Hiveserver2的代理执行之路

背景

对于一个数据平台的SQL查询服务，impala提供了优于hive/spark sql等一批sql-on-mr/spark性能的查询引擎，并且得益于impala可以直接共享hive的metastore，可以对用户提供"一套数据,多种引擎"的服务，目前我们计划在数据平台集成hive/spark/impala这几种SQL引擎。众所周知，hive无论是在稳定性还是在成熟度上都要优于后两者，不管是spark还是impala在使用的过程中总是需要对其进行一定的修改支持hive提供的特性，其中有一个对于平台服务最有用的特性——代理执行。

hiveserver2的代理访问可以使得平台端代理任意用户执行SQL操作就像该用户自己执行的操作一样（就像一个普通用户直接使用hive CLI执行操作），本文主要探索hiveserver2是如何使用代理的方式实现支持不同用户完成SQL操作，为修改impala支持对应的操作做铺垫。

HiveServer2的实现

在启动hive server2的时候，我们通常需要一个hive.keytab，这个用户在hadoop上被配置为可代理的用户（具体的配置是在namenode的core-site.xml中添加hadoop.proxyuser.hive.hosts=xxx和hadoop.proxyuser.hive.groups=xxx配置项，以确定hive用户可以对指定host的指定groups的用户执行代理），除了对hiveserver2配置principal和keytab之外，还需要设置hive.server2.enable.doAs参数为true（该配置项默认值就是true），该配置表示对于用户的操作，hiveserver2将以代理的方式访问HDFS和提交MR任务。

好了，以代理的方式配置好了hiveserver2，我们可以看一下利用这个特性能够做到什么。 在本例中，hive.keytab已经被配置了可代理的权限，其他用户全是普通用户。通过beeline连接hiveserver2，beeline作为client会以当前用户的kerberos cache中认证的kerberos用户作为被代理的账号执行，例如当前机器上的用户是nrpt：

```
>  klist
Ticket cache: FILE:/tmp/krb5cc_50997
Default principal: nrpt/dev@HADOOP.HZ.NETEASE.COM

Valid starting    Expires           Service principal
10/02/2017 09:30  11/02/2017 07:30  krbtgt/HADOOP.HZ.NETEASE.COM@HADOOP.HZ.NETEASE.COM
     renew until 11/02/2017 09:30
```

然后连接hiveserver2执行查询：

```
> beeline -u "jdbc:hive2://db-53.photo.163.org:10000/default;principal=hive/app-20.photo.163.org@HADOOP.HZ.NETEASE.COM"
Connecting to jdbc:hive2://db-53.photo.163.org:10000/default;principal=hive/app-20.photo.163.org@HADOOP.HZ.NETEASE.COM
Connected to: Apache Hive (version 1.2.1)
Driver: Hive JDBC (version 1.2.1)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 1.2.1 by Apache Hive
0: jdbc:hive2://db-53.photo.163.org:10000/def> select count(1) from foodmart.sales_fact_1997;
+--------+--+
|  _c0   |
+--------+--+
| 86837  |
+--------+--+
1 row selected (37.497 seconds)
```

连接成功，此时执行SQL查询，这个查询需要提交MR任务，通过hadoop任务管理界面，可以看到该任务的提交用户是nrpt，也就是被代理的用户。

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1483509789812_270639	nrpt	select count(1) from ...mart.sales_fact_1997(Stage-1)	MAPREDUCE	root.nrpt.default	2017/2/10 上午10:21:24	N/A	RUNNING	UNDEFINED		ApplicationMaster

除了任务提交，在访问hive的时候还会涉及到大量的HDFS操作，这种操作是否也是以被代理的账号执行的呢，可以通过如下的SQL验证，create as select会新建一个表并且让将查询结果写入该表。

```
0: jdbc:hive2://db-53.photo.163.org:10000/def> create table test_nrpt as select * from foodmart.sales_fact_1997;
No rows affected (28.992 seconds)
```

查看HDFS上的文件权限可以看到该表的数据的确是通过被代理用户(nrpt)执行的。

```
> hadoop fs -ls hdfs://hz-cluster2/user/nrpt/hive-server/test_nrpt
Found 1 items
-rw-r--r--   3 nrpt hdfs    2971680 2017-02-10 10:25 hdfs://hz-cluster2/user/nrpt/hive-server/test_nrpt/000000_0
> hadoop fs -ls hdfs://hz-cluster2/user/nrpt/hive-server/ | grep test_nrpt
drwxr-xr-x   - nrpt hdfs          0 2017-02-10 10:25 hdfs://hz-cluster2/user/nrpt/hive-server/test_nrpt
```

演示了这么多，为什么说这个特性是特别重要的呢？对于一个数据平台的开发，往往需要支持不同产品的用户执行SQL，而不同的产品通常使用不同的kerberos用户（对于不同用户的数据安全性的保证），利用hive server2的代理特性，就可以使得不同的用户使用同一个hiveserver2并且彼此之间做到数据和权限的隔离。

客户端代理

好了，那既然可以做到这样是不是就完事了，其实不然，一个数据平台的野心不会仅仅局限在提供一个hiveserver2让用户访问，往往维护封装成一个查询窗口（例如猛犸），用户不需要关心hiveserver2在哪里启动。在这种情况就需要平台端来创建到hiveserver2的连接，然后执行用户输入的查询，并且最重要的一点是**需要以用户的身份执行该查询！**前面我们看到，如果希望以用户A的身份执行查询那么就需要当前kerberos认证的用户是A，难不成平台端要保存全部的用户keytab，然后在执行不同用户操作的时候执行切换？

如果真的要这么笨重拿就没法玩了，既然hive用户可以在hiveserver2代理任意用户执行查询，那么客户端不是也可以通过hive代理任何用户执行SQL吗？我们需要做代理执行的时候执行逻辑是这样子的：

```
UserGroupInformation ugi = UserGroupInformation.createProxyUser(proxyUser, UserGroupInformation.getLoginUser());
System.out.println("Current kerberos user : " + ugi);
ugi.doAs(new PrivilegedExceptionAction<Void>() {
    public Void run() throws Exception {
        // do something with user proxyUser.
    }
});
```

在doAs中执行的操作都是以proxyUser用户的身份执行的，通常这里就是提交MR任务，访问HDFS之类的操作，相信hiveserver2的实现肯定使用的也是类似的方式，当然这里有一个前提条件就是getLoginUser()（通常就是kinit认证的用户或者在程序里面执行kerberos的用户）返回的用户必须具有代理权限，如果任意一个普通用户都可以createProxyUser，人人都变成超级账号了。我们把run方法中写入创建hive connection并且执行查询：

```
public Void run() throws Exception {
    Class.forName("org.apache.hive.jdbc.HiveDriver");
    Connection conn = DriverManager.getConnection(
        "jdbc:hive2://db-53.photo.163.org:10000/default;principal=hive/ app-20.photo.163.org@HADOOP.HZ.NETEASE.COM");
    Statement statement = conn.createStatement();
    statement.execute("select count(1) from foodmart.sales_fact_1997");
    return null;
}
```

执行发现并没有预想的那么顺利，出现了kerberos认证的错误：

```
Current kerberos user : nrpt (auth:PROXY) via hive/app-20.photo.163.org@HADOOP.HZ.NETEASE.COM (auth:KERBEROS)
17/02/10 10:56:31 INFO jdbc.Utils: Supplied authorities: db-53.photo.163.org:10000
17/02/10 10:56:31 INFO jdbc.Utils: Resolved authority: db-53.photo.163.org:10000
17/02/10 10:56:31 ERROR transport.TSaslTransport: SASL negotiation failure
javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Fa
    at com.sun.security.sasl.gsskerb.GssKrb5Client.evaluateChallenge(GssKrb5Client.java:212)
    at org.apache.thrift.transport.TSaslClientTransport.handleSaslStartMessage(TSaslClientTransport.java:94)
    at org.apache.thrift.transport.TSaslTransport.open(TSaslTransport.java:271)
    at org.apache.thrift.transport.TSaslClientTransport.open(TSaslClientTransport.java:37)
    at org.apache.hadoop.hive.thrift.client.TUGIAssumingTransport$1.run(TUGIAssumingTransport.java:52)
    at org.apache.hadoop.hive.thrift.client.TUGIAssumingTransport$1.run(TUGIAssumingTransport.java:49)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:415)
```

看来通过客户端的代理是不行了，此时就要祭出hiveserver2对于平台端一个非常重要的参数：hive.server2.proxy.user，仔细思考一下其实对于hiveserver2而言，它需要知道客户端的用户名，最直接的方式就是使用当前认证的用户执行，那么hiveserver2得到的用户名就是该连接中kerberos的认证用户，而对于代理执行的情况，如果当前连接的kerberos是一个可代理的账号，那么就可以通过hive.server2.proxy.user参数传递真正的代理用户，这样就不需要任何ugi的操作。使用该参数的前提仍然是当前的kerberos用户具有可代理权限，如果一个普通用户使用该参数创建连接，会出现如

下错误：

```
> beeline -u "jdbc:hive2://db-53.photo.163.org:10000/default;principal=hive/app-20.photo.163.org@HADOOP.HZ.NETEASE.COM;hive.server
Connecting to jdbc:hive2://db-53.photo.163.org:10000/default;principal=hive/app-20.photo.163.org@HADOOP.HZ.NETEASE.COM;hive.server
Error: Failed to validate proxy privilege of nrpt for da (state=08S01,code=0)
Beeline version 1.2.1 by Apache Hive
0: jdbc:hive2://db-53.photo.163.org:10000/def (closed)>
```

而如果当前kerberos账号是可代理的用户，执行连接则能够成功并且可以以被代理的用户身份执行SQL：

```
> kinit -kt hive.keytab  hive/app-20.photo.163.org@HADOOP.HZ.NETEASE.COM
> beeline -u "jdbc:hive2://db-53.photo.163.org:10000/default;principal=hive/app-20.photo.163.org@HADOOP.HZ.NETEASE.COM;hive.server
Connecting to jdbc:hive2://db-53.photo.163.org:10000/default;principal=hive/app-20.photo.163.org@HADOOP.HZ.NETEASE.COM;hive.server
Connected to: Apache Hive (version 1.2.1)
Driver: Hive JDBC (version 1.2.1)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 1.2.1 by Apache Hive
0: jdbc:hive2://db-53.photo.163.org:10000/def>
```

讨论到这里对于平台端如何使用hiveserver2代理任意用户执行查询已经比较清晰了，总结一下就是hiveserver2开启doAs的选项，平台端通过加入hive.server2.proxy.user参数代理任意用户执行查询，不需要拥有任何用户的keytab。但是这里还是存在一个问题，那就是不同用户的连接URL是不同的，这样服务端需要对每一个用户维护一个连接池，或者比较暴力的做法就是每次查询新创建一个connection，执行完成之后销毁该连接。

以上探讨了hiveserver2在使用过程中的代理方式，但是spark和impala就没有这么完善的功能了，下文再详细梳理impala是怎么做的，以及如何能够修改使其实现hiveserver2类似的功能。