# [Spark常见问题汇总](https://www.cnblogs.com/arachis/p/Spark_Exception.html)

原文地址：https://my.oschina.net/tearsky/blog/629201

摘要：

　　1、Operation category READ is not supported in state standby

　　2、配置spark.deploy.recoveryMode选项为ZOOKEEPER

　　3、多Master如何配置

　　4、No Space Left on the device（Shuffle临时文件过多）

　　5、java.lang.OutOfMemory, unable to create new native thread

　　6、Worker节点中的work目录占用许多磁盘空间

　　7、spark-shell提交Spark Application如何解决依赖库

　　8、Spark在发布应用的时候，出现连接不上master问题

　　9、开发spark应用程序（和Flume-NG结合时）发布应用时可能出现org.jboss.netty.channel.ChannelException: Failed to bind to: /192.168.10.156:18800

　　10、spark-shell 找不到hadoop so问题解决

　　11、ERROR XSDB6: Another instance of Derby may have already booted the database /home/bdata/data/metastore\_db.

　　12、java.lang.IllegalArgumentException: java.net.UnknownHostException: dfscluster

　　13、Exception in thread "main" java.lang.Exception: When running with master 'yarn-client' either HADOOP\_CONF\_DIR or YARN\_CONF\_DIR must be set in the environment.

　　14、Job aborted due to stage failure: Task 3 in stage 0.0 failed 4 times, most recent failure: Lost task 3.3 in

　　15、长时间等待无反应，并且看到服务器上面的web界面有内存和核心数，但是没有分配

　　16、内存不足或数据倾斜导致Executor Lost（spark-submit提交）

　　17、java.io.IOException : Could not locate executable null\bin\winutils.exe in the Hadoop binaries.（spark sql on hive 任务引发HiveContext NullPointerException）

　　18、The root scratch dir: /tmp/hive on HDFS should be writable. Current permissions are: rwx------

　　19、Exception in thread "main" org.apache.hadoop.security.AccessControlException : Permission denied: user=Administrator, access=WRITE, inode="/data":bdata:supergroup:drwxr-xr-x

　　20、运行Spark-SQL报错：org.apache.spark.sql.AnalysisException: unresolved operator 'Project‘

　　21、org.apache.spark.shuffle.MetadataFetchFailedException:Missing an output location for shuffle 0/Failed to connect to hostname/192.168.xx.xxx:50268

　　22、spark error already tried 45 time(s); maxRetries=45

　　23.cloudera 更改spark高级配置

　　24、spark Exception in thread "Thread-2" java.lang.OutOfMemoryError: PermGen space

　　25、pyspark TypeError: 'JavaPackage' object is not callable

　　26、spark als StackOverflowError

注意：如果Driver写好了代码，eclipse或者程序上传后，没有开始处理数据，或者快速结束任务，也没有在控制台中打印错误，那么请进入spark的web页面，查看一下你的任务，找到每个分区日志的stderr，查看是否有错误，一般情况下一旦驱动提交了，报错的情况只能在任务日志里面查看是否有错误情况了

### 1、Operation category READ is not supported in state standby

     org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.ipc.StandbyException): Operation category READ is not supported in state standby

     此时请登录Hadoop的管理界面查看运行节点是否处于standby

如登录地址是：

[http://192.168.50.221:50070/dfshealth.html#tab-overview](http://192.168.50.221:50070/dfshealth.html" \l "tab-overview)

     如果是，则不可在处于StandBy机器运行spark计算，因为该台机器为备分机器

### 2、配置spark.deploy.recoveryMode选项为ZOOKEEPER

     如果不设置spark.deploy.recoveryMode的话，那么集群的所有运行数据在Master重启是都会丢失，可参考BlackHolePersistenceEngine的实现。

### 3、多Master如何配置

     因为涉及到多个Master，所以对于应用程序的提交就有了一点变化，因为应用程序需要知道当前的Master的IP地址和端口。这种HA方案处理这种情况很简单，只需要在SparkContext指向一个Master列表就可以了，如spark://host1:port1,host2:port2,host3:port3，应用程序会轮询列表。

