

新浪广告系统 服务化优化

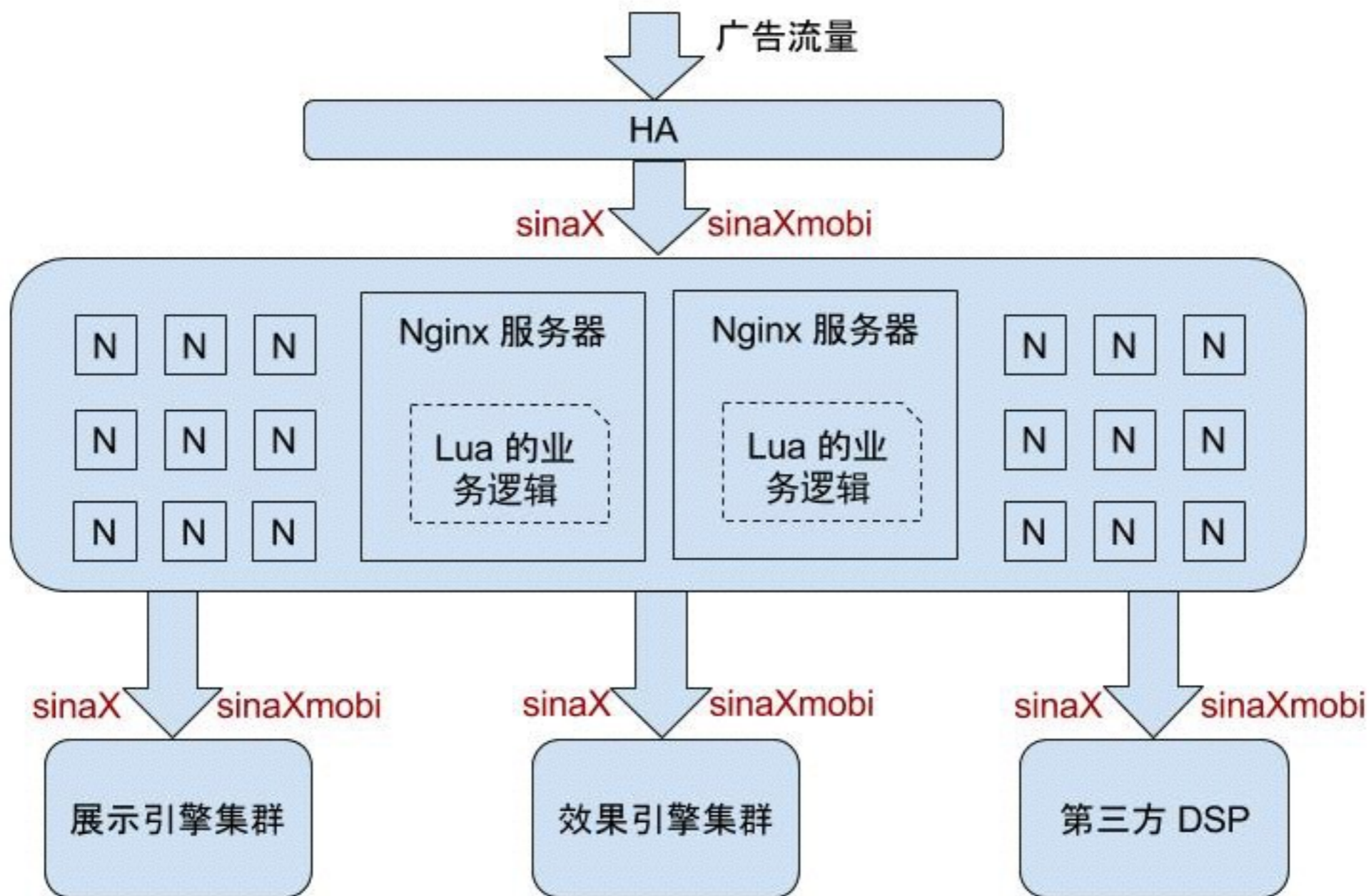
徐挺@sina

Agenda

- 新浪广告系统旧的技术架构和技术痛点
- 新浪广告系统解决痛点问题的技术分析与选择
- **新浪广告系统的服务化过程**
 - 技术上的权衡取舍
 - 对高可靠性和可扩展性的架构设计
- 服务化过程中的技术收获和经验总结

