1. 我们现在用的hive版本是2.0.0，spark版本是1.6.0，这两个都可以在对应的官网上找到。
2. 由于hive自带的元数据库derby不支持多用户同时访问hive，所以配置mysql做为hive的元数据库。
3. 用apt-get命令安装mysql（出现apt-get不能update的情况的话可以先将/var/lib/apt/lists这个路径下的所有文件都删除，然后再重新update，如果还是不行就只能更换apt-get的源了）
4. 使用用户账号密码登录mysql，这里用的是root，密码是mysql
5. 登陆后要设置用户权限允许远程登录

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

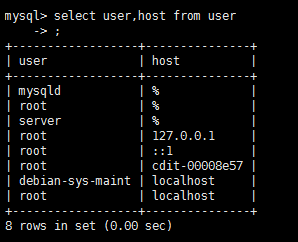
%表示所有的ip地址

FLUSH PRIVILEGES;

通过以上两条命令，我们就设置好了用户的远程登录权限

查看用户表select user,host from user;

可以看到root已经允许所有ip接入了。



1. 设置好用户远程登录权限之后，一般我们的mysql还是不允许远程登录，所以我们要解除mysql的本地登录绑定

修改/etc/mysql这个路径下的my.cnf文件，把这一行注释掉



新建一个hive数据库，以供hive使用。

1. 把hive解压到/opt目录下，仿照/conf/hive-default.xml.template新建一个hive-site.xml文件

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>

<property>

<name></name>

<value></value>

</property>

</configuration>

主要是需要这个格式。

在hive-siet.xml文件下配置以下参数

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:mysql://210.209.117.219:3306/hive?createDatabaseIfNotExist=true</value>

</property>

<property>

<name>javax.jdo.option.ConnectionDriverName</name>

<value>com.mysql.jdbc.Driver</value>

</property>

<property>

<name>javax.jdo.option.ConnectionUserName</name>

<value>root</value>

</property>

<property>

<name>javax.jdo.option.ConnectionPassword</name>

<value>mysql</value>

</property>

<property>

<name>datanucleus.schema.autoCreateTables</name>

<value>true</value>

</property>

<property>

<name>hive.execution.engine</name>

<value>spark</value>

</property>

<property>

<name>spark.master</name>

<value>spark://210.209.117.148:7070</value>

</property>

<property>

<name>spark.eventLog.enabled</name>

<value>true</value>

</property>

<property>

<name>spark.eventLog.dir</name>

<value>/opt/spark-1.6.0-bin-hadoop2.6/logs/</value>

</property>

<property>

<name>spark.serializer</name>

<value>org.apache.spark.serializer.KryoSerializer</value>

</property>

<property>

<name>hive.querylog.location</name>

<value>/usr/hive/log</value>

</property>

这些参数我暂时不是很清楚哪些是必须的，但是前几个关于mysql远程登录的参数是必须要有的。

1. 将hive-site.xml拷贝到spark的conf目录下
2. 将mysql的驱动拷贝到hive的lib目录下
3. 启动hive，第一次启动hive我们需要使用

hive --service metastore

以后启动直接用hive就可以了。

此时我们进入mysql的hive数据库下，应该就能够hive自动创建了很多表了。



1. 在hive里面创建一个test数据库，接下来我们如果能在pyspark shell里面查到，说明就搭建成功了。
2. 修改spark的spark-env.sh文件，将SPARK\_CLASSPATH这个参数设置成mysql驱动，例如我的是设置成

export

SPARK\_CLASSPATH=/opt/spark-1.6.0-bin-hadoop2.6/lib/mysql-connector-java-5.1.37-bin.jar

其实启动spark会出现一些警告信息，说是不支持这种方式，但是我按着它的方法修改并不能成功。如果不使用这种方式的话只能在每次启动mysql的时候后面加上一个

--driver-class-path /opt/apache-hive-2.0.0-bin/lib/mysql-connector-java-5.1.37-bin.jar

这样显得比较麻烦。

1. 打开pyspark shell
2. sqlContext.sql(“show databases”).collect()

此时应该已经能够看见有两个数据库default跟test。

1. 将hive文件夹整个发送到其他的节点上（要在相同的路径下），同时别忘了在所有节点下的spark/conf下都配置一个hive-site.xml（就跟之前的一样就行）
2. 已知缺陷

(a)现在在所有启动pyspark shell的路径下，只要我们在操作中用到了sqlContext，还是会出现一个derby.log文件