Hive on Spark编程入门指南

先说明一下，这里说的Hive on Spark是Hive跑在Spark上，用的是Spark执行引擎，而不是MapReduce，和Hive on Tez的道理一样。

从Hive 1.1版本开始，Hive on Spark已经成为Hive代码的一部分了，并且在spark分支上面，可以看这里https://github.com/apache/hive/tree/spark，并会定期的移到master分支上面去。关于Hive on Spark的讨论和进度，可以看这里https://issues.apache.org/jira/browse/HIVE-7292。

要想在Hive中使用Spark执行引擎，第一步当前就是环境设置，我们需要在Hive启动的时候加载spark-assembly-1.4.1-hadoop2.2.0.jar，最简单的方法是把spark-assembly-1.4.1-hadoop2.2.0.jar包直接拷贝到$HIVE_HOME/lib目录下。

不过在Hive的官方文档上面还提供了两种加载Spark相关包的方法：

```
hive> set spark.home=/location/to/sparkHome;
```

或者在启动Hive之前加上环境变量

```
export SPARK_HOME=/home/iteblog/spark-1.4.1-bin-2.2.0
```

这两种方法我都试过，都出现了一下异常：

```
java.lang.NoClassDefFoundError: io/netty/channel/EventLoopGroup
 at org.apache.hive.spark.client.SparkClientFactory.initialize(SparkClientFactory.java:56)
 at org.apache.hadoop.hive.ql.exec.spark.session.SparkSessionManagerImpl.setup(SparkSessionManagerImpl.java:86)
 at org.apache.hadoop.hive.ql.exec.spark.session.SparkSessionManagerImpl.getSession(SparkSessionManagerImpl.java:102)
 at org.apache.hadoop.hive.ql.exec.spark.SparkUtilities.getSparkSession(SparkUtilities.java:112)
 at org.apache.hadoop.hive.ql.exec.spark.SparkTask.execute(SparkTask.java:101)
 at org.apache.hadoop.hive.ql.exec.Task.executeTask(Task.java:160)
 at org.apache.hadoop.hive.ql.exec.TaskRunner.runSequential(TaskRunner.java:88)
 at org.apache.hadoop.hive.ql.Driver.launchTask(Driver.java:1653)
 at org.apache.hadoop.hive.ql.Driver.execute(Driver.java:1412)
```

```
 at org.apache.hadoop.hive.ql.Driver.runInternal(Driver.java:1195)
 at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1059)
 at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1049)
 at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:213)
 at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:165)
 at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:376)
 at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:736)
 at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:681)
 at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:621)
 at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
 at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
 at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
 at java.lang.reflect.Method.invoke(Method.java:606)
 at org.apache.hadoop.util.RunJar.main(RunJar.java:212)
Caused by: java.lang.ClassNotFoundException: io.netty.channel.EventLoopGroup
 at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
 at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
 at java.security.AccessController.doPrivileged(Native Method)
 at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
 at java.lang.ClassLoader.loadClass(ClassLoader.java:425)
 at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
 at java.lang.ClassLoader.loadClass(ClassLoader.java:358)
 ... 23 more
FAILED: Execution Error, return code -101 from org.apache.hadoop.hive.ql.exec.spark.SparkTa
sk. io/netty/channel/EventLoopGroup
```

还有一点需要注意，Hive on Spark模式用到的Spark assembly包必须是没有用-phive参数编译的，否则出现以下异常：

```
15/08/28 11:00:32 ERROR yarn.ApplicationMaster: User class threw exception: java.lang.NoSuc
hFieldError: SPARK_RPC_CLIENT_CONNECT_TIMEOUT
java.lang.NoSuchFieldError: SPARK_RPC_CLIENT_CONNECT_TIMEOUT
 at org.apache.hive.spark.client.rpc.RpcConfiguration.<clinit>(RpcConfiguration.java:46)
 at org.apache.hive.spark.client.RemoteDriver.<init>(RemoteDriver.java:146)
 at org.apache.hive.spark.client.RemoteDriver.main(RemoteDriver.java:556)
 at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
 at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
 at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
 at java.lang.reflect.Method.invoke(Method.java:606)
 at org.apache.spark.deploy.yarn.ApplicationMaster$$anon$2.run(ApplicationMaster.scala:483
)
15/08/28 11:00:32 INFO yarn.ApplicationMaster: Final app status: FAILED, exitCode: 15, (reaso
n: User
```

```
 class threw exception: java.lang.NoSuchFieldError: SPARK_RPC_CLIENT_CONNECT_TIMEOUT)
15/08/28 11:00:42 ERROR yarn.ApplicationMaster: SparkContext did not initialize after waiting for
100000 ms. Please check earlier log output for errors. Failing the application.
15/08/28 11:00:42 INFO yarn.ApplicationMaster: Unregistering ApplicationMaster with FAILED
(diag message: User class threw exception: java.lang.NoSuchFieldError: SPARK_RPC_CLIENT_CONNECT_TIMEOUT)
15/08/28 11:00:42 INFO yarn.ApplicationMaster: Deleting staging directory .sparkStaging/application_1440152921247_0079
15/08/28 11:00:42 INFO util.Utils: Shutdown hook called
```

