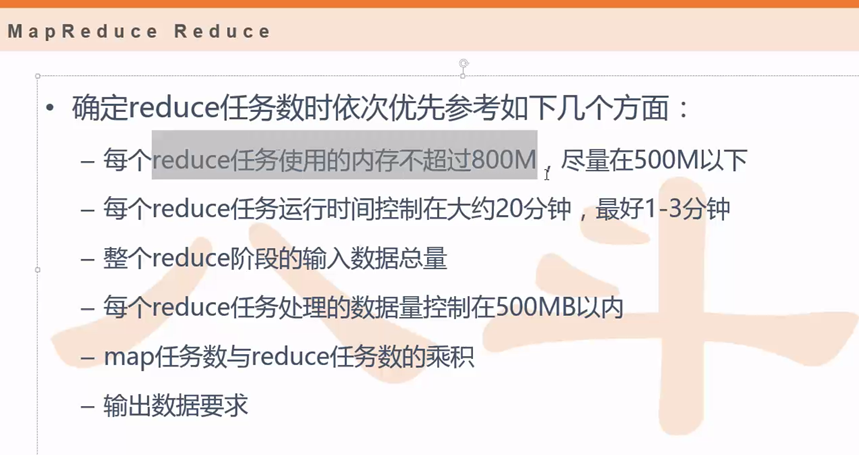
MapReduce进阶2

制作人：八斗大数据黄海峰

1. Reduce的注意事项（如下图1所示）

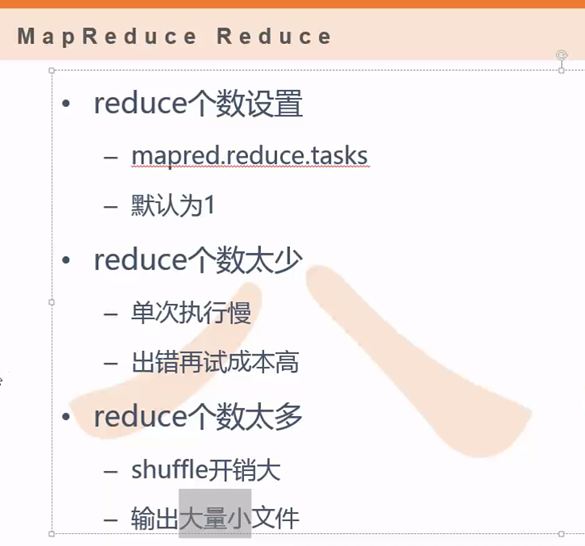
图1：



这个看一下就可以了，不作为重点

1. 那么如果说我们要对这个reduce的个数进行控制，可以通过一个参数去改（如下图2所示）

图2：



好了，MapReduce基础的部分就先到这，接下来我们来学习Streaming了

图三



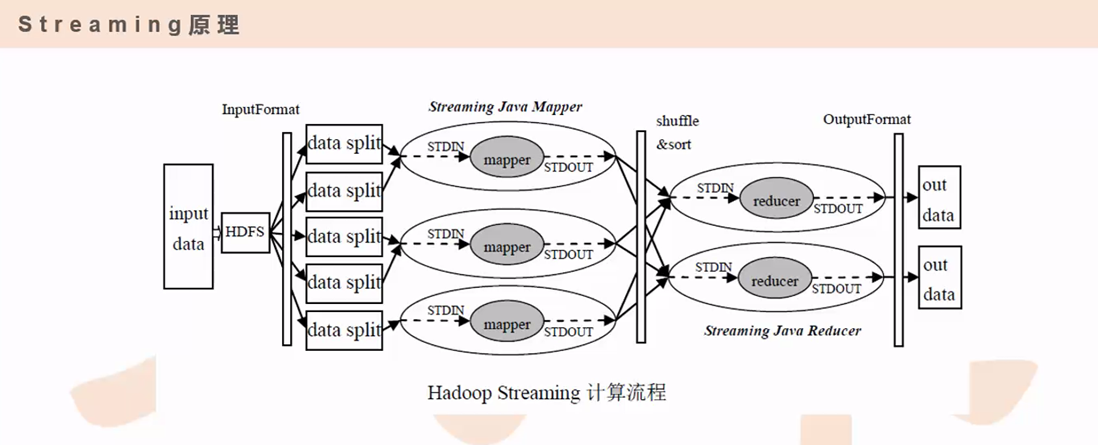
3.大家都知道，MapReduce和HDFS底层都采用java去实现，默认提供java编程接口

而Streaming框架是允许任何编程语言去开发大数据的（如图三所示）

如果用Streaming开发一套代码的时候在hadoop1.0和hadoop2.0都能跑（等于跨平台）

那么Streaming是怎么实现的呢？（如下图四所示）

图四：



之前我们学过Mapreduce的实现原理，图四的图大概看过，但是不一样，不一样的地方是图四里面的map和reduce都套了一个罩子，被一个椭圆形给罩起来了，这map的输入和输出一定是标准输入和标准输出，而reduce也是一样标准输入标准输出，但是这罩子是对接之前最原生态的对接方式，所以你会认为外面那个罩子是java版的，只不过是在这罩子里面开发了一个区域，这个区域我可以通过任何一个语言来开发，那你一旦你规定好了数据的一个标准输入和标准输出，是不是相当于它可以对语言不做一个任何强依赖了。跟上节课约定一样，整个图里面灰色的部分都是需要我们经过开发代码需要重点关注的地方。

