**数据实时监控**

**总结：**

**由于tableau只能同步mysql里的数据，所以我们需要在hadoop环境下建立一个脚本文件，将我们在hive中执行的sql代码的结果输入到txt文件中，并且导入到mysql环境里的表格中，从而同步到tableau**

**步骤：**

1. 在Xshell 里输入 pwd -》找到目标目录
2. 在目标目录新建文件夹，脚本文件，txt文件

mkdir liuliang

vi liuliang.sh 建立或者修改shell脚本

vi liuliang.txt

1. 写入脚本内容到sh文件，此处分情况（①hive下查询或②mysql下查询）

**① hive下查询**

**在hive里建立search表，利用sql语句**

create table if not exists search(

ymd string comment '时间',

show\_pv string comment '点击量',

show\_count string comment '直播数',

show\_uv string comment '访问量',

view\_uv string comment '观看点击',

view\_pv string comment '观看量',

short\_uv string comment '短停人数',

short\_pv string comment '短停数',

time string comment '时间',

enter\_uv string comment '进房间人数')

row format delimited fields terminated by '\t' lines terminated by '\n' stored as textfile

**Shell 脚本内容：直接写入liuliang.sh中**

date=`date -d "yesterday" +%Y%m%d`

hive -e"

insert overwrite table stg.search

select b.ymd,a.show\_pv,a.show\_count,a.show\_uv,b.view\_uv,b.view\_pv,b.short\_uv,b.short\_pv,b.time,c.enter\_uv

from

(select ymd,count(uid) as show\_pv,count(distinct token)as show\_count,count(distinct uid)as show\_uv

from hds.newapplog\_seach\_live\_show

where ymd=$date and token rlike 'rec\_9\_2\_1\_0' and cv rlike '6.0.0'

group by ymd)a

left join

(select ymd,count(uid) as view\_pv,count(distinct uid)as view\_uv,count(distinct case when unix\_timestamp(exit\_time)-unix\_timestamp(enter\_time)>60 then uid end) as short\_uv,count(case when unix\_timestamp(exit\_time)-unix\_timestamp(enter\_time)>60 then uid end) as short\_pv,sum(unix\_timestamp(exit\_time)-unix\_timestamp(enter\_time)) as time

from ana.dw\_live\_view\_info\_everyday\_enter

where ymd=$date and enter='search\_new\_rec' and token rlike 'rec\_9\_2\_1\_0' and cv rlike '6.0.0'

group by ymd)b

on a.ymd=b.ymd

left join

(select ymd,sum(uv) as enter\_uv

from bi.recommend\_rec\_basic\_os

where cv rlike'6.0.0' and ymd=$date and level rlike '全部' and logid rlike 'all'

group by ymd)c

on a.ymd=c.ymd

;"

## vi test.sh i 编辑 esc 退出编辑 :wq！保存编辑

查询：cd test ls cat log\*\*\*.log

**② mysql下查询**

**先在Xshell里进入mysql环境**

mysql -h rm-2ze6406t9hkur76rk.mysql.rds.aliyuncs.com -P 3306 -u inke\_db\_user -p

passward: Nx$X6^soiPi

use db\_test;

create table liuliang(

ymd varchar(15),

show\_pv varchar(10),

show\_count varchar(10),

show\_uv int,

view\_uv int,

view\_pv int,

short\_uv int,

short\_pv int,

alltime int,

enter\_uv int

)engine = innodb DEFAULT CHARSET=utf8; // 设置引擎和字符集，防止中文乱码。

### 编写liuliang.sh 脚本

date=`date -d "yesterday" +%Y%m%d`

hive -e"

select b.ymd,a.show\_pv,a.show\_count,a.show\_uv,b.view\_uv,b.view\_pv,b.short\_uv,b.short\_pv,b.time,c.enter\_uv

from

(select ymd,count(uid) as show\_pv,count(distinct token)as show\_count,count(distinct uid)as show\_uv

from hds.newapplog\_seach\_live\_show

where ymd=$date and token rlike 'rec\_9\_2\_1\_0' and cv rlike '6.0.0'

