**Hadoop问题集**

今天讲一讲关于Hadoop的问题、原因以及解决方法。

作者：王森丰|2016-12-13 16:36

[收藏](javascript:favorBox('open');)

[分享](javascript:;)

**1、问题：**

hadoop安装完以后，在执行命令时，经常会提示一下警告：

1. WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
2. using builtin-java classes where applicable

原因：

这个跟系统位数有关系，我们平常使用的是Centos 6.5 64位操作系统。

解决办法：

（1）如果你是hadoop2.6的可以下载下面这个：

http://dl.bintray.com/sequenceiq/sequenceiq-bin/hadoop-native-64-2.6.0.tar

（2）下载完以后，解压到hadoop的native目录下，覆盖原有文件即可。操作如下：

1. tar -x hadoop-native-64-2.4.0.tar -C hadoop/lib/native/

**2、hadoop HA namenode 无法启动错误**

问题：

1. 2014-12-15 14:27:55,224 INFO org.apache.hadoop.hdfs.server.namenode.FSImage: No edit log streams selected.
2. 2014-12-15 14:27:55,300 ERROR org.apache.hadoop.hdfs.server.namenode.FSImage: Failed to load image from FSImageFile(file=/usr/hadoop/tmp/dfs/name/current/fsimage\_0000000000000030732, cpktTxId=0000000000000030732)
3. java.io.IOException: Premature EOF from inputStream
4. at org.apache.hadoop.io.IOUtils.readFully(IOUtils.java:194)
5. at org.apache.hadoop.hdfs.server.namenode.FSImageFormat$LoaderDelegator.load(FSImageFormat.java:221)
6. at org.apache.hadoop.hdfs.server.namenode.FSImage.loadFSImage(FSImage.java:913)
7. at org.apache.hadoop.hdfs.server.namenode.FSImage.loadFSImage(FSImage.java:899)
8. at org.apache.hadoop.hdfs.server.namenode.FSImage.loadFSImageFile(FSImage.java:722)
9. at org.apache.hadoop.hdfs.server.namenode.FSImage.loadFSImage(FSImage.java:660)
10. at org.apache.hadoop.hdfs.server.namenode.FSImage.recoverTransitionRead(FSImage.java:279)
11. at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:955)
12. at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:700)
13. at org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:529)
14. at org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:585)
15. at org.apache.hadoop.hdfs.server.namenode.NameNode.(NameNode.java:751)
16. at org.apache.hadoop.hdfs.server.namenode.NameNode.(NameNode.java:735)
17. at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1407)
18. at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1473)
19. 2014-12-15 14:27:55,438 WARN org.apache.hadoop.hdfs.server.namenode.FSNamesystem: Encountered exception loading fsimage
20. java.io.IOException: Failed to load an FSImage file!
21. at org.apache.hadoop.hdfs.server.namenode.FSImage.loadFSImage(FSImage.java:671)
22. at org.apache.hadoop.hdfs.server.namenode.FSImage.recoverTransitionRead(FSImage.java:279)
23. at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:955)
24. at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:700)
25. at org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:529)
26. at org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:585)
27. at org.apache.hadoop.hdfs.server.namenode.NameNode.(NameNode.java:751)
28. at org.apache.hadoop.hdfs.server.namenode.NameNode.(NameNode.java:735)
29. at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1407)
30. at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1473)
31. 2014-12-15 14:27:55,440 INFO org.mortbay.log: Stopped HttpServer2$SelectChannelConnectorWithSafeStartup@HM0:50070
32. 2014-12-15 14:27:55,540 INFO org.apache.hadoop.metrics2.impl.MetricsSystemImpl: Stopping NameNode metrics system...
33. 2014-12-15 14:27:55,541 INFO org.apache.hadoop.metrics2.impl.MetricsSystemImpl: NameNode metrics system stopped.
34. 2014-12-15 14:27:55,541 INFO org.apache.hadoop.metrics2.impl.MetricsSystemImpl: NameNode metrics system shutdown complete.
35. 2014-12-15 14:27:55,541 FATAL org.apache.hadoop.hdfs.server.namenode.NameNode: Exception in namenode join
36. java.io.IOException: Failed to load an FSImage file!
37. at org.apache.hadoop.hdfs.server.namenode.FSImage.loadFSImage(FSImage.java:671)
38. at org.apache.hadoop.hdfs.server.namenode.FSImage.recoverTransitionRead(FSImage.java:279)
39. at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:955)
40. at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:700)
41. at org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:529)
42. at org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:585)
43. at org.apache.hadoop.hdfs.server.namenode.NameNode.(NameNode.java:751)
44. at org.apache.hadoop.hdfs.server.namenode.NameNode.(NameNode.java:735)
45. at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1407)
46. at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1473)

原因：

查看文件大小 fsimage\_0000000000000030732 大小为0，而在另一台服务器长度不为0，是这个fsimage文件的问题

解决：

可以从另一台服务器上拷贝该文件，或者删除这个文件及对应的md5文件即可。

建议从其他机器拷贝fsimage文件

**3、设置conbinner注意事项**

问题：

在使用combiner的时候，出现IntWritable is not Text，或者其他的类型不匹配问题

原因：

这是设置combiner的时候，map输出的keyvalue类型和combiner输出的keyvalue类型不一致导致的。

解决：

如果设置combiner，必须保证一点，map输出的keyvalue类型和combiner输出的keyvalue类型必须一致!!!

