# Hive-On-Spark

mapreduce -🡪 mapreduce 任务 运行在 yarn

hive -🡪 spark 任务 运行在spark的集群中

## HiveOnSpark简介

Hive On Spark （跟hive没太大的关系，就是使用了hive的标准（HQL， 元数据库、UDF、序列化、反序列化机制））

Hive原来的计算模型是MR,有点慢（将中间结果写入到HDFS中）

Hive On Spark 使用RDD（DataFrame），然后运行在spark 集群上

hive 元数据 mysql hive 数据 真正存储在hdfs中

真正要计算的数据是保存在HDFS中，mysql这个元数据库，保存的是hive表的描述信息，描述了有哪些database、table、以及表有多少列，每一列是什么类型，还要描述表的数据保存在hdfs的什么位置？

hive跟mysql的区别？

hive是一个数据仓库（存储数据并分析数据，分析数据仓库中的数据量很大，一般要分析很长的时间）

mysql是一个关系型数据库（关系型数据的增删改查（低延迟））

hive的元数据库中保存要计算的数据吗？

不保存，保存hive仓库的表、字段、等描述信息

真正要计算的数据保存在哪里了？

保存在HDFS中了

hive的元数据库的功能

建立了一种映射关系，执行HQL时，先到MySQL元数据库中查找描述信息，然后根据描述信息生成任务，然后将任务下发到spark集群中执行

hive on spark 使用的仅仅是hive的标准，规范，不需要有hive数据库一样可行。

hive : 元数据，是存放在mysql中，然后真正的数据是存放在hdfs中。

## 安装mysql

安装MySQL（hive的元数据库）

使用# yum -y install mysql mysql-server

启动服务

# service mysqld start

# 安全配置

# /usr/bin/mysql\_secure\_installation

连接

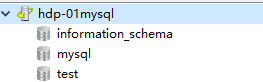
# mysql -uroot -p

授权root用户远程登录，并刷新权限

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;

FLUSH PRIVILEGES;

验证root用户的远程登录



到此为止，hive使用的元数据存放的mysql数据库，已经准备完毕。

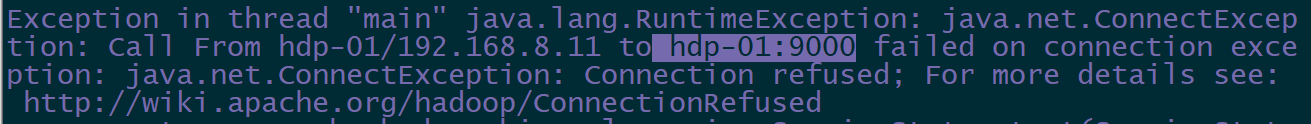
接下来该做何操作呢？？

## 配置HiveOnSpark

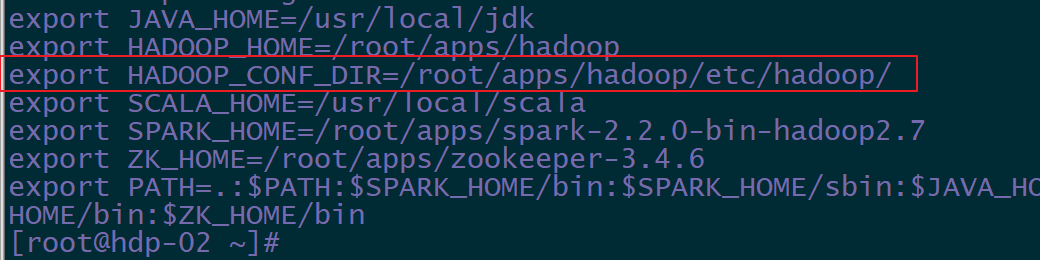
生成hive的元数据库表，根据hive的配置文件，生成对应的元数据库表。

spark-sql 是spark专门用于编写sql的交互式命令行。

当直接启动spark-sql以local模式运行时，如果报错：



是因为配置了Hadoop的配置参数导致的：

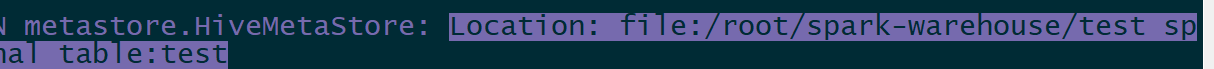


解决方案是直接启动hdfs即可。

执行测试命令:

create table test (name string);

