Impala作业题

业务背景

现有收集到用户的页面点击行为日志数据，数据格式如下：

用户id, 点击时间

A,2020-05-15 01:30:00

A,2020-05-15 01:35:00

A,2020-05-15 02:00:00

A,2020-05-15 03:00:10

A,2020-05-15 03:05:00

B,2020-05-15 02:03:00

B,2020-05-15 02:29:40

B,2020-05-15 04:00:00

A,2020-05-15 01:30:00

A,2020-05-15 01:35:00

A,2020-05-15 02:00:00

A,2020-05-15 03:00:10

A,2020-05-15 03:05:00

业务：

会话概念：用户的一次会话含义是指用户进入系统开始到用户离开算作一次会话，离开或者重新开始一次会话的概念

是指用户的两次行为事件差值大于30分钟，

比如以A用户为例：

第一次会话

第二次会话

判断条件是只要两次时间差值大于30分钟就属于两次会话。

需求

对用户的日志数据打上会话内序号，如下

user\_id click\_time

A,2020-05-15 01:30:00,1

A,2020-05-15 01:35:00,2

A,2020-05-15 02:00:00,3

A,2020-05-15 03:00:10,1

A,2020-05-15 03:05:00,2

B,2020-05-15 02:03:00,1

B,2020-05-15 02:29:40,2

B,2020-05-15 04:00:00,1

--创建表

drop table if exists user\_clicklog;

create table user\_clicklog (

user\_id string,

click\_time string

)

row format delimited fields terminated by ",";

--加载数据

load data local inpath '/root/impala\_data/clicklog.dat' into table user\_clicklog;

实现

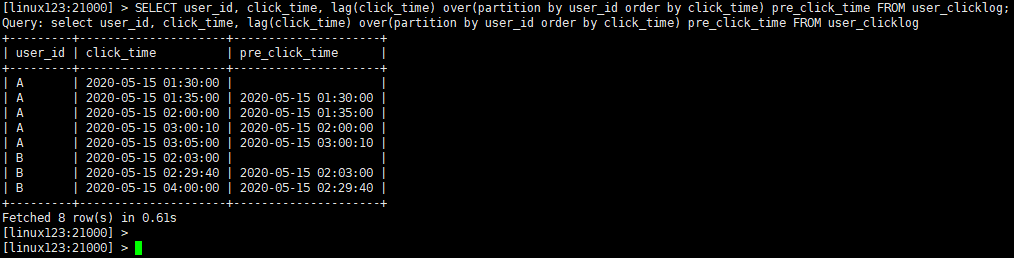
在Hive中完成数据加载

使用Impala sql完成指标统计

-- 1. 求得上一个时间

SELECT user\_id, click\_time, lag(click\_time) over(partition by user\_id order by click\_time) pre\_click\_time FROM user\_clicklog;

结果如下：



-- 2. 求两个时间差

WITH tmp AS (

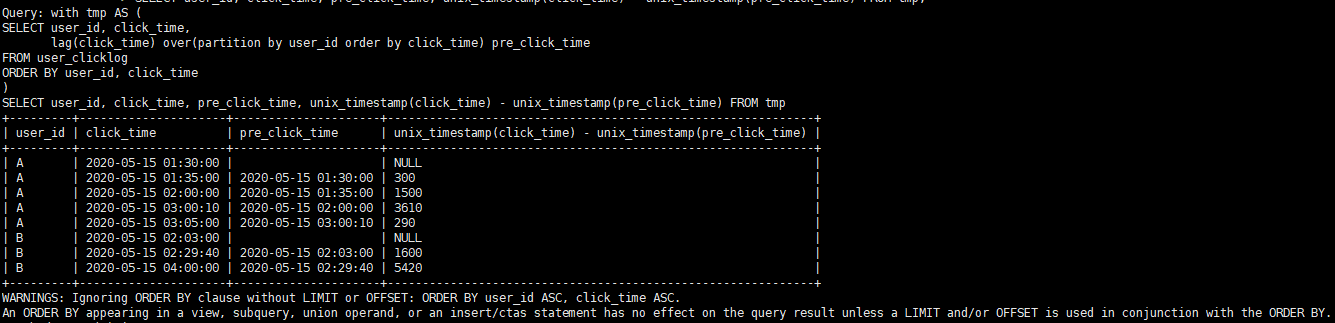
SELECT user\_id, click\_time,

lag(click\_time) over(partition by user\_id order by click\_time) pre\_click\_time

FROM user\_clicklog ORDER BY user\_id, click\_time

)

SELECT user\_id, click\_time, pre\_click\_time, unix\_timestamp(click\_time) -unix\_timestamp(pre\_click\_time) FROM tmp;

