

# DSP投放引擎设计与实现

—— 唐端荣

## 背景介绍

广告是目前大部分互联网公司的主要变现手段，根据其目的可分为合约（品牌）广告和效果广告，合约广告主要是让更多的人知晓品牌、以创造良好的品牌形象为目的。效果广告的主要目的短时间内有明显的ROI，比如电商后续产生的购买行为，游戏后续产生的下载激活等行为。

本文中介绍的DSP是直接面向广告主的效果广告平台，它的全称是“Demand Side Platform”，它是以定价或竞价的方式，实现广告的受众购买和程序化购买。支持实时竞价和人群定向，同时具备对投放效果的监测、分析和优化能力。本文主要介绍DSP系统的架构设计和核心功能的实现。

# 背景介绍

MARKETER  
广告主

PUBLISHER  
媒体

CONSUMER  
消费者

## DSP | 需求方平台



## Ad Exchanges & SSP | 程序化广告供应方



## Trading Desk & Tech 采购交易平台及技术



## Verification 广告验证



## Measurement & Analytics 监测分析工具



## Programmatic Creative 程序化创意



## Data Suppliers & Data Management 数据提供和数据管理



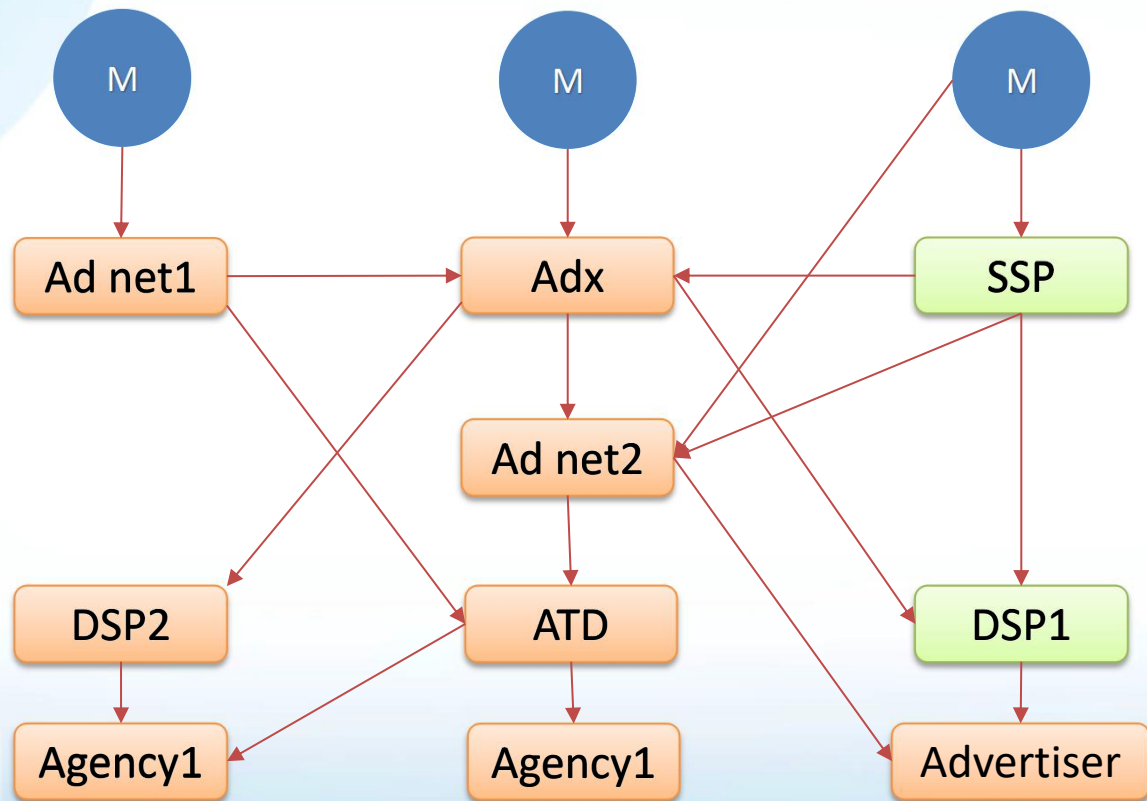
## "DSPAN" (DSP + Ad Network, Hybrid) | 扩展型DSP (DSP+媒体网络的混合体)



