[设计文档]

parent nodes:

月光宝盒离线计算设计文档

整体设计参见清无文档，此文档仅描述离线自身部分内容。http://red.lzdp.us/projects/effect-platform/wiki/Effect_Sprint1_Detail_Design

hadoop根目录位置为：hdfs_base_dir=/group/tbads/sds/linezing/effect_platform

一期不足和遗留未做功能：

只提供一天效果的计算，不支持多天效果

不支持按照4淘日志切分进行数据计算

同优先级来源效果归属规则中平均归属equal，不提供支持

2.0: 代码都在svn，考虑迁移到git

其他来源，其他路径计算和产品需求不符合

店铺收藏指标目前无法计算

下线任务，上线任务 只判断昨天？这个是不是有问题

多plan同时跑的问题？

1.参考资料

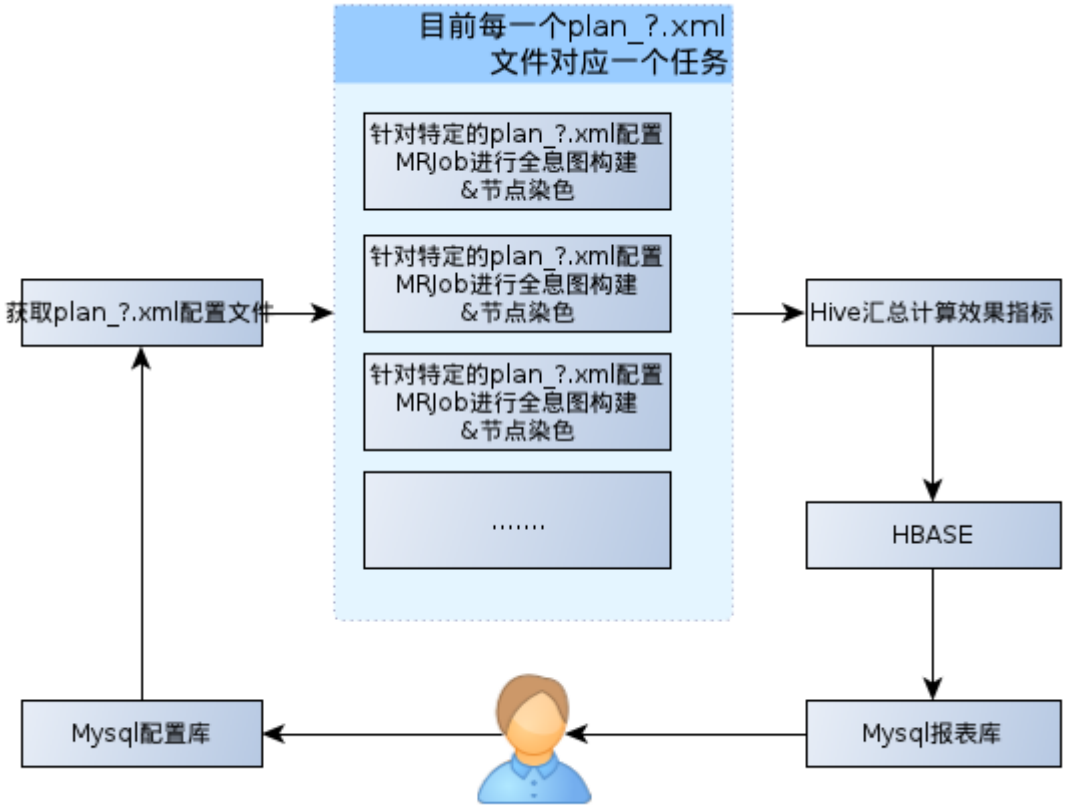
清无设计文档：http://red.lzdp.us/projects/effect-platform/wiki/Effect_Sprint1_Detail_Design

