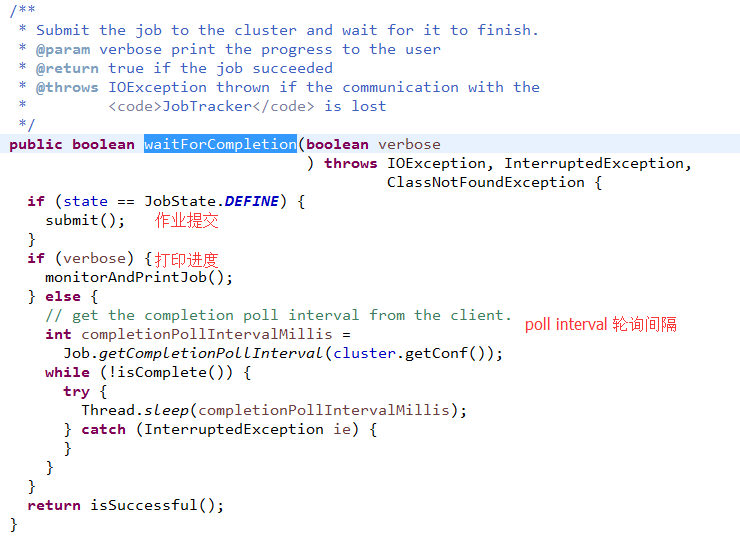
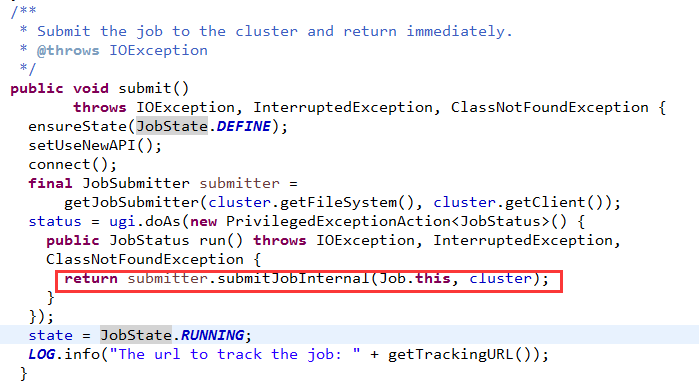
Hadoop作业提交

1：Job.waitForCompletion(boolean)

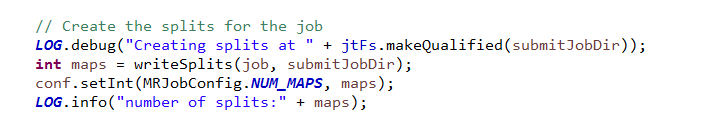


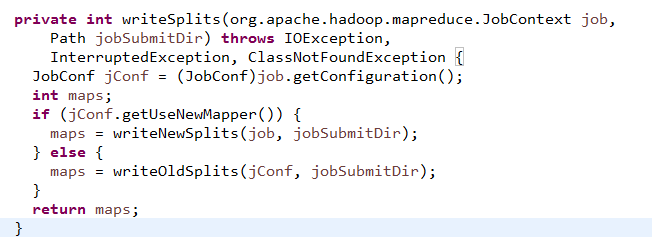
2：Job.submit()