在Hive中配置好环境之后，我们可以启动Hive了，如下：

```
[iteblog@www.iteblog.com hive]$ sudo -uiteblog bin/hive
hive>
```

这些和正常启动Hive没什么区别。下面才是最重要的，我们需要在Hive中设置任务的执行引擎，可以通过hive.execution.engine参数进行设置。目前Hive支持三种执行引擎：mr、tez、spark。因为我们需要使用Spark执行引擎，所以需要将hive.execution.engine设置为spark，具体如下：

```
hive> set hive.execution.engine=spark;
```

其实，我们还可以在Hive中设置很多关于Spark的配置，诸如spark.eventLog.enabled、spark.executor.memory以及spark.executor.instances等。如下：

```
hive> set spark.eventLog.enabled=true;
hive> set spark.eventLog.dir=hdfs://iteblog/spark-jobs/eventLog;
hive> set spark.executor.memory=4g;
hive> set spark.executor.cores=2;
hive> set spark.executor.instances=40;
hive> set spark.serializer=org.apache.spark.serializer.KryoSerializer;
```

这样设置是不是很麻烦？我们可以在$HIVE_HOME/conf里面创建spark-defaults.conf文件，然后加上需要设置的配置，最后设置到Hive启动的classpath环境中即可。当然，你完全可以将这些关于Spark的设置弄到hive-site.xml文件中，这样更方便。

