人群找回功能说明文档

内容目录

[何忠行提出的原始需求 1](#__RefHeading___Toc502_3750176473)

[功能实现方式 1](#__RefHeading___Toc504_3750176473)

[需求分析： 2](#__RefHeading___Toc506_3750176473)

[1. 前端界面改动 2](#__RefHeading___Toc508_3750176473)

[2. 前端界面对应的后端数据库存储。 2](#__RefHeading___Toc510_3750176473)

[3. 种cookie程序的适配工作。 2](#__RefHeading___Toc512_3750176473)

[4. 广告投放时， 2](#__RefHeading___Toc514_3750176473)

[可能涉及改动的模块： 3](#__RefHeading___Toc516_3750176473)

[实现过程 3](#__RefHeading___Toc518_3750176473)

[1. 解决已有的bug 3](#__RefHeading___Toc520_3750176473)

[2. 开发过程 4](#__RefHeading___Toc522_3750176473)

[PHP、前端部分 4](#__RefHeading___Toc524_3750176473)

[Java adp部分 4](#__RefHeading___Toc526_3750176473)

[csm种cookie部分 5](#__RefHeading___Toc528_3750176473)

[bidder广告部分 5](#__RefHeading___Toc530_3750176473)

[3.初步测试 5](#__RefHeading___Toc532_3750176473)

[涉及的svn版本提交 7](#__RefHeading___Toc534_3750176473)

[进一步的测试 8](#__RefHeading___Toc648_4099207128)

[额外增加的配置说明 9](#__RefHeading___Toc650_4099207128)

[config/bidder.ini 中 9](#__RefHeading___Toc652_4099207128)

[在CSM模块中 9](#__RefHeading___Toc654_4099207128)

[在ADP代码中 10](#__RefHeading___Toc656_4099207128)

[存在的问题和未来的工作 10](#__RefHeading___Toc2487_3571631225)

# 何忠行提出的原始需求

1，在配置订单时（新作）开启访客找回功能，填写客户网站顶级域名，eg.a.com。并获取对应埋点js代码添加到客户网站a.com页面下，该js支持客户回传客户自有用户id等参数。

2，用户访问客户a.com时，记录下用户cookie1，在redis中a.com对应的集合中添加该用户：a.com:set(cookie1)

3，广告投放时，进行广告订单筛选时，如果其中某个广告开启了访客找回功能，则判断该用户cookie是否在对应a.com下面存在。如果存在即可投放。"

**1+2即可实现3**

# 功能实现方式

图片中的地址为演示系统:

linquan@quanMacBook-Pro:~ » host lq.202m.com

lq.202m.com has address 115.29.173.86

<http://lq.202m.com:6505/ad.plan.add>



# 需求分析：

### 1. 前端界面改动

是否需要在New Campaign的创建界面中，

在 Add Compaign / Add policy页面增加:

一个<input> 用于输入客户网站顶级域名

一个<input>用于客户自定义客户ID的嵌入键名

获得一段自动生成的js代码

获得一个描述，用于讲述如何将js代码嵌入到客户的网页中。

获得一个描述，用于讲述如何将客户自有的用户ID添加到回传的信息中。

### 2. 前端界面对应的后端数据库存储。

包括以下字段：

客户的主域名

客户自定义ID的键名（可以hard-code?）

是否开启访客找回功能（which means, it can be paused）

### 3. 种cookie程序的适配工作。

从后端数据库中读取客户的自有域名，用户访问客户a.com时，记录下用户cookie1，在redis中a.com对应的集合中添加该用户：a.com:set(cookie1)

### 4. 广告投放时，

进行广告订单筛选时，如果其中某个广告开启了访客找回功能，则判断该用户cookie是否在对应a.com下面存在。如果存在即可投放。"

由此可以分析，访客找回，在前端界面上是个可选的功能。仅当开启此功能时，才显示需求分析1中的前端改动的内容。

# 可能涉及改动的模块：

1. New Campaign这个网站的代码改动，（广告管理网站？），本网站使用SlightPHP架构，设计PHP代码和前端代码的改动。

2. 后端ADP的改动，ADP使用Java on Apache Thrift技术。使用RPC模式与广告管理网站的PHP实现互动。

3. 种Cookie程序的适配性检查（可能不需要改动什么）

4. 广告投放程序的改动

# 实现过程

## 1. 解决已有的bug

如下图所示，现有的代码存在不少的问题:

这里应该用<input>标签的小数点内置属性。

前端代码搞定了，现在来看PHP后端：



现在的任务，是将新作的3个字段存入数据库。

在此时，也需要考虑thrift那边代码的问题。

AdGroup.thrift需要增加新的字段:

# 访客找回功能字段

 52: bool retargeting\_enabled, # 是否启用访客找回功能

 53: string retargeting\_domain, # 客户的嵌入代码的域名

 54: string retargeting\_generate\_code # 为客户生成的嵌入代码

## 2. 开发过程

### PHP、前端部分

随后在后端中嵌入了这三个字段的处理命令，并重新生成了thrift php code

但是发现数据库存储工作是放到java中进行的，至此，php的工作也告一段落，开始转入java部分的工作。

### Java adp部分

更改了 addAdgroup(com.adp.java.AdGroup adGroup) 接口，增加了3个新的字段。

更改了 updateAdGroup /接口，增加了3个新的字段的更新工作。

修改了，fillAdGroupByResultSet，增加了3个字段的解析工作。

至此，PHP、前端和后端Java处理都已完成。

现在转入csm和bidder代码的工作中。

### csm种cookie部分

学习了C++ fcgi程序的基本使用方法。

集成了redis\_wrapper的code

经实际使用发现，只有<img src=””>标签才没有跨站问题和https禁止加载的问题。（但是会提示加载了http的内容，导致页面安全评级降低），所以使用了ContentType: image/png

完成了种cookie及redis入库的工作

csm部分使用的信息来自网站的生成代码。

实际测试中，可以观察到cookie被成功种好。信息也正确的录入了Redis: (SADD命令)

### bidder广告部分

在AdDataAgent中读入了csm种cookie的Redis，通过配置文件引入相关信息，配置的段为[RETARGETING]

另，所有代码中涉及到人群找回的部分，命名都带有retargeting

引入的基本位置在src/bidder.cc

基本逻辑是，在已经有了ad\_id之后，通过判断该ad\_id对应的policy是否启用了retargeting特性，并从redis中读出是否之前已经种下了cookie，如果启用了retargeting特性且cookie之前出现过，则选广告。否则进入err label不选广告。(REDIS: SISMEMBER命令)

设置及逻辑部分均放到src/ad\_data\_agent.cc中，在AdDataService中引入了一个新的服务，获取一个ad\_id对应的domain信息。当有domain信息时，意味着开启了retargeting特性。否则为未开启或者未设置retargeting特性。

同时，回到了ADP中，添加了此新的服务的对饮的代码。

public String getRetargetingDomain(int ad\_id) {//…}

## 3.初步测试

**以实际启动的方式实际运行了修改后的代码。部分效果截图如下：**

****

**上图为编辑/新建 policy时，新增的3个字段的UI，Generate按钮可以生成所需的嵌入代码，代码中的src和domain均为对应好的信息。**

**src中的host，对应js.conf中的csm的host，是一个新增的配置。需填写csm服务的地址。domain即为上方客户的domain.**

**在bidder启动时，读取adp中的广告信息初始化时（以及dual\_buffer工作时）可以看到这部分retargeting功能的提示信息：**

**2019-01-30 12:44:42 INFO com.rtb.impl.AdDataServiceImpl:AdDataServiceImpl.getRetargetingDomain maps domain: test.lq.com to ad\_id: 921**

**2019-01-30 12:44:42 INFO com.rtb.impl.AdDataServiceImpl:AdDataServiceImpl.getRetargetingDomain maps domain: test.lq.com to ad\_id: 922**

**2019-01-30 12:44:42 INFO com.rtb.impl.AdDataServiceImpl:AdDataServiceImpl.getRetargetingDomain maps domain: test.lq.com to ad\_id: 923**

**如上所示，test.lq.com下，有921，922，923这3个广告启用了retargeting特性，因此这3个广告除之前的filter，新增了retargeting\_filter: 必须看cookie是否出现过。**

**以上就是初步测试的内容。**

**现在的开发版本预览状态:**

**bidder工作在:**

**ssh** **root@112.124.33.66****:/opt/baic/bidder\_text2018/bidder**

**adp工作在:**

**ssh** **linquan@115.29.165.122****:/opt/baic/ADP\_yinni/adp.jar**

**网站工作在:**

**ssh** **linquan@115.29.173.86****:/data/www/bs\_yinni\_adpush\_linquan/**

**with nginx conf file: /data/nginx/thinkphp\_linquan.conf**

## 涉及的svn版本提交

**URL: svn://115.29.174.6:36901/bcsvn/ams\_v1/bs\_yinni\_adpush**

**------------------------------------------------------------------------**

**r36079 | linquan | 2019-01-29 22:18:58 +0800 (二, 29 1 2019) | 1 line**

**change script generating logic**

**------------------------------------------------------------------------**

**r36063 | linquan | 2019-01-23 15:00:27 +0800 (三, 23 1 2019) | 1 line**

**retargeting feature completed**

**------------------------------------------------------------------------**

**r36056 | linquan | 2019-01-22 16:30:50 +0800 (二, 22 1 2019) | 1 line**

**add retargeting feature**

**URL: svn://115.29.174.6:36901/bcsvn/adp/ADP\_yinni**

**------------------------------------------------------------------------**

**r36081 | linquan | 2019-01-30 09:09:10 +0700 (Wed, 30 Jan 2019) | 1 line**

**retargeting feature tested commit**

**------------------------------------------------------------------------**

**r36078 | linquan | 2019-01-29 16:33:17 +0700 (Tue, 29 Jan 2019) | 1 line**

**retargeting: add service in addata**

**------------------------------------------------------------------------**

**r36065 | linquan | 2019-01-23 15:13:01 +0700 (Wed, 23 Jan 2019) | 1 line**

**replace from sun.misc.Lock to java.util.concurrent.locks.ReentrantLock**

**------------------------------------------------------------------------**

**r36064 | linquan | 2019-01-23 14:50:05 +0700 (Wed, 23 Jan 2019) | 1 line**

**add retargeting feature in addGroup, remove thrift date annotations**

**URL: svn://115.29.174.6:36901/bcsvn/rtb/trunk**

**------------------------------------------------------------------------**

**r36080 | linquan | 2019-01-29 22:02:59 +0700 (Tue, 29 Jan 2019) | 1 line**

**a untest prototype of retargeting feature(renqun zhaohui)**

**------------------------------------------------------------------------**

**r36075 | linquan | 2019-01-28 17:27:29 +0700 (Mon, 28 Jan 2019) | 1 line**

**1. fix zuopeng's typo, 2. add 'make debug' cmdline feature, you can type make debug to add BIDDER\_DEBUG flag now**

## 进一步的测试

进一步实际上是要验证如下情形：

1. 生成的代码可以种植cookie，记录cookie于redis库中。

经测试：可以生成cookie并正确留存，cookie在展示广告时，可以被bidder获取。

2. 在启用了人群找回的policy中的ad\_creative，只有在“是访客”的情况下才（可能）会展示。在“不是访客”的情况下一定不展示。为方便测试出展示的情况，频次控制等策略应当调整至最有利于展示广告。

经测试：只有在“是访客”的情况下才（可能）会展示。在“不是访客”的情况下一定不展示。

3. 当在Policy中（暂时）关闭retargeting（人群找回） feature时，（在配置更新后），bidder将把此policy的retargeting特性禁用。

经测试：暂时关闭此特性后，此policy不再进行人群找回过滤。

4. 人群找回特性不影响其他过滤器，如FSC等。

经测试：人群找回特性不影响FSC过滤，推定也不影响其他过滤器。因为FSC是其最靠近其判断逻辑的过滤器。

## 额外增加的配置说明

### config/bidder.ini 中

增加RETARGETING分组配置， 此配置决定了和CSM交互时使用哪个redis数据库，因此这里的配置需与CSM保持一致。

[RETARGETING]

host = 127.0.0.1

port = 6389

db\_name = 7

connect\_timeout = 1000

operate\_timeout = 1000

;the cookie key is the domain passed in the embed script, like: op.vm1.cn

cookie\_key = mkl\_sid

;the cookies are like: v1-744df85c8111439ae305332222c145f5,v1-e09c414c9ba0baf103e312e3a7d28d1e,...

