

Apache Presto对接FusionInsight

适用场景

Presto0.184 <-> FusionInsight HD V100R002C70SPC100

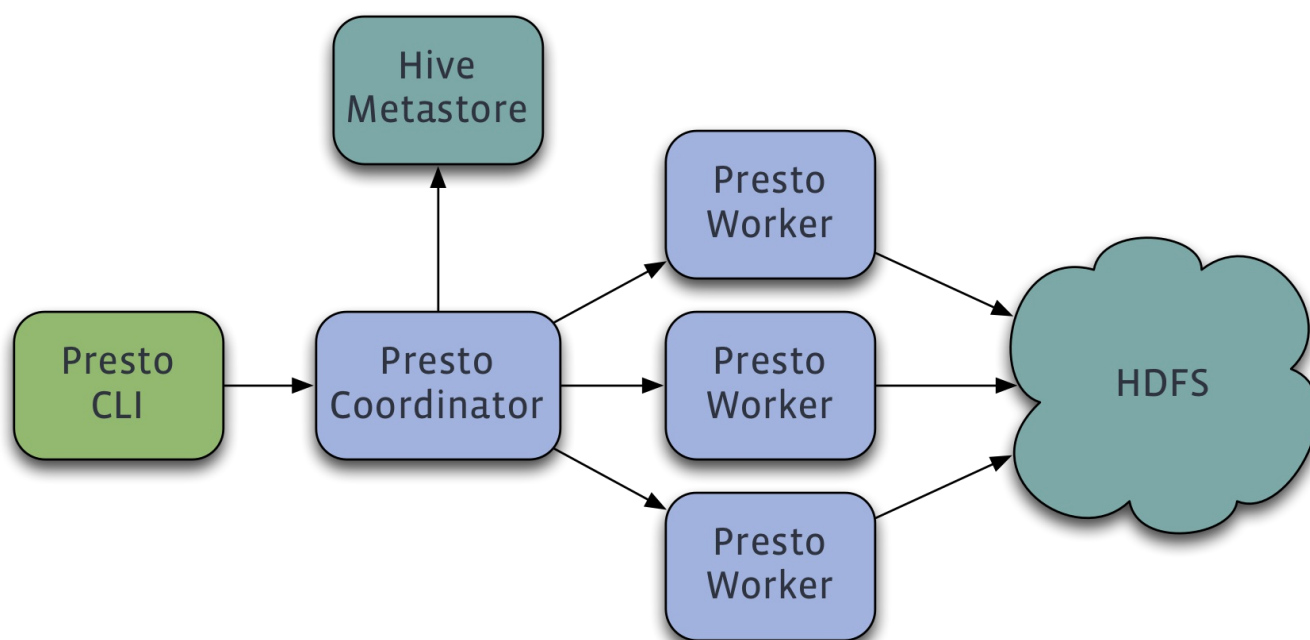
Presto0.196 <-> FusionInsight HD V100R002C80SPC100

说明

Presto是一个开源的分布式SQL查询引擎，适用于交互式分析查询，数据量支持GB到PB字节。

Presto的设计和编写完全是为了解决像Facebook这样规模的商业数据仓库的交互式分析和处理速度的问题

Presto主要与FusionInsight的Hive和HDFS进行对接



配置Hive Connector

Presto集群包括coordinator节点和无限数量的worker节点(coordinator节点也可同时为worker节点)，其中只需要在coordinator节点上配置Hive Connector即可。本文档中配置coordinator节点同时也是worker节点。

- 从该链接下载presto-server的安装包，并上传到presto coordinator的节点

<https://repo1.maven.org/maven2/com/facebook/presto/presto-server/0.184/presto-server-0.184.tar.gz>

将该压缩包解压后得到目录 `/opt/presto-server-0.184`。

- 在presto节点上安装华为FusionInsight HD V100R002C70SPC100的客户端，默认安装目录 `/opt/hadoopclient`
- presto 0.184版本要求jdk至少在1.8u60+以上，修改 `/etc/profile` 文件方式配置系统默认的java为FusionInsight HD客户端的jdk，并source环境变量，命令参考如下

在 `/etc/profile` 中增加以下行

```
export JAVA_HOME=/opt/hadoopclient/JDK/jdk
export JREHOME=/opt/hadoopclient/JDK/jdk/jre
export PATH=$JAVA_HOME/bin:$PATH
```

source 环境变量

```
source /etc/profile
```

- 创建Java keystore File for TLS，(后续步骤默认都在presto节点上执行)参考如下命令

```
source /opt/hadoopclient/bigdata_env
keytool -genkeypair -alias testuser -keyalg RSA -keystore /opt/presto.jks
```

alias后的值必须要跟后面创建的用户名称一致

first and last name必须写成presto节点的主机名

```
[root@presto-server opt]# keytool -genkeypair -alias testuser -keyalg RSA -keystore /opt/presto.jks
Enter keystore password:
Re-enter new password: Huawei@123
What is your first and last name?
[Unknown]: presto-server
What is the name of your organizational unit?
[Unknown]: it
What is the name of your organization?
[Unknown]: it
What is the name of your City or Locality?
[Unknown]: shenzhen
What is the name of your State or Province?
[Unknown]: shenzhen
What is the two-letter country code for this unit?
[Unknown]: 518000
Is CN=presto-server, OU=it, O=it, L=shenzhen, ST=shenzhen, C=518000 correct?
[no]: y

Enter key password for <testuser> Huawei@123
(RETURN if same as keystore password):
Re-enter new password:
[root@presto-server opt]#
```

- 通过FusionInsight HD的管理页面创建一个“机机”用户，具体请参见《FusionInsight HD管理员指南》的 [创建用户](#) 章节。例如，创建用户testuser，并选择hadoop和hive用户组，下载对应的密钥文件user.keytab以及krb5.conf文件，并上传到presto节点的 [/opt/hadoopclient](#) 目录下，将user.keytab改名为testuser.keytab。
- 参考如下命令在Huawei FusionInsight HD的Kerberos中创建一个新的principal，其名称为“testuser/presto-server”，其中presto-server为presto的coordinator节点的主机名，导出该principal的密钥文件为 [/opt/presto.keytab](#)。

```
[root@presto-server opt]# source /opt/hadoopclient/bigdata_env
[root@presto-server opt]#
[root@presto-server opt]# kinit admin
Password for admin@HADOOP.COM:
[root@presto-server opt]#
[root@presto-server opt]# kadmin -p kadmin/admin
Authenticating as principal kadmin/admin with password.
Password for kadmin/admin@HADOOP.COM:
kadmin: addprinc -randkey testuser/presto-server
NOTICE: no policy specified for testuser/presto-server@HADOOP.COM; assigning "default"
Principal "testuser/presto-server@HADOOP.COM" created.
kadmin: ktadd -k /opt/presto.keytab testuser/presto-server
Entry for principal testuser/presto-server with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to keytab WRFILE:/opt/presto.keytab.
Entry for principal testuser/presto-server with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to keytab WRFILE:/opt/presto.keytab.
kadmin:
```

执行kadmin -p kadmin/admin命令时初始密码Admin@123，修改后需严格牢记新密码。

```
[root@presto-server hadoopclient]# cd /opt
[root@presto-server opt]# ll
total 606848
drwxr-xr-x  3 root root    4096 Feb 24 09:46 client
drwxr-xr-x 16 root root    4096 Feb 24 11:10 hadoopclient
-rw-r--r--  1 root root 14836687 Feb 23 15:16 presto-cli-0.155-executable.jar
-rw-r--r--  1 root root    2240 Feb 24 10:09 presto.jks
-rw-----  1 root root    162 Feb 24 10:20 presto.keytab
drwxr-xr-x  6 root root    4096 Feb 24 10:25 presto-server-0.155
-rw-r--r--  1 root root 606543495 Feb 23 15:09 presto-server-0.155.tar.gz
drwxr-xr-x  2 root root    4096 May 17 2013 rh
```

- 创建目录/opt/presto-server-0.184/etc，在该目录下创建如下文件

config.properties参考如下

```
coordinator=true
node-scheduler.include-coordinator=true
http-server.http.port=8080
query.max-memory=50GB
query.max-memory-per-node=1GB
discovery-server.enabled=true
discovery.uri=http://presto-server:8080
http-server.authentication.type=KERBEROS
http.server.authentication.krb5.service-name=testuser
http.server.authentication.krb5.keytab=/opt/presto.keytab
http.authentication.krb5.config=/opt/hadoopclient/KrbClient/kerberos/var/krb5kdc/krb5.conf
http-server.https.enabled=true
http-server.https.port=7778
http-server.https.keystore.path=/opt/presto.jks
http-server.https.keystore.key=Huawei@123
```

jvm.config参考如下内容

```
-server
-Xmx16G
-XX:+UseG1GC
-XX:G1HeapRegionSize=32M
-XX:+UseGCOverheadLimit
-XX:+ExplicitGCInvokesConcurrent
-XX:+HeapDumpOnOutOfMemoryError
-XX:OnOutOfMemoryError=kill -9 %p
-Djava.security.krb5.config=/opt/hadoopclient/KrbClient/kerberos/var/krb5kdc/krb5.conf
```

node.properties参考如下内容

```
node.environment=production
node.id=ffffffff-ffff-ffff-ffff-ffffffffffffff
node.data-dir=/var/presto/data
```

log.properties参考如下内容

```
com.facebook.presto=INFO
```

- 创建目录/opt/presto-server-0.184/etc/catalog，在该目录下创建hive.properties文件

```
connector.name=hive-hadoop2
hive.metastore.uri=thrift://162.1.93.101:21088,thrift://162.1.93.102:21088
hive.metastore.service.principal=hive/hadoop.hadoop.com@HADOOP.COM
hive.metastore.authentication.type=KERBEROS
hive.metastore.client.principal=testuser/presto-server
hive.metastore.client.keytab=/opt/presto.keytab
hive.hdfs.authentication.type=KERBEROS
hive.hdfs.impersonation.enabled=false
hive.hdfs.presto.principal=testuser
hive.hdfs.presto.keytab=/opt/hadoopclient/testuser.keytab
hive.config.resources=/opt/presto-server-0.184/etc/catalog/core-site.xml,/opt/presto-server-0.184/etc/catalog/hdfs-site.xml
```

其中hive.metastore.uri的值从/opt/hadoopclient/Hive/config/hive-site.xml中查找

- 将FusionInsight HD客户端中的core-site.xml和hdfs-site.xml复制到 /opt/presto-server-0.184/etc/catalog 中

```
cp /opt/hadoopclient/HDFS/hadoop/etc/hadoop/core-site.xml /opt/presto-server-0.184/etc/catalog/
cp /opt/hadoopclient/HDFS/hadoop/etc/hadoop/hdfs-site.xml /opt/presto-server-0.184/etc/catalog/
```

- 按照下图修改hdfs-site.xml文件中的dfs.client.failover.proxy.provider.hacluster属性为org.apache.hadoop.hdfs.servernamenode.ha.ConfiguredFailoverProxyProvider

```
vi /opt/presto-server-0.184/etc/catalog/hdfs-site.xml
```

```
<property>
<name>dfs.client.failover.proxy.provider.hacluster</name>
<!--
<value>org.apache.hadoop.hdfs.server.namenode.ha.BlackListingFailoverProxyProvider</value>
-->
<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
</property>
```

- 修改/etc/hosts文件，将本机的IP与主机名解析以及Huawei FusionInsight HD集群节点的IP与主机名解析添加进去，例如

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
162.1.93.103 162-1-93-103
162.1.93.102 162-1-93-102
162.1.93.101 162-1-93-101
162.1.115.71 presto-server
~
```

- 安装maven:

```
wget http://apache.osuosl.org/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz
tar -xzf apache-maven-3.3.9-bin.tar.gz -C /opt/
```

修改profile文件 `vi /etc/profile` ,增加以下配置

```
export PATH=$PATH:/opt/apache-maven-3.3.9/bin
```

导入环境变量

```
source /etc/profile
```

执行 `mvn -v` 可以正确输出mvn版本

- 安装git

```
yum install -y git
```

- 参考如下命令，下载presto-server-0.184的源码

```
git clone https://github.com/prestodb/presto.git
git checkout 0.184
```

- 修改presto-hive/src/main/java/com/facebook/presto/hive/authentication/KerberosHiveMetastoreAuthentication.java的代码，将代码中"Sasl.QOP=auth"修改为"Sasl.QOP=auth-conf"

```
@Override
public TTransport authenticate(TTransport rawTransport, String hiveMetastoreHost)
    throws TTransportException
{
    try {
        String serverPrincipal = getServerPrincipal(hiveMetastoreServicePrincipal, hiveMetastoreHost);
        String[] names = SaslRpcServer.splitKerberosName(serverPrincipal);
        checkState(names.length == 3,
            "Kerberos principal name does NOT have the expected hostname part: %s", serverPrincipal);

        Map<String, String> saslProps = ImmutableMap.of(
            Sasl.QOP, "auth",
            Sasl.SERVER_AUTH, "true"
        );
    }
}
```

- 重新编译presto

```
cd presto-hive
```

```
mvn clean install -DskipTests
```

- 将编译后target目录下的presto-hive-0.184.jar文件替换/opt/presto-server-0.184/plugin/hive-hadoop2/presto-hive-0.184.jar文件
- 启动presto server, 跟踪/var/presto/data/var/log/server.log查看启动日志

```
sh /opt/presto-server-0.184/bin/launcher stop
sh /opt/presto-server-0.184/bin/launcher start
tailf /var/presto/data/var/log/server.log
```

通过Presto CLI连接Hive

使用Presto CLI连接Huawei FusionInsight HD的Hive, 使用presto自带的命令行工具执行SQL语句。

- 通过如下链接下载presto cli启动的jar包
<https://repo1.maven.org/maven2/com/facebook/presto/presto-cli/0.184/presto-cli-0.184-executable.jar>
- 并将该jar包上传到可与presto节点网络互通的节点上(也可将presto coordinator节点作为cli使用节点)。
- 配置cli节点的jdk为1.8u60+以上版本
- 配置cli节点的/etc/hosts文件, 将FI集群和presto coordinator节点的IP与主机名关系配置到cli节点
- 从presto节点拷贝presto.jks、presto.keytab、krb5.conf以及连接HDFS所需的core-site.xml和hdfs-site.xml文件到cli节点
- 将presto-cli-0.184-executable.jar包改为可执行文件

```
mv presto-cli-0.184-executable.jar presto
chmod u+x presto
./presto -h
```

- 创建presto cli启动脚本, 类似如下, 注意将相关文件的路径按实际位置替换

```
./presto \
--server https://presto-server:7778 \
--enable-authentication \
--krb5-config-path /opt/hadoopclient/krb5.conf \
--krb5-principal testuser/presto-server \
--krb5-keytab-path /opt/presto.keytab \
--krb5-remote-service-name testuser \
--keystore-path /opt/presto.jks \
--keystore-password Huawei@123 \
--catalog hive \
--schema default \
```

catalog后面的hive是和presto coordinator节点配置的hive.properties的文件名匹配的, 如果hive.properties改名为hivetest.properties, 则这里改为hivetest

- 通过cli执行SQL语句, 其他SQL语法请参考<https://prestodb.io/docs/0.184/sql.html>

```
[root@presto-server opt]# ./presto \
> --server https://presto-server:7778 \
> --enable-authentication \
> --krb5-config-path /opt/hadoopclient/krb5.conf \
> --krb5-principal testuser/presto-server \
> --krb5-keytab-path /opt/presto.keytab \
> --krb5-remote-service-name testuser \
> --keystore-path /opt/presto.jks \
> --keystore-password Huawei@123 \
> --catalog hive \
> --schema default \
> ;
presto:default> show tables;
  Table
-----
web_sales
workers_info
(2 rows)
```


查询表workers_info中数据:

```
presto:default> select * from workers_info ;
id | name | usd_flag | salary | address | entrytime
-----+-----+-----+-----+-----+-----
1 | Wang | R | 8000.01 | Chian:Shenzhen | 2014
2 | Chen | F | 2334.01 | Chian:Shanghai | 2015
3 | Zhang | R | 10000.01 | Chian:Suzhou | 2013
4 | Lili | F | 12334.01 | Chian:Shanghai | 2012
5 | Cina | R | 18000.01 | Chian:Zhejiang | 2013
6 | Wang | R | 8000.01 | Chian:Shenzhen | 2014
7 | Zhou | F | 12334.01 | Chian:Shanghai | 2011
8 | Zhang | R | 10000.01 | Chian:Suzhou | 2013
9 | Lili | F | 12334.01 | Chian:Shanghai | 2012
10 | Cina | R | 18000.01 | Chian:Zhejiang | 2013
(10 rows)

Query 20170224_032139_00006_rr37g, FINISHED, 1 node
Splits: 2 total, 2 done (100.00%)
0:00 [10 rows, 386B] [58 rows/s, 2.2KB/s]
```

百万记录数表web_sales查询:

```
presto:default> select * from web_sales ;
ws_sold_date_sk | ws_sold_time_sk | ws_ship_date_sk | ws_item_sk | ws_bill_customer_sk | ws_bill_demo_sk | ws_bill_hdemo_sk | ws_bill
-----+-----+-----+-----+-----+-----+-----+-----
2451843 | 56105 | 2451893 | 24355 | 96964 | 1531120 | 3676 |
2451843 | 56105 | 2451877 | 22253 | 96964 | 1531120 | 3676 |
2451843 | 56105 | 2451892 | 5504 | 96964 | 1531120 | 3676 |
2451843 | 56105 | 2451949 | 3620 | 96964 | 1531120 | 3676 |
2451843 | 56105 | 2451925 | 17288 | 96964 | 1531120 | 3676 |
2451843 | 56105 | 2451857 | 6221 | 96964 | 1531120 | 3676 |
2451843 | 56105 | 2451959 | 23975 | 96964 | 1531120 | 3676 |
2451621 | 29628 | 2451718 | 18494 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451640 | 9067 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451692 | 22601 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451710 | 13124 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451680 | 16387 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451697 | 16148 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451685 | 15980 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451725 | 6349 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451728 | 10502 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451663 | 3907 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451676 | 21025 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451727 | 23831 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451657 | 3175 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451715 | 16256 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451626 | 14999 | 117452 | 586110 | 6768 |
2451621 | 29628 | 2451658 | 18431 | 117452 | 586110 | 6768 |
2451966 | 78283 | 2452018 | 135 | 136490 | 131280 | 5717 |
2451966 | 78283 | 2451999 | 7473 | 136490 | 131280 | 5717 |
2451966 | 78283 | 2452020 | 18131 | 136490 | 131280 | 5717 |
:
```

通过Presto JDBC连接Hive

使用Presto JDBC接口连接Huawei FusionInsight HD Hive

- 从如下链接下载jdbc的驱动包

<https://repo1.maven.org/maven2/com/facebook/presto/presto-jdbc/0.184/presto-jdbc-0.184.jar>

- 参考<https://prestodb.io/docs/0.184/installation/jdbc.html>设置JDBC URL，用户名为任意字符，密码为空，在eclipse中调通的示例如下:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class PrestoTest {
    public static void main(String[] args) throws SQLException, ClassNotFoundException {
        Class.forName("com.facebook.presto.jdbc.PrestoDriver");
        Connection connection = DriverManager.getConnection(
            ("jdbc:presto://162.1.115.71:8080/hive/default", "root", null));
        Statement stmt = connection.createStatement();
        ResultSet rs = stmt.executeQuery("select * from workers_info limit 10");
        int col = rs.getMetaData().getColumnCount();
        while(rs.next())
        {
            // ...
        }
    }
}
```

```

        for (int i = 1; i <= col; i++) {
            System.out.print(rs.getString(i) + "\t");
            if ((i == 2) && (rs.getString(i).length() < 8)) {
                System.out.print("\n");
            }
        }
        System.out.println("");
    }
    rs.close();
    connection.close();
}

```

- 测试结果:

```

1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6
7
8 public class PrestoTest {
9     public static void main(String[] args) throws SQLException, ClassNotFoundException {
10         Class.forName("com.facebook.presto.jdbc.PrestoDriver");
11         Connection connection = DriverManager.getConnection("jdbc:presto://162.1.115.71:8080/hive/default","root",null);
12         Statement stmt = connection.createStatement();
13         ResultSet rs = stmt.executeQuery("select * from workers_info limit 10");
14         int col = rs.getMetaData().getColumnCount();
15         while(rs.next())
16         {
17             for (int i = 1; i <= col; i++) {
18                 System.out.print(rs.getString(i) + "\t");
19                 if ((i == 2) && (rs.getString(i).length() < 8)) {
20                     System.out.print("\n");
21                 }
22             }
23             System.out.println("");
24         }
25         rs.close();
26         connection.close();
27     }
28 }

```

Problems @ Javadoc Declaration Console

<terminated> PrestoTest [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2017年2月24日 下午3:12:15)

1	Wang	R	8000.01	Chian:Shenzhen	2014
2	Chen	F	2334.01	Chian:Shanghai	2015
3	Zhang	R	10000.01	Chian:Suzhou	2013
4	Lili	F	12334.01	Chian:Shanghai	2012
5	Cina	R	18000.01	Chian:Zhejiang	2013
6	Wang	R	8000.01	Chian:Shenzhen	2014
7	Zhou	F	12334.01	Chian:Shanghai	2011
8	Zhang	R	10000.01	Chian:Suzhou	2013
9	Lili	F	12334.01	Chian:Shanghai	2012
10	Cina	R	18000.01	Chian:Zhejiang	2013

- 百万记录数表web_sales查询:

```

1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6
7
8 public class PrestoTest {
9     public static void main(String[] args) throws SQLException, ClassNotFoundException {
10         Class.forName("com.facebook.presto.jdbc.PrestoDriver");
11         Connection connection = DriverManager.getConnection("jdbc:presto://162.1.115.71:8080/hive/default","root",null);
12         Statement stmt = connection.createStatement();
13         ResultSet rs = stmt.executeQuery("select * from web_sales limit 10");
14         int col = rs.getMetaData().getColumnCount();
15         while(rs.next())
16         {
17             for (int i = 1; i <= col; i++) {
18                 System.out.print(rs.getString(i) + "\t");
19                 if ((i == 2) && (rs.getString(i).length() < 8)) {
20                     System.out.print("\n");
21                 }
22             }
23             System.out.println("");
24         }
25         rs.close();
26         connection.close();
27     }
}

```

<terminated> PrestoTest [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2017年2月24日 下午3:15:50)																				
2451108	63244	2451185	2257	63605	1817417	2429	40241	69865	1887489	6924	44736	2	28	16	4	141	54566	100	24.74	47
2451108	63244	2451117	22702	63605	1817417	2429	40241	69865	1887489	6924	44736	1	8	15	3	232	54566	92	70.55	20
2451108	63244	2451199	6994	63605	1817417	2429	40241	69865	1887489	6924	44736	10	16	16	5	269	54566	27	95.11	13
2451108	63244	2451203	19234	63605	1817417	2429	40241	69865	1887489	6924	44736	7	13	16	3	307	54566	24	14.61	34
2451108	63244	2451211	11984	63605	1817417	2429	40241	69865	1887489	6924	44736	56	10	14	2	275	54566	88	44.17	57
2451108	63244	2451143	9931	63605	1817417	2429	40241	69865	1887489	6924	44736	40	8	14	1	268	54566	51	90.62	22
2451108	63244	2451128	14014	63605	1817417	2429	40241	69865	1887489	6924	44736	14	22	15	3	147	54566	46	63.36	91
2451108	63244	2451135	11515	63605	1817417	2429	40241	69865	1887489	6924	44736	55	10	6	4	87	54566	38	35.81	75
2451108	63244	2451144	11608	63605	1817417	2429	40241	69865	1887489	6924	44736	74	4	1	1	313	54566	93	15.67	40
2451408	67276	2451469	18736	85521	999043	2165	54087	73079	366866	5500	71822	61	7	16	1	299	54567	62	47.92	10