### 4、No Space Left on the device（Shuffle临时文件过多）

         由于Spark在计算的时候会将中间结果存储到/tmp目录，而目前linux又都支持tmpfs，其实就是将/tmp目录挂载到内存当中。

那么这里就存在一个问题，中间结果过多导致/tmp目录写满而出现如下错误

No Space Left on the device

解决办法

第一种：修改配置文件spark-env.sh,把临时文件引入到一个自定义的目录中去即可

export SPARK\_LOCAL\_DIRS=/home/utoken/datadir/spark/tmp

第二种：偷懒方式，针对tmp目录不启用tmpfs,直接修改/etc/fstab

cloudera manager 添加参数配置：筛选器=>高级=>搜索“spark\_env”字样，添加参数export SPARK\_LOCAL\_DIRS=/home/utoken/datadir/spark/tmp到所有配置项

### 5、java.lang.OutOfMemory, unable to create new native thread

Caused by: java.lang.OutOfMemoryError: unable to create new native thread

at java.lang.Thread.start0(Native Method)

at java.lang.Thread.start(Thread.java:640)

上面这段错误提示的本质是Linux操作系统无法创建更多进程，导致出错，并不是系统的内存不足。因此要解决这个问题需要修改Linux允许创建更多的进程，就需要修改Linux最大进程数。

[utoken@nn1 ~]$ulimit -a

临时修改允许打开的最大进程数

[utoken@nn1 ~]$ulimit -u 65535

临时修改允许打开的文件句柄

[utoken@nn1 ~]$ulimit -n 65535

永久修改Linux最大进程数量

[utoken@nn1 ~]$ vim /etc/security/limits.d/90-nproc.conf

\*          soft    nproc     60000

root       soft    nproc     unlimited

永久修改用户打开文件的最大句柄数，该值默认1024，一般都会不够，常见错误就是not open file

[utoken@nn1 ~]$ vim /etc/security/limits.conf

bdata  soft    nofile  65536

bdata  hard    nofile  65536

### 6、Worker节点中的work目录占用许多磁盘空间

目录地址：/home/utoken/software/spark-1.3.0-bin-hadoop2.4/work

这些是Driver上传到worker的文件，需要定时做手工清理，否则会占用许多磁盘空间

### 7、spark-shell提交Spark Application如何解决依赖库

spark-shell的话，利用--driver-class-path选项来指定所依赖的jar文件，注意的是--driver-class-path后如果需要跟着多个jar文件的话，jar文件之间使用冒号(:)来分割。

### 8、Spark在发布应用的时候，出现连接不上master问题，如下

15/11/19 11:35:50 INFO AppClient$ClientEndpoint: Connecting to master spark://s1:7077...

15/11/19 11:35:50 WARN ReliableDeliverySupervisor: Association with remote system [akka.tcp://sparkMaster@s1:7077] has failed, address is now gated for [5000] ms. Reason: [Disassociated]

解决方式

检查所有机器时间是否一致、hosts是否都配置了映射、客户端和服务器端的Scala版本是否一致、Scala版本是否和Spark兼容

检查是否兼容问题请参考官方网站介绍：

9、开发spark应用程序（和Flume-NG结合时）发布应用时可能出现org.jboss.netty.channel.ChannelException: Failed to bind to: /192.168.10.156:18800

15/11/27 10:33:44 ERROR ReceiverSupervisorImpl: Stopped receiver with error: org.jboss.netty.channel.ChannelException: Failed to bind to: /192.168.10.156:18800  
15/11/27 10:33:44 ERROR Executor: Exception in task 0.0 in stage 2.0 (TID 70)  
org.jboss.netty.channel.ChannelException: Failed to bind to: /192.168.10.156:18800

        at org.jboss.netty.bootstrap.ServerBootstrap.bind(ServerBootstrap.java:272)

Caused by: java.net.BindException: Cannot assign requested address

由于spark通过Master发布的时候，会自动选取发送到某一台的worker节点上，所以这里绑定端口的时候，需要选择相应的worker服务器，但是由于我们无法事先了解到，spark发布到哪一台服务器的，所以这里启动报错，是因为在 192.168.10.156:18800的机器上面没有启动Driver程序，而是发布到了其他服务器去启动了，所以无法监听到该机器出现问题，所以我们需要设置spark分发包时，发布到所有worker节点机器，或者发布后，我们去寻找发布到了哪一台机器，重新修改绑定IP，重新发布，有一定几率发布成功。详情可见《印象笔记-战5渣系列——Spark Streaming启动问题 - 推酷》

### 10、spark-shell 找不到hadoop so问题解决

[main] WARN  org.apache.hadoop.util.NativeCodeLoader  - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

在Spark的conf目录下，修改spark-env.sh文件，加入LD\_LIBRARY\_PATH环境变量，值为HADOOP的native库路径即可.

### 11、ERROR XSDB6: Another instance of Derby may have already booted the database /home/bdata/data/metastore\_db.

在使用Hive on Spark模式操作hive里面的数据时，报以上错误，原因是因为HIVE采用了derby这个内嵌数据库作为数据库，它不支持多用户同时访问,解决办法就是把derby数据库换成mysql数据库即可

变更方式

### java.lang.IllegalArgumentException: java.net.UnknownHostException: dfscluster

解决办法：

找不到hdfs集群名字dfscluster,这个文件在HADOOP的etc/hadoop下面，有个文件hdfs-site.xml，复制到Spark的conf下，重启即可

如：执行脚本，分发到所有的Spark集群机器中，

[bdata@bdata4 hadoop]foriin34,35,36,37,38;doscphdfs−site.xml192.168.10.

i:/u01/spark-1.5.1/conf/ ; done

### 13、Exception in thread "main" java.lang.Exception: When running with master 'yarn-client' either HADOOP\_CONF\_DIR or YARN\_CONF\_DIR must be set in the environment.

问题：在执行yarn集群或者客户端时，报以上错误，

[bdata@bdata4 bin]$ ./spark-sql --master yarn-client

Exception in thread "main" java.lang.Exception: When running with master 'yarn-client' either HADOOP\_CONF\_DIR or YARN\_CONF\_DIR must be set in the environment.

解决办法

根据提示，配置HADOOP\_CONF\_DIR or YARN\_CONF\_DIR的环境变量即可

export HADOOP\_HOME=/u01/hadoop-2.6.1

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

PATH=PATH:

HOME/.local/bin:HOME/bin:SQOOP\_HOME/bin:HIVEHOME/bin:

HADOOP\_HOME/bin

### 14、Job aborted due to stage failure: Task 3 in stage 0.0 failed 4 times, most recent failure: Lost task 3.3 in

[Stage 0:>                                                         (0 + 4) / 42]2016-01-15 11:28:16,512 [org.apache.spark.scheduler.TaskSchedulerImpl]-[ERROR] Lost executor 0 on 192.168.10.38: remote Rpc client disassociated

[Stage 0:>                                                         (0 + 4) / 42]2016-01-15 11:28:23,188 [org.apache.spark.scheduler.TaskSchedulerImpl]-[ERROR] Lost executor 1 on 192.168.10.38: remote Rpc client disassociated

[Stage 0:>                                                         (0 + 4) / 42]2016-01-15 11:28:29,203 [org.apache.spark.scheduler.TaskSchedulerImpl]-[ERROR] Lost executor 2 on 192.168.10.38: remote Rpc client disassociated

[Stage 0:>                                                         (0 + 4) / 42]2016-01-15 11:28:36,319 [org.apache.spark.scheduler.TaskSchedulerImpl]-[ERROR] Lost executor 3 on 192.168.10.38: remote Rpc client disassociated

2016-01-15 11:28:36,321 [org.apache.spark.scheduler.TaskSetManager]-[ERROR] Task 3 in stage 0.0 failed 4 times; aborting job

Exception in thread "main" org.apache.spark.SparkException : Job aborted due to stage failure: Task 3 in stage 0.0 failed 4 times, most recent failure: Lost task 3.3 in stage 0.0 (TID 14, 192.168.10.38): ExecutorLostFailure (executor 3 lost)

Driver stacktrace:

      at org.apache.spark.scheduler.DAGScheduler.orgapache

sparkscheduler

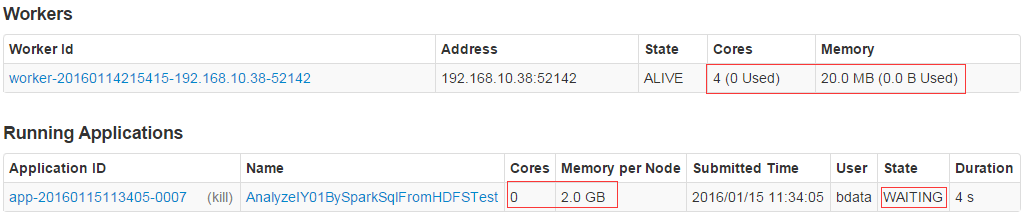
DAGScheduler$$failJobAndIndependentStages(DAGScheduler.scala:1283)

解决方案

这里遇到的问题主要是因为数据源数据量过大，而机器的内存无法满足需求，导致长时间执行超时断开的情况，数据无法有效进行交互计算，因此有必要增加内存

15、长时间等待无反应，并且看到服务器上面的web界面有内存和核心数，但是没有分配，如下图

[Stage 0:>                                                         (0 + 0) / 42]



或者日志信息显示：

16/01/15 14:18:56 WARN TaskSchedulerImpl: Initial job has not accepted any resources; check your cluster UI to ensure that workers are registered and have sufficient resources

解决方案

出现上面的问题主要原因是因为我们通过参数spark.executor.memory设置的内存过大，已经超过了实际机器拥有的内存，故无法执行，需要等待机器拥有足够的内存后，才能执行任务，可以减少任务执行内存，设置小一些即可

### 16、内存不足或数据倾斜导致Executor Lost（spark-submit提交）

TaskSetManager: Lost task 1.0 in stage 6.0 (TID 100, 192.168.10.37): java.lang.OutOfMemoryError: Java heap space

16/01/15 14:29:51 INFO BlockManagerInfo: Added broadcast\_8\_piece0 in memory on 192.168.10.37:57139 (size: 42.0 KB, free: 24.2 MB)

16/01/15 14:29:53 INFO BlockManagerInfo: Added broadcast\_8\_piece0 in memory on 192.168.10.38:53816 (size: 42.0 KB, free: 24.2 MB)  
16/01/15 14:29:55 INFO TaskSetManager: Starting task 3.0 in stage 6.0 (TID 102, 192.168.10.37, ANY, 2152 bytes)  
16/01/15 14:29:55 WARN TaskSetManager: Lost task 1.0 in stage 6.0 (TID 100, 192.168.10.37): java.lang.OutOfMemoryError: Java heap space  
        at java.io.BufferedOutputStream.<init>(BufferedOutputStream.java:76)  
        at java.io.BufferedOutputStream.<init>(BufferedOutputStream.java:59)  
        at org.apache.spark.sql.execution.UnsafeRowSerializerInstance$$anon$2.<init>(UnsafeRowSerializer.scala:55)  
        at org.apache.spark.sql.execution.UnsafeRowSerializerInstance.serializeStream(UnsafeRowSerializer.scala:52)  
        at org.apache.spark.storage.DiskBlockObjectWriter.open(DiskBlockObjectWriter.scala:92)  
        at org.apache.spark.shuffle.sort.BypassMergeSortShuffleWriter.insertAll(BypassMergeSortShuffleWriter.java:110)  
        at org.apache.spark.shuffle.sort.SortShuffleWriter.write(SortShuffleWriter.scala:73)  
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:73)  
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:41)  
        at org.apache.spark.scheduler.Task.run(Task.scala:88)  
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:214)  
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)  
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)  
        at java.lang.Thread.run(Thread.java:744)  
  
16/01/15 14:29:55 ERROR TaskSchedulerImpl: Lost executor 6 on 192.168.10.37: remote Rpc client disassociated  
16/01/15 14:29:55 INFO TaskSetManager: Re-queueing tasks for 6 from TaskSet 6.0  
16/01/15 14:29:55 WARN ReliableDeliverySupervisor: Association with remote system [akka.tcp://sparkExecutor@192.168.10.37:42250] has failed, address is now gated for [5000] ms. Reason: [Disassociated]

16/01/15 14:29:55 WARN TaskSetManager: Lost task 3.0 in stage 6.0 (TID 102, 192.168.10.37): ExecutorLostFailure (executor 6 lost)

16/01/15 14:29:55 INFO DAGScheduler: Executor lost: 6 (epoch 8)  
16/01/15 14:29:55 INFO BlockManagerMasterEndpoint: Trying to remove executor 6 from BlockManagerMaster.  
16/01/15 14:29:55 INFO BlockManagerMasterEndpoint: Removing block manager BlockManagerId(6, 192.168.10.37, 57139)  
16/01/15 14:29:55 INFO BlockManagerMaster: Removed 6 successfully in removeExecutor  
16/01/15 14:29:55 INFO AppClient$ClientEndpoint: Executor updated: app-20160115142128-0001/6 is now EXITED (Command exited with code 52)  
16/01/15 14:29:55 INFO SparkDeploySchedulerBackend: Executor app-20160115142128-0001/6 removed: Command exited with code 52

16/01/15 14:29:55 INFO SparkDeploySchedulerBackend: Asked to remove non-existent executor 6

.......

org.apache.spark.SparkException: Job aborted due to stage failure: Task 0 in stage 6.0 failed 4 times, most recent failure: Lost task 0.3 in stage 6.0 (TID 142, 192.168.10.36): ExecutorLostFailure (executor 4 lost)

......

WARN TaskSetManager: Lost task 4.1 in stage 6.0 (TID 137, 192.168.10.38): java.lang.OutOfMemoryError: GC overhead limit exceeded

解决办法：

     由于我们在执行Spark任务是，读取所需要的原数据，数据量太大，导致在Worker上面分配的任务执行数据时所需要的内存不够，直接导致内存溢出了，所以我们有必要增加Worker上面的内存来满足程序运行需要。

在Spark Streaming或者其他spark任务中，会遇到在Spark中常见的问题，典型如Executor Lost 相关的问题(shuffle fetch 失败，Task失败重试等)。这就意味着发生了内存不足或者数据倾斜的问题。这个目前需要考虑如下几个点以获得解决方案：

A、相同资源下，增加partition数可以减少内存问题。 原因如下：通过增加partition数，每个task要处理的数据少了，同一时间内，所有正在运行的task要处理的数量少了很多，所有Executor占用的内存也变小了。这可以缓解数据倾斜以及内存不足的压力。

B、关注shuffle read 阶段的并行数。例如reduce,group 之类的函数，其实他们都有第二个参数，并行度(partition数)，只是大家一般都不设置。不过出了问题再设置一下，也不错。

C、给一个Executor 核数设置的太多，也就意味着同一时刻，在该Executor 的内存压力会更大，GC也会更频繁。我一般会控制在3个左右。然后通过提高Executor数量来保持资源的总量不变。

### 16、 Spark Streaming 和kafka整合后读取消息报错：OffsetOutOfRangeException

解决方案：如果和kafka消息中间件结合使用，请检查消息体是否大于默认设置1m，如果大于，则需要设置fetch.message.max.bytes=1m，这里需要把值设置大些

### 17、java.io.IOException : Could not locate executable null\bin\winutils.exe in the Hadoop binaries.（spark sql on hive 任务引发HiveContext NullPointerException）

解决办法

在开发hive和Spark整合的时候，如果是Windows系统，并且没有配置HADOOP\_HOME的环境变量，那么可能找不到winutils.exe这个工具，由于使用hive时，对该命令有依赖，所以不要忽视该错误，否则将无法创建HiveContext，一直报Exception in thread "main" java.lang.RuntimeException:java.lang.NullPointerException

因此，解决该办法有两个方式

A、把任务打包成jar，上传到服务器上面，服务器是配置过HADOOP\_HOME环境变量的，并且不需要依赖winutils,所以只需要通过spark-submit方式提交即可，如：

[bdata@bdata4 app]$ spark-submit --class com.pride.hive.HiveOnSparkTest --master spark://bdata4:7077 spark-simple-1.0.jar

B、解决winutils.exe命令不可用问题，配置Windows上面HADOOP\_HOME的环境变量，或者在程序最开始的地方设置HADOOP\_HOME的属性配置,这里需要注意，由于最新版本已经没有winutils这些exe命令了，我们需要在其他地方下载该命令放入HADOOP的bin目录下，当然也可以直接配置下载项目的环境变量，变量名一定要是HADOOP\_HOME才行

下载地址：<https://github.com/srccodes/hadoop-common-2.2.0-bin/archive/master.zip> （记得FQ哦）

任何项目都生效，需要配置Windows的环境变量，如果只在程序中生效可在程序中配置即可，如

//用于解决Windows下找不到winutils.exe命令

System. setProperty("hadoop.home.dir", "E:\\Software\\hadoop-common-2.2.0-bin" );