-- 结果如下：

-- 3. 标识是否超 30 分钟

WITH tmp AS (

SELECT t.user\_id, t.click\_time, t.pre\_click\_time, unix\_timestamp(t.click\_time) - unix\_timestamp(t.pre\_click\_time) diff FROM (

SELECT user\_id, click\_time,

lag(click\_time) over(partition by user\_id order by click\_time) pre\_click\_time

FROM user\_clicklog

ORDER BY user\_id, click\_time

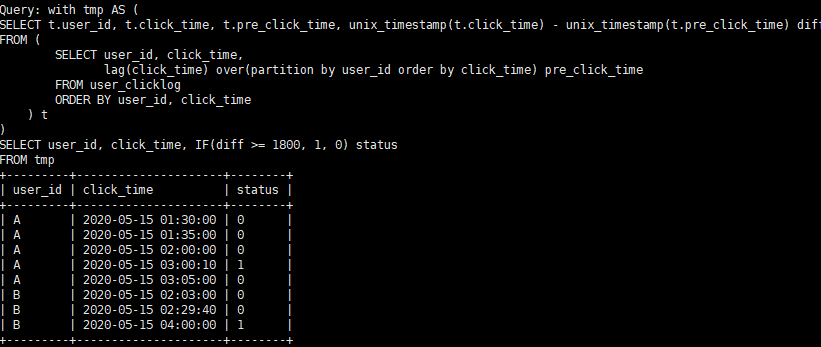
) t

)

SELECT user\_id, click\_time, IF(diff >= 1800, 1, 0) status

FROM tmp;

-- 结果如下：



-- 4. 求和分组

WITH tmp AS (

SELECT t2.user\_id, t2.click\_time, IF(diff >= 1800, 1, 0) status

FROM (

SELECT t1.user\_id, t1.click\_time, t1.pre\_click\_time, unix\_timestamp(t1.click\_time) - unix\_timestamp(t1.pre\_click\_time) diff

FROM (

SELECT user\_id, click\_time,

lag(click\_time) over(partition by user\_id order by click\_time) pre\_click\_time

FROM user\_clicklog

ORDER BY user\_id, click\_time

) t1

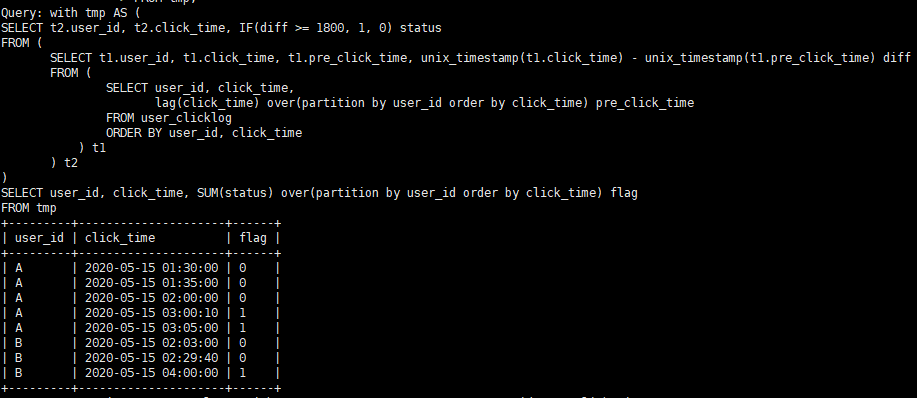
) t2

)

SELECT user\_id, click\_time, SUM(status) over(partition by user\_id order by click\_time) flag

FROM tmp;

-- 结果如下：



-- 5. 按照 user\_id 和 sum 分组，求得结果

-- flag 表示：连续求和

WITH tmp AS (

SELECT t3.user\_id, t3.click\_time, SUM(t3.status) over(partition by t3.user\_id order by t3.click\_time) flag

FROM (

SELECT t2.user\_id, t2.click\_time, IF(diff >= 1800, 1, 0) status

FROM (

SELECT t1.user\_id, t1.click\_time, t1.pre\_click\_time, unix\_timestamp(t1.click\_time) - unix\_timestamp(t1.pre\_click\_time) diff

FROM (

SELECT user\_id, click\_time,

lag(click\_time) over(partition by user\_id order by click\_time) pre\_click\_time

FROM user\_clicklog

ORDER BY user\_id, click\_time

) t1

) t2

) t3

)

SELECT user\_id, click\_time, row\_number() over(partition by CONCAT(user\_id, CAST(flag as string)) order by click\_time)

FROM tmp;

-- 结果如下：