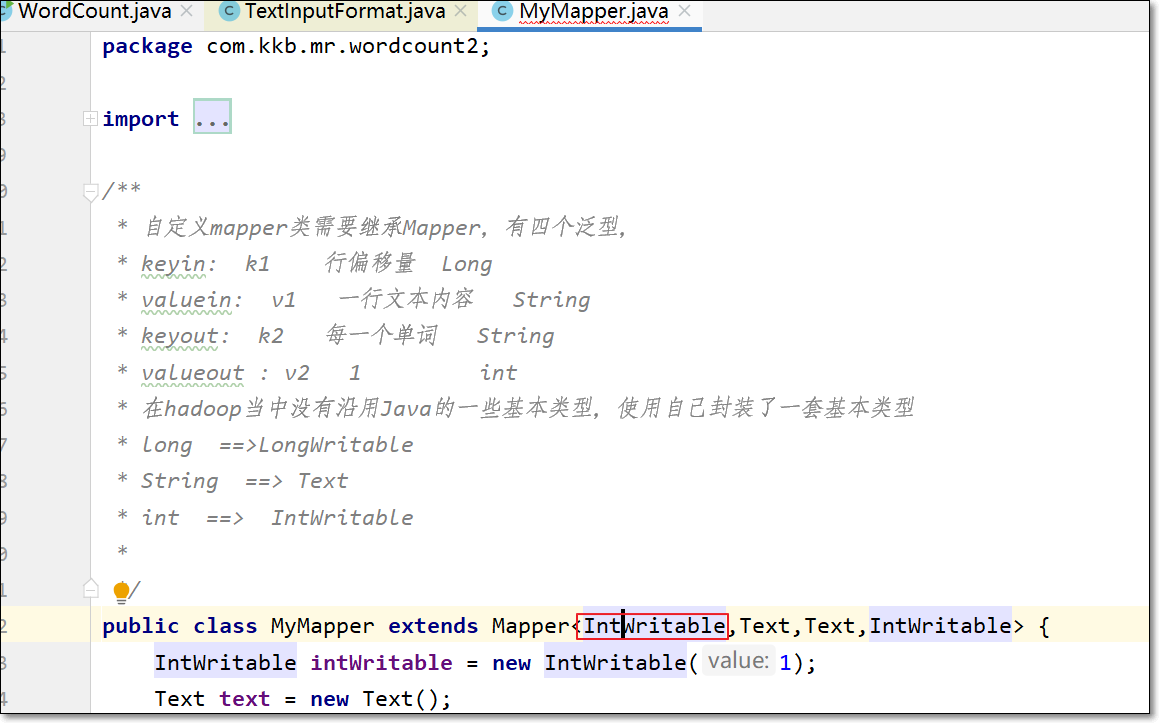
另外：java 8的在线文档 <https://docs.oracle.com/javase/8/docs/api/index.html>