insert into test values(“xxtest”);



local模式下，默认使用derby数据库，数据存储于本地位置。

**要想使用hive的标准，需要把hive的配置文件放到spark的conf目录下**

cd /root/apps/spark-2.2.0-bin-hadoop2.7/conf/

vi hive-site.xml

hive-site.xml文件：

|  |
| --- |
| <**configuration**>  <**property**>  <**name**>javax.jdo.option.ConnectionURL</**name**>  <**value**>jdbc:mysql://hdp-01:3306/hive?createDatabaseIfNotExist=true</**value**>  <**description**>JDBC connect string for a JDBC metastore</**description**>  </**property**>   <**property**>  <**name**>javax.jdo.option.ConnectionDriverName</**name**>  <**value**>com.mysql.jdbc.Driver</**value**>  <**description**>Driver class name for a JDBC metastore</**description**>  </**property**>   <**property**>  <**name**>javax.jdo.option.ConnectionUserName</**name**>  <**value**>root</**value**>  <**description**>username to use against metastore database</**description**>  </**property**>   <**property**>  <**name**>javax.jdo.option.ConnectionPassword</**name**>  <**value**>123456</**value**>  <**description**>password to use against metastore database</**description**>  </**property**> </**configuration**> |

把该配置文件，发送给集群中的其他节点：

|  |
| --- |
| cd /root/apps/spark-2.2.0-bin-hadoop2.7/conf/  for i in 2 3 ;do scp hive-site.xml hdp-0$i:`pwd` ;done |

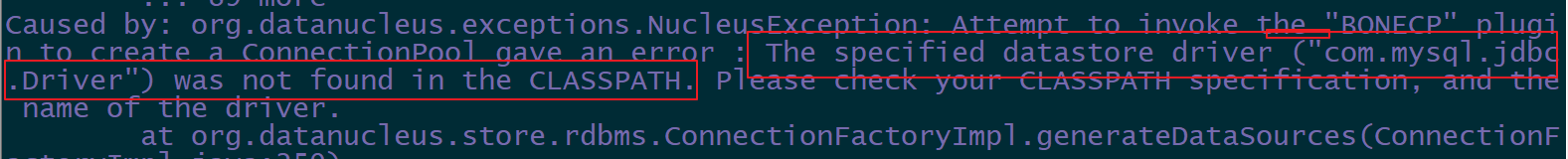
重新停止并重启spark: start-all.sh

启动spark-sql时，

出现如下错误是因为操作mysql时缺少mysql的驱动jar包，

解决方案1：--jars 或者 --driver-class-path 引入msyql的jar包

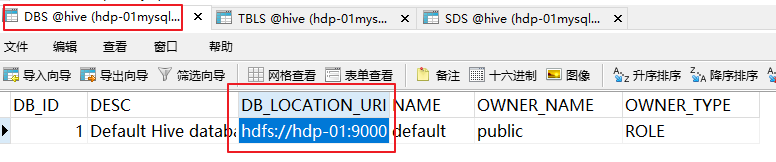
解决方案2： 把mysql的jar包添加到$spark\_home/jars目录下



启动时指定集群：（如果不指定master，默认就是local模式）

spark-sql --master spark://hdp-01:7077 --jars /root/mysql-connector-java-5.1.38.jar

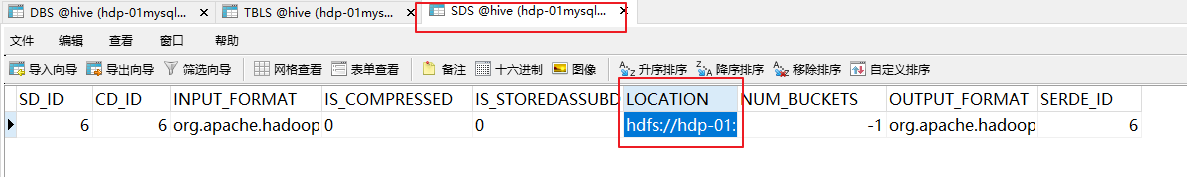
sparkSQL会在mysql上创建一个database，需要手动改一下DBS表中的DB\_LOCATION\_UIR改成hdfs的地址



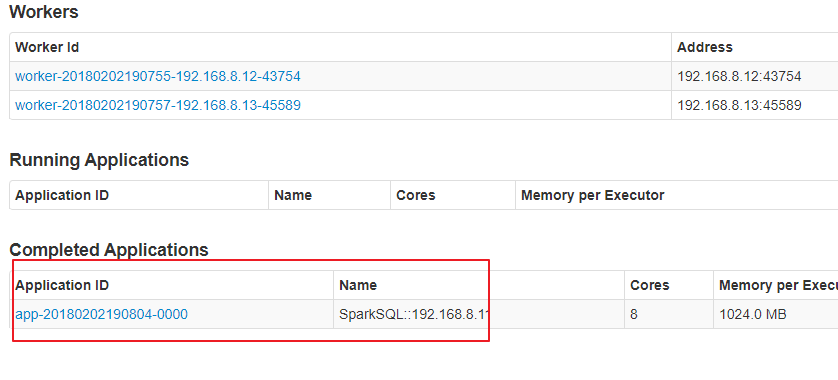
hdfs://hdp-01:9000/user/hive/spark-warehouse

还需要查看一下，自己创建的数据库表的存储路径是否是hdfs的目录。如果是本地目录，需要修改为hdfs目录，不然写入该表的数据，依然会存储到本地目录。

所有的数据表在SDS表中。



执行spark-sql任务之后：可以在集群的监控界面查看



同样 ，会有SparkSubmit进程存在。

## IDEA编程

**要先开启spark对hive的支持**

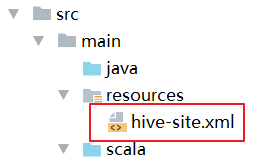
|  |
| --- |
| *//如果想让hive运行在spark上，一定要开启spark对hive的支持* **val** session = SparkSession.*builder*()  .master(**"local"**)  .appName(**"xx"**)   **.enableHiveSupport()** *// 启动对hive的支持, 还需添加支持jar包* .getOrCreate() |

要添加spark对hive的兼容jar包

|  |
| --- |
| *<!--sparksql对hive的支持-->* <**dependency**>  <**groupId**>org.apache.spark</**groupId**>  <**artifactId**>spark-hive\_2.11</**artifactId**>  <**version**>${spark.version}</**version**> </**dependency**> |

在本地运行，还需把hive-site.xml文件拷贝到resource目录下。

resources目录，存放着当前项目的配置文件



编写代码，local模式下测试：

|  |
| --- |
| *// 执行查询* **val** query = session.sql(**"select \* from t\_access\_times"**) query.show() *// 释放资源* session.close() |

创建表的时候，需要伪装客户端身份

报错：



|  |
| --- |
| System.*setProperty*(**"HADOOP\_USER\_NAME"**, **"root"**) *// 伪装客户端的用户身份为root*  *// 或者添加运行参数 –DHADOOP\_USER\_NAME=root* |

基本操作

|  |
| --- |
| *// 求每个用户的每月总金额  // session.sql("select username,month,sum(salary) as salary from t\_access\_times group by username,month")  // 创建表  // session.sql("create table t\_access1(username string,month string,salary int) row format delimited fields terminated by ','")   // 删除表  // session.sql("drop table t\_access1")   // 插入数据  // session.sql("insert into t\_access1 select \* from t\_access\_times")  // .show()  // 覆盖写数据  // session.sql("insert overwrite table t\_access1 select \* from t\_access\_times where username='A'")   // 覆盖load新数据  // C,2015-01,10  // C,2015-01,20  // session.sql("load data local inpath 't\_access\_time\_log' overwrite into table t\_access1")   // 清空数据  // session.sql("truncate table t\_access1")   // .show()   // 写入自定义数据* **val** access: Dataset[String] = session.createDataset(*List*(**"b,2015-01,10"**, **"c,2015-02,20"**))   **val** accessdf = access.map({  t =>  **val** lines = t.split(**","**)  (lines(0), lines(1), lines(2).toInt)  }).toDF(**"username"**, **"month"**, **"salary"**)   *// .show()* accessdf.createTempView(**"t\_ac"**)  *// session.sql("insert into t\_access1 select \* from t\_ac")*  *// insertInto的api 入库* accessdf.write.insertInto(**"databaseName.tableName"**)session.close() |

集群运行：

需要把hive-site.xml配置文件，添加到$SPARK\_HOME/conf目录中去,上传一个mysql连接驱动（sparkSubmit也要连接MySQL，获取元数据信息）

spark-sql --master spark://hdp-01:7077 --driver-class-path /root/mysql-connector-java-5.1.38.jar

--class xx.jar