在这里你会发现Streaming原理很简单。

1. 通过Streaming开发的代码效率要比java高很多，一方面这代码可以少写很多，不需要依赖于很多的jar包，另外的是你需要通过像脚本语言python等去开发的话是不需要编译的，现学现用，调试非常方便

另外调试也是非常重要的，相当于是我们在开发的时候，如果要是每次开发，你把每一次开发的任务你都去提交到你的集群上，但是不能够保证你的任务是否跑的通，因为启动一次MapReduce又加上排队的时间又加上启动的时间，其实整个时间成本调试起来是很慢的，所以建议在去写这个MapReduce的时候，特别是那种不是特别熟练的时候，一定要写完一个小的模块，你在本地就要自己测试一下，看看是否跑的动（如下图5所示）

如果本地测试没有问题，再去集群上面跑

怎么测试呢？就是按照下面图5里面写的红色的命令

测试命令：cat input | mapper | sort |reducer >output

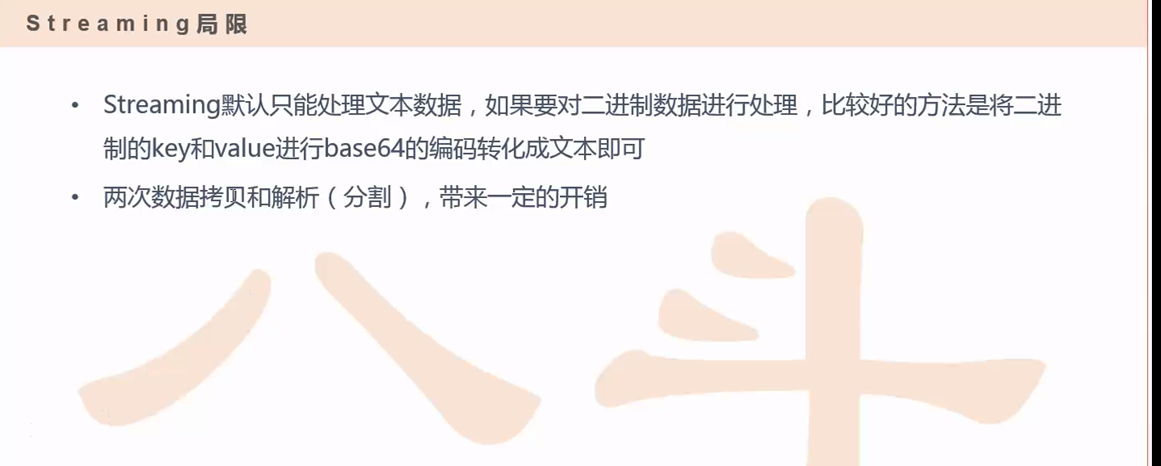
图5：



Streaming局限：

这个我们上节课也说了，Streaming默认方式是处理文本数据，对于这种二进制的数据进行处理就不太方便，另外一个它的效率没有java开发的效率高。（如下图6所示）

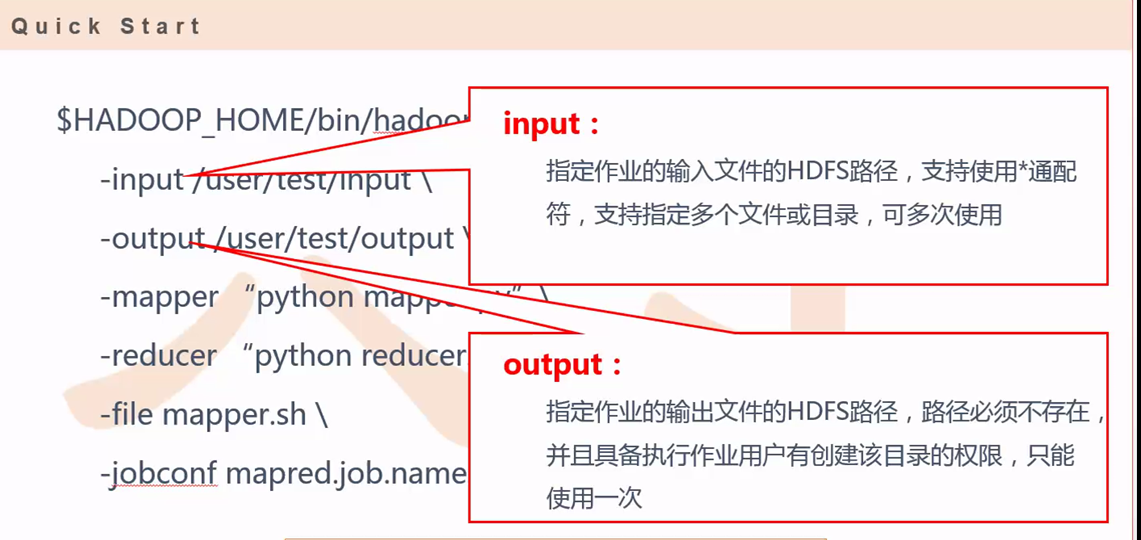
图6：



但是我们既然可以用MapReduce去开发去跑任务，那对于拷贝和解析带来的一些开销是可以忽略的，因为MapReduce本身实效性就不高。

1. 好了，我们真正来看一下在做HadoopStreaming开发的时候需要注意的地方（如下图7所示）

图7：



我们上次不是做了一个wordcount例子吗？然后提交这个任务需要写一个run的一个脚本，在这个脚本里面需要指定一些参数，参数有些必不可少的参数：比如这个input.

-input输入就是说你这个任务里面，它读的源数据是在哪里，一般数据源都在HDFS上，所以数据源得指定HDFS上的路径。

那输入指定完以后，那你要执行的这个大数据处理，要做完相关的数据处理之后，要把最后的结果输出出来，那么输出的话就通过output指定一个输出路径，这个输出路径也可以指定到HDFS路径。

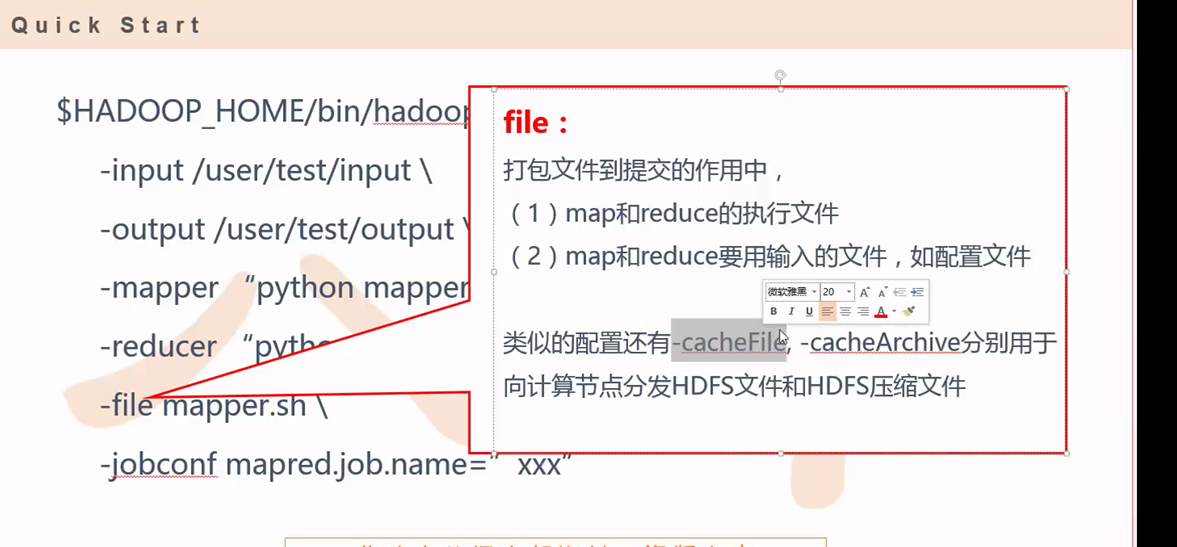
那么这个-mapper需要自己指定一个，因为毕竟map和reduce两个阶段，那么这个map就要指定你是通过怎样的方式去运行你的map进程，那么这个是通过一个python解析器 后面通过mapper.py这样的一个脚本文件。那么这些个参数呢都是Streaming帮我们提供好的与外部友好交互的一个入口。

那么-reduce是必须的吗？不是必须要写的，有很多的mapreduce根本就不需要reduce，你比如说只是说做一个简单的过滤，所以这个reduce根据自己的情况来写，reduce是可选的

那么-file 就是你本地写的代码，分发上去，那么-file只适合小的文件分发，如果你的这个文件比较大，建议用图8-cacheFile和-cacheArchive的这两种方式，这两个参数有一个共同点是它要分发的这个数据是已经在HDFS上了，然后这两个细节有什么区别呢？cacheFile就是指的是一个具体的文件，那-cacheArchive它相当于是一个文件体系（目录），这个目录里面可能有很多个文件或文件夹，我们去执行mapreduce的时候它是分布到整个集群的不同的节点上去运行的，那你写完了一个代码，你的代码只是在你本地，那你的代码你想运行的时候可以通过-mapper去执行，但是真正在一些命令在远程节点上运行，但是远程节点上并没有你这个-mapper这个脚本并没有你开发的代码没有你写的程序怎办？所以需要把你写的程序通过-file的方式去分发，分发到去执行你这个任务的节点上去。

然后还有一些个参数配置（jobconf），通过jobconf参数去激活一些个功能，比如可以为这个当前的Mapreduce取一个名字，或者你可以通过这样的方式去控制你reduce的个数，或者是你可以指定一些个分隔符，然后指定key的那个区域，那这个我们会在实践中会说

图8：



在下图9列出来了很多很重要的参数

图9：



这些参数我们工作中用的基本上就是这些，但是也不局限在这些。