group by ymd)a

left join

(select ymd,count(uid) as view\_pv,count(distinct uid)as view\_uv,count(distinct case when unix\_timestamp(exit\_time)-unix\_timestamp(enter\_time)>60 then uid end) as short\_uv,count(case when unix\_timestamp(exit\_time)-unix\_timestamp(enter\_time)>60 then uid end) as short\_pv,sum(unix\_timestamp(exit\_time)-unix\_timestamp(enter\_time)) as time

from ana.dw\_live\_view\_info\_everyday\_enter

where ymd=$date and enter='search\_new\_rec' and token rlike 'rec\_9\_2\_1\_0' and cv rlike '6.0.0'

group by ymd)b

on a.ymd=b.ymd

left join

(select ymd,sum(uv) as enter\_uv

from bi.recommend\_rec\_basic\_os

where cv rlike'6.0.0' and ymd=$date and level rlike '全部' and logid rlike 'all'

group by ymd)c

on a.ymd=c.ymd

;" >liuliang.txt

sed -i '/^WARN.\*/d' liuliang.txt

mysql -hrm-2ze6406t9hkur76rk.mysql.rds.aliyuncs.com -P3306 -uinke\_db\_user -p'Nx$X6^soiPi' -e "LOAD DATA LOCAL INFILE '/home/huanghongyu/liuliang.txt' INTO TABLE db\_inke\_pm\_strategy.liuliang CHARACTER SET utf8 fields terminated by '\t' lines terminated by '\n';"

PS：hive里面不创建表，所以hive语句里就写hive里执行的东西就可以了

**### 执行crontab实时任务**

15 08 \* \* \* sh /home/huanghongyu/liuliang.sh > /home/huanghongyu/liuliang/log\_`date -d last-day +\%Y\%m\%d`.log 2>&1

1. 然后在pwd路径执行Linux系统中crontab定时任务

## crontab –e

执行任务后，我们就可以直接在系统里查找它

## cd test

查找日志

## cat log\_\*\*\*\*\*

**通过定时脚本来实现hive查询可以有以下情况，可根据具体需求来选择：**

**一、hive中SQL的查询结果可从执行日志中查看。**

**二、将hive中SQL的查询结果插入到stg库的临时表中。**

**三、将hive中SQL的查询结果保存到hdfs上的某个文本文件，再将文本文件中的内容load到mysql的db\_inke\_pm\_strategy库中。**

大致思路都是编写一个shell脚本，让它定时跑，从而实现数据的定时更新。

**大致步骤如下：**

* 登陆hadoop（以XShell，windows为例）：进入XShell后打开一个会话
* 输入命令：**pwd**

得到当前所在目录的路径。

* 在这个目录下新建一个shell脚本。  
  输入命令：**vi test.sh**  （ vi 命令用于编辑某个文件。若test.sh已存在，则可以进行修改；若test.sh不存在，则会自动新建一个test.sh文件）
* 进入脚本的编辑状态：**i**
* **编辑脚本：将写好的shell脚本粘贴进来。本文末尾有示例。**
  + **（本文开头提到的三种查询情况，是通过执行不同的脚本来实现。）**
* 编辑完成后，按下 Esc 退出编辑状态，并输入 :wq  保存修改内容。（若不想保存修改内容，输入 :q ）
* 输入命令 **crontab -e**  ，并输入 i 进入编辑状态，添加一个定时任务。示例如下：