2.离线部署

参见'清无设计文档' 1.2节

lz_effect_platform_start为天网起始节点

3.任务部署调度流程



4.数据交互

参见'清无设计文档' 5节

包括三部分：配置文件获取，数据产出入HBASE，广告信息datax直接入mysql

5.数据计算调度模块

1. 新plan自动运行：

   每天从/home/lz/effect_platform/conf/report/%(YYYY)s/%(MM)s/%(DD)s得到新审批通过的配置文件，上传到云梯/group/tbads/sds/linezing/effect_platform/hadoop/effect_platform

   程序执行时，遍历所有plan文件，并行启动任务。

2. plan自动下线

   每天从/home/lz/effect_platform/conf/overdue/%(YYYY)s/%(MM)s/%(DD)s得到需要下线的配置文件，从云梯目录删除对应文件。

6.用户轨迹全息图构建和染色

参见'清无设计文档' 3节

7.效果指标归属逻辑

染色：用户轨迹全息图中符合路径规则的后续访问节点。

引导宝贝/店铺页面：效果页后续访问的第一个店内页面。

归属优先级：用户指定路径的优先级，数值越小优先级越高。

归属顺序：last, first, all, equal(不支持)。其中last, first是按照归属点的时间戳时间判定。

归属点：用户指定的路径中的某一跳。用此节点时间戳作为路径的时间戳进行比较。（1.0默认为效果页）

7.1.流量指标

用户访问路径内染色的节点认为是流量效果节点，进行计算。

7.2.成交归属逻辑

参见'清无设计文档' 4.3节

7.3.收藏归属逻辑

同成交逻辑

7.4.购物车归属逻辑

同成交逻辑

7.5.淘外成交归属逻辑（仅etao有此需求）

对etao浏览日志url进行trade_track_info的提取

成交表：s_dw_sh_etao_pay_order，取相同trade_track_info的数据，按照时间戳顺序归属给最近的浏览日志。

7.6.返利数据归属逻辑（需求实现，此处仅作为文档记录）

返利成交表：s_dw_etao_cps_order 中返利数据(is_settle=1 and gmv_num>0)，包含 亿起发(source=1）站外B2C(source=4）和淘客(source=2）这三种类型.

亿起发和站外B2C 都是站外引导

归属时使用etao浏览日志url提取出的 trade_track_info 字段。 成交浏览trade_track_info相同数据，按找时间戳顺序归属给最近的浏览日志。

淘客 属于 站内效果,归属时使用买家user_id

按照user_id，按时间戳归属给宝贝浏览页面的user_id相同的浏览记录

8.核心计算模块（构建全息图，染色，归属）

核心模块使用MR任务进行开发，部分计算使用后端实时提供接口共用。

功能描述：完成用户全息轨迹图的构建，归属用户来源路径匹配规则，完成染色

需要参考章节：2，3，4

8.0.输入和输出的规定

1. 最小输入单元

流量access日志

字段名称	字段类型	字段说明
timestamp	bigint	时间戳h
url	string	
refer_url	string	
shop_id	string	店铺ID
auction_id	string	宝贝ID
user_id	string	访客ID
cookie	string	cookie用来标识一个访问用户（atpanle日志中用mid）
session	string	一次会话的标识（atpanle日志中用sid）
cookie2	string	用来计算uv使用（atpanle日志中用mid_uid, 其他情况直接使用cookie）

成交trade日志

字段名称	字段类型	字段说明
gmv_trade_timestamp	bigint	拍下时间戳
shop_id	string	店铺ID
auction_id	string	宝贝ID
user_id	string	访客ID
ali_corp	bigint	网站类型（0 未知或非阿里系 1 淘宝 2 天猫 3 一淘 4 聚划算）
gmv_trade_num	int	拍下笔数
gmv_trade_amt	float	拍下金额
gmv_auction_num	int	拍下件数
alipay_trade_num	int	成交比数

alipay_trade_amt	float	成交金额
alipay_auction_num	int	成交件数

收藏collect日志

字段名称	字段类型	字段说明
collect_timestamp	bigint	收藏时间戳
type	int	收藏类型（0宝贝 1店铺）
shop_id	string	店铺ID
auction_id	string	宝贝ID(店铺收藏此处留空)
user_id	string	访客ID
ali_corp	bigint	网站类型（0 未知或非阿里系 1 淘宝 2 天猫 3 一淘 4 聚划算）
collect_num	int	收藏次数