[](http://s4.51cto.com/wyfs02/M01/8B/7F/wKioL1hP2C2hx8DbAAEgzMyFudc464.jpg-wh_651x-s_1304471733.jpg)

**4、error：server IPC version 9 cannot communicate with client version 4**

问题：

error：server IPC version 9 cannot communicate with client version 4

原因：

（1）如果是在使用插件操作hdfs时报错，是因为，eclipse插件和eclipse的版本不匹配

（2）如果是在执行mapreduce时报错：是因为jar包不匹配

解决：

重新编译hadoop插件，使用自己的hadoop和eclipse版本

**5、Could not locate executable null\bin\winutils.exe in the Hadoop binaries.**

问题：

Could not locate executable null\bin\winutils.exe in the Hadoop binaries.

原因：

在使用hadoop插件的时候，会在本地找winutils.exe这个文件，而这个文件是hadoop安装包下的文件，必须配置eclipse插件的hadoop的windows本地路径才行。

解决：

配置windows的hadoop环境变量，并且重启机器，让环境变量生效

**6、问题：Shuffle Error: Exceeded MAX\_FAILED\_UNIQUE\_FETCHES; bailing-out**

原因：

程序里面需要打开多个文件，进行分析，系统一般默认数量是1024，(用ulimit -a可以看到)对于正常使用是够了，但是对于程序来讲，就太少了。

解决：

修改2个文件。

/etc/security/limits.conf

1. vi /etc/security/limits.conf

加上：

1. \* soft nofile 102400
2. \* hard nofile 409600
3. $cd /etc/pam.d/
4. $sudo vi login

添加session required /lib/security/pam\_limits.so

**7、问题：Too many fetch-failures**

原因:

出现这个问题主要是结点间的连通不够全面。

解决：

1) 检查/etc/hosts

要求本机ip对应服务器名

要求要包含所有的服务器ip+服务器名

2) 检查 .ssh/authorized\_keys

要求包含所有服务器(包括其自身)的public key

**8、问题：处理速度特别的慢 出现map很快 但是reduce很慢 而且反复出现 reduce=0%**

解决:

修改 conf/hadoop-env.sh 中的export HADOOP\_HEAPSIZE=4000

**9、问题：能够启动datanode，但无法访问，也无法结束的错误**

原因：

可能原因很多，需要查看日志定位，可能的原因是namespaceId不一致

解决：

（1）删除namenode的本地文件夹：${hadoop}/namedir/\*

（2）删除datanode的本地文件夹：${hadoop}/datadir/\*

（3）重新格式化

（4）注意：删除是个很危险的动作，不能确认的情况下不能删除!!做好删除的文件等通通备份!!

**10、问题：java.io.IOException: Could not obtain block: blk\_194219614024901469\_1100 file=/user/hive/warehouse/src\_20090724\_log/src\_20090724\_log**

原因：

出现这种情况大多是结点断了，没有连接上。

解决：

检查datanode是否有宕机情况，恢复

**11、问题：java.lang.OutOfMemoryError: Java heap space**

原因：

出现这种异常，明显是jvm内存不够得原因，要修改所有的datanode的jvm内存大小。

解决：

执行命令

1. Java -Xms1024m -Xmx4096m

一般jvm的最大内存使用应该为总内存大小的一半，我们使用的8G内存，所以设置为4096m，这一值可能依旧不是最优的值。

**12、问题：单个node新加硬盘**

原因：

datanode的硬盘不够用了

解决：

（1）修改需要新加硬盘的node的dfs.data.dir，用逗号分隔新、旧文件目录

（2）重启dfs

**13、问题：hdfs的namenode内存飙升，不够用**

原因：

可能是上传了过多的小文件

解决：

用命令合并HDFS小文件

1. hadoop fs -getmerge **<src>** **<dest>**

**14、问题：hadoop OutOfMemoryError**

解决：

1. **<property>**
2. **<name>**mapred.child.java.opts**</name>**
3. **<value>**-Xmx800M -server**</value>**
4. **</property>**

With the right JVM size in your hadoop-site.xml , you will have to copy this

to all mapred nodes and restart the cluster.

或者：

1. hadoop jar jarfile [main class] -D mapred.child.java.opts=-Xmx800M

**15、问题：Hadoop name -format后Incompatible namespaceIDS 错误解决办法**

Hadoop集群在namenode格式化(bin/hadoop namenode -format)后重启集群会出现如下错误：

Incompatible namespaceIDS in ... :namenode namespaceID = ... ,datanode namespaceID=...

原因：

是格式化namenode后会重新创建一个新的namespaceID,以至于和datanode上原有的不一致。

解决：

（1）删除datanode dfs.data.dir目录(默认为tmp/dfs/data)下的数据文件，

（2）hadoop namenode -format

（3）修改/home/hadoop/data/current/VERSION 文件，把namespaceID修成与namenode上相同即可(log错误里会有提示)

另外的解决方案

（1）查看集群的所有的namespaceid

namenode：${hadoop}/namenode/current/VERSION

datanode：${hadoop}/data/current/VERSION

（2）找出和集群namespaceid不一样的，改成一样

**16、问题：hadoop：RemoteException**

启动Hadoop集群测试HBase时候，发现三台DataNode只启动成功了两台，未启动成功的那一台日志中出现了下列异常：

1. 2012-09-07 23:58:51,240 WARN org.apache.hadoop.hdfs.server.datanode.DataNode:
2. DataNode is shutting down: org.apache.hadoop.ipc.RemoteException: org.apache.hadoop.hdfs.protocol.
3. UnregisteredDatanodeException: Data node 192.168.100.11:50010 is attempting to report storage ID DS-1282452139-218.196.207.181-50010-1344220553439.
4. Node 192.168.100.12:50010 is expected to serve this storage.
5. at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getDatanode(FSNamesystem.java:4608)
6. at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.processReport(FSNamesystem.java:3460)
7. at org.apache.hadoop.hdfs.server.namenode.NameNode.blockReport(NameNode.java:1001)
8. at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
9. at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
10. at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
11. at java.lang.reflect.Method.invoke(Method.java:601)
12. at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:563)
13. at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:1388)
14. at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:1384)
15. at java.security.AccessController.doPrivileged(Native Method)
16. at javax.security.auth.Subject.doAs(Subject.java:415)
17. at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1121)
18. at org.apache.hadoop.ipc.Server$Handler.run(Server.java:1382)
20. at org.apache.hadoop.ipc.Client.call(Client.java:1070)
21. at org.apache.hadoop.ipc.RPC$Invoker.invoke(RPC.java:225)
22. at $Proxy5.blockReport(Unknown Source)
23. at org.apache.hadoop.hdfs.server.datanode.DataNode.offerService(DataNode.java:958)
24. at org.apache.hadoop.hdfs.server.datanode.DataNode.run(DataNode.java:1458)
25. at java.lang.Thread.run(Thread.java:722)

原因：

两台DataNode的storageID出现了冲突，应该是因为我直接备份安装的原因吧。

解决：

直接将出现异常的那台机器的data目录删除!

**17、问题：namenode无法启动，不报错**

原因：

可能原因是：之前用root用户启动过，导致current文件夹的权限和所属用户更改了，需要更改回来

解决：

current文件夹位于hadoop安装目录同级目录的tmp/dfs/namesecondary

**18、问题：WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platfo**

1. [root@db96 hadoop]# hadoop dfs -put ./in
2. DEPRECATED: Use of this script to execute hdfs command is deprecated.
3. Instead use the hdfs command for it.
4. 14/07/17 17:07:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
5. put: `./in': No such file or directory

原因：

查看本地文件：

1. [root@db96 hadoop]# file /usr/local/hadoop/lib/native/libhadoop.so.1.0.0
2. /usr/local/hadoop/lib/native/libhadoop.so.1.0.0: ELF 32-bit LSB shared object,
3. Intel 80386, version 1 (SYSV), dynamically linked, not stripped

是32位的hadoop，安装在了64位的linux系统上。lib包编译环境不一样，所以不能使用。

解决：

重新编译hadoop.就是重新编译hadoop软件。 (本例文是在从库db99上编译。你也可以在master db96上编译 //只要机器的环境一直。)

参考手动编译步骤：64位的linux装的hadoop是32位的，需要手工编译

http://blog.csdn.net/wulantian

**19、Hadoop 报错be replicated to 0 nodes, instead of 1**

问题：

Hadoop 报错be replicated to 0 nodes, instead of 1

原因

（1）namespaceid不相同

（2）没有足够的硬盘

解决

（1）停止datanode

（2）删除datadir下所有数据。

（3）重启datanode

**20、The ratio of reported blocks 0.0000 has not reached the threshold 0.9990. Safe m**

问题：

Hadoop集群启动的时候一切正常，但一直处于safemode，只能读不能写，这种时候应该查看namenode的logs，当然这可能会出现不同的情况... ：

1. org.apache.hadoop.hdfs.server.namenode.SafeModeException: Cannot delete /home/hadoop/tmp/mapred/system. Name node is in safe mode.
2. The ratio of reported blocks 0.0000 has not reached the threshold 0.9990. Safe mode will be turned off automatically.

原因：

由日志可以看出无法删除/home/hadoop/tmp/mapred/system.(其实这只是一种假象，往往我们会去纠结于这个目录，其实不然)

解决：

（1）：终极办法强制退出安全模式(safemode)

hadoop dfsadmin -safemode leave

这种方式虽然快，但会有遗留问题，我在用habse的时候就遇到过，很麻烦,然后你就用“hadoop fsck /”工具慢慢恢复吧。

（2）：删除namenode下/home/hadoop/tmp下的所有文件，重新format，当然这种方式非常暴力，因为你的数据完全木有了

（3）：参考源码可发现这个错误是在检查file的时候抛出来的，基本也就是file的block丢失、错误等原因造成的。

这种情况在副本数为1的情况下会很棘手，其他的时候hadoop基本能自行解决，错误数很多的情况下就会一直处于safemode下，当然你关于集群修改配置文件后的分发，本人写了一个配置文件分发工具可以强制离开安全模式，先保证正常读写，然后再启用“hadoop fsck /”工具慢慢修复。

**21、Access denied for user 'root'@'hadoop1master' (using password: YES)**

问题：

Access denied for user 'root'@'hadoop1master' (using password: YES)

原因：

没有除本地用户的其他用户远程连接

解决：

修改mysql表，将localhost修改为%

**22、运行本地的wordcount报错**

(1)此时使用Run as->Java Application运行，会报如下类似错误：

1. 2015-01-22 15:31:47,782 [main] WARN org.apache.hadoop.util.NativeCodeLoader (NativeCodeLoader.java:62) - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2. 2015-01-22 15:31:47,793 [main] ERROR org.apache.hadoop.util.Shell (Shell.java:373) - Failed to locate the winutils binary in the hadoop binary path
3. java.io.IOException: Could not locate executable D:\hbl\_study\hadoop2\hadoop-2.6.0\bin\winutils.exe in the Hadoop binaries.
4. at org.apache.hadoop.util.Shell.getQualifiedBinPath(Shell.java:355)
5. at org.apache.hadoop.util.Shell.getWinUtilsPath(Shell.java:370)
6. ......

该错误是找不到winutils.exe，需要将winutils.exe拷贝到hadoop2.6.0/bin目录下。

(2)再次运行报错类似：

1. Exception in thread "main" java.lang.UnsatisfiedLinkError: org.apache.hadoop.io.nativeio.NativeIO$Windows.access0(Ljava/lang/String;I)Z
2. at org.apache.hadoop.io.nativeio.NativeIO$Windows.access0(Native Method)
3. at org.apache.hadoop.io.nativeio.NativeIO$Windows.access(NativeIO.java:557)
4. at org.apache.hadoop.fs.FileUtil.canRead(FileUtil.java:977)
5. ......

该错误是缺少hadoop.dll(hadoop2.6.0编译的版本)文件，需要将hadoop.dll拷贝到hadoop2.6.0/bin目录下。

再次运行没有报错。

**23、在eclipse中运行hdfs的api报错**

问题：

运行api的时候报了权限问题，使用的用户是hadoop，而我们想使用root用户

原因：

配置环境变量中设置了HADOOP\_USER\_NAME=hadoop或者在run configuration中设置的-DHADOOP\_USER\_NAME=hadoop

解决：

将配置环境变量中设置成HADOOP\_USER\_NAME=root或者在run configuration中设置的-DHADOOP\_USER\_NAME=root

**24、org.apache.hadoop.dfs.SafeModeException:Name node is in safe mode安全模式**

运行hadoop程序时，有时候会报以下错误：

1. org.apache.hadoop.dfs.SafeModeException: Cannot delete /user/hadoop/input. Name node is in safe mode

分析问题：

Hadoop的NameNode处在安全模式下。

那什么是Hadoop的安全模式呢?

在分布式文件系统启动的时候，开始的时候会有安全模式，当分布式文件系统处于安全模式的情况下，文件系统中的内容不允许修改也不允许删除，直到安全模式结束。

安全模式主要是为了系统启动的时候检查各个DataNode上数据块的有效性，同时根据策略必要的复制或者删除部分数据块。

运行期通过命令也可以进入安全模式。在实践过程中，系统启动的时候去修改和删除文件也会有安全模式不允许修改的出错提示，只需要等待一会儿即可。

现在就清楚了，那现在要解决这个问题，我想让Hadoop不处在safe mode模式下，能不能不用等，直接解决呢?

答案是可以的，只要在Hadoop的目录下输入：

解决方法：

1. bin/hadoop dfsadmin -safemode leave

也就是关闭Hadoop的安全模式，这样问题就解决了。

**25、用java -jar执行hadoop的job报错**

问题：

用java -jar执行hadoop的jar包时报错

1. [root@centos ~]# java -jar test04.jar
2. Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/hadoop/conf/Configuration
3. at cn.hc.hadoop.text.MyJob.main(MyJob.java:28)
4. Caused by: java.lang.ClassNotFoundException: org.apache.hadoop.conf.Configuration
5. at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
6. at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
7. at java.security.AccessController.doPrivileged(Native Method)
8. at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
9. at java.lang.ClassLoader.loadClass(ClassLoader.java:425)

原因：

用hadoop的mapreduce变成，在执行的时候需要依赖hadoop的大部分依赖，所以上述错误是缺少hadoop的依赖包

解决：

（1）建议使用hadoop -jar 执行job

（2）如果使用java -jar，需要使用java -cp 把hadoop依赖的所有jar拼接到路径里面去

（3）如果使用java -jar，另一种是在打包的时候把hadoop依赖的jar一起打包进去

**26、windows平台调用yarn报错：no job control一般解决方法**

在使用windows调用Hadoop yarn平台的时候，一般都会遇到如下的错误：

1. WARN org.apache.hadoop.yarn.server.nodemanager.DefaultContainerExecutor: Exception from container-launch with container ID: container\_1401177251807\_0034\_01\_000001 and exit code: 1
2. org.apache.hadoop.util.Shell$ExitCodeException: /bin/bash: line 0: fg: no job control
3. at org.apache.hadoop.util.Shell.runCommand(Shell.java:505)
4. at org.apache.hadoop.util.Shell.run(Shell.java:418)
5. at org.apache.hadoop.util.Shell$ShellCommandExecutor.execute(Shell.java:650)
6. at org.apache.hadoop.yarn.server.nodemanager.DefaultContainerExecutor.launchContainer(DefaultContainerExecutor.java:195)
7. at org.apache.hadoop.yarn.server.nodemanager.containermanager.launcher.ContainerLaunch.call(ContainerLaunch.java:300)
8. at org.apache.hadoop.yarn.server.nodemanager.containermanager.launcher.ContainerLaunch.call(ContainerLaunch.java:81)
9. at java.util.concurrent.FutureTask.run(FutureTask.java:262)
10. at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
11. at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
12. at java.lang.Thread.run(Thread.java:744)

（1）首先这个问题是在windows的eclipse作为客户端提交任务到linux Hadoop集群才会出现的问题，如果是linux的eclipse提交任务到linux Hadoop集群则不会出现这样的问题。那么一个很直观的想法就是同时使用两个客户端运行一个任务，然后每个步骤都调试，来确定其中的不同点。这么做，肯定是可以的。但是这么做肯定也是比较费时的(而且还要自己在一个linux上装个eclipse，麻烦);

（2）按照1的做法，一般就可以看到有两点的不同，一个是java命令的不同，还有一个就是classpath的不同。先说下断点的地方：

1)java命令的断点：

YarnRunner.java的这一行：

1. // Setup the command to run the AM
2. List**<String>** vargs = new ArrayList**<String>**(8);
3. vargs.add(Environment.JAVA\_HOME.$() + "/bin/java");

这里打上断点后，然后运行到下面这一行，就可以看到vargs是如下的样子(或者看vargsFinal这个变量)：

1. [%JAVA\_HOME%, -Dlog4j.configuration=container-log4j.properties, -Dyarn.app.container.log.dir=**<LOG\_DIR>**, -Dyarn.app.container.log.filesize=0, -Dhadoop.root.logger=INFO,CLA, , -Xmx1024m, org.apache.hadoop.mapreduce.v2.app.MRAppMaster, 1**><LOG\_DIR>**/stdout, 2**><LOG\_DIR>**/stderr, null, null]

2)classpath的断点：YarnRunner.java中，查看environment的值，可以看到起值为：

1. {CLASSPATH=%PWD%;$HADOOP\_CONF\_DIR;$HADOOP\_COMMON\_HOME/\*;$HADOOP\_COMMON\_HOME/lib/\*;$HADOOP\_HDFS\_HOME/\*;$HADOOP\_HDFS\_HOME/lib/\*;$HADOOP\_MAPRED\_HOME/\*;$HADOOP\_MAPRED\_HOME/lib/\*;$HADOOP\_YARN\_HOME/\*;$HADOOP\_YARN\_HOME/lib/\*;%HADOOP\_MAPRED\_HOME%\share\hadoop\mapreduce\\*;%HADOOP\_MAPRED\_HOME%\share\hadoop\mapreduce\lib\\*;job.jar/job.jar;job.jar/classes/;job.jar/lib/\*;%PWD%/\*}

（3）看到2中的两个值就可以确定，windows和linux的不同之处了，主要有两个：

* %%和$的区别;
* 正反斜杠的区别(这个好像不区别也行);

（4）看出上面两个地方的区别后，如果直接把这两个值改为：

1. [$JAVA\_HOME/bin/java -Dlog4j.configuration=container-log4j.properties -Dyarn.app.container.log.dir=**<LOG\_DIR>** -Dyarn.app.container.log.filesize=0 -Dhadoop.root.logger=INFO,CLA -Xmx1024m org.apache.hadoop.mapreduce.v2.app.MRAppMaster 1**><LOG\_DIR>**/stdout 2**><LOG\_DIR>**/stderr ]

和

1. {CLASSPATH=$PWD:$HADOOP\_CONF\_DIR:$HADOOP\_COMMON\_HOME/\*:$HADOOP\_COMMON\_HOME/lib/\*:$HADOOP\_HDFS\_HOME/\*:$HADOOP\_HDFS\_HOME/lib/\*:$HADOOP\_MAPRED\_HOME/\*:$HADOOP\_MAPRED\_HOME/lib/\*:$HADOOP\_YARN\_HOME/\*:$HADOOP\_YARN\_HOME/lib/\*:$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/\*:$HADOOP\_MAPRED\_HOME/share/hadoop/mapreduce/lib/\*:job.jar/job.jar:job.jar/classes/:job.jar/lib/\*:$PWD/\*}

那么应该是可以运行的;

（5）怎么改呢?

