广告系统 DMP

编辑:田瑞林

业务背景需求

需求概述:

建立用户、tag 两个属性库及后台查询系统，它们将为推荐系统提供最基本的数据积累，也为以后对这些数据进行深度挖掘，分析，给用户带来更好的体验提供帮助，同时还能满足其他业务对用户精分的扩展需求。

可以提供支持广告方向的在线查询和用户匹配等

支持场景：

- 通过用户 id，实时查询对应 TAG，单条访问（第一期不用实现）
- 通过用户 id，非实时查询（或写入）对应 TAG，批量访问
- 通过 TAG，非实时查询所有的用户 ID，批量访问

概要设计

设计目标

- 通过用户采集和其他数据源提供的数据简历用户信息库
- 构建用户库，包含用户的基本属性和信息
- 提供离线和在线读写 API
- 实现后台查询系统计划做实时对接

数据规模假设

用户和数据量级达到一定程度后，需要选用可以支持该量级的技术架构，如果这

两个参数的量级都在百万以下，选用常规的关系型数据库就能满足需求，如 mysql, 但是业务背景是基于大数据的，量级可能上千万和亿级别，甚至更高，为了满足以后数据爆发的扩容和其他大数据系统的兼容，固选用 Hbase 系统

功能指标

1. 存储能力

- 支持亿级别用户存储包含用户信息
- 支持千万级的用户 tag

2. 运算能力和查询能力

- 支持前端业务至少每秒 500 量级的试试查询
- 支持每秒 1W+的写入小路
- 支持遍历查询，小时级别

3. 逻辑计算

- 支持多维度组合查询
- 支持二级索引查询
- 支持协处理器查询和过滤

扩展性

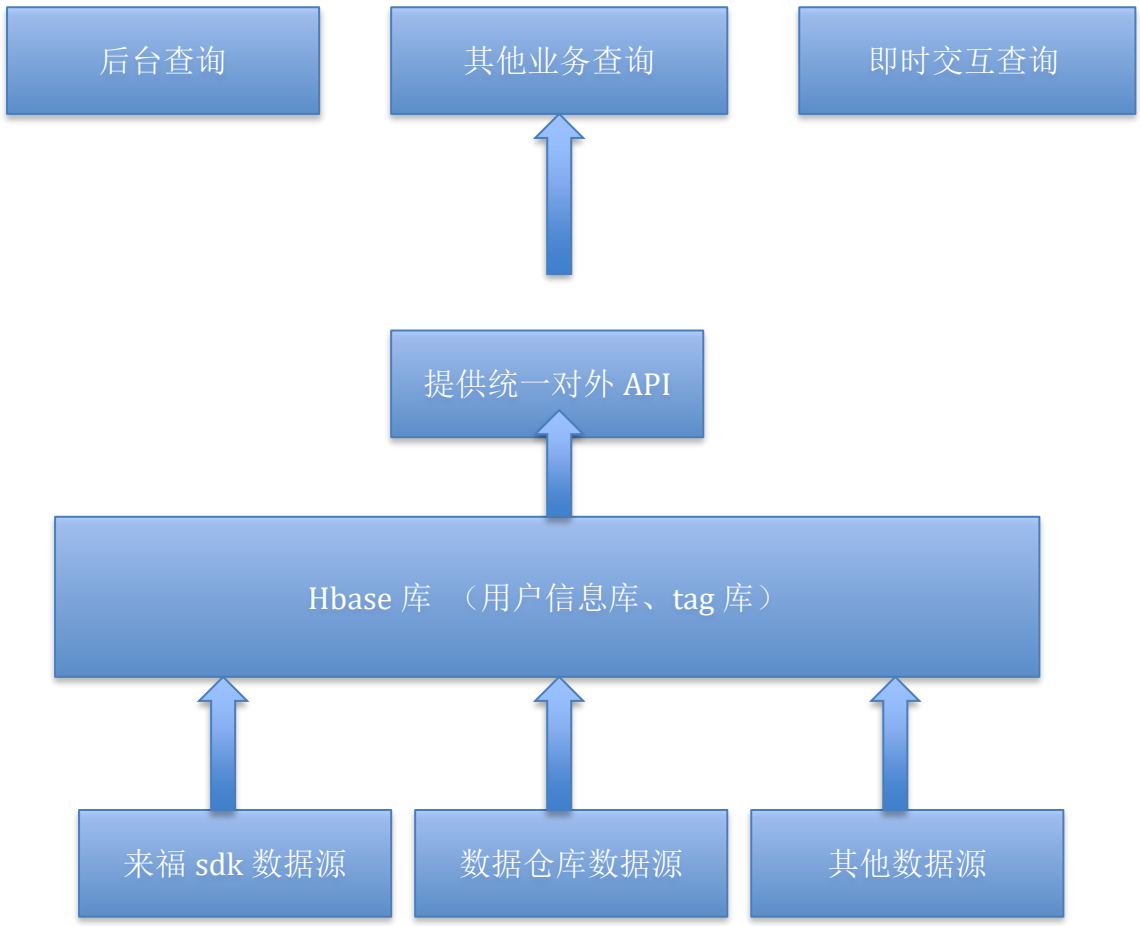
- Hbase 以 KV 存储为主能映射 hive 表同步数据，可借助工具提供 sql 查询
- 基于 Hbase 上层可接 redis 做 memory Cache 方便提供 top 快速查询
- 可以对接 spark-streaming 对 hbases 表进行读写操作，并计算