**30 08 \* \* \* sh /home/linyuxin/test.sh > /home/linyuxin/test/log\_`date -d last-day +\%Y\%m\%d`.log 2>&1**

* + 代表的意思分别是：

分  时  天  月  年  命令   脚本位置/脚本名称   >   输出日志的位置/日志命名   2>&1

* + 示例的含义为：  
       每天早上8：30  执行/home/linyuxin 目录下的 test.sh 脚本，将输出日志重定向到/home/linyuxin/test 目录下，且文件名称命名为 log\_昨天的日期.log  （备注：例如，若无 test 文件夹，可通过  mkdir test  命令新建文件夹）
  + 如果不想输出日志，则只需输入   30 08 \* \* \* sh /home/linyuxin/test.sh
* 编辑完成后，按下 Esc 退出编辑状态，并输入 :wq  保存修改内容。（若不想保存修改的内容，输入 :q ）
* 定时更新设置完成。
* 所需执行的脚本在每天定时执行结束后，可以在/home/linyuxin/test  目录下查看到每天的日志文件：
  + 登陆hadoop后，输入命令  cd test  ，可进入到 test  文件夹（若想退回上一级目录，输入命令 cd ..  即可）
  + 输入命令 ls  可查看 test 文件夹下的所有文件
  + 将所需要的日志文件下载到本地（以windows、XShell为例）：**sz** log\_20180524.log   
    备注：日志记录的是这个脚本执行的过程，保存日志的目的一方面是可以从日志里取出你的SQL查询结果，另一方面，如果某天的数据有问题，可以通过查看日志来定位问题，排查错误。

**三种shell脚本：**

**一、hive中SQL的查询结果可从执行日志中查看。**

脚本内容示例：（双斜杠//后是对该语句的解释说明，便于理解，脚本里应将其删掉）

date=`date -d "yesterday" +%Y%m%d`    // 设置一个date变量。例如每天早上8点跑昨天的数据，应将date的日期设置为昨天。

hive -e"

select ymd,count (uid) as show\_pv,  
count (distinct token) as show\_count,  
count(distinct uid)as show\_uv  
from hds.newapplog\_seach\_live\_show  
where ymd=$date and token rlike 'rec\_9\_2\_1\_0' and cv rlike  '6.0.0'   // 引用date变量。注意不要遗漏$符号。  
group by ymd  
  
;"

**二、将hive中SQL的查询结果插入到stg库的临时表中。**

首先，先在stg库中创建一个临时表。可参照wiki： [hive建表新人参考](http://wiki.inkept.cn/pages/viewpage.action?pageId=20983480)

然后，再编写脚本，示例如下：

date=`date -d "yesterday" +%Y%m%d`

hive -e "

**insert into table stg.xxx**   
select concat($date,'\_a'),  
count(distinct case when gender=1 then t1.uid else null end) as list\_num\_male,  
count(distinct case when gender=0 then t1.uid else null end) as list\_num\_female,  
count(distinct case when gender is null then t1.uid else null end) as list\_num\_null from   
(select uid from hallone\_rec.potential\_anchor\_pool1 where ymd=$date  
union all  
select uid from hallone\_rec.final\_potential\_anchor\_pool2 where ymd=$date  
)t1   
left join  
(select uid,gender from hdw.u\_user\_certification where ymd=$date  
)t3 on t1.uid=t3.uid

;"

**三、将hive中SQL的查询结果保存到hdfs上的某个文本文件，再将文本文件中的内容load到mysql的db\_inke\_pm\_strategy库中。**

**思路：**

1、用shell脚本把hive的查询结果输出到文本文件中。

2、根据需要处理该文本，留下要用的数据。

3、将文本文件的内容load到mysql的表中。（备注：mysql建表见wiki：[mysql建表新人参考](http://wiki.inkept.cn/pages/viewpage.action?pageId=22944234)）

4、用crontab命令定时跑这个脚本。

脚本内容示例：

date=`date -d "$1 -1 day " +%Y%m%d`     // 设置一个date变量。此处是将date的日期设置为昨天。

hive -e"

select ymd,info\_content\_type,count(distinct info\_live\_uid)   
from hds.view\_newserverlog\_rechall\_potential\_anchor\_msg   
where ymd = $date  
group by ymd,info\_content\_type

;" >> sixin\_mysql.txt           //  将hive的查询结果追加到sixin\_mysql.txt文本中。注：> 是覆盖原文件，>> 是追加内容。

sed -i '/^WARN.\*/d' sixin\_mysql.txt     // 本次查询结果的末两行有warn警告信息，此命令是删除这两行无用的信息。

mysql -[h](http://hrm-2ze6406t9hkur76rk.mysql.rds.aliyuncs.com/)(输入服务器地址，不包含括号) -P(输入端口号) -u(输入用户名) -p'(输入密码)' -e "LOAD DATA LOCAL INFILE '/home/linyuxin/sixin\_mysql.txt' INTO TABLE mysql的库名.表名 CHARACTER SET utf8 fields terminated by '\t' lines terminated by '\n';"     // 将sixin\_mysql.txt的内容load到mysql的表中