1. 目标，将gpuenabler代码运行在spark2.4.3 上面，下面是用到的其他依赖的版本

Cuda 10.1

Spark 2.4.3

Scala 2.11.12

Java 1.8.0\_241

Jcuda 10.1

二．首先，修改gpuenable代码中的maven配置文件，将里面的依赖的版本都修改为我们需要的版本。

1.Gpuenabler/pom.xml

2.Gpuenabler/examples/pom.xml

3.Gpuenabler/gpu-enabler/pom.xml

4.Gpuenabler/gpu-enabler/dependency-reduced-pom.xml

5.mavenized-jcuda/pom.xml

6.mavenized-jcuda-old/pom.xml

三．修改好配置文件之后，用idea打开Gpuenabler/gpu-enabler工程，开始修改代码

首先要配置idea的编译器。删除自动加载的maven依赖的jars，重新配置scala版本为2.11.12，使用spark的版本为2.4.3，依次打开各个源文件，查看有无编译错误

1. CUDADSFunctionSuite.scala和TestJavaCUDASuite.java由于使用scalateset进行测试会出错，所以先全部注释掉
2. CUDACodeGenerator.scala：把import里的asm5改为asm6即可
3. GPUMemoryManager.scala：把出错的askWithRetry改为askSync
4. JCUDACodeGen.scala：这个文件里修改最多，首先修改显示的错误

Spark2.4.3之后，boxedType 和javaType两个函数都改到了CodeGenerator包里面，所以需要把ctx都改成CodeGenerator。954行和974行，CodeGenerator.compile(code)返回结果变为了一个Tuple，所以要用.\_1活的对应的结果。此时编译错误都解决了，此时加入一个测例子来运行，检验结果。

1. 在com.ibm.gpuenabler中加入一个test文件，粘贴GpuEnablerExample.scala中的代码。

1.第一个错误是val ptxURL = getClass.getResource("/GpuEnablerExamples.ptx")

这一行是由于文件路径没有找到GpuEnablerExamples.ptx文件，所以要把文件放到对应的位置。

1. 把test文件内容改为GpuDSArrayMult.scala文件进行测试，发现更多错误

经过多方查找资料，这个错误主要是由于构造函数的问题。要么构造函数内部数据类型出错，要么因为构造函数是private的，要么因为构造函数里面访问了访问不到的private变量。我们由于将spark2.1.0的代码迁移到spark2.4.3上，许多spark源文件内容发生了改变，有的函数从public变味了private，所以会报这个错误。

但是定位报错的位置是generated.java，这个是gpuenabler运行时生成的中间代码，在JCUDACodeGen.java代码中加入几行代码，将这个文件写出来，以对错误进行调试。

编译发现JCUDACodeGen.java代码中错误很多，主要集中在

org.apache.spark.sql.catalyst.expressions.codegen.BufferHolder

org.apache.spark.sql.catalyst.expressions.codegen.UnsafeRowWriter

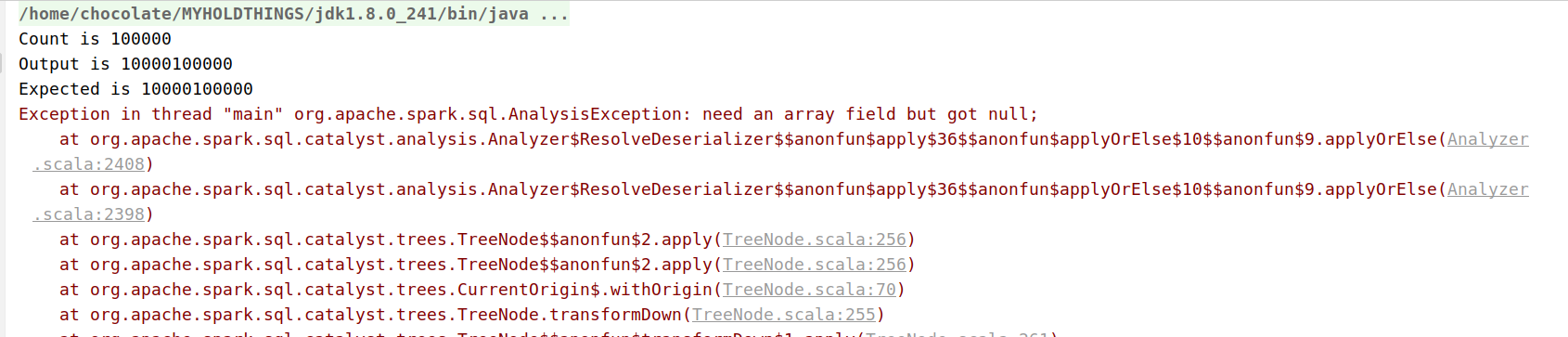
org.apache.spark.sql.catalyst.expressions.codegen.UnsafeArrayWriter