系统流程

DMP 系统根据数据流可以分为以下几层:

- 数据来源层：来福公司数据仓库的数据、日志服务器端的数据、其他第三方提供的数据源
- 数据处理层：数据采集端过滤、数据仓库清洗、其他 ETL 处理
- 数据存储层：Hbase 核心层、存放处理后的结构化数据，包含用户信息和 TAG 标签库
- 数据应用层：对接 redis 做 cache 层，提供后台和 API 查询

架构设计如下图:



表结构设计

基础用户表

- ad_user_info_base : 广告用户基础信息表，数据存储对象为所有广告用户的基本信息，按照规则更新写入到 hbase 中,其中用户信息表分为 Android、IOS 和 PC 表
  1. android_user_info_base　安卓基础用户信息表
  2. ios_user_info_base　IOS 基础用户信息表
  3. pc_user_info_base　PC 基础用户信息表
- user_tags:用户标签库表，存储广告标签数据，这部分数据，需要迭代更新
- user_tags_info；用户标签综合表，存储用户信息和标签对应关系

基础表逻辑结构

android_user_info_base 表结构如下

	android_user_info_base 表结构			
主键、列族	字段名称	字段解释	字段值举例	其他
rowkey		表主键用户标识 (uuid)	24ba02d30e5ed72d303	
timestamp		时间戳		
info		列族:用户信息		base_info
	damid			
	sdkversion	速度快版本		
	appid	广告 id		
	publiserid	开发中 id		
	model	设备型号	iPhone	

	machine	机器类型	iPhone5,2	
	osversion	设备系统版本	4.3.2	
	imei			
	ram	设备的内存容量		
	rom	设备的磁盘容量		
	carrier	运营商信息		
	simnumber	SIM 卡序列号		
	time	当前系统时间		
	isjailbroken	是否破解		
	uid	设备唯一标识		
	deviceid	设备标识		
expand1		扩展列族 1		
expand2		扩展列族 2		

建表语句

create

'android_user_info_base',{NAME=> 'info',VERSION=>21473647,COMPRESSION=>

'LZO',BLOOMFILTER='ROW'},{NAME=>

'expand1',VERSION=>21473647,COMPRESSION=>

'LZO',BLOOMFILTER='ROW'},{NAME=>

'expand1',VERSION=>21473647,COMPRESSION=> 'LZO',BLOOMFILTER='ROW'}

ios_user_info_base 表结构如下

	ios_user_info_base 表结构			
主键、列族	字段名称	字段解释	字段值举例	其他
rowkey		表主键用户标识 (uuid)	24ba02d30e5ed72d303	
timestamp		时间戳		
info		列族:用户信息		base_info
	damid			

	sdkversion	速度快版本		
	appid	广告 id		
	publiserid	开发中 id		
	model	设备型号	iPhone	
	machine	机器类型	iPhone5,2	
	osversion	设备系统版本	4.3.2	
	imei			
	ram	设备的内存容量		
	rom	设备的磁盘容量		
	carrier	运营商信息		
	simnumber	SIM 卡序列号		
	time	当前系统时间		
	isjailbroken	是否破解		
	uid	设备唯一标识		
	deviceid	设备标识		
expand1		扩展列族 1		
expand2		扩展列族 2		