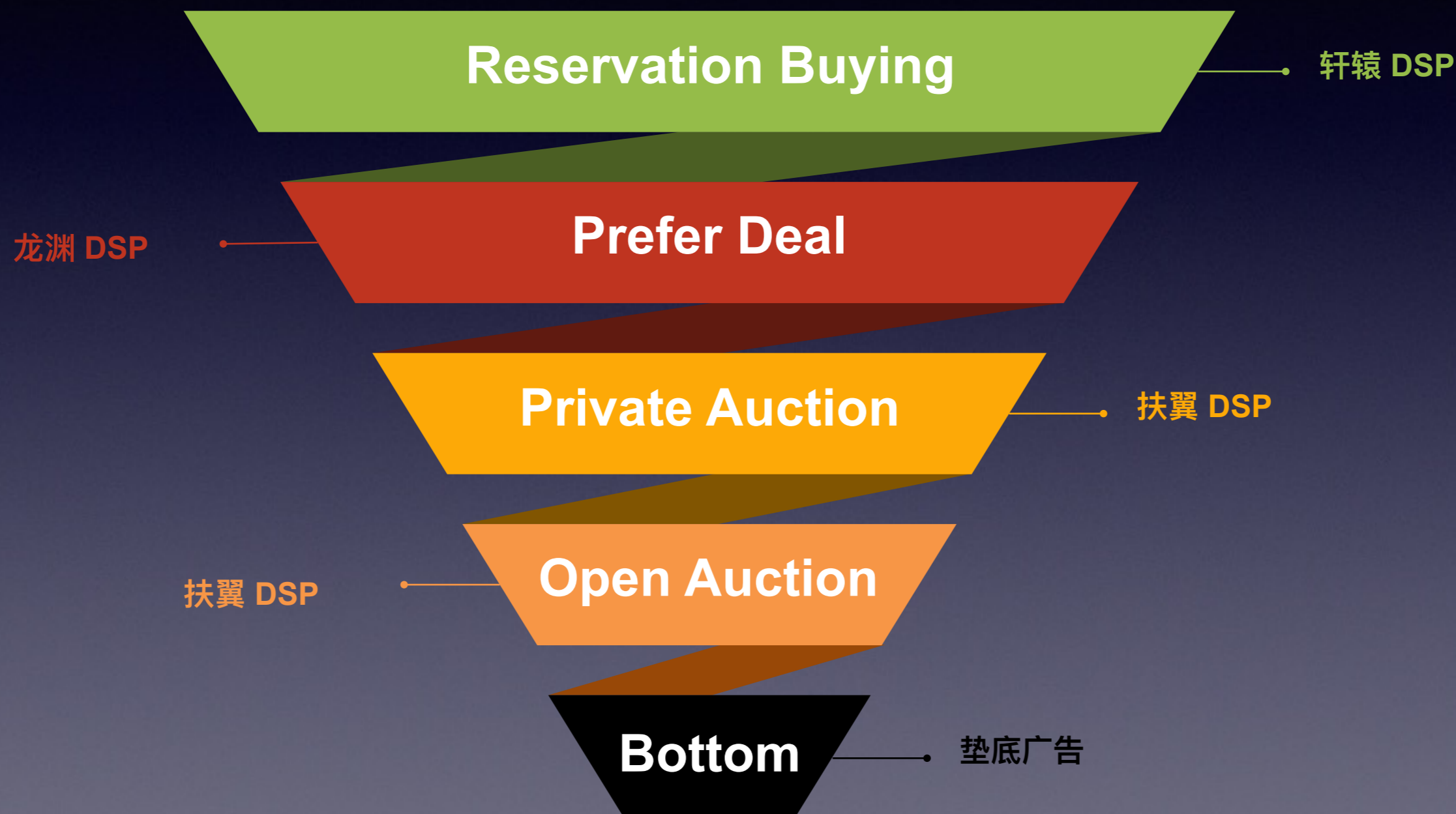
Agenda

- 新浪广告系统旧的技术架构和技术痛点
- ~~新浪广告系统解决痛点问题的技术与选择~~
- **新浪广告系统的服务化过程**
 - ~~技术上的权衡取舍~~
 - ~~对高可靠性和可扩展性的架构设计~~
- ~~服务化过程中的技术收获和经验总结~~