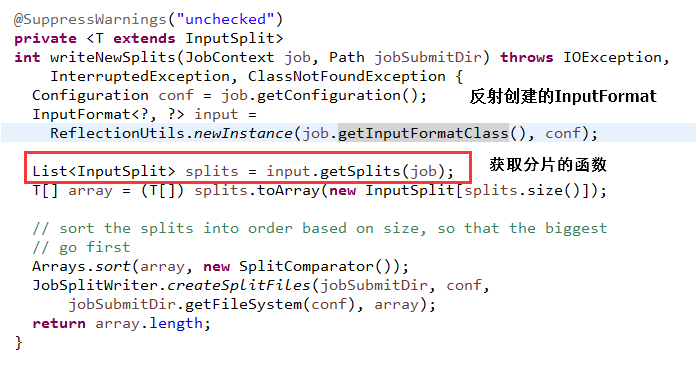


3: JobSubmitter.submitJobInternal(Job, Cluster)中的如下部分：

（1）获取map任务数量，由于一个split对应一个map任务，所以split数目就是map数目。

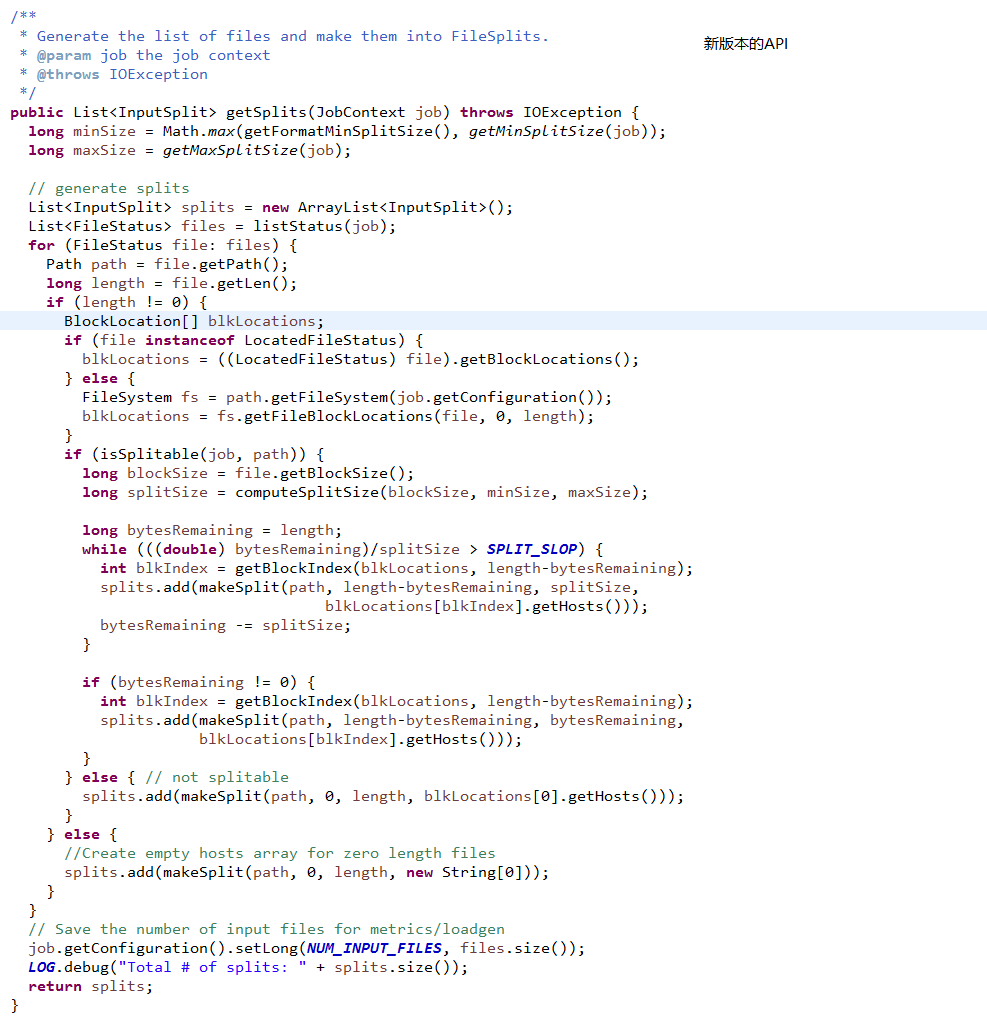






备注：大神写的代码就是健全！通过调用：

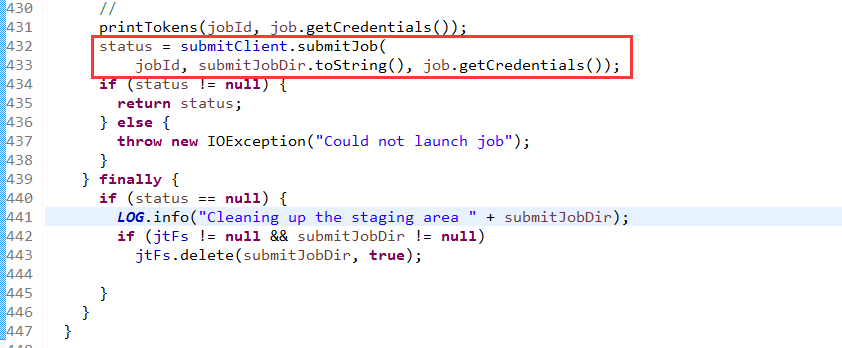
List<InputSplit> splits = input.getSplits(job);获取分片，然后转成数组，排序将大的split优先进行处理并返回split数目。