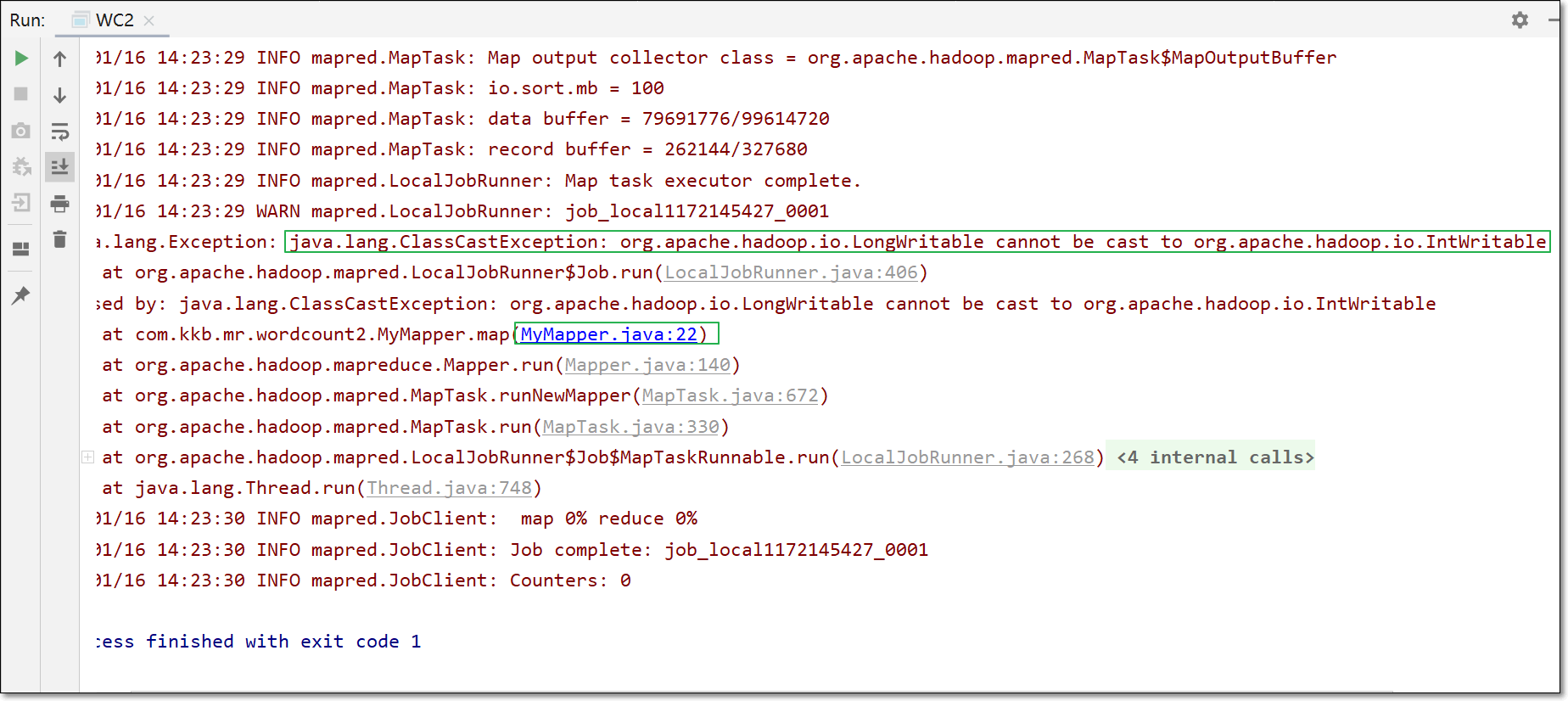
问：map文件读取，读取行，定义intwritable 好像就计算不出来为啥（没报错，我排查了好久，改为long就好了）@老游～讲师

答：因为我们使用的输入格式是TextInputFormat，而它的父类FileInputFormat<LongWritable, Text>，输入的键是LongWritable类型



那么，如果自定义的Mapper类，输入的键设置成IntWritable的话，会类型转换异常





# MAVEN

# 关于基础

问： 1是关于学习的，我现在工作用的Python，大学学的java已经忘得一干二净了，所以老师你昨天讲的课，虽然我都能看懂每一步的原理，但是让我写估计写不出来。我平时用spark也是使用python调用接口和jar包运行，所以我想问下后面如果需要开发相关程序是否一定要重新学习java。还是学习比如类似java的scala。答：肯定要学习的；大数据好多框架是用java开发的，许多是用java语言做开发；尤其如果想向高层架构发展，需要阅读源码，如果java不懂，根本无从下手，影响自己的职业晋升

问：2 关于hive的问题，我用sqoop导出hive数据表（报表层）到mysql，这个增量导出如何实现，这个只能在hive表添加一个主键列按照主键唯一导入，因为我的聚合好的数据是没有主键ID列的。还请老师指导下

答：增量导出，参考帖子：<https://blog.csdn.net/wiborgite/article/details/80958201>