1)在我们的工程中新建一个YarnRunner类，该类与源码的YarnRunner类一模一样(包路径，代码内容都一样);

2)把下面替换为(这里默认Hadoop 集群是在linux环境下的)：

即把

1. vargs.add(Environment.JAVA\_HOME.$() + "/bin/java");

替换为

1. vargs.add("$JAVA\_HOME/bin/java");

3)添加：

1. replaceEnvironment(environment);

这个方法放在最后面，为：

1. private void replaceEnvironment(Map**<String**, String**>** environment) {
2. String tmpClassPath = environment.get("CLASSPATH");
3. tmpClassPathtmpClassPath=tmpClassPath.replaceAll(";", ":");
4. tmpClassPathtmpClassPath=tmpClassPath.replaceAll("%PWD%", "\\$PWD");
5. tmpClassPathtmpClassPath=tmpClassPath.replaceAll("%HADOOP\_MAPRED\_HOME%", "\\$HADOOP\_MAPRED\_HOME");
6. tmpClassPathtmpClassPath= tmpClassPath.replaceAll("\\\\", "/" );
7. environment.put("CLASSPATH",tmpClassPath);
8. }

这样替换完成后，在windows的eclipse中向linux Hadoop集群中提交任务就可以执行了。

**27、运行mr程序报UnsatisfiedLinkError:nativeio.NativeIO$Windows.access0(Ljava/lang/String**

有的同学在开发工具里面跑mr程序时候，遇到了下

Exception in thread "main" java.lang.UnsatisfiedLinkError: org.apache.hadoop.io.nativeio.NativeIO$Windows.access0(Ljava/lang/String;I)Z

一般这个问题是由本地hadoop环境变量照成的。

需要设置hadoop\_home变量的值。

注意hadoop安装目录下，bin目录中缺少hadoop.dll和winutils.exe等动态库。还要增加bin目录到path路径。

另外编辑器需要添加hadoop环境 还要注意jdk等是否正确安装。

**28、hdfs API文件找不到异常**

问题：

在使用hdfs的copyfromlocal上传文件到hdfs时，爆出本地文件找不到异常，但是查看本地文件确实存在

原因：

windows设置了隐藏已知文件的扩展名功能，导致上传的文件没有写扩展名

解决：

在上传文件的地方添加上扩展名即可。

**29、hadoop节点启动异常**

问题：

在执行hadoop-deamon.sh start xxx时报错

错误：找不到或无法加载主类

原因：

启动的时候，节点名写错了

解决：

修改名字，名字有， namenode datanode等

**30、UnsatisfiedLinkError: org.apache.hadoop.io.nativeio.NativeIO$Windows.access0(Ljava/lang/Strin**

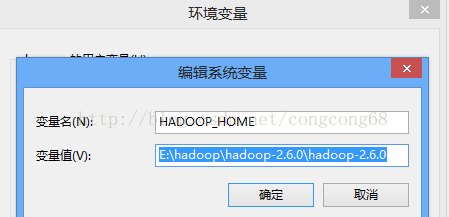
好多同学在windows平台Eclipse上跑MR本地程序的时候，都报了如下的异常：

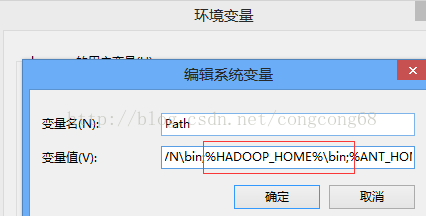
Exception in thread "main" java.lang.UnsatisfiedLinkError: org.apache.hadoop.io.nativeio.NativeIO$Windows.access0(Ljava/lang/String;I)Z

解决步骤如下：

一、检查自己是否配置环境变量HADOOP\_HOME 和path，如下所示：

注意要配置Windows版编译的hadoop。



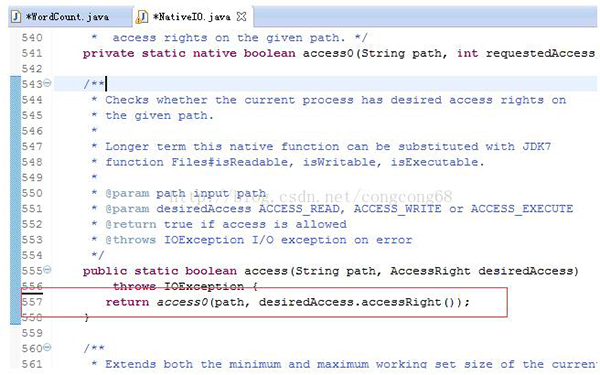


二、检查C:\Windows\System32下是否缺少hadoop.dll,如果缺少的话把这个文件拷贝到C:\Windows\System32下面即可

然后重启电脑，也许还没那么简单，还是出现这样的问题。

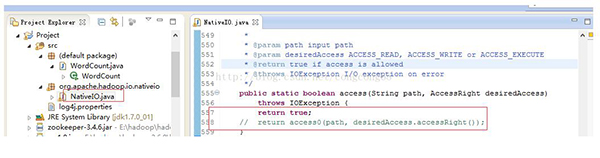
我们再继续分析：

我们在出现错误的的atorg.apache.hadoop.io.nativeio.NativeIO$Windows.access(NativeIO.java:557)我们来看这个类NativeIO的557行，如图所示：

[](http://s3.51cto.com/wyfs02/M02/8B/82/wKiom1hP1U7RyIOiAALoOJCn9_w772.jpg)