## Programmatic TV | 程序化电视广告



### 程序化售卖生态链



## 背景介绍

广告投放技术经历了由静态页面纯手工的替换，到系统自动按排期投放，再到现在的RTB实时交易等三个阶段；

计费方式：

CPT 按时段付费

CPM 按千次曝光付费

CPC 按点击付费

CPA 按转化行为付费



电梯厢内的纯人工CPT广告

## 业务特点

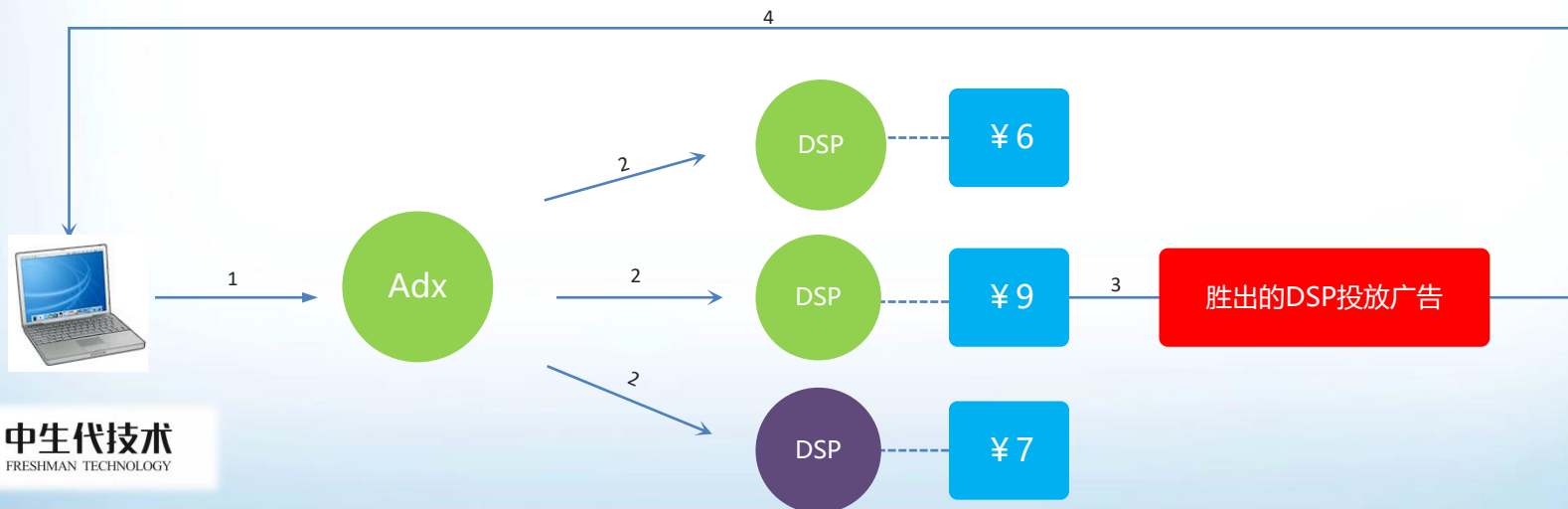
DSP业务特点:

高并发：广告请求量大

低延迟：adx对广告返回有限制，页面渲染时间制限

高时效：实时响应广告主的投放需求，实时投放效果反馈

满足各种定制化投放需求：支持N个定向维度

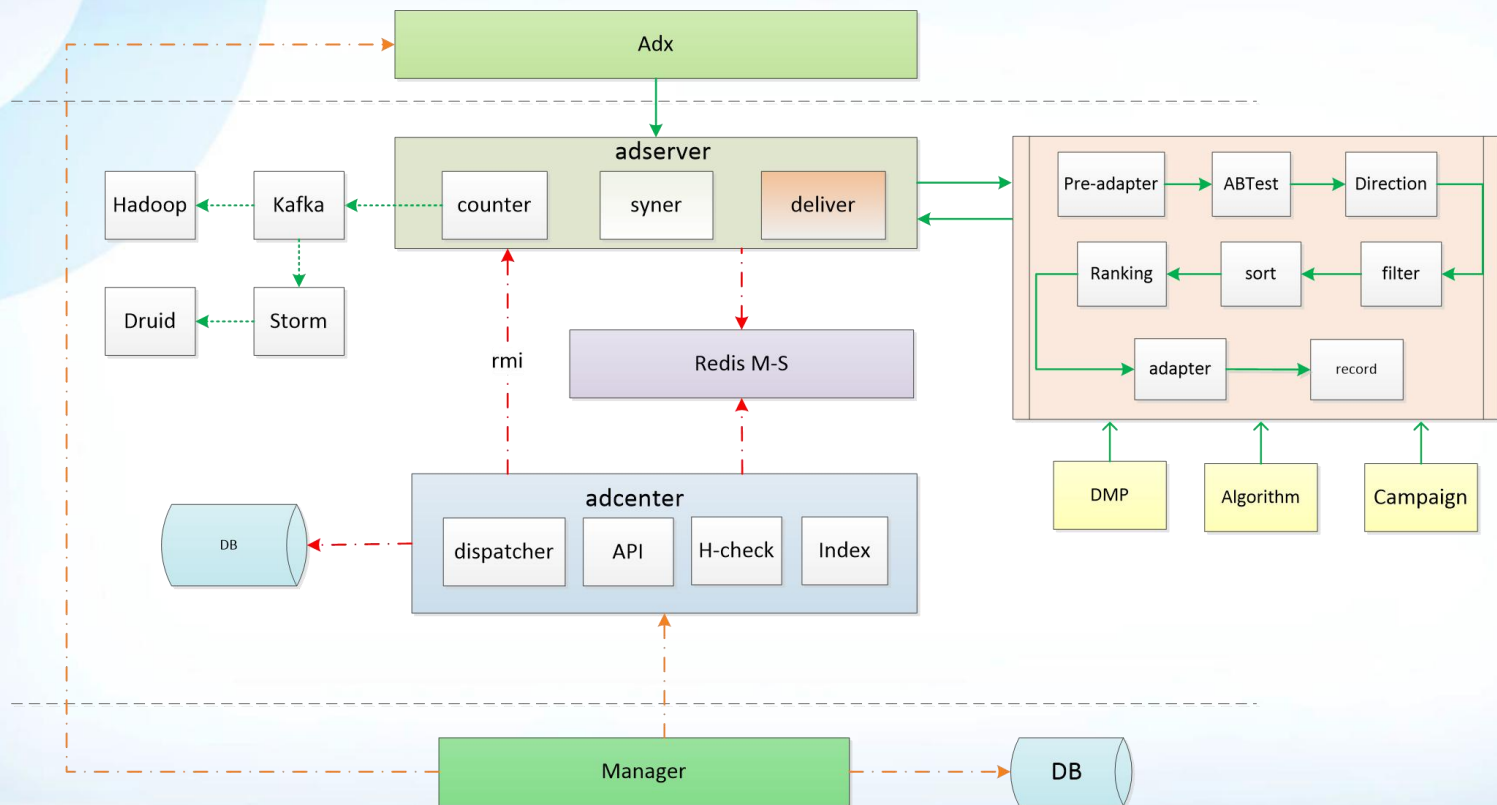




## 设计思路

- 支持灵活的AB实验：
  - 支持插拔式的算法实验
- 满足数据时效要求：
  - 新增广告要实时生效、用户暂停投放要实时响应，投放效果要实时呈现
- 7\*24小时服务
  - 稳定性保证
- Adx的应答率标准
  - 应答率影响adx的派量算法，超时太多会被adx控量
- 灾备和故障恢复
  - 在线容灾，健康检查、自动功能降级、灰度发布、负载均衡
- 多维监控方案
  - 软硬件监控、业务异常报警、投放线索追踪

# 系统架构





### ➤ 业务分割：

在线计算与离线计算分离，使用cache隔离

内网与外网分离

前后端分离，使用API交互数据。

### ➤ 灵活plugin：

流量切分配置化

基于接口编程，功能配置化

### ➤ 缓存隔离：

根据数据时效要求和重要程度分为3级cache：

Heap：广告数据、在线反作弊数据

Redis-MS：广告数据、消费数据

Redis-Cluster：DMP、频控数据

## 节点去中心化

adcenter :

VIP

主辅运行时互备

无状态

adserver :

LVS负载均衡

多机同时提供服务

心跳监控

无状态

**\*\* 主辅运行时互备\*\*:**

利用Redis **SET key value EX seconds NX** 实现

Key为锁名，

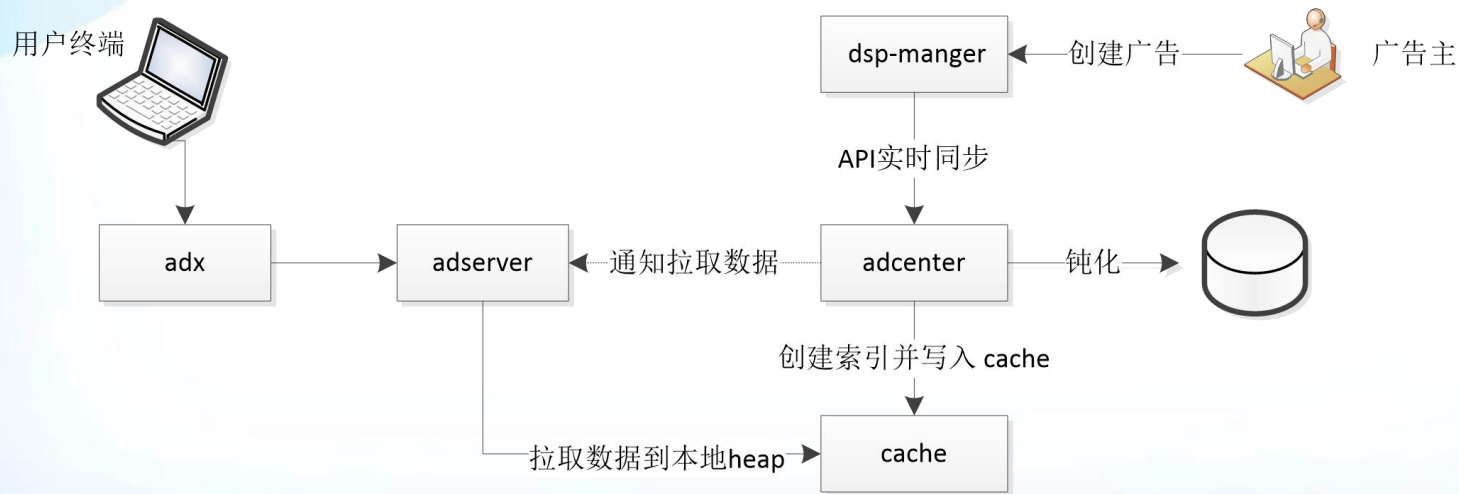
value建议是本机IP（如果单机多实例需要使用ip:port作为value）

EX seconds: 租约锁的失效时长，失效后锁自动删除

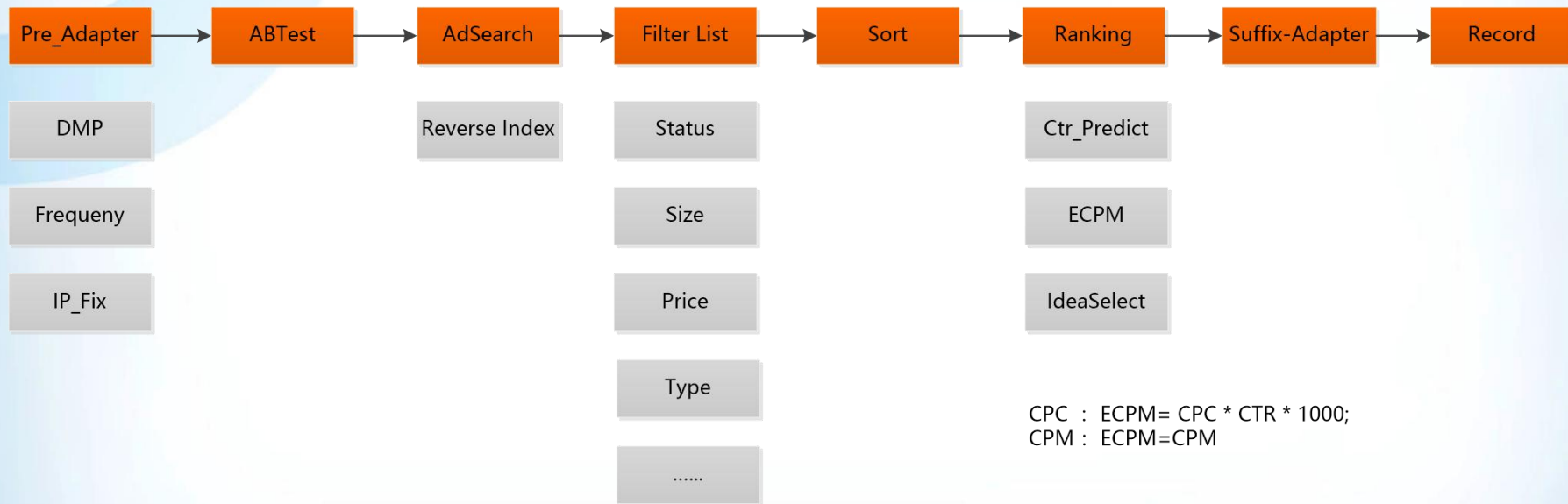
NX: 只有key不存在时才能set成功

## 广告的投放过程

- 新增广告分钟内可见
- 使用cache隔离离线计算



# 广告筛选



## 广告筛选

**AUC**匹配计算:

A-广告

C-媒体内容

U-用户

媒体属性与广告要求的匹配  
用户特征与广告要求的匹配



流量特征与广告要求进行匹配

广告要求: 时间、地域、关键字、人群等等.....

媒体属性: 广告位、女性、动漫、房产、教育.....

用户特征: 地域、性别、年龄、兴趣标签.....

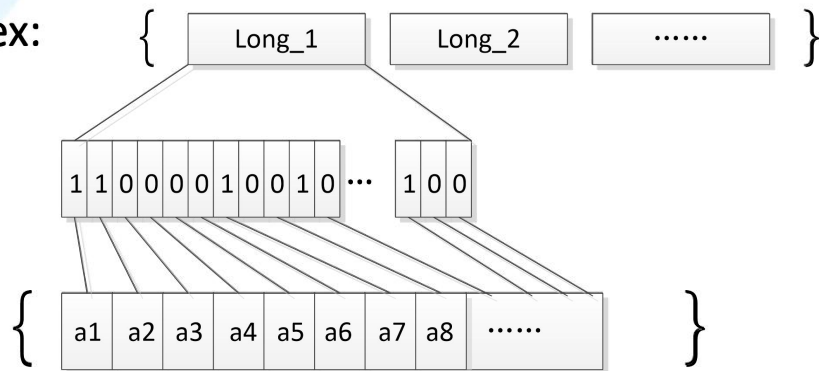
## 广告筛选

### ➤ 广告检索：索引构建和查找

#### 倒排索引 – BitMap

效率担当!

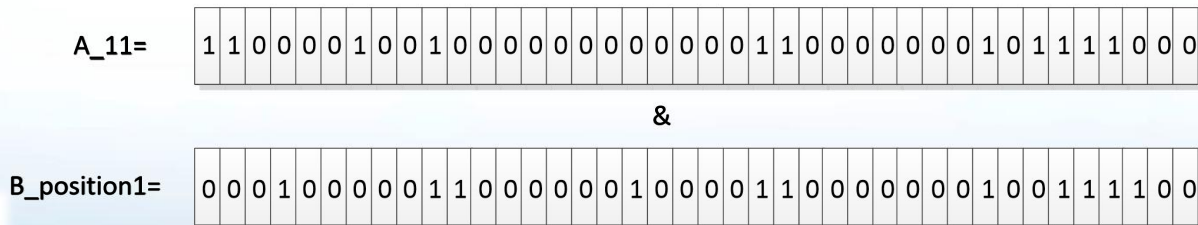
Build Index:



Key : 定向维度+维度值  
value : BitSet

索引同步:  
倒排索引: 定时全量同步, 分钟生效  
正排索引: 实时增量同步, 秒级生效

Search:





## 通用过滤器

- 按状态过滤
- 按预算过滤
- 按尺寸过滤
- 按素材类型过滤
- 按频次过滤
- 按行业类别过滤

Status filter

budget filter

size filter

..... filter

## 实时扣费

### ➤ 扣费：扣数和扣费

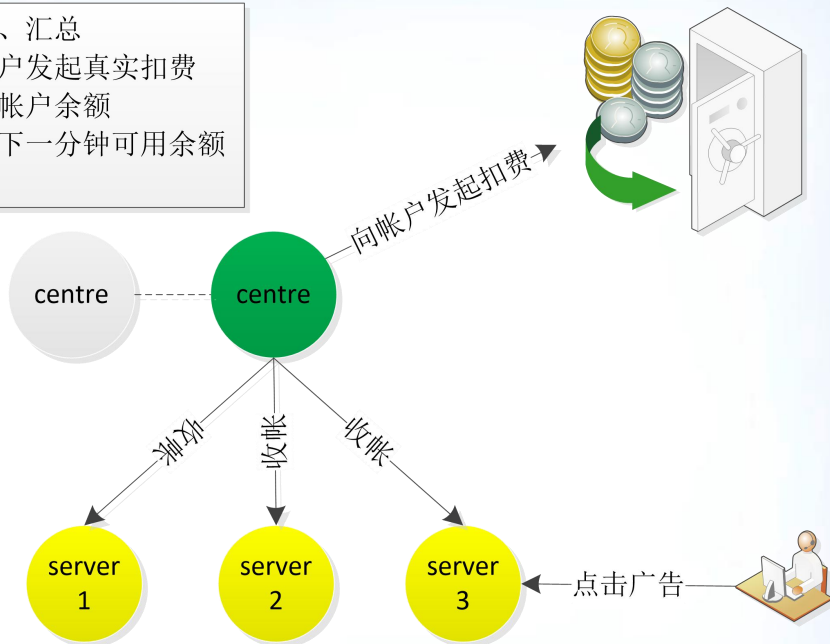
- 多线程异步队列
- Failover异步队列
- 流水落本地磁盘
- 本地数据库扣费原子操作
- 无状态多机运行时互备

### ➤ 止损：投放与点击之间的时差

- 设置最小投放余额
- 流控
- 预算平滑



- 1、收帐、汇总
- 2、向帐户发起真实扣费
- 3、更新帐户余额
- 4、计算下一分钟可用余额
- 5、派帐



- 1、反作弊计算
- 2、本地扣费
- 3、推送到异步队列
- 4、流水落本地磁盘

➤ Dubbo协议

➤ CTR预估：优化ROI

出价ECPM公式：

竞价ECPM (cpm) = cpm 出价

竞价ECPM (cpc) = cpc出价 × 1000 × ctr (预估)

扣费计算：

扣费 (cpm) = ecpm2 (第二高价) / 1000

扣费 (cpc) = ecpm2 (第二高价) / 1000 / ctr (预估)

\*\* 小贴士\*\*：

- 1、按CPM售卖一般是通过服务费盈利，需要把流量采买的损耗考虑进去。
- 2、如果按CPC卖，预估CTR高于实际CTR，DSP会赔钱；反之估的过于低了有可能会买不到流量。
- 3、只有adx知道真正的第二高价是多少。

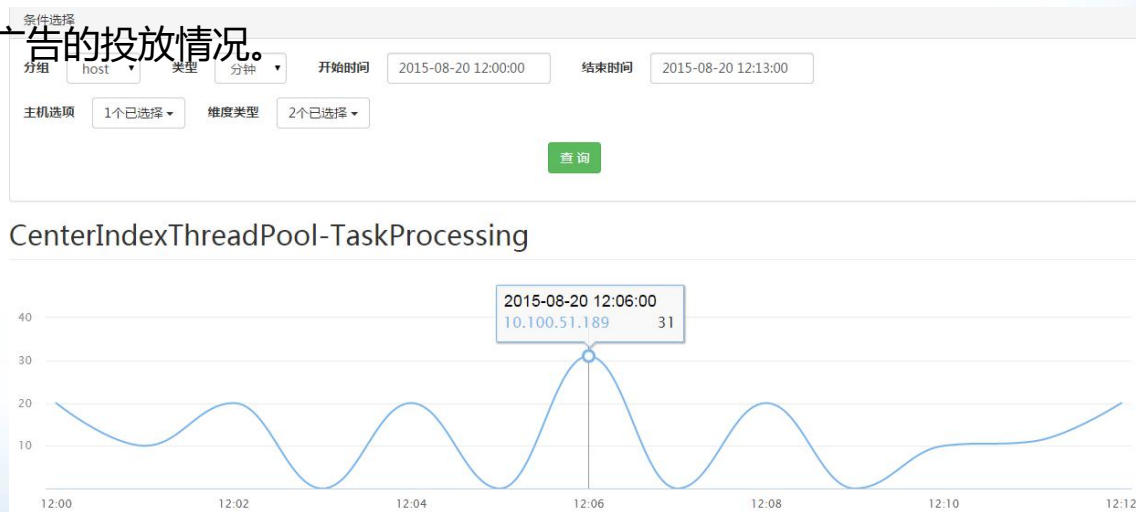
➤ 用户画像：打标签



- 1、按CPM投放的广告是否需要做CTR优化？
- 2、谁该对adx与dsp的曝光gap负责？
- 3、广告主是否真的需要预算平滑？

## 监控手段

- 内存数据可视化
- 硬件监控：监控服务器cpu、内存使用率、连接数、网卡流量，服务器负载等。
- 软件监控：监控qps、rt、jvm、队列大小等。
- 业务数据监控
- 线上问题快速响应，跟踪每只广告的投放情况。



- JVM优化
- 长连接
- 多级缓存，就近访问：按数据重要程度和时效要求设计3级缓存
- JSON 选型：在我们的业务场景中 Jackson > Gson 约3倍
- Ranking耗时控制
- 代码质量控制

生产环境性能：

rt : P99.998 < 50ms; P99.81 < 10ms

qps : 10000+

## 研发总结

- 单元测试、交叉编码
- 协同开发中必须统一 Code Style
- 较大变更需要过压测
- 关键策略必须形成文档，所有人理解一致后再编码
- 允许有多个方案，按实验结果择优选用
- 信息透明，互为备份



Thanks!