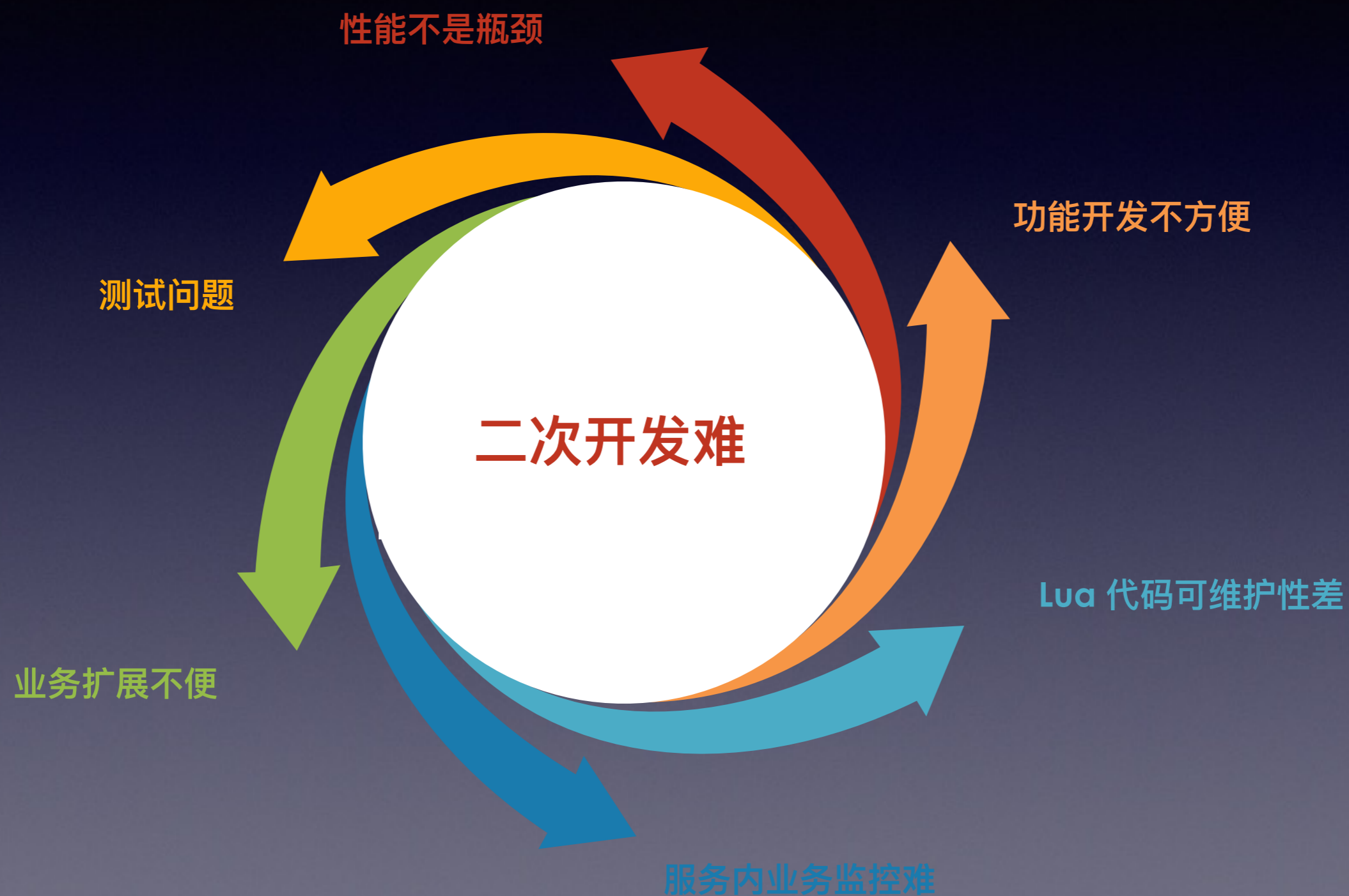