cookie\_prefix = v1

password = bcdata@2701

### 在CSM模块中

增加了redis配置，此配置应当与bidder中的RETARGETING分组保持一致。

[csm]

cookie\_prefix = v1

#30\*24\*60\*60

cookie\_timeout = 2592000

media\_cookie\_path = /

media\_cookie\_domain = 112.124.33.66

[redis]

redis\_host = 127.0.0.1

redis\_port = 6389

redis\_db\_name = 7

redis\_password = bcdata@2701

connect\_timeout = 1000

operate\_timeout = 1000

### 在ADP代码中

增加了新的配置文件：

* hibernate.cfg.xml

 这是Hibernate ORM的基本配置

* hibernate.properties

 这是Hibernate的连接配置

* hikari.properties

 这是HikariCP的连接配置（使用HikariCP取代了旧的连接池）

## 存在的问题和未来的工作

1. 单线条过滤

人群找回功能只是提供了一个过滤器，串在了一连串的已有的过滤器上。这样做的逻辑简单可靠，但是不一定是设计者的最终目的。所以也许需要改动过滤的方式。(2019年2月18日，已进行调整。)

2. 对于“访客”的广告投放，并未留下显著的日志。

也许设计者希望看到一个管理界面，了解有多少访客被投放了广告。

3. “嵌入代码”的操作未提供说明，未提供https方式的cookie种植。

未提供说明可能使某些操作者感到疑惑。

未提供https方式的cookie种植地址可能降低客户网页的安全评级。（加载了不是https的页面内容）

2019年2月19日的补充说明

启用访客找回功能的条件，如果cookie是目标人群cookie，只需要进一步判断终端（为了适配素材大小）和频次，满足的话这个广告就不被过滤掉，跟其他满足条件的广告一块进入到最后判断优先级的环节。

在info.js参数未给定ad\_id具体值的时候，实际上进行的是从全部可用的广告中选出可用广告的过程，此时主要通过了以下过滤器：

* NoBidFilterByAdConstraint —— 基本广告条件约束过滤器
* FilterByAdslIP —— ADSL、IP过滤器
* FilterByAdOperatorArea
* FilterByUserLacAndCellid
* FilterByAppChannels
* FilerByAdFrequence —— 频次过滤器
* FilterByAdSmoothStatus —— 平滑过滤器
* FilterByUserTags

本次实现的实际上也是一种访客找回过滤器，因此访客找回过滤器的代码位于基本广告条件过滤器之后，并且由于何忠行的设计，如果通过了这个过滤器的人群找回(retargeting)广告，则获得免于被平滑过滤器(smooth filter)过滤的权利。

实现中，设计了before和after两个状态存储，前后做一个差集，计算一下哪些被平滑过滤器过滤了，然后用预先生成的可用人群找回广告补回去。这样如果以后有多个类似的免于处理的过滤器，在写这些过滤器的时候就不必考虑人群找回的例外特性，只需要考虑自己的过滤规则即可。将相应的before和after放置到代码块的前后位置即可。

本次改动的测试：

1. （相比较与前次的测试）移除了aid（即为ad\_id），使得进行选广告过滤。测试时选广告基本功能正常，可以选出广告。在满足了retargeting广告的条件之后，可以选出retargeting的广告。

2. 利用移除浏览器cookie的办法，在移除了retargeting广告的条件后，发现retargeting广告已经从可用广告中erase掉。符合设计要求。即不满足条件则不选出retargeting广告。

3. retargeting广告满足条件后，测试发现的确免于被平滑过滤器过滤。

完成这些测试后，代码已提交:

r36097 | linquan | 2019-02-18 19:08:30 +0700 (Mon, 18 Feb 2019) | 1 line

retargeting feature enabled when adid is not set

本文档完

林泉

2019年2月15日