购物车cart日志

字段名称	字段类型	字段说明
cart_timestamp	bigint	添加购物车时间戳
shop_id	string	店铺ID
auction_id	string	宝贝ID(店铺收藏此处留空)
user_id	string	访客ID
ali_corp	bigint	网站类型（0 未知或非阿里系 1 淘宝 2 天猫 3 一淘 4 聚划算）
cart_num	int	购物车宝贝件数

站外成交out_trade日志(没有宝贝件数，因为一淘从来不算)

字段名称	字段类型	字段说明
gmv_create_ori_ts	bigint	时间戳
trade_no	string	交易号
trade_track_info	string	订单跟踪id
seller_id	string	卖家ID
auction_id	string	宝贝ID
user_id	string	访客ID
gmv_trade_num	int	拍下笔数
gmv_trade_amt	float	拍下金额
pay_trade_num	int	成交比数
pay_trade_amt	float	成交金额

2. 输入单元的扩展属性

每个表最后有两个附加字段useful_extra，extra。最终都会附加到产出表里

字段名称	字段类型	字段说明
useful_extra	string	使用key+ctrlC+value+ctrlB+...方式存储.key为字段名，value为内容
extra	string	自身使用ctrl+B分割

3. 最小输出单元

字段名称	字段类型	字段说明
index_root_path	string	路径列表 （留空）
ts	string	时间戳
analyzer_id	bigint	制定推广计划的用户ID
plan_id	bigint	推广计划ID
src	string	来源路径实例，可作为ID
url	string	url
refer	string	refer url
shop_id	string	店铺ID
auction_id	string	宝贝ID
user_id	string	访客ID
ali_corp	bigint	0 未知或非阿里系 1 淘宝 2 天猫 3 一淘 4 聚划算
cookie	string	cookie用来标识一个访问用户（atpanle日志中用mid）
session	string	一次会话的标识（atpanle日志中用sid）
cookie2	string	用来计算uv使用（atpanle日志中用mid_uid, 其他情况直接使用cookie）
is_effect_page	bigint	是否为效果页 1为true
ref_is_effect_page	bigint	是否为效果页下一跳 1为true
is_leaf	bigint	是否为树的叶子节点 1为true
jump_num	int	引导宝贝相对效果页跳数(从0开始，二期支持到4)
index_type	int	指标类型标识(0非店铺，1单品， 2单品同店，3单店， 4淘外成交(仅etao才有)，5单品其他， 6单店其他)
pv	float	
gmv_amt	float	拍下金额
gmv_auction_num	float	拍下件数
gmv_trade_num	float	拍下笔数
alipay_amt	float	成交金额
alipay_auction_num	float	成交件数
alipay_trade_num	float	成交笔数
item_collect_num	float	收藏宝贝数
shop_collect_num	float	收藏店铺数
cart_auction_num	float	购物车宝贝数