设置好了Spark引擎之后，我们来执行一个SQL：

```
hive> select count(*) from ewaplog limit 10;
Query ID = datadev_20150828112049_3a5a6d78-83d9-4cea-b5e2-6bd2d8836d01
Total jobs = 1
Launching Job 1 out of 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=</number><number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=</number><number>
Starting Spark Job = d5a8ff30-9fb5-401c-beca-1043e38f9362

Query Hive on Spark job[0] stages:
0
1

Status: Running (Hive on Spark job[0])
Job Progress Format
CurrentTime StageId_StageAttemptId: SucceededTasksCount(+RunningTasksCount-
FailedTasksCount)/TotalTasksCount [StageCost]
2015-08-28 11:21:35,736 Stage-0_0: 0/5554 Stage-1_0: 0/1
2015-08-28 11:21:36,743 Stage-0_0: 0(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:39,759 Stage-0_0: 6(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:40,765 Stage-0_0: 27(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:41,773 Stage-0_0: 86(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:42,793 Stage-0_0: 142(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:43,845 Stage-0_0: 197(+51)/5554 Stage-1_0: 0/1
2015-08-28 11:21:44,851 Stage-0_0: 256(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:45,857 Stage-0_0: 298(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:46,862 Stage-0_0: 359(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:47,868 Stage-0_0: 427(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:48,874 Stage-0_0: 486(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:49,879 Stage-0_0: 551(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:50,883 Stage-0_0: 610(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:51,888 Stage-0_0: 672(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:52,895 Stage-0_0: 737(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:53,903 Stage-0_0: 804(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:54,908 Stage-0_0: 859(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:55,919 Stage-0_0: 913(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:56,942 Stage-0_0: 974(+50)/5554 Stage-1_0: 0/1
```

```
2015-08-28 11:21:57,947 Stage-0_0: 1032(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:58,953 Stage-0_0: 1094(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:21:59,961 Stage-0_0: 1159(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:00,971 Stage-0_0: 1223(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:01,988 Stage-0_0: 1281(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:02,993 Stage-0_0: 1341(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:03,997 Stage-0_0: 1400(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:05,003 Stage-0_0: 1449(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:06,016 Stage-0_0: 1506(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:07,027 Stage-0_0: 1571(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:08,032 Stage-0_0: 1634(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:09,038 Stage-0_0: 1691(+51)/5554 Stage-1_0: 0/1
2015-08-28 11:22:10,042 Stage-0_0: 1749(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:11,048 Stage-0_0: 1816(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:12,052 Stage-0_0: 1878(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:13,057 Stage-0_0: 1942(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:14,061 Stage-0_0: 2010(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:15,065 Stage-0_0: 2073(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:16,069 Stage-0_0: 2115(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:17,077 Stage-0_0: 2184(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:18,081 Stage-0_0: 2244(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:19,086 Stage-0_0: 2305(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:20,090 Stage-0_0: 2362(+50)/5554 Stage-1_0: 0/1
2015-08-28 11:22:21,094 Stage-0_0: 2421(+51)/5554 Stage-1_0: 0/1
2015-08-28 11:22:22,098 Stage-0_0: 2478(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:23,102 Stage-0_0: 2536(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:24,106 Stage-0_0: 2599(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:25,111 Stage-0_0: 2634(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:26,115 Stage-0_0: 2682(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:27,128 Stage-0_0: 2740(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:28,135 Stage-0_0: 2790(+47)/5554 Stage-1_0: 0/1
2015-08-28 11:22:29,140 Stage-0_0: 2847(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:30,144 Stage-0_0: 2889(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:31,148 Stage-0_0: 2950(+47)/5554 Stage-1_0: 0/1
2015-08-28 11:22:32,152 Stage-0_0: 3014(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:33,157 Stage-0_0: 3064(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:34,160 Stage-0_0: 3116(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:35,194 Stage-0_0: 3166(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:36,198 Stage-0_0: 3219(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:37,202 Stage-0_0: 3271(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:38,205 Stage-0_0: 3315(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:39,210 Stage-0_0: 3355(+47)/5554 Stage-1_0: 0/1
2015-08-28 11:22:40,214 Stage-0_0: 3424(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:41,217 Stage-0_0: 3481(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:42,221 Stage-0_0: 3528(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:43,224 Stage-0_0: 3585(+47)/5554 Stage-1_0: 0/1
```

```
2015-08-28 11:22:44,228 Stage-0_0: 3636(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:45,232 Stage-0_0: 3685(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:46,235 Stage-0_0: 3739(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:47,238 Stage-0_0: 3796(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:48,246 Stage-0_0: 3839(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:49,275 Stage-0_0: 3899(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:50,279 Stage-0_0: 3952(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:51,282 Stage-0_0: 4010(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:52,286 Stage-0_0: 4063(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:53,290 Stage-0_0: 4114(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:54,293 Stage-0_0: 4158(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:55,297 Stage-0_0: 4202(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:56,300 Stage-0_0: 4251(+47)/5554 Stage-1_0: 0/1
2015-08-28 11:22:57,304 Stage-0_0: 4310(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:58,307 Stage-0_0: 4358(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:22:59,312 Stage-0_0: 4395(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:00,323 Stage-0_0: 4444(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:01,326 Stage-0_0: 4500(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:02,348 Stage-0_0: 4555(+47)/5554 Stage-1_0: 0/1
2015-08-28 11:23:03,352 Stage-0_0: 4598(+47)/5554 Stage-1_0: 0/1
2015-08-28 11:23:04,355 Stage-0_0: 4648(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:05,358 Stage-0_0: 4684(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:06,361 Stage-0_0: 4745(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:07,364 Stage-0_0: 4803(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:08,376 Stage-0_0: 4847(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:09,379 Stage-0_0: 4907(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:10,389 Stage-0_0: 4959(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:11,397 Stage-0_0: 5012(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:12,400 Stage-0_0: 5063(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:13,403 Stage-0_0: 5110(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:14,407 Stage-0_0: 5163(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:15,410 Stage-0_0: 5219(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:16,416 Stage-0_0: 5263(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:17,419 Stage-0_0: 5312(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:18,423 Stage-0_0: 5350(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:19,429 Stage-0_0: 5397(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:20,432 Stage-0_0: 5453(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:21,435 Stage-0_0: 5506(+2)/5554 Stage-1_0: 0/1
2015-08-28 11:23:22,438 Stage-0_0: 5508(+0)/5554 Stage-1_0: 0/1
2015-08-28 11:23:24,468 Stage-0_0: 5508(+46)/5554 Stage-1_0: 0/1
2015-08-28 11:23:25,478 Stage-0_0: 5517(+37)/5554 Stage-1_0: 0/1
2015-08-28 11:23:26,484 Stage-0_0: 5526(+28)/5554 Stage-1_0: 0/1
2015-08-28 11:23:27,487 Stage-0_0: 5546(+8)/5554 Stage-1_0: 0/1
2015-08-28 11:23:28,491 Stage-0_0: 5554/5554 Finished Stage-1_0: 0(+1)/1
2015-08-28 11:23:30,497 Stage-0_0: 5554/5554 Finished Stage-1_0: 1/1 Finished
Status: Finished successfully in 151.80 seconds
```

OK
1251538181
Time taken: 161.475 seconds, Fetched: 1 row(s)