下面是分片方法FileInputFormat.getSplits(JobContext) ：



JobSubmitter.submitJobInternal(Job, Cluster)的部分代码：



客户端提交作业的代码，进入submitJob方法：submitJob的实现有两套。

一套是：

org.apache.hadoop.mapred.LocalJobRunner.submitJob(JobID,String, Credentials)实现。是单机模式的启动，本地调试。

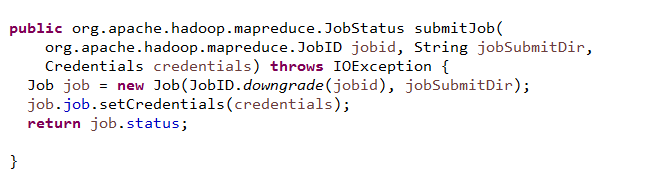
另外一套是：

org.apache.hadoop.mapred.YARNRunner.submitJob(JobID,String,Credentials)的实现

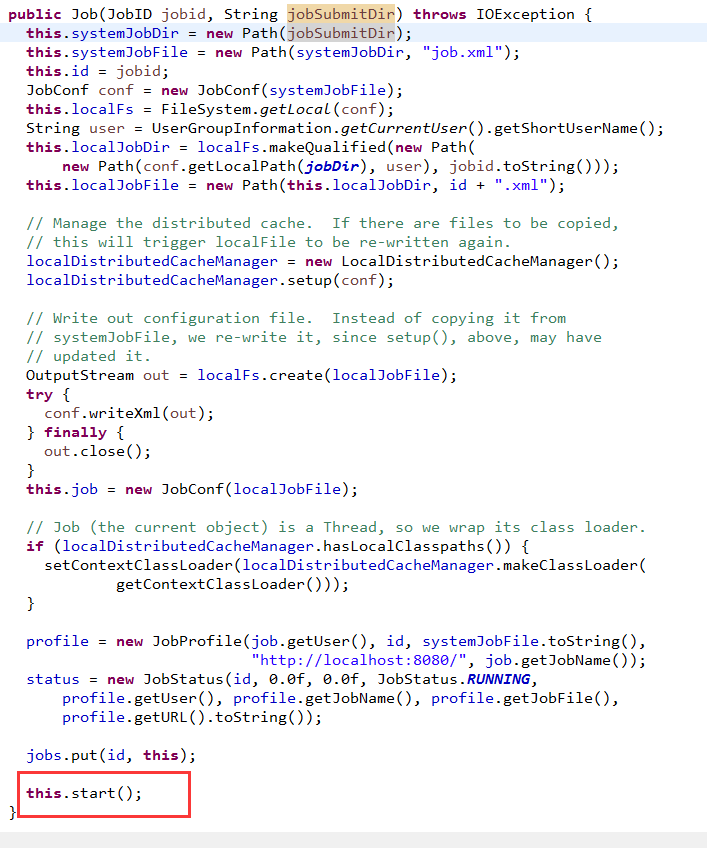


分别对应两种提交方式。