4. 输出单元的扩展属性

2.0只支持浏览日志扩展字段的输出

字段名称	字段类型	字段说明
access_useful_extra	string	从输入直接继承

access_extra	string	从输入直接继承

8.1.hadoop任务之间数据交互规则

统一使用proto buf进行数据交互：LzEffect.proto （规则如下）

```
 0 package lz.dw.effect.proto;
1
2     option java_package = "com.lz.dw.effect.proto";
3     option java_outer_classname = "LzEffectProto";
4
5     message TreeNodeValue{
6         message KeyValueS {
7                optional string key = 1;
8                optional string value = 2;
9         }
10         message KeyValueI {
11                optional string key = 1;
12                optional int32 value = 2;
13         }
14
15         optional int64 ts = 1; // ■■■■■■■
16         optional int32 log_type = 2; // ■■ 0  ■■ 1 ■■ 2 ■■■■ 3 ■■■ 4
17
18         optional string index_root_path = 3; // ■■root_path
19         optional bool is_leaf = 4; // ■■■■■■■■
20         optional bool is_root = 5; // ■■■■■■■
21
22         optional string url = 6;
23         optional string refer = 7;
24         optional string shop_id = 8;
25         optional string auction_id = 9;
26         optional string user_id = 10;
27         optional int32 ali_corp = 11; // ■■■■4■■■■0 ■■■■■■■■ 1 ■■ 2 ■■ 3 ■■ 4 ■■■■
28
29         optional string cookie = 12;
30         optional string session = 13;
31         optional string cookie2 = 14;  // ■■uv■■■■
32
33         message TypeRef {
34            optional int32 analyzer_id = 1;  // ■■id
35            optional int32 plan_id = 2;  // ■■id
36
37            optional bool is_matched = 3; // url match■■■■
38            optional int32 rtype = 4; // ■■■■■■■■id
39            optional int32 ptype = 5;  // ■■■■id
40            repeated KeyValueI source_info = 6; // ■■match■■■■■■■PTLogEntry■source
41            repeated KeyValueS captured_info = 7; // ■■match■■■■■■■PTLogEntry■captured
42
43            message TypePathInfo {
44                optional string src = 1;
45                optional int64 first_ts = 2;
46                optional int64 last_ts = 3;
47                optional int32 priority = 4;
48                optional bool is_effect_page = 5; // true ■■■■
49                optional bool ref_is_effect_page = 6; // true ■■■■■■■
50                optional int32 first_guide_jump_num = 7; // ■■■■■■■■■■
51                optional string first_guide_auction_id = 8; // ■■■■■■■■■■ID
52                optional string first_guide_shop_id = 9; // ■■■■■■■■■■ID
53                optional int32 last_guide_jump_num = 10; // ■■■■■■■■■■
54                optional string last_guide_auction_id = 11; // ■■■■■■■■■■ID
55                optional string last_guide_shop_id = 12; // ■■■■■■■■■■ID
56            }
57            repeated TypePathInfo path_info = 8; // ■■■■■■■
58         }
59         repeated TypeRef type_ref = 15; // ■■■■■■■■plan■■■■■■■■
60
61         repeated KeyValueS access_useful_extra = 16;
62         optional string access_extra = 17;
63     }
```

提供LzEffectProtoUtil类，提供两种序列化方法

二进制序列化：serialize，deserialize

字符串序列化：toString，fromString

不需要的字段不需要添加内容

8.2.单元测试

MR任务全部使用MRUnit进行单元测试。

8.3.调度使用方法

参数： "Usage : EffectOwnership【InputAccessPath】【OutputPath】【config_paths】【numOfMappers】【numOfReducers】【mid_path】<gmv=GmvLogPath> <collect=CollectLogPath><period=1(归属周期，默认1)><tree_split=none(默认)><runner_job=1/2><files=(本地路径)>

8.4.特殊指标定义

被引导页面的宝贝/店铺ID获取方法：

从效果页开始，查看效果页是否有auction_id/shop_id。有则填写jump_num=0

效果页没有则看效果页下一跳是否有auction_id/shop_id。有则填写jump_num=1

以此类推，一直到jump_num=4为止，还没有的话则不填写引导宝贝/店铺信息。jump_num=-1。

8.5.详细计算流程

job1:URL MATCHER & Forest & Colorized

1. 类： com.lz.dw.effect.EffectNodeFinder
2. 开发设计:
   1. map阶段完成 url matcher 任务。
      调用URLMatcher.java模块【鸣柯】，获取type_ref, ali_corp信息
      key = new TextPair(new Text(mid+"_"+sid), new Text(ts))
      value = LzEffect.proto
   2. sort & partition
      mid+"_"+sid 分组， ts 顺序
   3. reduce阶段完成 forest & colorized 部分
      调用外部模块【清无&民瞻】，参见清无文档3.1
      只有染色节点才输出数据
      key = new Text("")
      value = LzEffect.proto（toString）

