课程综合实验：Hive 分析搜狗搜索日志与结果可视化

一、实验描述

利用 hive 命令行完成搜狗日志各项数据分析，使用 Python 进行数据可视化。主要步骤包括：安装部署 Hive、启动 Hadoop 集群、进入 Hive 命令行、创建数据库和数据表、加载或导入数据、用 Hive SQL 完成需求、使用 Python 实现数据可视化。

二、实验目的

1.掌握安装 Hive 的方法；

2.掌握 Hive 创建数据库、导入数据的方法；

3.学会使用 Hive SQL 分析数据；

4.学会数据可视化的方法。

三、实验环境

1.虚拟机数量：3；

2.系统版本：Centos 7.5；

3. Hadoop 版本：Apache Hadoop 2.7.3；

4. MySQL 版本：MySQL 5.7.30；

5. Hive 版本：Apache Hive 2.1.1。

四、实验步骤

实验开始前，请确保 Hadoop 集群已经安装成功(可参考实验三 Hadoop 集群安装部署部分)。接下来的步骤主要是：元数据库 Mysql 安装、Hive 安装部署、Hive SQL 数据分析、数据可视化。

4.1 元数据库 Mysql 安装

本实验安装 MySQL 是为了给 Hive 提供元数据存储库，主要包括：yum 安装 MySQL、修改 MySQL root 密码、添加 zkpk 用户并赋予远程访问权限、修改数据库默认编码。

步骤 1：查看并卸载系统自带的 mariadb-lib 数据库：

[root@master ~]# rpm -qa|grep mariadb

mariadb-5.5.68-1.el7.aarch64

mariadb-libs-5.5.68-1.el7.aarch64

*[root@master ~]# yum -y remove mariadb-**

步骤 2：添加 MySQLyum 源（首先确保能访问网络）；

1)使用 WinSCP 上传 mysql 安装包(或使用 wget 下载，命令为：*wget https://obs-mirror-ftp4.obs.cn-north-4.myhuaweicloud.com/database/mysql-5.7.30.tar.gz*)：

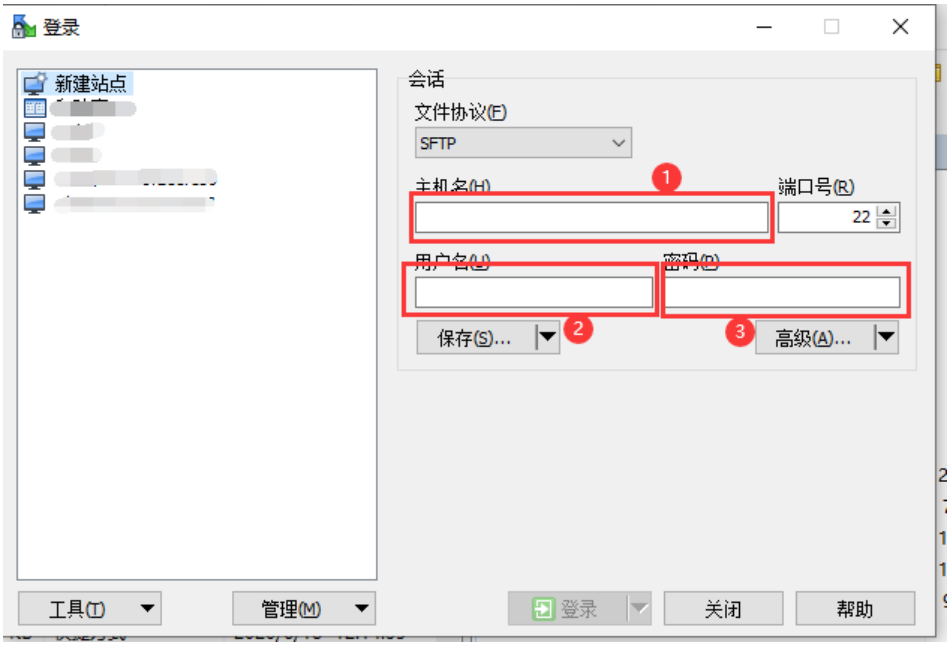


图 1 填写 ip、用户名和密码新建站点

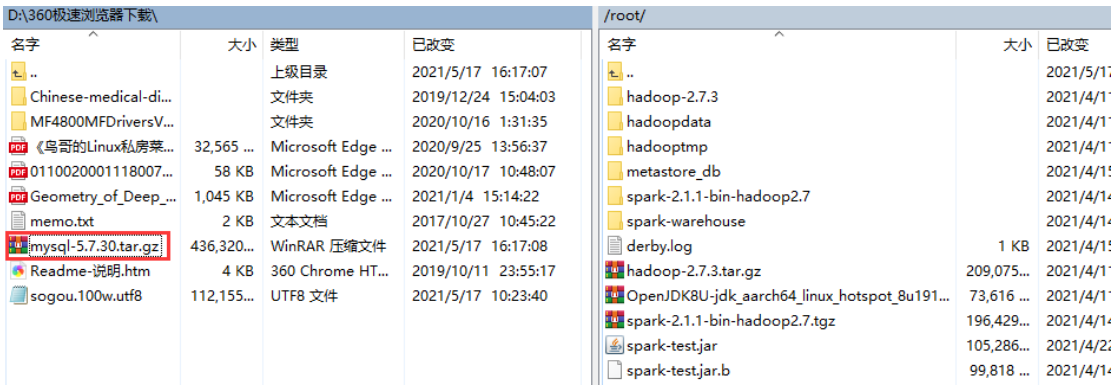


图 2 上传 mysql 安装包

2)安装 mysql 所需依赖：

yum install -y perl openssl openssl-devel libaio perl-JSON autoconf

3)解压 mysql 安装包：

tar -xvf mysql-5.7.30.tar.gz

4)进入 aarch64 目录，对 rpm 包进行安装:

cd aarch64

*yum install *.rpm*

步骤 3：启动 MySQL 服务：

1)命令

systemctl start mysqld

2)查看启动状态

systemctl status mysqld

```
[root@master aarch64]# systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-05-17 16:37:16 CST; 1min 16s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
  Process: 3205 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid $M
ode=exited, status=0/SUCCESS)
  Process: 3150 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
 Main PID: 3208 (mysqld)
   CGroup: /system.slice/mysqld.service
           └─3208 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid
```

步骤 4：修改 root 默认密码：

1)查看 mysql 安装生成的随机默认密码（/var/log/mysqld.log 文件中）

[root@master ~]# grep 'temporary password' /var/log/mysqld.log

```
2021-05-17T08:37:13.599399Z 1 [Note] A temporary password is generated for root@localhost: ex6#sz8flotR
```

2)登录 mysql（密码为上图红框标注部分）

mysql -uroot -p

3)修改 mysql 密码为:MyNewPass4!

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';

注意：mysql5.7 默认安装了密码安全检查插件（validate_password），默认密码检查策略要求密码必须包含：大小写字母、数字和特殊符号，并且长度不能少于 8 位。否则会提示 ERROR

步骤 5：修改 mysql 密码策略

1) 查看 msyql 密码策略的相关信息:

mysql> show variables like '%password%';

```
mysql> show variables like '%password%';
+-----------------------------------------+-----------------+
| Variable_name                           | Value           |
+-----------------------------------------+-----------------+
| default_password_lifetime               | 0               |
| disconnect_on_expired_password          | ON              |
| log_builtin_as_identified_by_password   | OFF             |
| mysql_native_password_proxy_users       | OFF             |
| old_passwords                           | 0               |
| report_password                         |                 |
| sha256_password_auto_generate_rsa_keys  | ON              |
| sha256_password_private_key_path        | private_key.pem |
| sha256_password_proxy_users             | OFF             |
| sha256_password_public_key_path         | public_key.pem  |
| validate_password_check_user_name       | OFF             |
| validate_password_dictionary_file       |                 |
| validate_password_length                | 8               |
| validate_password_mixed_case_count      | 1               |
| validate_password_number_count          | 1               |
| validate_password_policy                | MEDIUM          |
| validate_password_special_char_count    | 1               |
+-----------------------------------------+-----------------+
```

步骤 6：关闭密码策略；

1)禁用密码策略，向 my.cnf 文件中[mysqld]下添加如下配置（/etc/my.cnf）：

[root@master ~]# vim /etc/my.cnf

[mysqld]

validate_password = off

2) 重新启动 mysql 服务使配置生效：

[root@master ~]# systemctl restart mysqld

步骤 7：配置默认编码为 utf8

1)修改/etc/my.cnf 配置文件，在[mysqld]下添加编码配置（注意红色部分）；

2)并且在 my.cnf 中添加 client 模块，输入 client 模块的相关编码格式;

vim /etc/my.cnf

```
validate_password = off
init_connect='SET NAMES utf8'
#
# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option: logging
# changes to the binary log between backups.
# log_bin
```

```
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
[client]
default-character-set=utf8
```

3)重新启动 mysql 服务

[root@master ~]# systemctl restart mysqld

4)登录 mysql:

[root@master ~]# mysql -uroot -p

5)查看编码

mysql> show variables like '%character%';

```
mysql> show variables like '%character%';
+--------------------------+----------------------------+
| Variable_name            | Value                      |
+--------------------------+----------------------------+
| character_set_client     | utf8                       |
| character_set_connection | utf8                       |
| character_set_database   | latin1                     |
| character_set_filesystem | binary                     |
| character_set_results    | utf8                       |
| character_set_server     | latin1                     |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
8 rows in set (0.01 sec)

mysql> exit
Bye
[root@master aarch64]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.138  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::f816:3eff:fe28:412  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:28:04:12  txqueuelen 1000  (Ethernet)
        RX packets 591621  bytes 857254317 (817.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 171403  bytes 12360568 (11.7 MiB)
```

实验截图 1

4.2 Hive 安装部署

本节内容是 Hive 安装部署，主要内容包括：启动 hadoop 集群、解压并安装 Hive、创建 Hive 的元数据库、修改配置文件、添加并生效环境变量、初始化元数据

步骤 1：启动 Hadoop 集群

1)在 master 启动 Hadoop 集群:

[root@master ~]# start-all.sh

2)在 master、slave01、slave02 运行 JPS 指令，查看 Hadoop 是否启动成功；

```
[root@master ~]# jps
4292 Jps
3672 NameNode
4027 ResourceManager
3867 SecondaryNameNode
1070 WrapperSimpleApp
```

```
[root@slave01 ~]# jps
1170 WrapperSimpleApp
2178 Jps
1900 DataNode
2013 NodeManager
```

```
[root@slave02 ~]# jps
1105 WrapperSimpleApp
2018 Jps
1740 DataNode
1853 NodeManager
```

步骤 2:解压并安装 Hive

1) 使用 WinScp 上传 apache-hive-2.1.1-bin.tar.gz(或使用 :*wget http://archive.apache.org/dist/hive/hive-2.1.1/apache-hive-2.1.1-bin.tar.gz* 进行下载)

2)解压并安装 Hive

[root@master ~]# tar -zxvf /root/apache-hive-2.1.1-bin.tar.gz

步骤 3: 向 MySQL 中添加 hadoop 用户和创建名为（hive）的数据库；

1)登录 mysql

[root@master ~]# mysql –uroot -p

2) 创建 hadoop 用户（密码：hadoop）:

*mysql>grant all on *.* to hadoop@'%' identified by 'hadoop';*

*mysql>grant all on *.* to hadoop@'localhost' identified by 'hadoop';*

*mysql>grant all on *.* to hadoop@'master' identified by 'hadoop';*

mysql>flush privileges;

```
mysql> grant all on *.* to hadoop@'%' identified by 'hadoop';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> grant all on *.* to hadoop@'localhost' identified by 'hadoop';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> grant all on *.* to hadoop@'master' identified by 'hadoop';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

3)创建数据库连接

mysql> create database hive;

步骤 4：配置 Hive

1) 进入 hive 安装目录下的配置目录:

[root@master ~]# cd /root/apache-hive-2.1.1-bin/conf/

2) 创建 hive 配置文件:

[root@master conf]# vim hive-site.xml

3)添加如下内容：

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
        <name>hive.metastore.local</name>
        <value>true</value>
</property>
    <property>
        <name>javax.jdo.option.ConnectionURL</name>
         <value>jdbc:mysql://master:3306/hive?characterEncoding=UTF-8</value>
    </property>
<property>
        <name>javax.jdo.option.ConnectionDriverName</name>
        <value>com.mysql.jdbc.Driver</value>
    </property>
<property>
        <name>javax.jdo.option.ConnectionUserName</name>
        <value>hadoop</value>
    </property>
<property>
        <name>javax.jdo.option.ConnectionPassword</name>
```

```
        <value>hadoop</value>
    </property>
    <property>
        <name>hive.server2.authentication</name>
        <value>NOSASL</value>
    </property>
</configuration>
```

步骤 5：复制 MySQL 连接驱动到 hive 根目录下的 lib 目录中：

[root@master conf]# cp /root/mysql-connector-java-5.1.28.jar /root/apache-hive-2.1.1-bin/lib/

[root@master conf]# cd apache-hive-2.1.1-bin/lib/

[root@master conf]# ll | grep mysql-connector-java-5.1.28.jar

```
[root@master lib]# ll | grep mysql-connector-java-5.1.28.jar
-rw-r--r-- 1 root root   875336 May 17 18:56 mysql-connector-java-5.1.28.jar
```

步骤 6：配置系统 zkpk 用户环境变量

1)打开配置文件：

[root@master lib]# cd

[root@master ~]# vim /root/.bash_profile

2) 将下面两行配置添加到环境变量中:

#HIVE

export HIVE_HOME=/root/apache-hive-2.1.1-bin

export PATH=$PATH:$HIVE_HOME/bin

3)使环境变量生效

[root@master ~]# source /root/.bash_profile

步骤 7：启动并验证 Hive 安装：

1)初始化 Hive 元数据库

说明：该命令是把 hive 的元数据都同步到 mysql 中

[root@master ~]# schematool -dbType mysql -initSchema

```
Starting metastore schema initialization to 2.1.0
Initialization script hive-schema-2.1.0.mysql.sql
Initialization script completed
schemaTool completed
```

2)启动 hive 客户端

hive

```
[root@master ~]# hive
which: no hbase in (/usr/java/jdk8u191-b12/bin:/root/hadoop-2.7.3/bin:/root/ha
a/jdk8u191-b12/bin:/root/hadoop-2.7.3/bin:/root/hadoop-2.7.3/sbin:/usr/java/jd
oop-2.7.3/bin:/root/hadoop-2.7.3/sbin:/usr/java/jdk8u191-b12/bin:/root/hadoop-
.7.3/sbin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/root/sp
bin:/root/bin:/root/spark-2.1.1-bin-hadoop2.7/bin:/home/zkpk/apache-hive-2.1.1
/spark-2.1.1-bin-hadoop2.7/bin:/home/zkpk/apache-hive-2.1.1-bin/bin:/root/bin:
doop2.7/bin:/root/apache-hive-2.1.1-bin/bin)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/root/apache-hive-2.1.1-bin/lib/log4j-slf4j-
j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/root/hadoop-2.7.3/share/hadoop/common/lib/s
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanatio
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/root/apache-hive-2.1.1-bi
jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future vers
ifferent execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> exit;
[root@master ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.138  netmask 255.255.255.0  broadcast 192.168.0.255
```

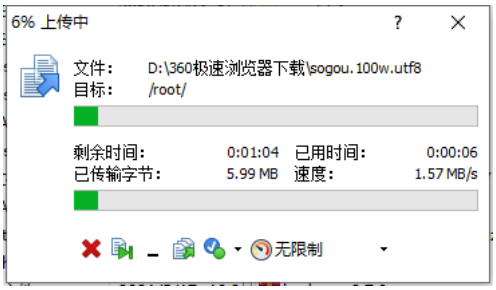
实验截图 2

4.3 Hive SQL 数据分析

利用 hive 命令行完成搜狗日志各项数据分析，本节内容包括：进入 Hive 命令行、创建数据库和数据表、加载或导入数据、用 Hive SQL 完成需求。

步骤 1：数据预处理

1)使用 WinScp 上传 sougo.100w.utf8 文件和 extract.py 文件



对 sougo.100w.utf8 文件，用学号做种子，随机抽取 50 万条，构成新的数据集。执行以下指令(2020140528 需替换为你的学号):

[root@master ~]# python extract.py 2020140528

2) 查看数据内容

[root@master ~]# head -1 sogou.100w.utf8

```
[root@master ~]# head -1 sogou.100w.utf8
20111230000005  57375476989eea12893c0c3811607bcf        奇艺高清        1       1       http://www.qiy
i.com/
```

3) 查看总行数

[root@master ~]# wc -l sogou.100w.utf8

```
[root@master ~]# wc -l sogou.100w.utf8
1000000 sogou.100w.utf8
```

4）将时间字段拆分，添加年、月、日、小时字段

[root@master ~]# awk -F '\t' '{print $0"\t"substr($1,1,4)"\t"substr($1,5,2)"\t"substr($1,7,2)"\t"substr($1,9,2)}' sogou.100w.utf8 > sogou.100w.utf8.1

5)查看拓展后的字段

[root@master ~]# head -3 sogou.100w.utf8.1

```
[root@master ~]# head -3 sogou.100w.utf8.1
20111230000005  57375476989eea12893c0c3811607bcf        奇艺高清        1       1       http://www.qiyi.com/
  2011    12      30      00
20111230000005  66c5bb7774e31d0a22278249b26bc83a        凡人修仙传      3       1       http://www.booksky.o
rg/BookDetail.aspx?BookID=1050804&Level=1 2011    12      30      00
20111230000007  b97920521c78de70ac38e3713f524b50        本本联盟        1       1       http://www.bblianmen
g.com/    2011    12      30      00
```

6)在数据扩展的结果上，过滤第 2 个字段（UID）或者第 3 个字段（搜索关键词）为空的行

[root@master ~]# awk -F"\t" '{if($2 != "" && $3 != "" && $2 != " " && $3 != " ") print $0}' sogou.100w.utf8.1 > sogou.100w.utf8.2

7）重命名数据文件

[root@master ~]# cp sogou.100w.utf8.2 sogou.100w.utf8

步骤 2：加载数据到 HDFS 上:

1）在 hdfs 上创建目录/sogou_ext/20111230

[root@master ~]# hadoop fs -mkdir -p /sogou_ext/20111230

```
[root@master ~]# hadoop fs -ls /
21/05/17 20:35:07 WARN util.NativeCodeLoader: Unable to load native-hadoop
g builtin-java classes where applicable
Found 4 items
drwxr-xr-x   - root supergroup          0 2021-05-17 20:23 /sogou_ext
drwxr-xr-x   - root supergroup          0 2021-04-14 11:56 /spark_test
drwx------   - root supergroup          0 2021-04-14 21:59 /tmp
drwxr-xr-x   - root supergroup          0 2021-04-11 22:49 /user
```

2)上传数据

[root@master ~]# hadoop fs -put sogou.100w.utf8 /sogou_ext/20111230

```
[root@master ~]#  hadoop fs -put sogou.100w.utf8 /sogou_ext/20111230
21/05/17 20:37:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your plat
g builtin-java classes where applicable
[root@master ~]#  hadoop fs -ls /sogou_ext/20111230
21/05/17 20:38:23 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your plat
g builtin-java classes where applicable
Found 1 items
-rw-r--r--   2 root supergroup  128845849 2021-05-17 20:37 /sogou_ext/20111230/sogou.100w.utf8
```

步骤 3：基于 Hive 构建日志数据的数据仓库

1) 进入 hive 客户端命令行:

[root@master ~]# hive

2)查看数据库

hive> show databases;

```
hive> show databases;
OK
default
Time taken: 0.945 seconds, Fetched: 1 row(s)
```

3)创建数据库表

hive> create database sogou_100w;

4)使用数据库：

hive> use sogou_100w;

5) 创建扩展 4 个字段（年、月、日、小时）数据的外部表 sogou_ext_20111230

```
hive> CREATE EXTERNAL TABLE sogou_ext_20111230(
ts STRING,
uid STRING,
keyword STRING,
rank INT,
`order` INT,
url STRING,
year INT,
month INT,
day INT,
hour INT
)
COMMENT 'This is the sogou search data of extend data'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/sogou_ext/20111230';
```

```
hive> CREATE EXTERNAL TABLE sogou_ext_20111230(
    > ts STRING,
    > uid STRING,
    > keyword STRING,
    > rank INT,
    > `order` INT,
    > url STRING,
    > year INT,
    > month INT,
    > day INT,
    > hour INT
    > )
    > COMMENT 'This is the sogou search data of extend data'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > STORED AS TEXTFILE
    > LOCATION '/sogou_ext/20111230';
OK
Time taken: 0.219 seconds
```

6) 创建分区表 sogou_partition

```
hive> CREATE EXTERNAL TABLE sogou_partition(
ts STRING,
uid STRING,
keyword STRING,
rank INT,
`order` INT,
url STRING
)
COMMENT 'This is the sogou search data by partition'
partitioned by (
year INT,
month INT,
day INT,
hour INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE;
```

```
hive> CREATE EXTERNAL TABLE sogou_partition(
    > ts STRING,
    > uid STRING,
    > keyword STRING,
    > rank INT,
    > `order` INT,
    > url STRING
    > )
    > COMMENT 'This is the sogou search data by partitior
    > partitioned by (
    > year INT,
    > month INT,
    > day INT,
    > hour INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > STORED AS TEXTFILE;
OK
Time taken: 0.159 seconds
```

7)开启动态分区

hive> set hive.exec.dynamic.partition=true;

hive> set hive.exec.dynamic.partition.mode=nonstrict;

8) 往分区表中灌入表 sogou_ext_20111230 的数据

*hive> INSERT OVERWRITE TABLE sogou_partition PARTITION(year,month,day,hour) select * from sogou_ext_20111230;*

```
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using
a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_20210517205442_353a15f1-3d27-4234-9d90-c328ae7c0658
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1621242975148_0001, Tracking URL = http://master:18088/proxy/application_1621242975148_00
01/
Kill Command = /root/hadoop-2.7.3/bin/hadoop job  -kill job_1621242975148_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2021-05-17 20:54:52,108 Stage-1 map = 0%,  reduce = 0%
2021-05-17 20:55:01,514 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.76 sec
MapReduce Total cumulative CPU time: 6 seconds 760 msec
Ended Job = job_1621242975148_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://master:9000/user/hive/warehouse/sogou_100w.db/sogou_partition/.hive-staging_
hive_2021-05-17_20-54-42_310_7731095892966213588-1/-ext-10000
Loading data to table sogou_100w.sogou_partition partition (year=null, month=null, day=null, hour=null)


        Time taken to load dynamic partitions: 2.872 seconds
        Time taken for adding to write entity : 0.003 seconds
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 6.76 sec   HDFS Read: 128851259 HDFS Write: 114848276 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 760 msec
OK
Time taken: 26.08 seconds
hive>
```

9)查询分区表的结果：

*hive> select * from sogou_partition limit 3;*

```
hive> select * from sogou_partition limit 3;
OK
20111230000005  57375476989eea12893c0c3811607bcf        奇艺高清        1       1       http://www.qiyi.com/
2011    12      30      0
20111230000005  66c5bb7774e31d0a22278249b26bc83a        凡人修仙传      3       1       http://www.booksky.o
rg/BookDetail.aspx?BookID=1050804&Level=1       2011    12      30      0
20111230000007  b97920521c78de70ac38e3713f524b50        本本联盟        1       1       http://www.bblianmen
g.com/  2011    12      30      0
Time taken: 0.154 seconds, Fetched: 3 row(s)
hive>
```

步骤 4：数据分析需求

1)计总条数

hive> select count(*) from sogou_ext_20111230;

```
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-17 20:58:51,737 Stage-1 map = 0%,  reduce = 0%
2021-05-17 20:58:57,029 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.6 sec
2021-05-17 20:59:04,248 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.01 sec
MapReduce Total cumulative CPU time: 4 seconds 10 msec
Ended Job = job_1621242975148_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.01 sec   HDFS Read: 128854313 HDFS Write: 107 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 10 msec
OK
1000000
Time taken: 19.475 seconds, Fetched: 1 row(s)
```

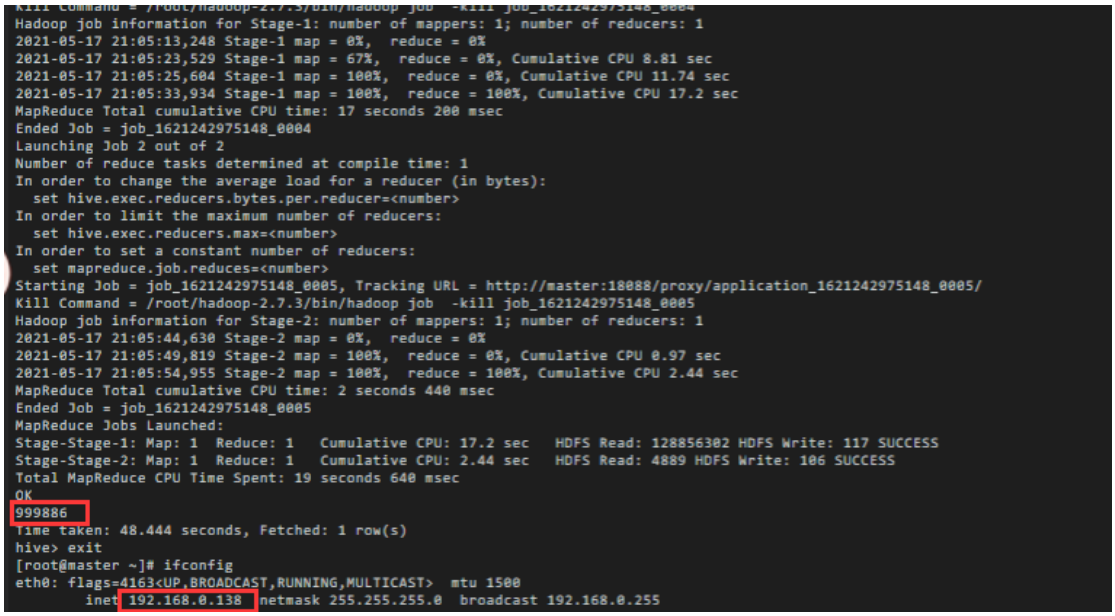
2）统计非空查询条数，即关键词不为空或者 null,用 where 语句排除关键词为 null 和空的字段，在此基础上用 count（）函数统计.

select count(*) from sogou_ext_20111230 where keyword is not null and keyword !='';

```
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-17 21:01:22,201 Stage-1 map = 0%,  reduce = 0%
2021-05-17 21:01:29,423 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.35 sec
2021-05-17 21:01:34,588 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.96 sec
MapReduce Total cumulative CPU time: 4 seconds 960 msec
Ended Job = job_1621242975148_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.96 sec   HDFS Read: 128855138 HDFS Write: 107 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 960 msec
OK
1000000
Time taken: 19.201 seconds, Fetched: 1 row(s)
```

3）无重复总条数（根据 ts、uid、keyword、url）先按照以上四个字段分组，再用 having 语句选出 count（）数为 1 的记录为表 a，再用 count（）函数统计表 a 中的条数。

hive> select count(*) from (select ts,uid,keyword,url from sogou_ext_20111230 group by ts,uid,keyword,url having count(*) =1) a;



实验截图 3

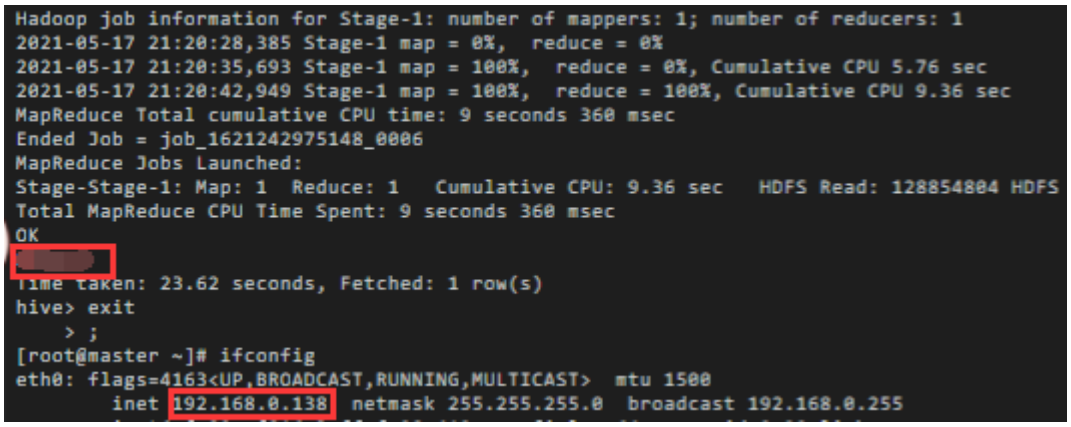
4)独立 UID 条数使用 distinct（）函数对 uid 字段去重，再用 count（）函数统计出条数:

重新进入 hive 并切换数据库：

[root@master ~]# hive

hive> use sogou_100w

自行设计查询语句



实验截图 4

步骤 5：实际需求分析：

1）查询关键词平均长度统计。提示:先使用 split 函数对关键词进行切分，然后用 size（）函数统计关键词的大小，然后再用 avg 函数获取长度的平均值：

重新进入 hive 并切换数据库：

[root@master ~]# hive

hive> use sogou_100w

自行设计查询语句

```
Hadoop job information for Stage-1: number of mappers: 1; number of reducers:
2021-05-17 21:26:06,838 Stage-1 map = 0%,  reduce = 0%
2021-05-17 21:26:15,085 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.82
2021-05-17 21:26:20,238 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.
MapReduce Total cumulative CPU time: 7 seconds 30 msec
Ended Job = job_1621242975148_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.03 sec   HDFS Read: 1288
Total MapReduce CPU Time Spent: 7 seconds 30 msec
OK
[redacted]
Time taken: 21.053 seconds, Fetched: 1 row(s)
hive> exit;
[root@master ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.138  netmask 255.255.255.0  broadcast 192.168.0.255
```

实验截图 5

2)查询频度排名（频度最高的前 50 词）对关键词做 groupby，然后对每组进行 count（），再按照 count（）的结果进行倒序排序。

hive> select keyword,count(*) as cnt from sogou_ext_20111230 group by keyword order by cnt desc limit 50;

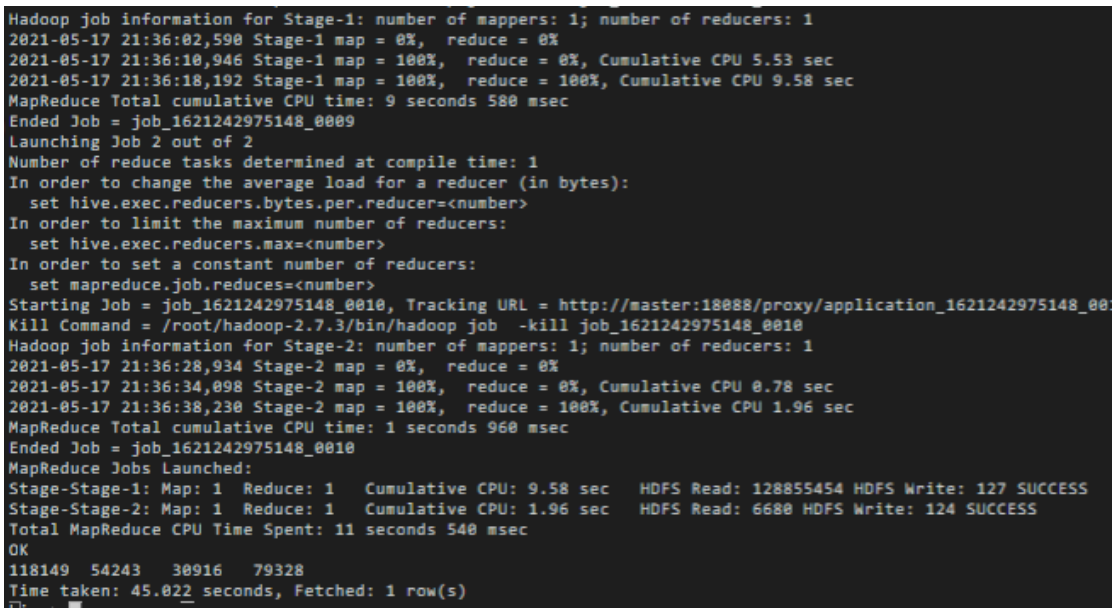
```
Hadoop job information for Stage-2: number of mappers: 1; number of reducers:
2021-05-17 22:09:17,985 Stage-2 map = 0%,  reduce = 0%
2021-05-17 22:09:25,222 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.05
2021-05-17 22:09:31,457 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 5.4
MapReduce Total cumulative CPU time: 5 seconds 470 msec
Ended Job = job_1621242975148_0014
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 11.81 sec   HDFS Read: 1288
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 5.47 sec   HDFS Read: 15058
Total MapReduce CPU Time Spent: 17 seconds 280 msec
OK
百度    7564
baidu   3652
人体艺术        2786
馆陶县县长闫宁的父亲    2388
4399小游戏      2119
qq空间  2083
优酷    1948
新亮剑  1946
百度一下 你就知道       1537
公安卖萌        1459
百度一下        1435
4399    1332
qq网名  1184
魏特琳  1167
儿子与母亲不正当关系    1136
黑狐    1134
7k7k小游戏      1100
新疆暴徒被击毙图片      1067
新浪微博        1017
123     984
hao123  950
李宁香体        906
nba     862
qq个性签名      848
cf官网  807
qq头像  800
dnf官网 775
龙门飞甲        758
全国各省最低工资标准    749
```

步骤 6：UID 分析

1) UID 的查询次数分布（查询 1 次的 UID 个数，2 次的，3 次的，大于 3 次的 UID 个数）

先按照 uid 分组，并用 count（）函数对每组进行统计，然后再用 sum、if 函数对查询次数为 1，查询次数为 2，查询次数为 3，查询次数大于 3 的 uid 进行统计

```
hive> select SUM(IF(uids.cnt=1,1,0)),SUM(IF(uids.cnt=2,1,0)),SUM(IF(uids.cnt=3,1,0)),SUM(IF(uids.cnt>3,1,0)) from
(select uid,count(*) as cnt from sogou_ext_20111230 group by uid) uids;
```



2) 查询次数大于 2 次的用户总数。提示：先对 uid 进行 groupby，并用 having 函数过滤出查询次数大于 2 的用户，然后再用 count 函数统计用户的总数。

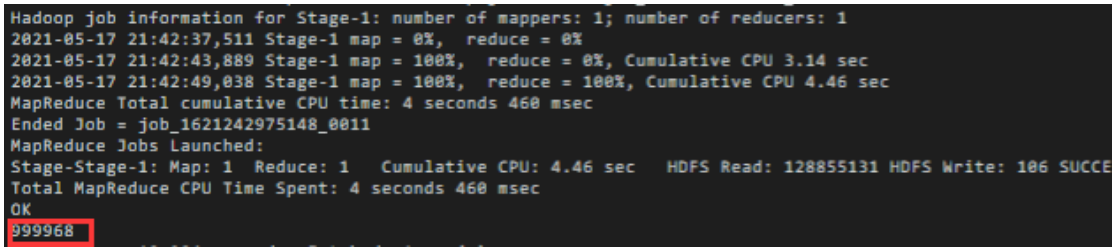
自行设计查询语句(结果中需有 ip)

实验截图 6

步骤 7：用户行为分析

1) 点击次数与 Rank 之间的关系分析：Rank 在 10 以内的点击次数

```
hive> select count(*) from sogou_ext_20111230 where rank < 11;
```



2)直接输入 url 查询的点击次数。通过 where 条件选出直接通过 url 查询的记录再用 count 函数进行统计

hive> select count(*) from sogou_ext_20111230 where keyword like '%www%';

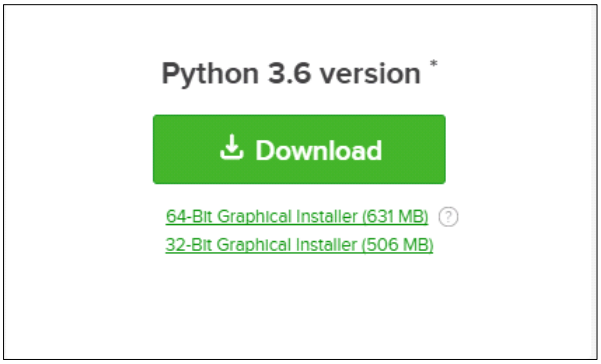
```
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-05-17 21:47:56,296 Stage-1 map = 0%,  reduce = 0%
2021-05-17 21:48:03,564 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.57 sec
2021-05-17 21:48:08,797 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.03 sec
MapReduce Total cumulative CPU time: 5 seconds 30 msec
Ended Job = job_1621242975148_0012
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.03 sec   HDFS Read: 128855116 HDFS Write: 105
Total MapReduce CPU Time Spent: 5 seconds 30 msec
OK
15206
Time taken: 18.368 seconds, Fetched: 1 row(s)
```

4.4 数据可视化

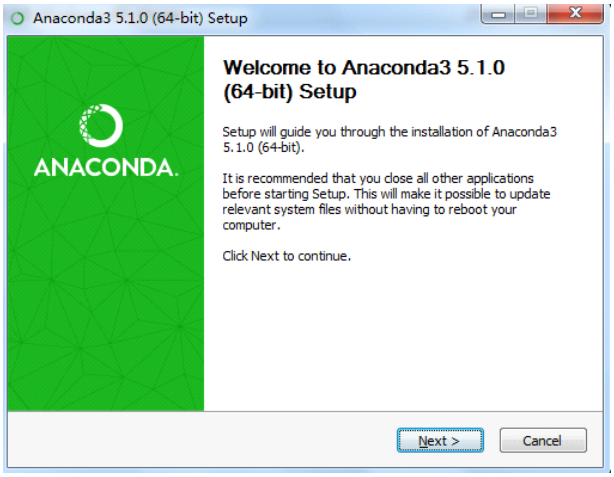
4.4.1 基于 Python 的数据可视化

步骤 1：安装 Anaconda

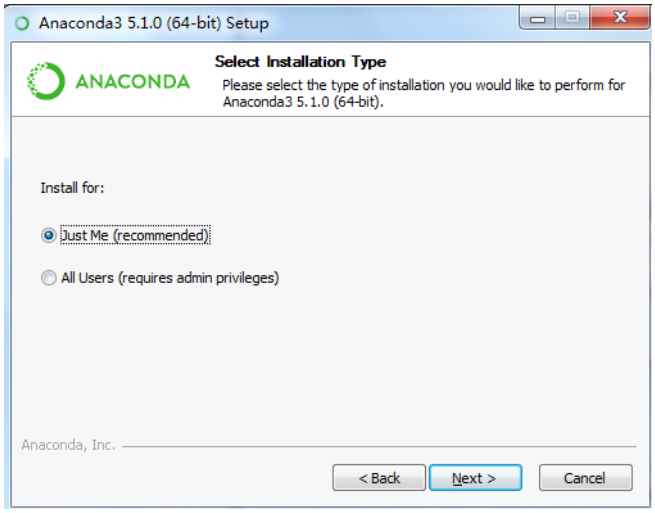
1)登陆 Anaconda 官网下载安装包 https://www.anaconda.com/download/，选择 Python3.6 version ，点击 64-Bit 下载（32 位电脑请下载 32-Bit）。



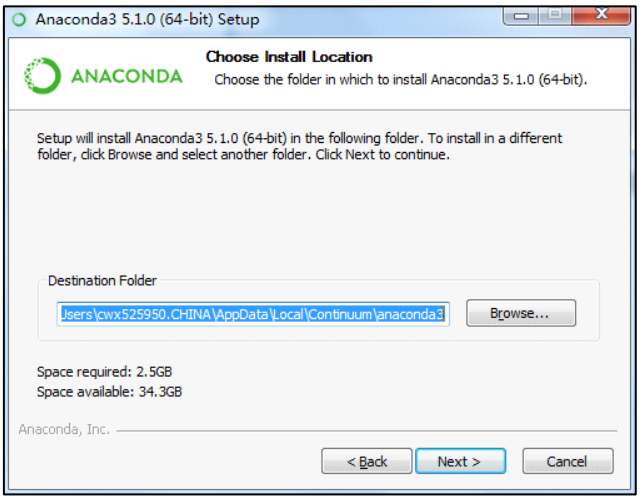
2)双击下载好的 Anaconda3-x.x.x-Windows-x86_64.exe 文件，出现如下界面，点击 Next;



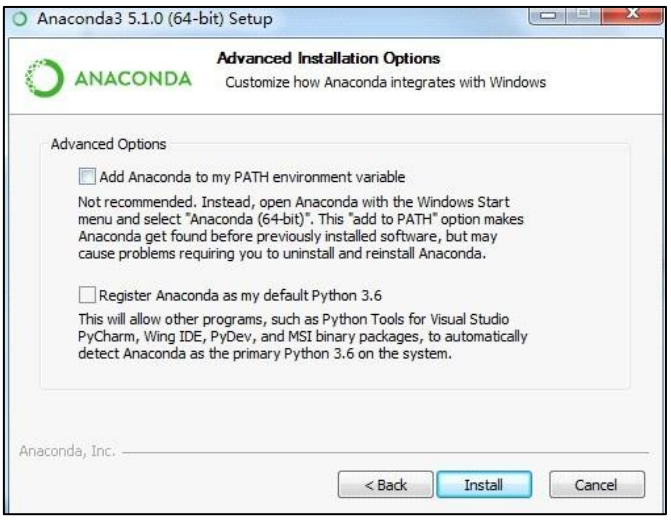
Install for: Just me 还是 All Users,这里直接 Just Me,继续点击 Next 。



3)选择软件安装地址，继续 Next;



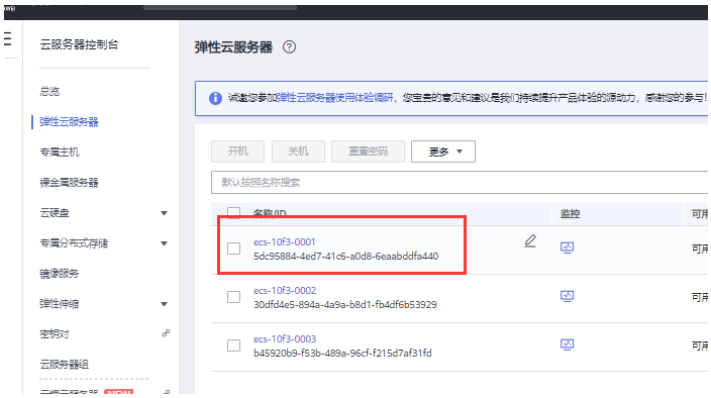
4) 两个都不勾选，第一个是加入环境变量，第二个是默认使用 Python 3.6，我们使用 Anaconda 自带的环境，点击 Install 开始安装;



等待完成安装，点击 finish 即可。

步骤 2：开放服务器的 10000 端口

1)登录华为云->控制台，点击安装 Hive 的服务器；



2)点击安全组

3)配置规则

4)添加如方向规则

5)添加规则

步骤 3：修改 Hadoop 集群配置

1)在/root/hadoop-2.7.3/etc/hadoop 路径下，修改文件 core-site.xml 下，添加如下内容：

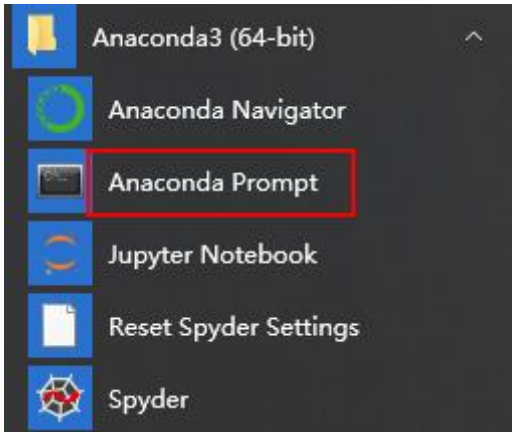
```
<property>
        <name>hadoop.proxyuser.root.hosts</name>
        <value>*</value>
</property>
<property>
        <name>hadoop.proxyuser.root.groups</name>
        <value>*</value>
</property>
```

步骤 4:开启 Hive 远程模式：

```
[root@master hadoop]# hive --service metastore &
[root@master hadoop]# hive --service hiveserver2 &
```

步骤 5：Python 可视化程序编写

1）在安装好的 Anaconda 文件夹中打开 Anaconda Prompt:



2)使用 pip 安装 matplotlib 包

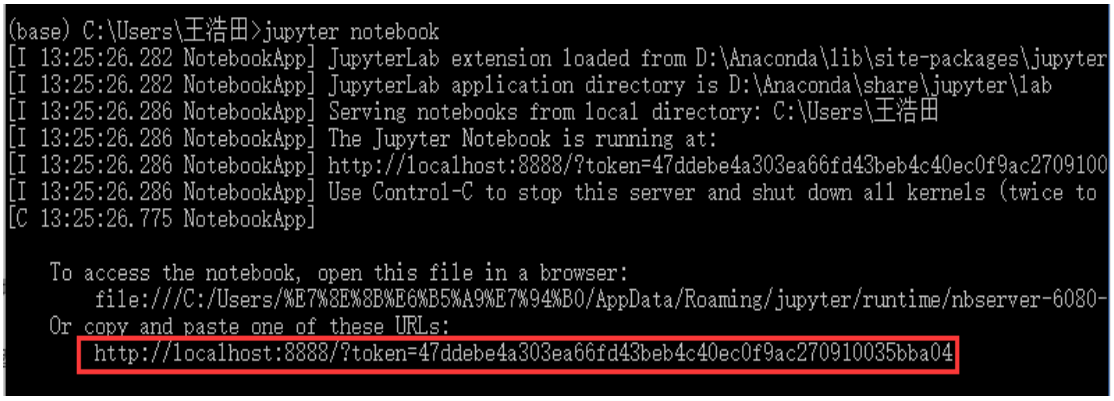
3)使用 pip 安装 pyhive(python 远程连接 hive)

4)使用 pip 安装 thrift(python 远程连接 hive)

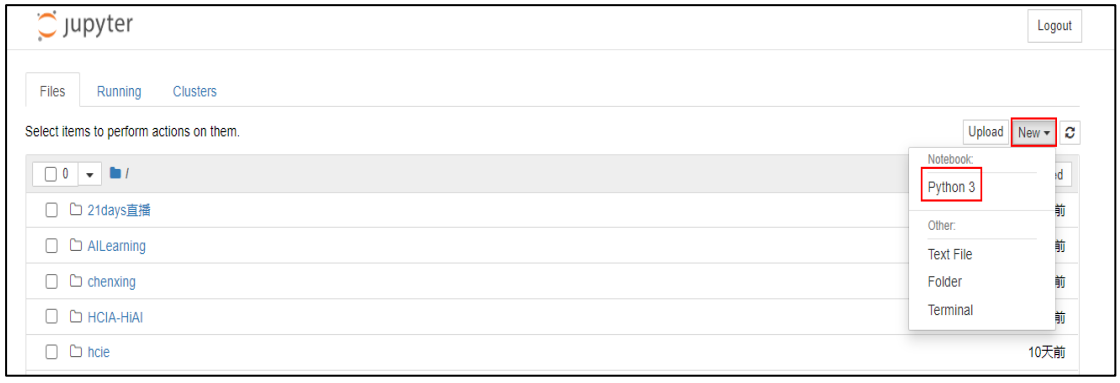
5)使用 pip 安装 sasl(python 远程连接 hive)(若 pip 安装失败，可使用 conda install sasl 安装)

4)输入：jupyter notebook，回车，浏览器会启动 Jupyter notebook,此对话框不要关闭。

5)拷贝 url 到浏览器



6)新建 python 文件

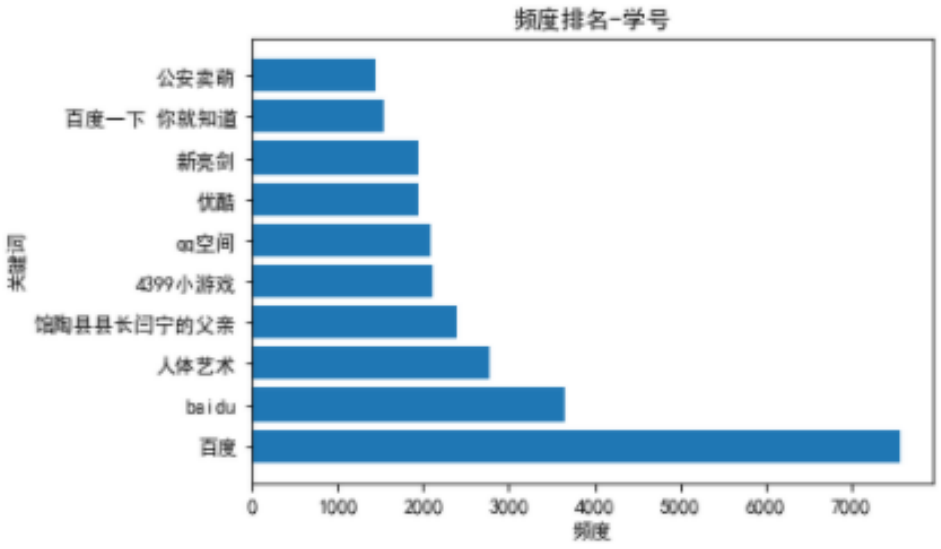


7)编写如下程序，完成关键词搜索前 10 的可视化

```
from pyhive import hive
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei'] # 步骤一（替换 sans-serif 字体）
plt.rcParams['axes.unicode_minus'] = False  # 步骤二（解决坐标轴负数的负号显示问题）

conn = hive.Connection(host='121.36.94.37',port=10000,auth='NOSASL',username='root')
cursor = conn.cursor()
cursor.execute('select keyword,count(*) as cnt from sogou_100w.sogou_ext_20111230 group by keyword order by cnt desc limit 10')
keywords = []
frequency = []
for result in cursor.fetchall():
    keywords.append(result[0])
    frequency.append(result[1])
cursor.close()
conn.close()
plt.barh(keywords, frequency)
plt.title('频度排名-学号') #学号请替换为你的学号，例 2020160750
plt.xlabel('频度')
plt.ylabel('关键词')
plt.show()
```

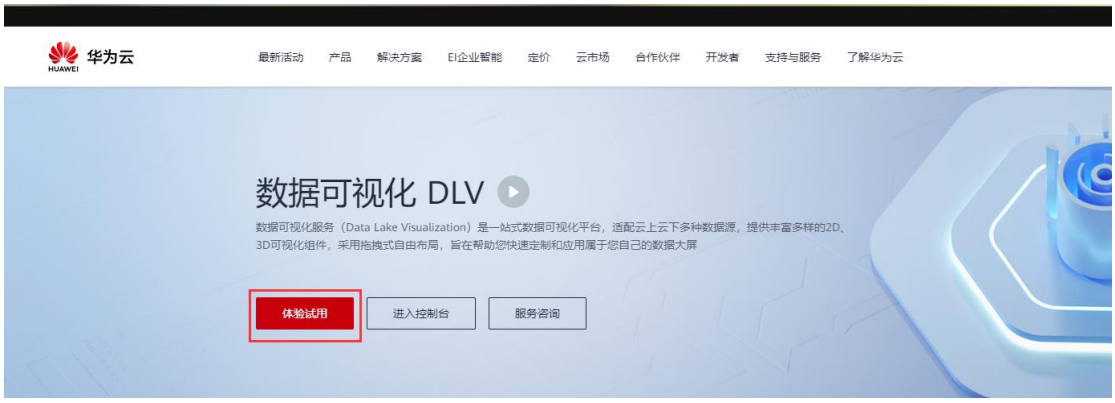


实验截图 7

4.4.2 基于华为云 DLV 的数据可视化

步骤 1：开启 DLV 数据可视化平台

1)打开 https://www.huaweicloud.com/product/dlv.html



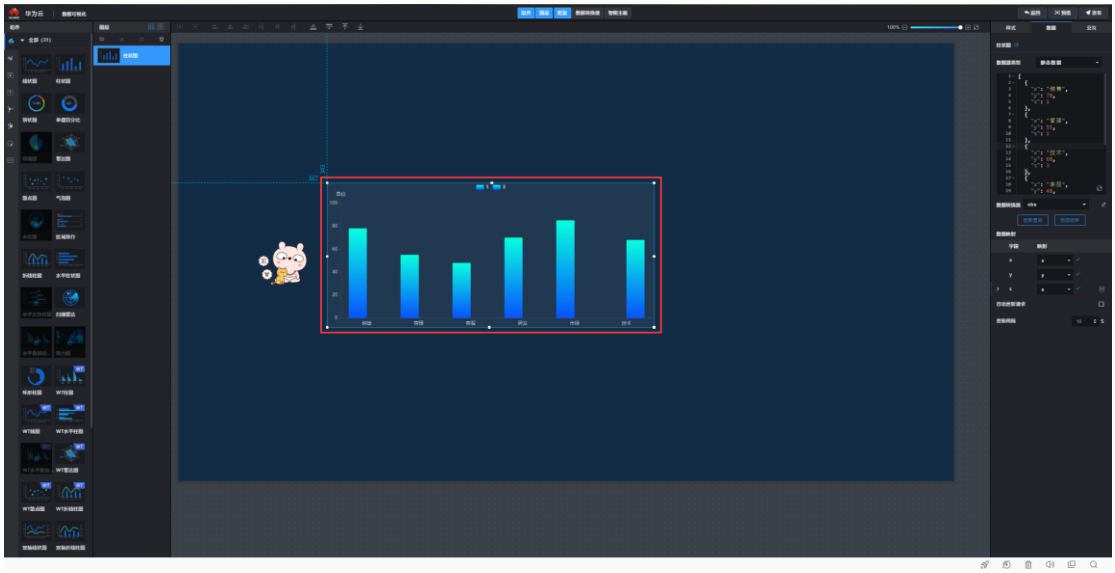
2)点击进入控制台；

3)新建大屏

4)空白模板

5)插入适合分析数据的图组件，这里使用柱状图组件

6)点击统计图



7)设置 y 轴坐标名称

8)将前 6 个热度排名的词填入左侧的静态数据

注意：这里每个 x、y 的取值，使用 在“4.3 Hive SQL 数据分析”中 “步骤 5”的中完成的结果，来进行可视化—— *“2)查询频度排名（频度最高的前 50 词）对关键词做 groupby，然后对每组进行 count（），再按照 count（）的结果进行倒序排序。”*

因此这里面的 x、y、s 的取值，同学们之前应有所不同。

```
[
  {
    "x": "百度",
    "y": 7564,
    "s": "替换为你的学号"
  },
  {
    "x": "baidu",
    "y": 3652,
    "s": "替换为你的学号"
  },
  {
    "x": "人体艺术",
    "y": 2786,
    "s": "替换为你的学号"
  },
  {
    "x": "4399 小游戏",
    "y": 2119,
```

```
    "s": "替换为你的学号"
  },
  {
    "x": "优酷",
    "y": 1948,
    "s": "替换为你的学号"
  },
  {
    "x": "新亮剑",
    "y": 1946,
    "s": "替换为你的学号"
  }
]
```


9)点击预览

实验截图 8

五、实验结果与分析

1.粘贴实验中要求的 8 张截图(按照要求截图，每图 3 分，共 24 分)

注意：sougo.100w.utf8 文件原始是 100 万条数据，要求按照学号作为种子，随机抽取 50 万条，所以不同同学的统计结果应有所区别。提交的实验结构，截图必须在指定位置包含学号，不包含学号不能得分。

实验截图 1:元数据库 Mysql 安装的步骤 7 中查看 mysql 编码的截图；

实验截图 2：Hive 安装部署中的步骤 7 中 hive 启动成功的截图；

实验截图 3-6：Hive SQL 查询的截图；

实验截图 7：python 可视化截图；

实验截图 8：华为 DLV 可视化截图；

3.分析 rank 与用户点击次数之间的关系(6 分，结合数据集的内容、查询统计值，给出细节分析说明和论述);

4.你还能从数据中挖掘哪些有价值的信息？定义2种不同挖掘目标,完成自定义实验，每一种完成以下要求的五个步骤给 5 分。完成 2 种不同挖掘目标，给 10 分。

1）设计挖掘目标；2）完成 HiveSQL 代码；3）运行 HiveSQL 代码，形成执行结果；4）完成基于结果的 Python 可视化代码的编写；5）对结果进行分析。(每个步骤 1 分);