**数据分析结构体系文档**

作者: 数据分析组

日期: 2014/02/10

（本文档只适用于中科云智内部交流）

目录

[1. 数据源 3](#_Toc379890732)

[2. 结构体系 3](#_Toc379890733)

[3. 模块介绍 4](#_Toc379890734)

[3.1 DSP实时数据接口 4](#_Toc379890735)

[3.1.1 存储转发 4](#_Toc379890736)

[3.1.1.1 存储转发电商数据 4](#_Toc379890737)

[3.1.1.2 存储转发百度数据 5](#_Toc379890738)

[3.1.2 响应查询 5](#_Toc379890739)

[3.2 Radius客户端 6](#_Toc379890740)

[3.3 分析平台 7](#_Toc379890741)

[3.3.1 平台架构 7](#_Toc379890742)

[3.3.2 程序入口 7](#_Toc379890743)

[3.3.3 主要功能 7](#_Toc379890744)

[3.3.4 整体设计 7](#_Toc379890745)

[3.3.5 接口模块 8](#_Toc379890746)

[3.3.5.1 全流量样本 8](#_Toc379890747)

[3.3.5.2 网站分类 9](#_Toc379890748)

[3.3.5.3 搜索引擎 9](#_Toc379890749)

[3.3.5.4 共现词 9](#_Toc379890750)

[3.3.5.5 相似网站 9](#_Toc379890751)

[3.3.5.6 电商 10](#_Toc379890752)

[3.3.5.7 用户行为 10](#_Toc379890753)

[4. 总结 11](#_Toc379890754)

## 数据源

现有的数据主要有下面三种类型，采集方式如下：

1. access数据

通过在用户的浏览器中插入js, 从而读取用户正在浏览的页面，记录该用户的user agent和该页面的url, title, refer等信息，同时还包括时间，地域，源IP, 发送数据的机器IP, cookie等信息。每隔5分钟向一批IP插入js.

1. nstat数据

直接在分光点上采集数据，包括PC端和移动端两种类型的数据，主要有时间，地域，源IP, 目标IP, host, url, refer, uid等信息。采集电商数据的比例是100%，百度搜索数据是10%，其他类型数据是1%，这个比例会根据需要调整。

1. clk数据

通过在用户的浏览器中插入js, 捕捉用户的点击事件，会记录当前页面的url和refer, 用户点击的页面url等信息。

## 结构体系



图1. 数据结构体系图

图1是整个数据结构体系图，包括数据采集、数据分析和平台界面上各种功能数据的呈现等过程，以及DSP广告投放整个过程，下面将分别来介绍各个模块。

## 模块介绍

### DSP实时数据接口

Rest Server 主要负责两个任务

1. 存储转发：接收用户电商数据、百度搜索实时数据数据，将数据保存到SSDB数据库中。实时数据接收后将新输入发送给RTB模块
2. 响应查询：响应RTB模块的查询请求。

#### 存储转发

##### 存储转发电商数据



添加电商信息以及向RTB发送电商信息均通过http协议实现。

下表是添加电商信息的协议格式：

|  |  |
| --- | --- |
| 输入 | http://171.8.71.147:8000/dmp/add\_tbi?s=<base64\_encode\_str>  base64\_encode\_str=uid \tTBITEM \t atime \t item\_id \t title \t cats\ titemkeys|attrkeys |
| 输出 | {“err”:0} |

下表是向RTB发送电商信息的协议格式：

|  |  |
| --- | --- |
| url | http://171.8.71.147:18019/rt |
| 数据 | {"cates":["女装男装","女式上装","衬衫"],"attrs":["平跟","明星款","真皮"],"items":["女靴","马丁靴"],"uid":"949fb562d63fa0f1a7b4f6e58fe24bde"} |

向SSDB存储的数据如下:

SSDB key:value标示：

ec:i:[userid] 用户商品标签

ec:a:[userid] 用户属性标签

ec:c:[userid] 用户类别标签

tg:i:[md5tag]商品标签用户

tg:a:[md5tag] 属性标签用户

tg:c:[md5tag]类别标签用户

##### 存储转发百度数据

添加百度信息以及向RTB发送百度信息均通过http协议实现。

下表是添加百度信息的协议格式：

|  |  |
| --- | --- |
| 输入 | http://171.8.71.147:8000/dmp/add\_bd?s=<base64\_encode\_str>  base64\_encode\_str=uid \tBAIDU\t atime \t searchstring \t kw1 kw2 |
| 输出 | {“err”:0} |

下表是向RTB发送百度信息的协议格式：

|  |  |
| --- | --- |
| url | http://171.8.71.147:18019/baidu |
| 数据 | {"bdcates":["数码"],"bdkeys":["手机","三键"],"uid":"ba35edeabd9001767724652e5c273ed4"} |

向SSDB存储的数据如下:

SSDB key:value标示：

bd:c[userid] 百度用户类别标签

bd:k:[userid] 百度用户关键词标签

bdtg:c[md5tag] 百度类别用户标签

bdtg:k[md5tag] 百度关键词用户标签

#### 响应查询



RestServer 接收RTB的用户或标签查询请求，查找ssdb并返回结果

RTB查询请求也是通过http协议实现，下表是根据标签查询的输入输出：

|  |  |  |
| --- | --- | --- |
| 输入 | http://171.8.71.147:8000/tagusers.json?tag=女装 | |
| 输出 | {“bdcates\_uid”:[],“bdkeys\_uid”:[],"attrs\_uid":[],"cates\_uid":[],"err":0,"items\_uid":[],"tag":"女装"} | |
| 字  段  描  述 | tag | 查询标签名称UTF-8 编码 |
| items\_uid | 商品标签用户列表 |
| attrs\_uid | 属性标签用户列表 |
| cates\_uid | 类别标签属性列表 |
| err | 错误号 0 成功 |

下表是根据用户ID查询的输入输出：

|  |  |  |
| --- | --- | --- |
| 输入 | http://171.8.71.147:8000/usertags.json?uid=008613f6e1779285f1e1026eadead5df | |
| 输出 | {“bdcates”:[],“bdkeys”:[],"attrs":[],"cates":[],"err":0,"items":[],"uid":"008613f6e1779285f1e1026eadead5df"} | |
| 字  段  描  述 | uid | 用户ID |
| items | 商品标签列表 |
| attrs | 属性标签列表 |
| cates | 类别标签类别 |
| err | 错误号 0 成功 |

### Radius客户端

客户端启动后连接浙江SIG Rasius服务器，发送验证及心跳信息，实时获取服务器推送过来的Radius用户数据(用户ID、上/下线以及心跳信息)，将相关信息保存到memcache缓存服务器中并定时写入本地文件中。



从Radius服务器接收数据通过TCP 协议实现。接收数据格式如下：

typedef struct RadiusInfo

{

unsigned long user\_ip; /\* 用户IP\*/

unsigned short user\_status; /\* 1：start，2：stop, 3：ka\*/

unsigned short user\_name\_len; /\*帐号长度\*/

unsigned char user\_name[0]; /\* 帐号信息，大小不固定，由user\_name\_len决定\*/

} RADIUSINFO;

向Memcache存储数据格式如下:

用户上线添加key:valueuserip:useiduserid:userip

用户下线删除key:value userip userid

写入本地文件格式如下：

IP地址\t 在线\离线状态\t 用户ID\t 接收时间戳

183.155.88.212 1 shJHkuoXXAqo59/qEDFFnQ== 2014-01-23 15:27:16

122.242.84.167 1shJHkuoXXAoYqJLyjlcMpA== 2014-01-23 15:27:16

183.152.8.53 1 c0sdDEpvEh39huVbnNjTHQ== 2014-01-23 15:27:16

### 分析平台

在下面的章节中，只基于nstat数据进行介绍。

#### 平台架构

以内网hadoop集群为例，共有20台1u的机架服务器，IP地址为192.168.210.201-208, 192.168.210.210-221, 其中192.168.210.205是主节点，192.168.210.218-221 4台服务器主要做存储（可以配置参与少量计算），其他机器是计算节点。

#### 程序入口

将程序调用加入到crontab每天自动执行，如：

30 6 \* \* \* hadoop (cd /home/hadoop/click\_ana/nstat; ./sync\_nstat.sh yesterday; php ./nstat\_app.php yesterday)

sync\_nstat.sh用于同步分光点日志数据，主程序是nstat\_app.php，进行数据的预处理，并调用子模块函数接口提取所需数据，直接运行下面任一种方式即可：

# php nstat\_app.php yesterday

# php nstat\_app.php 20131015

#### 主要功能

nstat平台主要提供下面七种功能数据：

1. 全流量数据
2. 网站分类数据
3. 搜索引擎数据，包括百度、谷歌和360等搜索数据
4. 共现词数据
5. 相似网站数据
6. 电商数据，包括京东、天猫、淘宝、苏宁、易讯和亚马逊等六大电商数据
7. 用户行为数据

#### 整体设计

下面是分析平台总体结构流程图：



图2.分析平台流程图

#### 接口模块

nstat平台处理程序主要处理上面7种类型的数据，在nstat\_app.php的execute()函数中分别调用相应的类函数，下面分别介绍每个模块。

##### 全流量样本

全流量模块在主函数中的调用接口是stat\_host ()函数，输入是nstat数据，处理过程：

1. 提取分类数据，代码：

INSERT OVERWRITE DIRECTORY '$this->HOST\_STAT\_INH' SELECT area, sip, dip, host, url, refer, cookie, loc FROM $this->RAWHIVE where dt='$this->day'

1. 将数据插入表名含有随机数的hive表"pipe\_host\_stat\_$day"."\_".rand().rand()中
2. 提取这几种类型的数据：
3. pvip信息，有aid, pvs, ipvs等字段
4. host信息，有host, pvs, ipvs等字段
5. url信息，有host, url, pvs, ips等字段
6. url refer信息，有host, url, refer, pvs, ips等字段

##### 网站分类

网站分类模块在主函数中的调用接口是cate\_host ()函数，输入是nstat数据，处理过程：

1. 提取分类数据，只提取host就可以，对应代码如下

INSERT OVERWRITE DIRECTORY '$this->HOSTS\_H' SELECT distinct(host) FROM $this->RAWHIVE where dt='$this->day'

1. 对数据分类，调用pipe\_host\_cate.php, 输入: host
2. 将结果数据存放于pipe\_host\_cate hive表, 格式：host category

##### 搜索引擎

搜索引擎模块在主函数中的调用接口是se\_analysis()函数，以百度为例，输入是raw\_se/baidu数据，处理函数有4个：analyze\_keywords(), analyze\_user\_keywords(), analyze\_links(), analyze\_user\_links().每个函数都有一个map和reduce过程。

1. analyze\_keywords()函数

map过程首先提取百度搜索词匹配的url，解析出搜索词，输出：keyword ip

reduce过程记录搜索词的pv, ips, 输出：keyword pv ips

1. analyze\_links()函数

map过程提取百度搜索词匹配的链接url，解析出链接，输出：keyword hosturl ip

reduce过程记录链接的pv, ips, 输出：keyword hosturl pv ips

相应地，analyze\_user\_keywords()和analyze\_user\_links()与上面两个函数类似，只是多了userid和atime两个字段。

##### 共现词

共现词模块在主函数中的调用接口是cocur\_words()函数，输入是pipe\_host\_info hive表中某天的数据，处理过程分两个阶段：

1. 对输入文本进行分词(seg\_input\_text()), 计算相关词(get\_cocur\_keywords()), 将计算结果输出并加载进pipe\_cocur\_keyword hive表中
2. 生成单词->host的倒排文件(get\_invert\_wordhostfile())，将结果输出并load到hive 表pipe\_hostword\_invert中

##### 相似网站

相似网站模块在主函数中的调用接口是similiar\_hosts()函数，输入是pipe\_host\_info hive表中某天的数据，处理过程分两个阶段：

1. 生成单词->host@num的倒排文件(get\_invert\_wordhostfile())，将结果输出并加载到hive表pipe\_hostwordcount\_invert中
2. 生成相关host(get\_similar\_hosts()), 默认采用内积计算相似度，将结果输出并加载到hive表pipe\_similar\_host中

##### 电商

电商模块在主函数中的调用接口是monitor\_data ()函数，输入是raw\_ec数据，处理过程如下：

文件monitor\_utils.php存放电商程序的工具类，六大电商程序分别存放在monitor\_data\_\*.php下，每个monitor\_data\_\*.php文件都有一个prepare()函数和execute()函数。

以京东为例，prepare()函数从电商hive表monitor\_data中提取出京东的数据，存放于monitor\_data\_jd hive表中。execute()函数调用每种类型数据的提取函数，主要有：

|  |  |
| --- | --- |
| **函数名称** | **功能** |
| whole\_traffic\_jd() | 提取全站流量 |
| channel\_traffic\_jd() | 提取频道页流量 |
| product\_traffic\_jd() | 提取四级页面流量 |
| refer\_traffic\_jd() | 提取站外流量来源 |
| area\_traffic\_jd() | 提取流量来源城市 |
| whole\_pageview\_jd() | 提取页面总浏览量 |
| homepage\_pageview\_jd() | 提取首页浏览量 |
| leimu\_pageview\_jd() | 提取类目页面浏览量 |
| product\_pageview\_jd() | 提取四级页面url |
| cart\_pageview\_jd() | 提取购物车页面浏览量 |
| search\_traffic\_jd() | 提取站内搜索的使用人数 |
| search\_keywords\_pageview\_jd() | 提取站内搜索词 |
| pay\_pageview\_jd() | 提取支付方式使用情况 |
| sale\_pageview\_jd() | 提取单品销售top100 |
| crawl\_jd() | 提取分品类四级页面浏览量 |
| search\_keywords\_cookie\_jd() | 提取用户搜索词 |
| products\_cookie\_jd() | 提取用户浏览商品页面 |

其他电商的函数类型和京东类似。

##### 用户行为

用户行为模块在主函数中的调用接口是user\_hosts()函数，输入是hive表中的nstat数据，处理过程分两个阶段：

1. 先提取每天的用户分类属性、用户点击页面、用户访问主机、用户搜索关键词和用户搜索点击链接数据
2. 再提取最近一周的各种用户数据。

## 总结

本文档介绍了现有的数据结构体系，从结构体系图出发，介绍了数据源、DSP实时数据接口、Radius客户端和分析平台等各模块接口，同时也对分析平台的7种功能数据进行了简要的描述。

本文档还有很多不足之处，欢迎大家提出修改意见，在以后的工作中会对本文档进行完善。