**UFP设计文档**

1. 系统设计目标：

支撑每天1000万IMEI用户的访问，每天3000万次请求

支持每天6000万的log日志

1. 系统架构

部署结构如下，初期需要4台服务器，后续压力增大时可添加更多JavaServer。



部署结构图



典型四层部署结构：

Nginx

Resin

Mysql5

Redis

JDK6

Ubuntu

改进点：

* 1. 使用Nginx，更快更稳定
  2. 使用Resin Java server
     1. 更新稳定，速度是Tomcat的3倍以上
     2. 并发处理、启动停止机制优于Tomcat
  3. 加入Redis Cache server，尽量减少Mysql的压力，保证数据处理速度。

1. SDK前后端交互协议

参照google docs文档：

<https://docs.google.com/a/umeng.com/document/d/1stj-fKwE-IO2k0QlpS440U2zPZvR08M0DKnof4L-kGc/edit>

改进：

1. 目前协议传输使用POST请求，考虑改为Get请求（URL最长支持为1024Byte，可满足需求）。

好处：便于Nginx/Apache记录日志到Raw Log；便于调试分析；Post请求并不能避免Spam。

待讨论：如何将数组扁平化，多promoters中合并action

1. 服务端返回数据格式中加入更多参数，目标是将配置后台化。
2. 后台系统模块设计

典型的Spring MVC三层结构，系统并不复杂，拿过来直接可以用。

和我们目前基于rails交换架构类似，但由于Java项目模块划分更容易也更清晰一些。

1. 系统模块设计



1. UFP Serving流程设计



接入第三方广告网络流程：

1. 服务器端根据广告位的设置策略，如果有接入广告网络，将ad\_network\_key、request\_count、layout和用户设备信息等通过API调用。
2. 将广告网络返回的结果字段mapping到ufp协议，同时对每条广告追加一个category（0：交换、1交换自主、2交换付费、21 UFP、31 umeng ads），然后将结果返回给SDK
3. SDK正常展示所有广告。
4. UFP Logging流程设计

Logging流程

1. ufp Nginx打印log到文件中。（ufp的batch根据此文件定时分析文件入库到mysql）
2. UFP LogService根据Log中的category识别是否含有非UFP广告，如果有将log（只包含广告网络的ads）转发给各广告网络。
3. 返回200成功结果给SDK。

TODO：

计划Q2开始，Logging转发一份给数据平台保存。后续再作分析。

1. 技术路线

JDK6

JSTL+HTML+JQuery

Spring Action

Spring Core + Transaction

MyBatis (ORM)

Mysql 5

Redis

Python

未来改进：

* + 1. 考虑使用Google Guice替换Spring，超轻量级的依赖注入容器、速度快代码少，很容易进行单元测试。（liqun）
    2. 前端可以考虑Google GWT，Java代码来替代javascript，便于实现和维护复杂的前端逻辑。（hai）

1. 数据统计模块
   1. Raw Log记录在日志文件，以小时进行切割，文件名如request\_2011101610.log。
   2. 初期使用python或ruby脚本进行日志解析入库，每小时一次进行累加。先计算AdReportSummary、AdReportPromoterSummary、AdReportDetail，再汇聚到AdSlot、AdOrder、App、Account。
   3. 每日对RawLog进行重新计算一次，覆盖当天计算的数据。
   4. Raw Log入库，并追加附加信息。（MyISAM）

Issues:

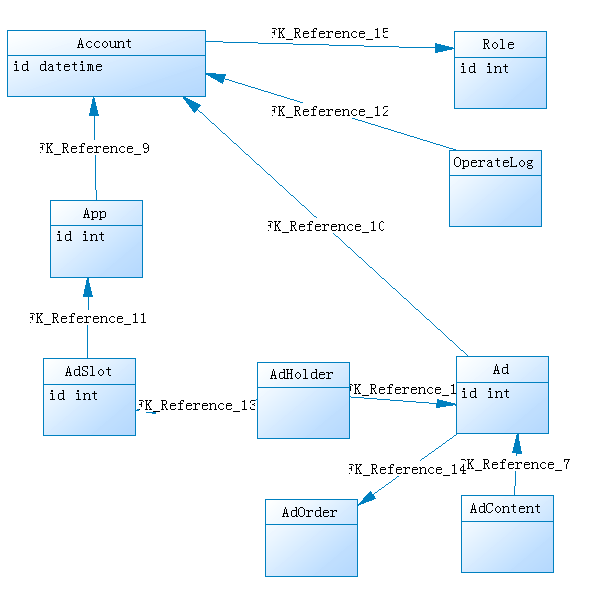
* 1. 测试期间如何快速看到数据

1. 数据库详细设计

数据库我们使用MySQL，因为除日志外数据规模不大，稳定性和性能更可靠，运维和优化技术相对成熟。存储引擎使用支持事务的InnoDB，保证广告系统数据的一致性。

* 1. 数据库ER关系设计

账户和广告信息相关表



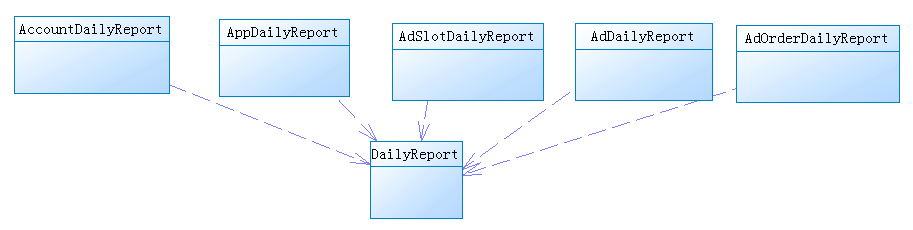
1. 数据库表字段设计

由于UFP目前只面对Publisher，用户角色较少，因此权限系统设计相对简单。

|  |
| --- |
|  |
| Create database umeng\_ufp;  Use umeng\_ufp;  create table account  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  email varchar(64) not null comment ‘账户地址’,  password varchar(64) not null comment ‘密码，MD5加密后’,  type enum(‘admin’,'admin\_view',’admin\_op’,'publisher') not null comment ‘用户组’,  name varchar(32) comment ‘账户名称’,  phone varchar(32) comment ‘电话或手机’,  info varchar(128) comment ‘更多信息’,  logo\_url varchar(128) comment ‘账户定制的Logo地址’,  last\_login\_time datetime,  login\_times int,  PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table role  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  type enum(‘admin’,'admin\_view',’admin\_op’,'publisher') not null comment ‘用户组’,  permission vachar(128) comment ‘权限设置’,  PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table app  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  app\_key varchar(32) NOT NULL comment ‘应用的key与umeng主站对应或新的’,  platform enum(‘android’,'ios') NOT NULL comment ‘android/ios’,  name varchar(32) not null comment ‘应用名称’,  PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table ad\_slot  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  app\_id int NOT NULL comment ‘所属应用’,  name varchar(32) not null comment ‘广告位名称’,  size\_landing varchar(32) comment ‘入口尺寸’,  size\_content varchar(32) comment ‘内容尺寸’,  platform enum(‘android’,'ios') NOT NULL comment ‘平台’,    style varchar(32) comment ‘样式’,  device varchar(32) comment ‘子平台、适用设备’,  channels varchar(32) comment ‘适用渠道’,  areas varchar(32) comment ‘适用地域’,  timeslots varchar(32) comment ‘适用时段’,  display\_rate varchar(32) comment ‘出现概率’,  enable\_xp varchar(32) comment ‘是否接入交换’,  xp\_rate int comment ‘交换比例’,  xp\_id int comment ‘交换ID’,  enable\_adnet varchar(32) comment ‘是否接入广告网络’,  adnet\_rate int comment ‘广告网络比例’,  adnet\_id int comment ‘广告网络ID’,  status varchar(32) NOT NULL comment ‘状态’,  PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table ad  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  name varchar(32) not null comment ‘广告名称’,  size varchar(32) comment ‘尺寸’,  type varchar(32) comment ‘类型’,  ad\_order\_id int comment ‘所属订单的id’,  prior varchar(32) comment ‘优先级’,  start\_time datetime,  end\_time datetime,    device varchar(32) comment ‘子平台、适用设备’,  channels varchar(32) comment ‘适用渠道’,  areas varchar(32) comment ‘地域’,  network varchar(32) comment ‘适用网络’,  keywords varchar(32) comment ‘关键词’,  landing\_type varchar(32) comment ‘打开方式’,  timeslots varchar(32) comment ‘适用时段’,    budget varchar(32) comment ‘预算’,    status varchar(32) NOT NULL comment ‘状态’,  PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table ad\_holder  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  ad\_id int not null comment ‘广告的ID’,  ad\_slot\_id int NOT NULL comment ‘广告位的ID’,    PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table ad\_content  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  platform varchar(32) NOT NULL comment ‘平台’,  title varchar(32) NOT NULL comment ‘应用名称’,  ad\_words varchar(32) comment ‘广告语’,  description varchar(512) comment ‘描述’,  provider varchar(32) comment ‘开发商’,  icon varchar(32) comment ‘图标地址’,  url varchar(32) comment ‘下载地址’,  size int comment ‘大小’,  price varchar(32) comment ‘在商店的标价，默认为0’,  app\_package\_name varchar(32) comment ‘’,  app\_version\_code varchar(32) comment ‘’,  app\_version\_name varchar(32) comment ‘’,  app\_store\_id varchar(32) comment ‘’,    PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table ad\_order  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  start\_at datetime,  end\_at datetime,  name varchar(32) NOT NULL comment ‘广告公司名称’,  contact varchar(32) NOT NULL comment ‘广告公司联系人’,    PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table user\_op\_log  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  op\_date date,  op\_time time,    account\_id int comment ‘账户的编号’,  ad\_id int comment ‘广告的编号’,  ad\_order\_id int comment ‘广告订单的编号’,  ad\_slot\_id int comment ‘广告位的编号’,  app\_id int comment ‘应用的编号’,    admin\_id int comment ‘管理员id，如果不是管理员操作，就显示0’,  op\_content varchar(32) comment ‘1、流量 2、推广类型 3、推广样式 4、应用名称/网站名称 5、推广文字/网站简介 6、详细描述 7、开发商 8、icon地址/网站logo地址/图片地址 9 apk地址/appstore地址/网页地址 10、跳转方式 11、优先级 12、是否参接入 13、启用/暂停 14、广告价格 15、付费方式 16、投放网络 17 投放时段 18、投放时间 19、预算 20、入口类型 21、入口尺寸 22、内容尺寸 23、渠道 24、出现概率 25、广告类型 26、名称 26、所属应用 27、所属订单 28、关键词 29、适用设备30、打开方式 31、地域’,  op\_type varchar(32) comment ‘1：新增 2：删除 3：修改 4：暂停 5：启用’,  ad\_type varchar(32) comment ‘1、交换 2、友盟广告 3、付费应用 4、自销广告’,  ad\_subtype varchar(32) comment ‘1：广告位 2：广告 3、应用 4、订单 5、账户’,  old\_value varchar(32) comment ‘修改前的值’,  new\_value varchar(32) comment ‘修改后的值’,  PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci; |

外键：优点：缺点：

索引：



DailyReport分三个表，分别为publisher\_summary，promoter\_summary，detail

|  |
| --- |
| Report |
| create table report\_publisher\_summary  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  publisher\_id int NOT NULL comment ‘媒体Id’,  date\_s date,  impression int not null default 0,  click int not null default 0,  download int not null default 0,  appstore\_click int not null default 0,    PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table report\_promoter\_summary  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  promoter\_id int NOT NULL comment ‘被推广的ID：广告、广告位、应用、订单’,  date\_s date,  impression int not null default 0,  click int not null default 0,  download int not null default 0,  appstore\_click int not null default 0,    PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;  create table report\_detail  (  id int(11) NOT NULL auto\_increment comment ‘自增的ID’,  created\_at datetime,  updated\_at datetime,  publisher\_id int NOT NULL comment ‘媒体Id’,  promoter\_id int NOT NULL comment ‘被推广的ID：广告、广告位、应用、订单’,  date\_s date,  impression int not null default 0,  click int not null default 0,  download int not null default 0,  appstore\_click int not null default 0,    PRIMARY KEY (`id`)  ) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci; |

1. 工作和人员计划

10.10-10.15 一周 系统设计、数据库设计（ke、hai）

10.17-10.29 二周 搭系统框架、模块、数据读写模块（ke）

10.31-11.19 三周 UFP交互实现，数据统计实现，功能实现（Ke+FTE+Intern）

11.10-11.25 两周 开始和SDK连调，基于交换的SDK

11月底 一期版本可用

人员计划：

11月前务必到岗一个Java工程师，主要做UFP的前端编程。

11月初到岗一个Python/Ruby实习工程师(maybe 鄢舒元)，主要用于对UFP的日志进行统计，核对数据。（目标是保证数据无误，后期量起来后会迁移到Haddop平台计算）