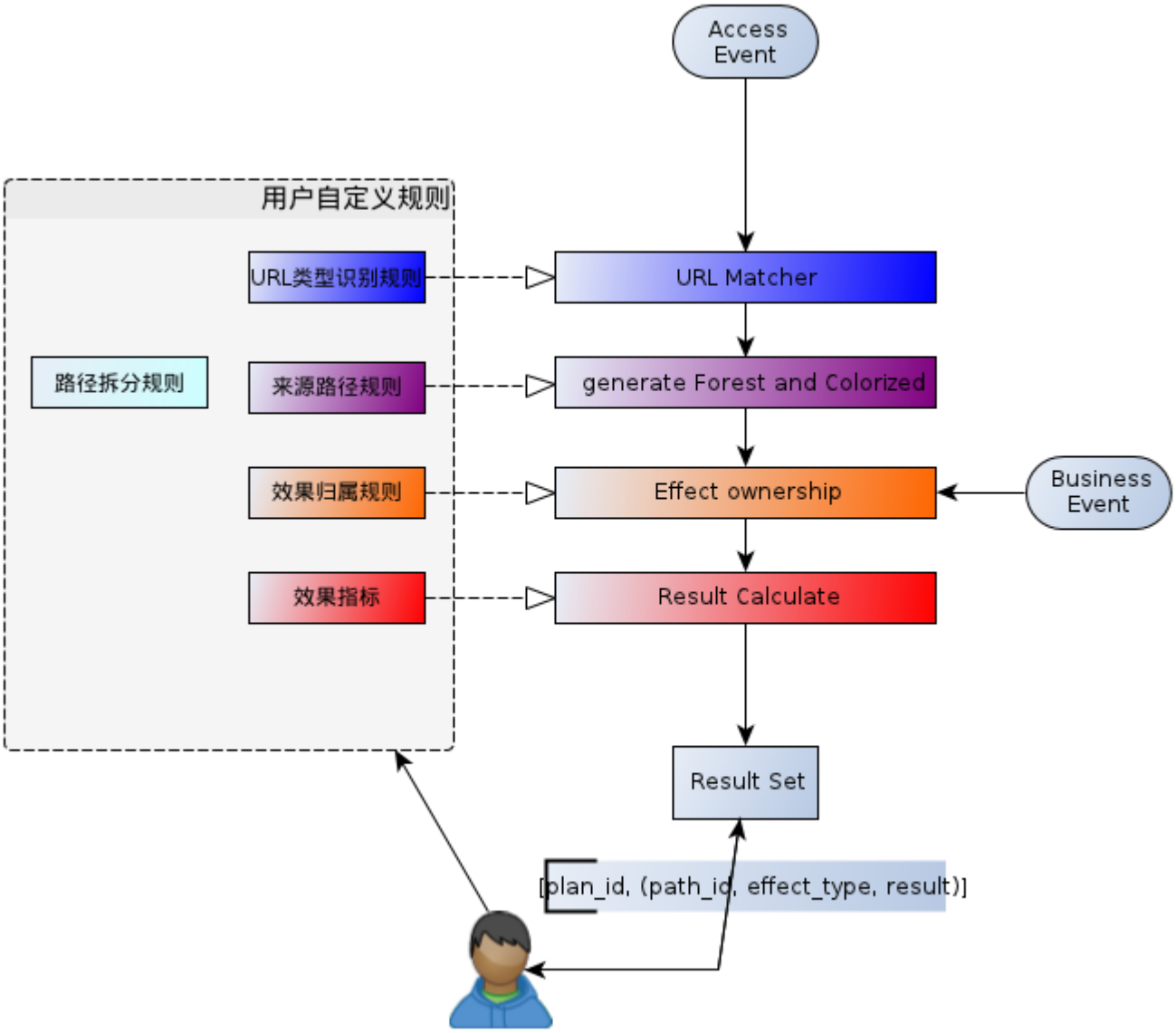
job2:EFFECT OWNERSHIP

1. 类： com.lz.dw.effect.EffectOwnership
2. 开发设计:
   1. map阶段
      对浏览日志，成交日志进行ETL
      浏览日志仅保留已经标颜色的部分
      成交全部保留
      key = new TextPair(user_id, ts)
      value = LzEffect.proto
   2. sort & partition
      user_id, auction_id 分组， ts 倒序
   3. reduce阶段
      通过 同优先级来源效果归属规则 成交效果归属逻辑 进行归属
      保留成交list，按归属规则找到其他店铺path_info，间接成交path_info，直接成交path_info。最终把成交属性附加到某个树节点中。