这三个import中。经过对比spark2.1.0和spark2.4.3的源码，codegen下的代码改动很大，组织结构也发生了变化，其中更有很多方法由原来的public变为了private，所以导致了上面的问题。经过多次改正无果，决定不使用spark2.4.3中的codegen，将引入的代码注释掉，然后把spark2.1.0中的codegen相关代码粘贴过来。改正位置在JCUDACodeGen.scala中，这个文件后面有一段很长的代码是生成的generated.java的内容。

加入的代码为BufferHolder，UnsafeRowWriter，UnsafeArrayWriter三个文件的全部内容。此时再根据编译出来的generated.java代码，修改因为这个导致的相关错误。

3.接着继续调试

首先遇到data.json路径问题，修改好后出现

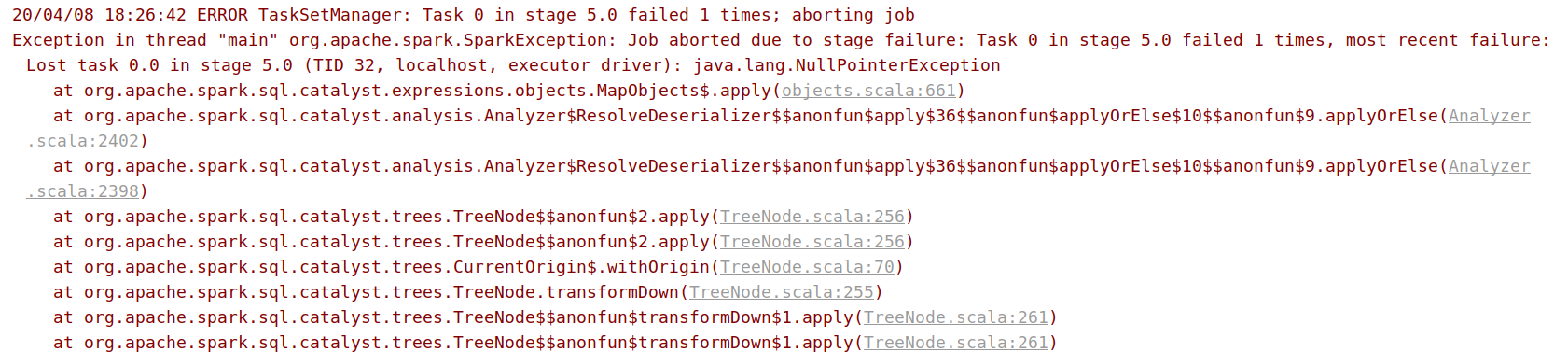


原因出现在

val dds = ds.withColumn("result", lit(null: Array[Double] )).as[inputData]

这一行代码。在spark2.1.0中可以编译通过，但spark2.4.3出现问题，推测是Null的问题，然后将这一行代码修改成lit( Array[Long](3) )，目前还不知道原因是什么。

4.最后一个问题，所有不通过测例都出现的问题，卡了三个星期的问题，目前也不知道什么原因的问题



问题定位在

2020-04-08 18-28-18 的屏幕截图

百思不得其解，终于在某一天，尝试将第155行注释掉，PASS！

1. 修改jcuda版本
2. 修改GpuEnabler/pom.xml文件，仿照jcuda80，加入jcuda10，不过skip.jcuda.natives要修改一下
3. GpuEnabler/修改compile.sh文件，加入cuda10.1和jcuda10的选项
4. 从网上下载jcuda10.1的包，按照mavenized-jcuda/repo里jucuda8.0的形式，加入我们的jcuda10.1。
5. 之后编译成功，不过出现warning,只需要把mavenized-jcuda/repo/里面的md5和sha文件删除即可。最终编译结果：