再来看看yarn WEB UI界面上的显示：

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1440152921247_0085	iteblog	Hive on Spark	SPARK	root.iteblog	2015/8/28 上午 11:35:02	N/A	RUNNING	UNDEFINED		ApplicationMaster

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

注意看里面的作业名称（Name）变成了Hive on Spark，而且应用类型（Application Type）变成了SPARK。

从计算速度来看，执行同一个SQL分别在MapReduce执行引擎和Spark执行引擎上运行，Spark执行引擎确实是要比MapReduce执行引擎快，不过就稳定性而已，Hive on Spark还在开发中，可能有一些地方不太稳定。

2016年5月6日更新：

1、本文上面的Hive on Spark的各组件版本是：Hive 0.12.0，Spark 1.4.1， Hadoop 2.2.0，如果大家在使用的过程中遇到各种异常，请先确认Hive的版本和Spark是否兼容！
2、如何确定Hive启动的时候已经加载了Spark相关的类包？一个简单的方法就是，在启动hive的时候，观察输出的日志是否有类似于SLF4J: Found binding in [jar:file:/home/q/spark/spark-1.5.1-bin-2.2.0/lib/spark-assembly-1.5.1-hadoop2.2.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]相关的信息。
3、如果出现类似于以下的异常信息，请确保你Spark相关类包在Hive启动classpath中。

```
Exception in thread "main" java.lang.NoClassDefFoundError: io/netty/util/concurrent/GenericFutureListener
  at org.apache.hive.spark.client.SparkClientFactory.initialize(SparkClientFactory.java:56)
  at org.apache.hadoop.hive.ql.exec.spark.session.SparkSessionManagerImpl.setup(SparkSessionManagerImpl.java:86)
  at org.apache.hadoop.hive.ql.exec.spark.session.SparkSessionManagerImpl.getSession(SparkSessionManagerImpl.java:102)
  at org.apache.hadoop.hive.ql.exec.spark.SparkUtilities.getSparkSession(SparkUtilities.java:112)
  at org.apache.hadoop.hive.ql.optimizer.spark.SetSparkReducerParallelism.process(SetSparkReducerParallelism.java:115)
  at org.apache.hadoop.hive.ql.lib.DefaultRuleDispatcher.dispatch(DefaultRuleDispatcher.java:90)
  at org.apache.hadoop.hive.ql.lib.DefaultGraphWalker.dispatchAndReturn(DefaultGraphWalke
```

```
r.java:95)
  at org.apache.hadoop.hive.ql.lib.DefaultGraphWalker.dispatch(DefaultGraphWalker.java:79)
  at org.apache.hadoop.hive.ql.lib.DefaultGraphWalker.walk(DefaultGraphWalker.java:133)
  at org.apache.hadoop.hive.ql.lib.DefaultGraphWalker.startWalking(DefaultGraphWalker.java:
110)
  at org.apache.hadoop.hive.ql.parse.spark.SparkCompiler.optimizeOperatorPlan(SparkCompil
er.java:128)
  at org.apache.hadoop.hive.ql.parse.TaskCompiler.compile(TaskCompiler.java:102)
  at org.apache.hadoop.hive.ql.parse.SemanticAnalyzer.analyzeInternal(SemanticAnalyzer.java:
10125)
  at org.apache.hadoop.hive.ql.parse.CalcitePlanner.analyzeInternal(CalcitePlanner.java:207)
  at org.apache.hadoop.hive.ql.parse.BaseSemanticAnalyzer.analyze(BaseSemanticAnalyzer.jav
a:227)
  at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:424)
  at org.apache.hadoop.hive.ql.Driver.compile(Driver.java:308)
  at org.apache.hadoop.hive.ql.Driver.compileInternal(Driver.java:1122)
  at org.apache.hadoop.hive.ql.Driver.runInternal(Driver.java:1170)
  at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1059)
  at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1049)
  at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:213)
  at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:165)
  at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:376)
  at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:736)
  at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:681)
  at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:621)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43
)
  at java.lang.reflect.Method.invoke(Method.java:606)
  at org.apache.hadoop.util.RunJar.main(RunJar.java:212)
Caused by: java.lang.ClassNotFoundException: io.netty.util.concurrent.GenericFutureListener
  at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
  at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
  at java.security.AccessController.doPrivileged(Native Method)
  at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
  at java.lang.ClassLoader.loadClass(ClassLoader.java:425)
  at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
  at java.lang.ClassLoader.loadClass(ClassLoader.java:358)
  ... 32 more
```

4、如果在使用Hive on Spark中遇到其他不确定的问题，最好打开Hive的debug日志，这里面会有很多详细的信息，具体如何配置请参见：《Hive日志调试》

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。
本文链接: 【】（）