8.6:性能优化方案

1. 方案
   Finder时reduce先把session存入队列，如果是否有效果页is_matched=true进行建树处理，否则抛弃。
   Finder时reduce先把session存入对垒，按照plan多少进行多次重复建树操作。完成一次输出一次。（因为染色节点毕竟是少数，所以产出条数会减少很多）
2. 提升效果
   TODO

9.数据计算



9.1.ETL操作

1. 流量日志
   规则：refer like '/1.gif%'
   上游：r_atpanel_log
   产出：lz_fact_ep_browse_log
   分区：dt=20120603，logsrc='atpanel'
2. 点击日志
   规则：url like '%ju.atpanel.com%' and url like '%tb_market_id%' and not refer like '/1.gif%'
   上游：r_atpanel_log
   产出：lz_fact_ep_ad_click_log
   分区：dt=20120603，logsrc='atpanel'
3. 成交日志
   必须字段：gmv_ts(拍下时间戳)，alipay_ts，shop_id，auction_id，user_id，gmv_amt，gmv_auction_num，alipay_amt，alipay_auction_num，ali_corp
   上游：lz_fact_user_trade，r_bmw_shops(lz_dim_sellers)
   产出：lz_fact_ep_trade_info
   分区：dt=20120603，round=1，logsrc='taobao'
4. 表详情见Power Designer

9.2 输出中间表

输出中间表lz_fact_ep_ownership，参见【核心计算模块】

9.3.效果指标计算

1. 需要关联点击日志，汇总点击数据

2. 最终结果

plan_id, day, dim, 指标

plan_id, day, dim, src_id, src_ext, path_id, 指标

指标有如下：http://red.lzdp.us/projects/effect-platform/wiki/_Effect_Sprint1_Design_index

可以汇总为：广告位3个指标，效果页5个指标， 效果指标42个（效果页含不同页面前端可适配）。

9.4. 广告点击数据处理

1. 开发设计:
2. 参数:【InputPath】【OutputPath】【numOfMappers】【numOfReducers】【config_paths】
3. 进行广告点击数据的计算
4. 表1：lz_fact_ep_ad_click_info_temp

字段名称	字段类型	字段说明
ts	string	
analyzer_id	bigint	制定推广计划的用户ID
plan_id	bigint	推广计划ID
adid	string	外投广告id
cookie	string	用来计算uv（取自mid_uid 有uid就是uid，没有就是mid）
pv	float	

1. 表2：lz_fact_ep_ad_click_info

字段名称	字段类型	字段说明
ts	string	
analyzer_id	bigint	制定推广计划的用户ID
plan_id	bigint	推广计划ID
adid	string	外投广告id
pv	float	
uv	float	

9.5. 广告数据mysql入库

1. 导入r_act_media_adid_site表到前端mysql
2. 表：lz_fact_ep_ad_config

字段名称	字段类型	字段说明
ad_id	string	外投广告id
ad_site_name	string	
ad_page_name	string	
ad_position_name	string	
ad_creative_name	string	
ad_activity_name	string	
ad_activity_id	string	

9.6. 结果数据导入到hbase

1. hbase定义见清无设计文档
2. 进行业务指标到hbase的指标映射。hbase指标id可参见：http://red.lzdp.us/projects/effect-platform/wiki/_Effect_Sprint1_Design_index
3. 例子：

如direct_pv 需映射到：200，405

如indirect_pv 需映射到：211，300，416

10.SPM信息

* 站点ID信息：s_spm_site
* 页面信息: s_spm_page
* 模块信息: s_spm_module
* 位置信息: