

返现广告系统设计

Bill

February 25, 2024

Contents

1	需求	2
2	主要业务数据流	2
3	系统技术架构	4
4	模块简述	4
4.1	广告数据平台	4
4.2	广告拉取模块	5
4.3	用户管理模块	5
4.3.1	用户匿名化策略	5
4.3.2	用户注册相关 API 接口	5
4.3.3	其它说明	5
4.4	广告投放模块	6
5	非功能性目标	6
5.1	可扩展性	6
5.2	性能	6
5.3	安全性	6
5.4	数据合规	6
6	其它说明	6

1 需求

Task :“系统设计挑战:奖励返现应用的广告分发后端”

- 目标: 设计一个可扩展且高效的奖励返现应用后端系统, 该系统需要与第三方广告平台集成。系统应管理用户注册、处理并匿名化用户数据以用于广告定位, 并处理广告分发。
- 要求:
 - 用户注册与数据管理:
 - * 设计用于用户注册的 API 端点, 包括数据验证和存储。实现每个用户生成本地 DID (去中心化标识符) 的机制, 以确保隐私和数据主权。
 - * 提出一种策略, 用于在使用用户数据进行广告定位之前匿名化用户数据, 以确保用户隐私。
 - 广告分发系统:
 - * 设计一个系统, 从第三方广告平台获取广告, 并根据匿名用户数据将广告分发给应用用户。
 - * 确保系统可以扩展以支持高量级的广告请求和用户数据处理。
 - 数据存储与处理:
 - * 提出适合存储用户数据、广告元数据和交易日志的数据库架构。
 - * 设计一个数据处理管道, 用于分析用户活动和偏好, 以进行有针对性的广告分发。
 - 系统可扩展性和性能:
 - * 详细说明扩展后端以处理日益增长的用户数量和增加的数据量的方法。
 - * 包括负载均衡、缓存、数据库优化和微服务(如果适用)的考虑。
 - 安全性和合规性:
 - * 概述保护用户数据和系统完整性的安全措施。
 - * 考虑遵守数据保护法规(例如 GDPR)。
 - 技术栈:
 - * 鉴于您在 Rust 和 Go 方面的熟练程度, 请选择其中一种语言进行系统实现。根据系统要求证明您的选择。
 - * 描述其他技术(数据库、云平台、工具)的选择以及它们如何集成到架构中。
- 交付物
 - 一个展示所有组件及其相互作用的高层次系统架构图。
 - 每个组件的详细描述, 包括其角色、使用的技术以及如何扩展。
 - 关于系统如何确保用户数据隐私并遵守相关数据保护法律的简要说明。
 - 设计过程中做出的任何假设。