建表语句

create 'ios_user_info_base',{NAME => 'info',VERSION=>21473647,COMPRESSION=> 'LZO',BLOOMFILTER='ROW'},{NAME => 'expand1',VERSION=>21473647,COMPRESSION=> 'LZO',BLOOMFILTER='ROW'},{NAME => 'expand1',VERSION=>21473647,COMPRESSION=> 'LZO',BLOOMFILTER='ROW'}

pc_user_info_base 表结构如下

	pc_user_info_base 表结构			
主键、列族	字段名称	字段解释	字段值举例	其他
rowkey		表主键用户标识	24ba02d30e5ed72d303	

		(uuid)		
timestamp		时间戳		
info		列族:用户信息		base_info
	damid			
	sdkversion	速度快版本		
	appid	广告 id		
	publiserid	开发中 id		
	model	设备型号	iPhone	
	machine	机器类型	iPhone5,2	
	osversion	设备系统版本	4.3.2	
	imei			
	ram	设备的内存容量		
	rom	设备的磁盘容量		
	carrier	运营商信息		
	simnumber	SIM 卡序列号		
	time	当前系统时间		
	isjailbroken	是否破解		
	uid	设备唯一标识		
	deviceid	设备标识		
expand1		扩展列族 1		
expand2		扩展列族 2		

建表语句

create 'pc_user_info_base',{NAME => 'info',VERSION=>21473647,COMPRESSION=> 'LZO',BLOOMFILTER='ROW'},{NAME => 'expand1',VERSION=>21473647,COMPRESSION=> 'LZO',BLOOMFILTER='ROW'},{NAME => 'expand1',VERSION=>21473647,COMPRESSION=> 'LZO',BLOOMFILTER='ROW'}

user_tags 表结构如下

user_tags 表结构			
主键、列族	字段名称	字段解释	字段值举例
rowkey		表主键	
timestamp		时间戳，版本控制	
tags		列族	
	stags	用户标签	[{"kinds":"young","tags":["20 岁-30 岁"]}]
	ctags	抓取标签	同 stags,存储为 json 窜
cf1		扩展列族	

建表语句

create
'user_tags',{NAME
=>'tags',BLOOMFILTER='ROW',VERSION=>21473647,COMPRESSION=>
'LZO'},{NAME
=>'cf1',BLOOMFILTER='ROW',VERSION=>21473647,COMPRESSION=> 'LZO'}

user_tags_info 表结构如下

	user_tags_info 表结构			
主键、列族	字段名称	字段解释	字段值举例	其他
rowkey		表主键用户标识(uuid)	24ba02d30e5ed72d303	
timestamp		时间戳		

		列族:用户信息		base_info
	damid			
	sdkversion	速度快版本		
	appid	广告 id		
	publiserid	开发中 id		
	model	设备型号	iPhone	
	machine	机器类型	iPhone5,2	
	osversion	设备系统版本	4.3.2	
info	imei			
	ram	设备的内存容量		
	rom	设备的磁盘容量		
	carrier	运营商信息		
	simnumber	SIM 卡序列号		
	time	当前系统时间		
	isjailbroken	是否破解		
	uid	设备唯一标识		
	deviceid	设备标识		
	username	用户名		string
	gender	性别		string
	age	年龄		string
	region	地域		[]
	prof	职业		string
attr	income	收入		string
	edu	学历		string
	mar	婚姻状况		0,1
	hobby	兴趣		json
	adt	商业纬度	跑男	string
	pert	个性标签	白富美	json
expand1		扩展列族 1		
expand2		扩展列族 2		

建表语句

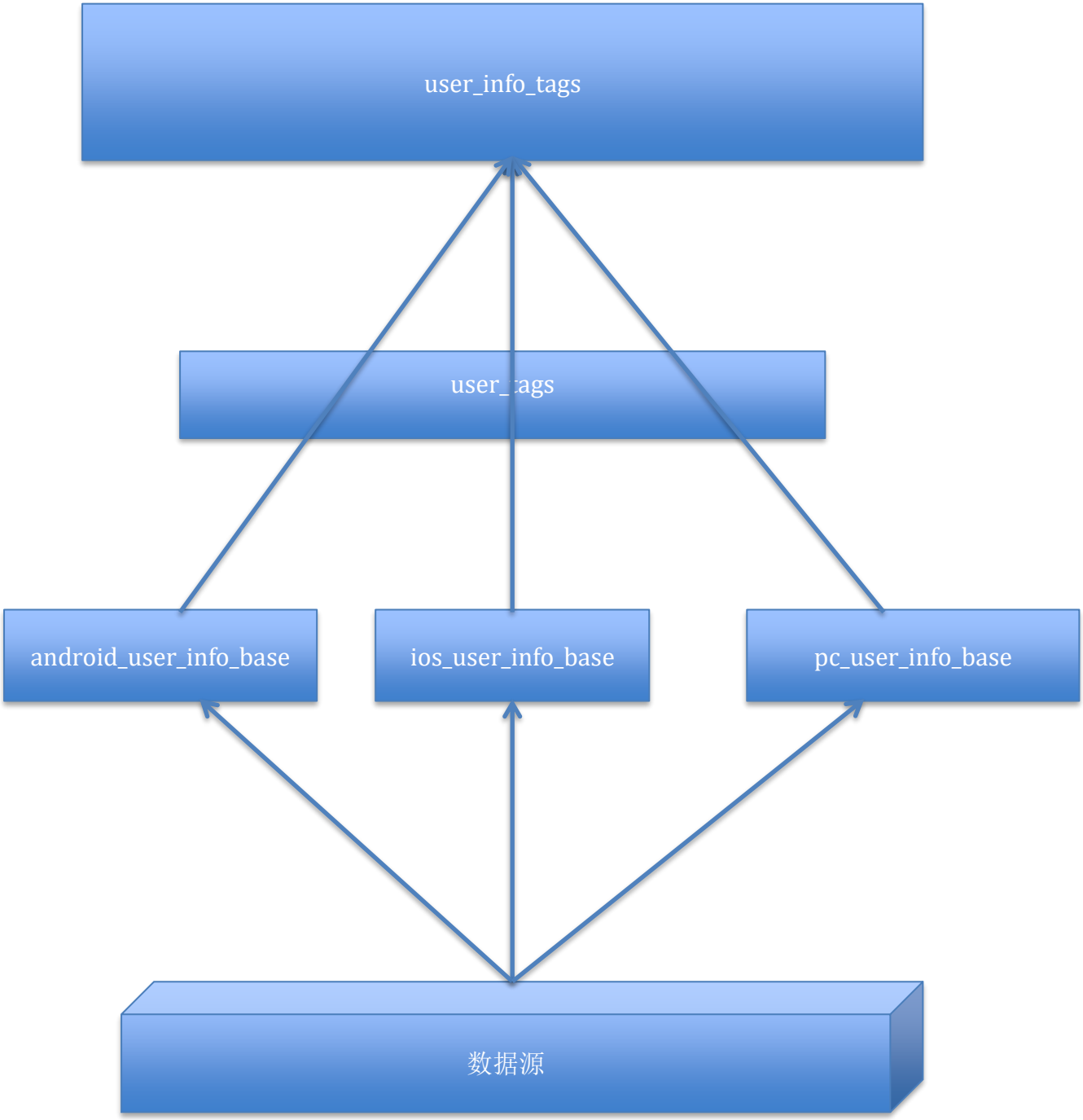
```
create 'user_tags_info',{NAME => 'info',VERSION=>21473647,COMPRESSION=>
'LZO',BLOOMFILTER='ROW'},{NAME =>
'attr',VERSION=>21473647,COMPRESSION=>
'LZO',BLOOMFILTER='ROW'},{NAME =>
'expand1',VERSION=>21473647,COMPRESSION=>
'LZO',BLOOMFILTER='ROW'},{NAME =>
'expand1',VERSION=>21473647,COMPRESSION=> 'LZO',BLOOMFILTER='ROW'}
```

数据加载

数据加载模块优先使用 distcp 传输数据到 hdfs 上，通过 MR 或者 API 方式加载到 hbase 对应的表中

数据计算逻辑

整体数据流加载和计算逻辑图



说明

数据通过 MR 和 Hbase API 分别更新写入到 android_user_info_base、ios_user_iinfo_base 和 pc_user_info_base 表中，然后这三张基础表关联 user_tags 表做逻辑计算，匹配用户信息并更新用户标签，同时把数据写入到 user_info_tags,这张表是用户和标签关联并映射表，此表作为产出表