Windows的唯一方法用于检查当前进程的请求，在给定的路径的访问权限，所以我们先给以能进行访问，我们自己先修改源代码，return true 时允许访问。

我们下载对应hadoop源代码，hadoop-2.6.0-src.tar.gz解压，hadoop-common-project\hadoop-common\src\main\java\org\apache\hadoop\io\nativeio下NativeIO.java 复制到对应的Eclipse的project，然后修改557行为return true如图所示：

[](http://s5.51cto.com/wyfs02/M02/8B/7F/wKioL1hP1W-wL4c-AAEtBZx_nmI662.jpg)

这样通过终极解救大法。我们就可以愉快的在Windows上跑MR了。

**31、Error: java.io.IOException: Unable to initialize any output collector**

错误如下所示：

1. Error: java.io.IOException: Unable to initialize any output collector
2. at org.apache.hadoop.mapred.MapTask.createSortingCollector(MapTask.java:412)
3. at org.apache.hadoop.mapred.MapTask.access$100(MapTask.java:81)
4. at org.apache.hadoop.mapred.MapTask$NewOutputCollector.**<init>**(MapTask.java:695)
5. at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:767)
6. at org.apache.hadoop.mapred.MapTask.run(MapTask.java:341)
7. at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:163)
8. at java.security.AccessController.doPrivileged(Native Method)
9. at javax.security.auth.Subject.doAs(Subject.java:415)
10. at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1656)
11. at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:158)

解决思路：

错误都是出现在Mapper的task里面。错误说明输出的类型错误。请优先排查自己mr程序中的输出类型跟你在job中设置的输出类型是否一致。

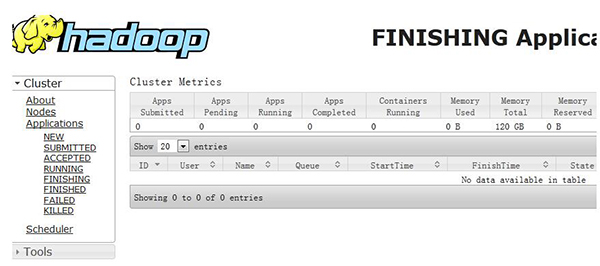
就是一致的情况下，要排除自己导入的包是否正确，比如import里面不是import的org.apache.hadoop.io.Text类，而是另外一个不知所云的包下面的Text。

最后还有一种情况，如果是自定义的情况下，请回头检查自己写的Key类，发现有没有显式地定义类的构造函数。

一般在添加无参数的构造函数后，问题即可解决。因为必须有默认的构造器皿，这样Mapreduce方法才能创建对象，然后通过readFields方法从序列化的数据流中读出进行赋值。

**32、hadoop 8088 看不到mapreduce 任务的执行状态，无数据显示。**

错误页面显示如下：

[](http://s1.51cto.com/wyfs02/M01/8B/83/wKiom1hP1dPwI_uiAAE8XU784Dc709.jpg)

解决方法：

（1）首先检查自己的集群中配置$HADOOP\_HOME/conf/mapred-site.xml是否存在。

其中的mapreduce.framework.name是否配置。

（2）如果还不行的话，请在$HADOOP\_HOME/conf/mapred-site.xml中原来的配置文件基础之上再添加下面

1. **<property>**
2. **<name>**mapreduce.jobhistory.address**</name>**
3. **<value>**master:10020**</value>**
4. **</property>**
5. **<property>**
6. **<name>**mapreduce.jobhistory.webapp.address**</name>**
7. **<value>**master:19888**</value>**
8. **</property>**

**33、org.apache.hadoop.fs.ChecksumException: Checksum error: file:.....**

有的同学在练习中遇到了这样的错误：

1. [root@hadoop01 zookeeper-3.4.5]# hadoop fs -put /root/xx.oo /wc/input
2. 16/09/23 19:37:17 INFO fs.FSInputChecker: Found checksum error: b[0, 53]=69206c6f766520796f750a77656c636f6d6520796f750a686f772061726520796f750a7468697320697320796f7520686f6d650a0a
3. org.apache.hadoop.fs.ChecksumException: Checksum error: file:/root/xx.oo at 0 exp: -1023732142 got: -897233467
4. at org.apache.hadoop.fs.FSInputChecker.verifySums(FSInputChecker.java:323)
5. at org.apache.hadoop.fs.FSInputChecker.readChecksumChunk(FSInputChecker.java:279)
6. at org.apache.hadoop.fs.FSInputChecker.read1(FSInputChecker.java:228)
7. at org.apache.hadoop.fs.FSInputChecker.read(FSInputChecker.java:196)
8. at java.io.DataInputStream.read(DataInputStream.java:100)
9. at org.apache.hadoop.io.IOUtils.copyBytes(IOUtils.java:78)
10. at org.apache.hadoop.io.IOUtils.copyBytes(IOUtils.java:52)
11. at org.apache.hadoop.io.IOUtils.copyBytes(IOUtils.java:112)
12. at org.apache.hadoop.fs.shell.CommandWithDestination$TargetFileSystem.writeStreamToFile(CommandWithDestination.java:466)
13. at org.apache.hadoop.fs.shell.CommandWithDestination.copyStreamToTarget(CommandWithDestination.java:391)
14. at org.apache.hadoop.fs.shell.CommandWithDestination.copyFileToTarget(CommandWithDestination.java:328)
15. at org.apache.hadoop.fs.shell.CommandWithDestination.processPath(CommandWithDestination.java:263)
16. at org.apache.hadoop.fs.shell.CommandWithDestination.processPath(CommandWithDestination.java:248)
17. at org.apache.hadoop.fs.shell.Command.processPaths(Command.java:306)
18. at org.apache.hadoop.fs.shell.Command.processPathArgument(Command.java:278)
19. at org.apache.hadoop.fs.shell.CommandWithDestination.processPathArgument(CommandWithDestination.java:243)
20. at org.apache.hadoop.fs.shell.Command.processArgument(Command.java:260)
21. at org.apache.hadoop.fs.shell.Command.processArguments(Command.java:244)
22. at org.apache.hadoop.fs.shell.CommandWithDestination.processArguments(CommandWithDestination.java:220)
23. at org.apache.hadoop.fs.shell.CopyCommands$Put.processArguments(CopyCommands.java:267)
24. at org.apache.hadoop.fs.shell.Command.processRawArguments(Command.java:190)
25. at org.apache.hadoop.fs.shell.Command.run(Command.java:154)
26. at org.apache.hadoop.fs.FsShell.run(FsShell.java:287)
27. at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:70)
28. at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:84)
29. at org.apache.hadoop.fs.FsShell.main(FsShell.java:340)
30. put: Checksum error: file:/root/xx.oo at 0 exp: -1023732142 got: -897233467

错误原因：

启动任务的命令中包含一个参数“-files tb\_steps\_url\_path\_dim.txt”

Hadoop客户端需要将机器本地磁盘中的tb\_steps\_url\_path\_dim.txt文件上传到DFS中。

在上传的过程中，Hadoop将通过FSInputChecker判断需要上传的文件是否存在进行校验的crc文件，即.tb\_steps\_url\_path\_dim.txt.crc，如果存在crc文件，将会对其内容一致性进行校验，如果校验 失败，则停止上传该文件。最终导致整个MR任务无法执行。

crc文件来源

DFS命令：hadoop fs -getmerge srcDir destFile

这类命令在执行的时候，会将srcDir目录下的所有文件合并成一个文件，保存在destFile中，同时会在本地磁盘生成一个. destFile.crc的校验文件。

DFS命令：hadoop fs -get -crc src dest

这类命令在执行的时候，会将src文件，保存在dest中，同时会在本地磁盘生成一个. dest.crc的校验文件。

如何避免

在使用hadoop fs -getmerge srcDir destFile命令时，本地磁盘一定会(没有参数可以关闭)生成相应的.crc文件。

所以如果需要修改getmerge获取的文件的内容，再次上传到DFS时，可以采取以下2种策略进行规避：

（1）删除.crc文件 ls -a 查看，同级目录下有个crc校验文件，删除即可。

（2）将getmerge获取的文件修改后重新命名，如使用mv操作，再次上传到DFS中。

**34、security.AccessControlException: Access denied for user sunqw. Superuser privilege is required。**

问题：

Windows|Eclipse 运行HDFS程序之后，报：

1. org.apache.hadoop.security.AccessControlException: Permission denied: user=sunqw, access=WRITE, inode="":hadoop:supergroup:rwxr-xr-x

或者

Windows|Eclipse 运行HDFS程序之后，报：

1. org.apache.hadoop.security.AccessControlException: Access denied for user sunqw. Superuser privilege is required

解决方法：

方式一：

在系统环境变量中增加HADOOP\_USER\_NAME，其值为root;

或者 通过java程序动态添加，如下：

?1System.setProperty("HADOOP\_USER\_NAME", "root");

方式二：

使用Eclipse在非hadoop运行的用户下进行写入hdfs文件系统中时，由于sunqw对"/"目录没有写入权限，所以导致异常的发生。解决方法即开放hadoop中的HDFS目录的权限，命令如下：hadoop fs -chmod 777 / 。

方式三：

修改hadoop的配置文件：conf/hdfs-core.xml，添加或者修改 dfs.permissions 的值为 false。

方式四：

将Eclipse所在机器的用户的名称修改为root，即与服务器上运行hadoop的用户一致。

【本文为51CTO专栏作者“王森丰”的原创稿件，转载请注明出处】

【编辑推荐】

1. [外媒速递：强化Windows 10以实现最大安全性的几个关键性角度](http://zhuanlan.51cto.com/art/201609/517974.htm)
2. [专家教你如何在SQL Server中通过PowerUpSQL获得Windows自动登录密码](http://zhuanlan.51cto.com/art/201610/518473.htm)
3. [外媒速递：Windows 10中的七项企业级移动管理功能](http://zhuanlan.51cto.com/art/201610/519623.htm)
4. [谷歌安全又来曝光Windows的0day漏洞了-这次依旧席卷所有windows系统](http://zhuanlan.51cto.com/art/201611/520809.htm)
5. [Apache Hadoop 3.0.0-alpha1主要改进](http://zhuanlan.51cto.com/art/201612/524872.htm)