### 18、The root scratch dir: /tmp/hive on HDFS should be writable. Current permissions are: rwx------

解决办法

1、程序中设置环境变量：System.setProperty("HADOOP\_USER\_NAME", "bdata")

2、修改HDFS的目录权限

Update the permission of your /tmp/hive HDFS directory using the following command

#hadoop dfs -chmod 777 /tmp/hive

此问题暂未解决，估计是17点解决winutils有问题，建议最好把任务程序发布到服务器上面解决

### 19、Exception in thread "main" org.apache.hadoop.security.AccessControlException : Permission denied: user=Administrator, access=WRITE, inode="/data":bdata:supergroup:drwxr-xr-x

解决办法

1、在系统的环境变量或java JVM变量里面添加HADOOP\_USER\_NAME，如程序中添加System.setProperty("HADOOP\_USER\_NAME", "bdata");，这里的值就是以后会运行HADOOP上的Linux的用户名，如果是eclipse，则修改完重启eclipse，不然可能不生效

2、hdfs dfs -chmod 777 修改相应权限地址

### 20、运行Spark-SQL报错：org.apache.spark.sql.AnalysisException: unresolved operator 'Project

解决办法：

在Spark-sql和hive结合时或者单独Spark-sql，运行某些sql语句时，偶尔出现上面错误，那么我们可以检查一下sql的问题，这里遇到的问题是嵌套语句太多，导致spark无法解析，所以需要修改sql或者改用其他方式处理；特别注意该语句可能在hive里面没有错误，spark才会出现的一种错误。

### 21、org.apache.spark.shuffle.MetadataFetchFailedException:Missing an output location for shuffle 0/Failed to connect to hostname/192.168.xx.xxx:50268

解决方法：

1.优化shuffle操作（比如groupby,join）

2.加大executor内存（spark.executor.memory）

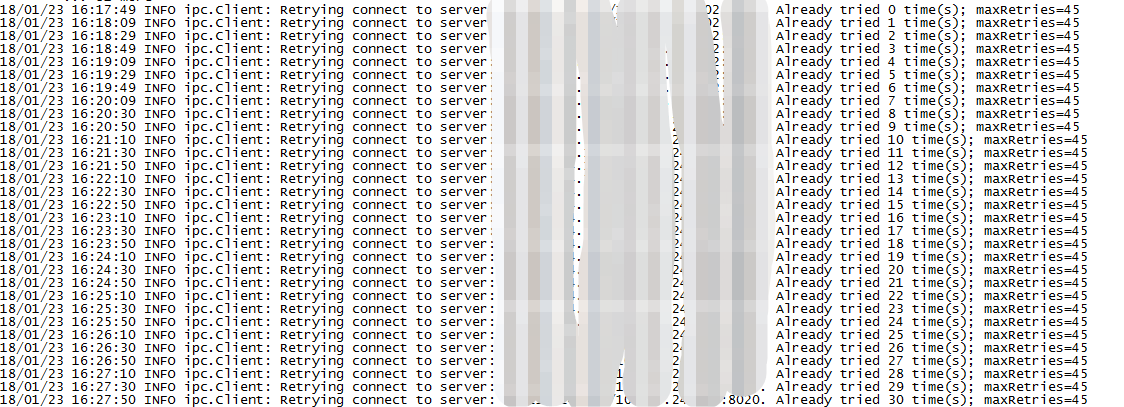
3.加大并行化参数（spark.default.parallelism）

参考链接：

http://blog.csdn.net/lsshlsw/article/details/49155087

### 22、spark error already tried 45 time(s); maxRetries=45

场景描述：spark提交任务，发现9个节点用8个运行都很快，只有一个节点一直失败，查看ui页面也没有发现数据倾斜  
问题定位：通过查看节点的work日志发现如下输出：

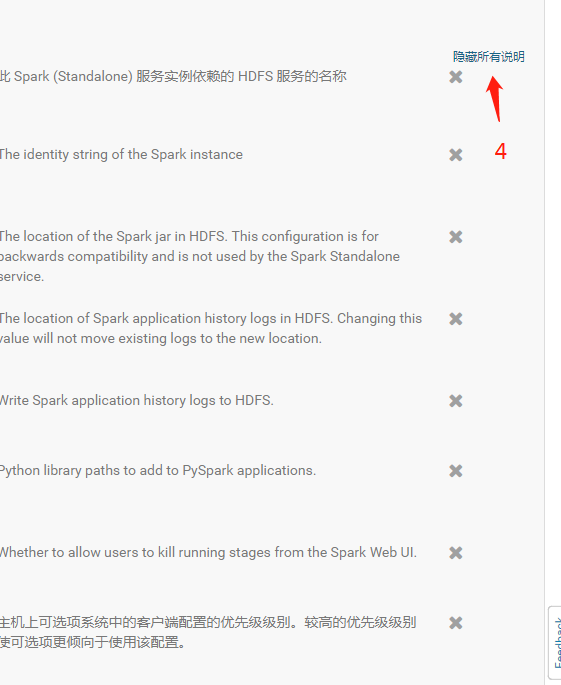


问题解决：修改spark的最大重试参数：spark.port.maxRetries 值从45=>16

### 23.cloudera 更改spark高级配置：

业务场景：在使用cloudera  manager搭建spark集群的过程中总是有些参数在配置界面中搜索不到

问题解决：集群=>配置=>高级，找到右侧的显示所有说明，找到你要修改的配置文件就可以了



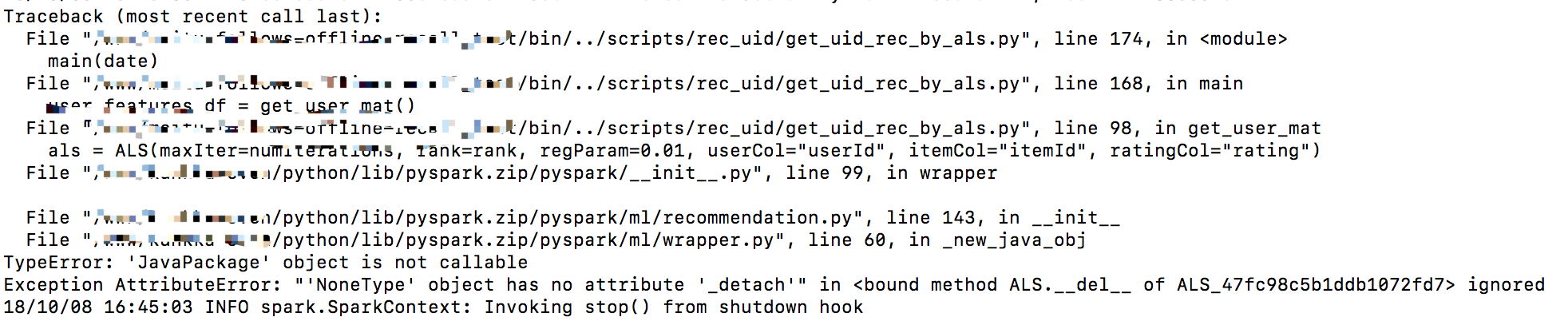
### 24.Exception in thread "Thread-2" java.lang.OutOfMemoryError: PermGen space

问题原因：在JVM中，PermGen用于保存加载的类，yarn Client模式的持久代默认大小是32M；

解决方案：spark-submit时增加如下参数：“--driver-java-options -XX:MaxPermSize=256m"

### 25.pyspark TypeError: 'JavaPackage' object is not callable

场景：pyspark使用py4j调用java对象ALS,这里有兴趣同学可以根据报错去跟读源码



问题原因：运行时找不到ALS类，也即没有spark-mllib包

解决方案：spark-submit时添加如下参数“--jars /path/to/spark-mllib\_scala.version−

{spark.version}.jar”

### 26、spark als StackOverflowError

场景：

als = ALS(maxIter=15, rank=rank, regParam=0.01, userCol="userId", itemCol="itemId", ratingCol="rating")

model = als.fit(ratings)

解决方案：

一般情况下ALS收敛是非常快的，将iteration设置为<30的数字就可以了，绝大多数情况下，设置为15就行。次数过多的情况下就会出现：java.lang.StackOverflowError。当然，也可也通过设置checkpoint来解决。

spark.sparkContext.setCheckpointDir("/tmp/als\_checkpoint")

als = ALS(maxIter=numIterations, rank=rank, regParam=0.01, userCol="userId", itemCol="itemId", ratingCol="rating",checkpointInterval=2)

model = als.fit(ratings)