首先查看LocalJobRunner.submitJob(JobID,String, Credentials)实现：

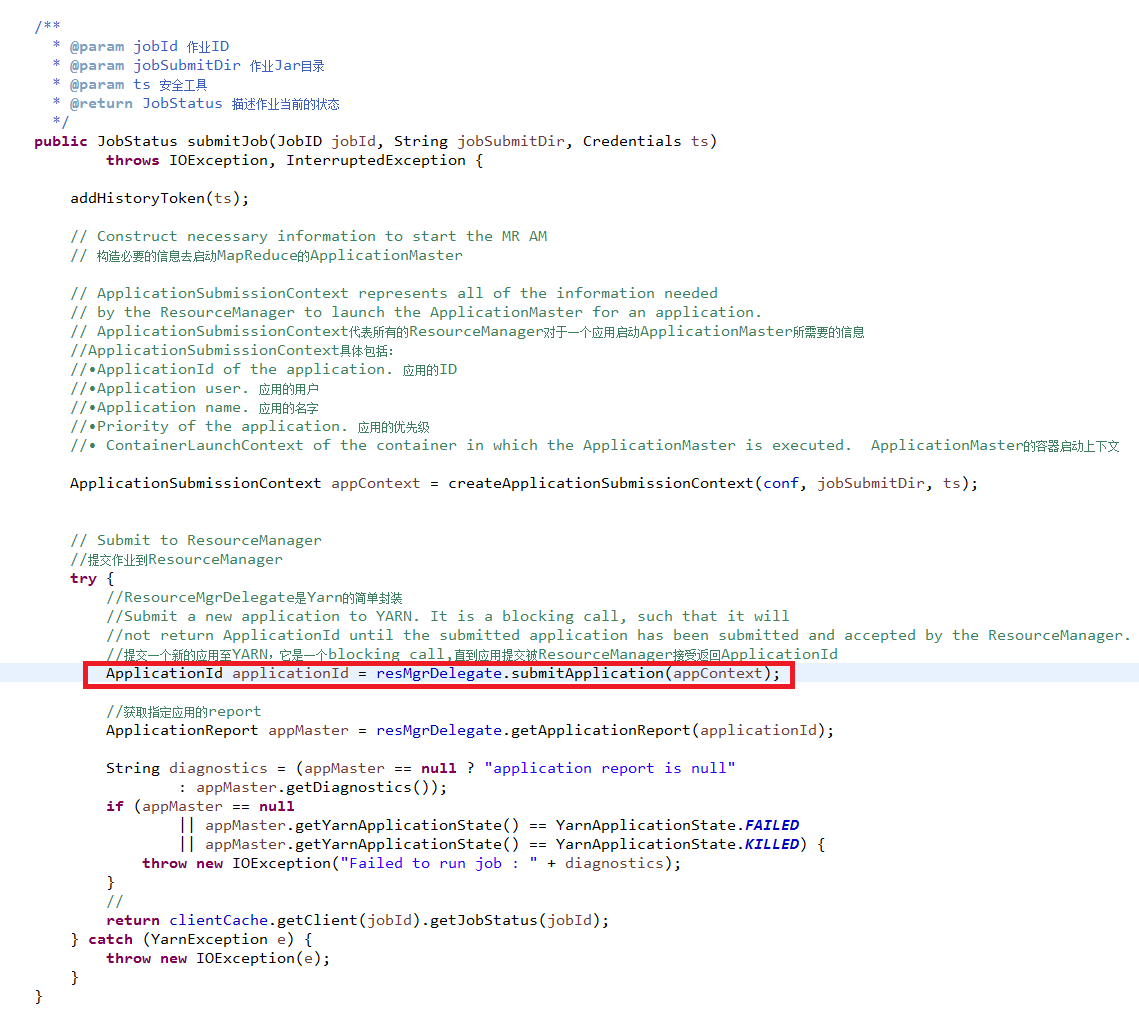
单机模式的启动，本地调试。

真正作业的启动是在Job构造函数中：this.start()启动作业！

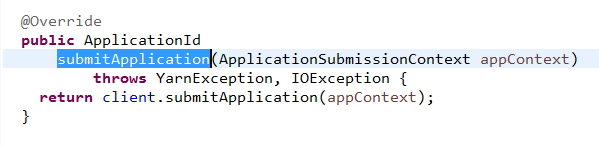


查看YARNRunner.submitJob(JobID,String,Credentials)的实现

（重点！！！）



ResourceMgrDelegate.submitApplication(ApplicationSubmissionContext)

其中 client为 YarnClient类型。

调用YarnClientImpl.submitApplication(ApplicationSubmissionContext) 实现作业提交