2 主要业务数据流

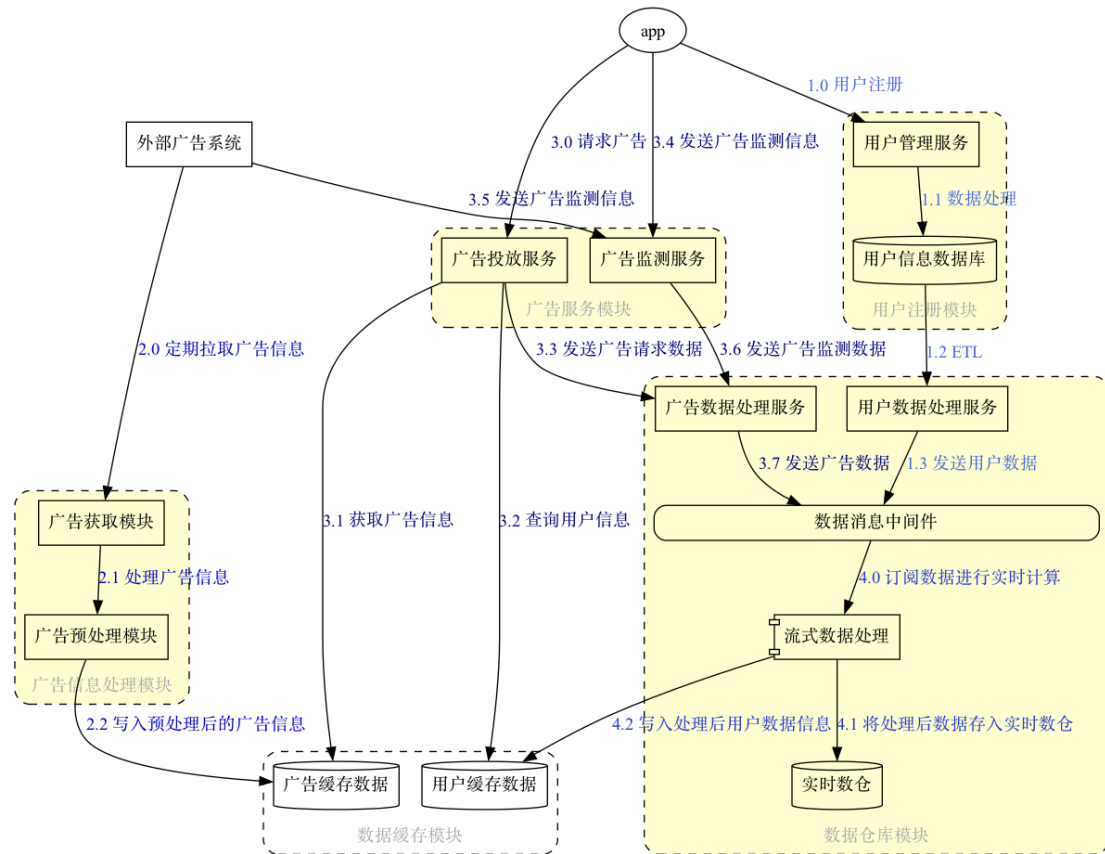
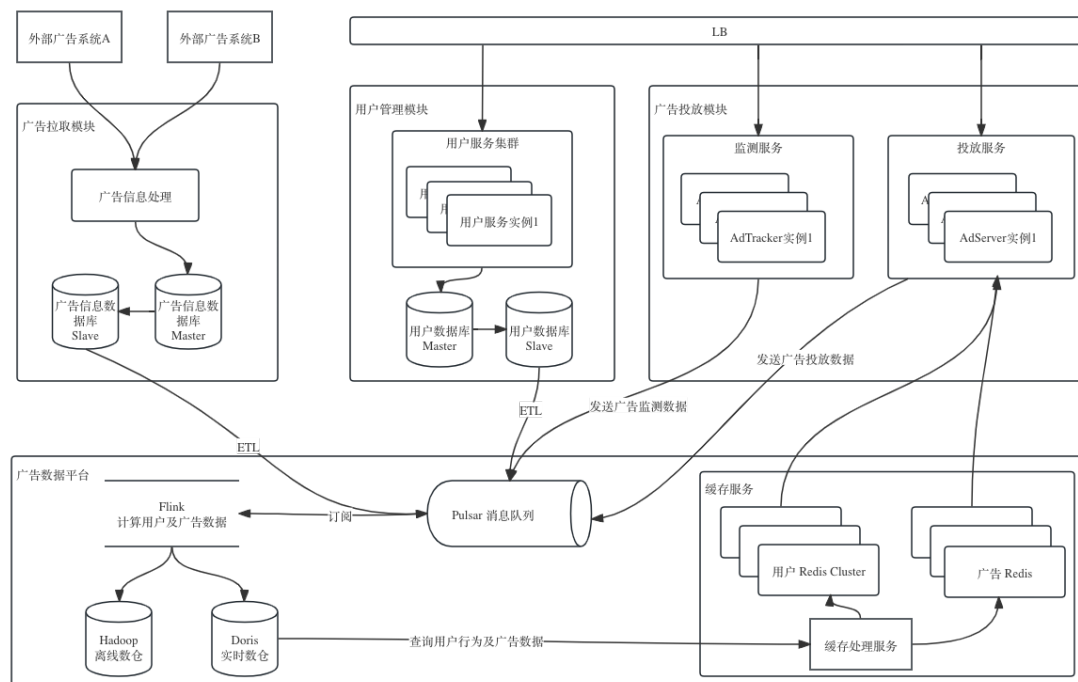


Figure 1: 业务数据流

系统主要涉及四个业务数据流, 分别为:

- 1. 用户注册
- 2. 广告信息同步
- 3. 广告投放
- 4. 用户及广告数据处理

3 系统技术架构



• 系统技术架构简要说明：

1. 系统分为三大模块:广告信息获取模块、广告投放模块和广告数据平台。
2. 在架构图中只体现主要模块的核心应用,在实际实施中可能需要对模块经一步细分。

• 相关技术选型

1. 广告数据平台：

- 首选开发语言:Java
- 消息队列:Pulsar
- 实时数仓:Apache Doris
- 流式计算:Flink
- 离线数仓:基于 Hadoop, 相关组件选择视具体业务场景
- 缓存服务:Redis & Redis Cluster

2. 广告拉取模块：

- 首选开发语言:Java
- 数据库:PostgreSQL (主从部署)

3. 用户管理模块：

- 首选开发语言:Java
- 数据库:PostgreSQL (主从部署、分库分表)

4. 广告投放模块

- 首选开发语言:Rust

• 技术选型不依赖于某家云服务厂商,可支持多云环境。

4 模块简述

4.1 广告数据平台

广告数据处理平台,负责对广告生命周期内所有产生数据的处理和分析。主要包含以下一个模块:

- 消息队列

- 流式计算引擎
- 实时数仓:数据更新频率为分钟级别,用于反馈、修正投放策略。
- 离线数仓:数据更新频率为小时级别,长时间周期存储,用于各类离线复杂数据处理和分析。
- 算法模块:依据各方数据,实时调整投放策略。
- 缓存服务:数据信息查询加速。

4.2 广告拉取模块

主要用于从第三方广告平台拉取广告信息。该模块主要有以下注意事项:

- 需异步获取广告信息,避免在广告投放流程中同步获取。
- 依据不同的广告平台,控制好广告信息更新机制。确保广告信息的有效性。
- 对不同广告平台的广告信息进行整理,保证广告信息数据模型一致,简化广告投放逻辑。

4.3 用户管理模块

提供用户注册及用户信息匿名化管理等能力。

4.3.1 用户匿名化策略

1. 广告 SDK 或终端信息采集 SDK (后续都简称为:ADSDK),收集用户设备敏感性较低的设备指纹信息。包含不限于以下内容:
 - 设备品牌和型号
 - CPU、内存、存储等
 - 系统版本、UserAgent 等
 - 系统更新时间,启动时间等
2. 依据终端所提交的信息,为用户设备生成一个设备唯一标识符。标识符由可变信息生成、不可变信息生成及生成时间三部分组成。
3. 将生成的设备唯一标识写入用户设备中。
4. 用户发起注册时:
 - (a) ADSDK 先尝试获取设备唯一标识,获取成功进入子步骤 2,否则进入子步骤 3。
 - (b) 依据设备唯一标识,判断唯一标识是否已经过期。如过期进入子步骤 3,否则进入子步骤 4。
 - (c) 重新采集设备信息,生成新的设备唯一标识。
 - (d) 使用用户输入信息及 ADSDK 获取到有效设备唯一标识,发起注册请求。

设备唯一标识符生成流程图:

- 待补充

4.3.2 用户注册相关 API 接口

主要注意以下几点:

- 接口的幂等性
- 请求及字段合法性校验
- 数据通信安全,可以考虑端到端的数据加密。

4.3.3 其它说明

1. 设备唯一标识生成,仅基于此前经验。会有思考不周或更优方案
2. 因时间原因,未提供相关 API 接口设计详情。

4.4 广告投放模块

广告投放模块, 提供了广告分发、广告监测及广告预算控制功能(在返现系统中, 可能不需要)

- 广告分发模块: 负责将广告信息高效的发放至用户应用中, 该模块主要有以下难点:

1. 高并发、低延迟
 - 使用 Rust 语言, 消除 GC 机制带来的延迟影响
 - 预处理及缓存, 避免在广告分发过程中进行不必要的计算
 - 设计无状态服务, 可快速横向扩展, 应对高并发及流量波动规律
2. 如何使 eCPM 达到最优
 - 尽量及时的数据反馈机制
 - 有效的投放策略算法

5 非功能性目标

5.1 可扩展性

广告系统并不涉及单点服务, 每个模块都可进行高效横向扩展。但在用户管理模块需注意用户数据分片管理能力, 以应对不断增长的用户数。

5.2 性能

对性能要求比较高的是: 广告投放模块。除了设计角度需要满足高性能要求, 还需要用:

- 日常编码中, 在涉及广告分发流程中的功能, 时刻要保证警惕性, 理解所加入的代码, 对性能产生的影响。
- 形成有效的压测机制, 当涉及广告分发流程代码修改时, 需做好性能压测和分析。

5.3 安全性

- 定期安全漏洞扫描
- 充分依赖云厂商能力, 抵抗 DDOS 等攻击
- 代码实现上, 需要注意设计及编码上产生的功能漏洞

5.4 数据合规

对各类数据合规要求, 了解的比较少。需要在实际业务中, 补充学习。以下几点应该是都需要的:

1. 敏感性信息脱敏
2. 数据传输、保存加密
3. 仅采集必要的数据库

6 其它说明

- 系统比较复杂, 涉及的模块也比较多。时间原因, 无法做到详尽叙述。
- 以上仅依据本人之前相关广告工作经验而设计, 未做相关 research 工作。在实际设计时, 会依据当前广告生态进行相应调整。