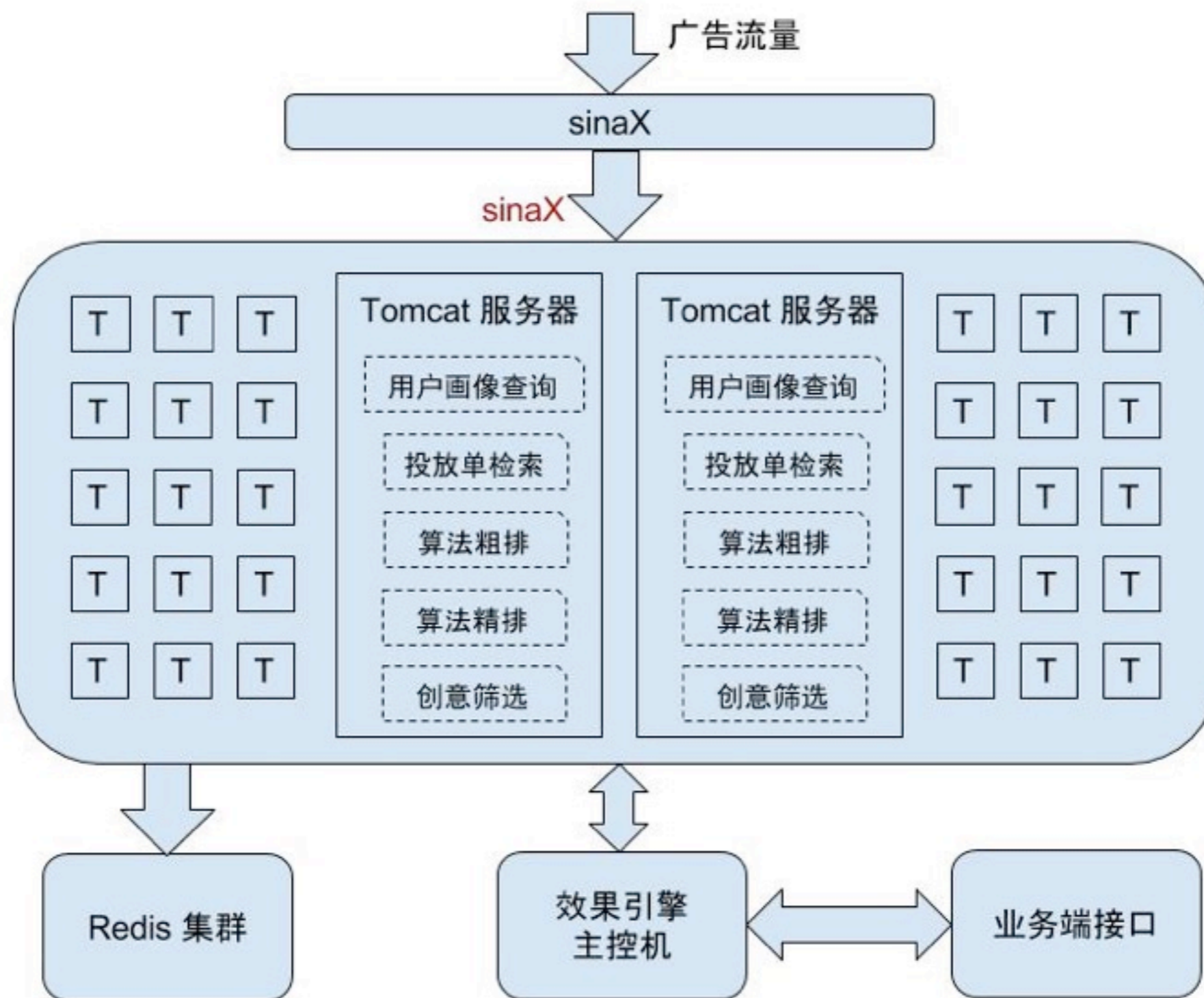
sinaX 服务化之前的架构

流量漏斗型模型



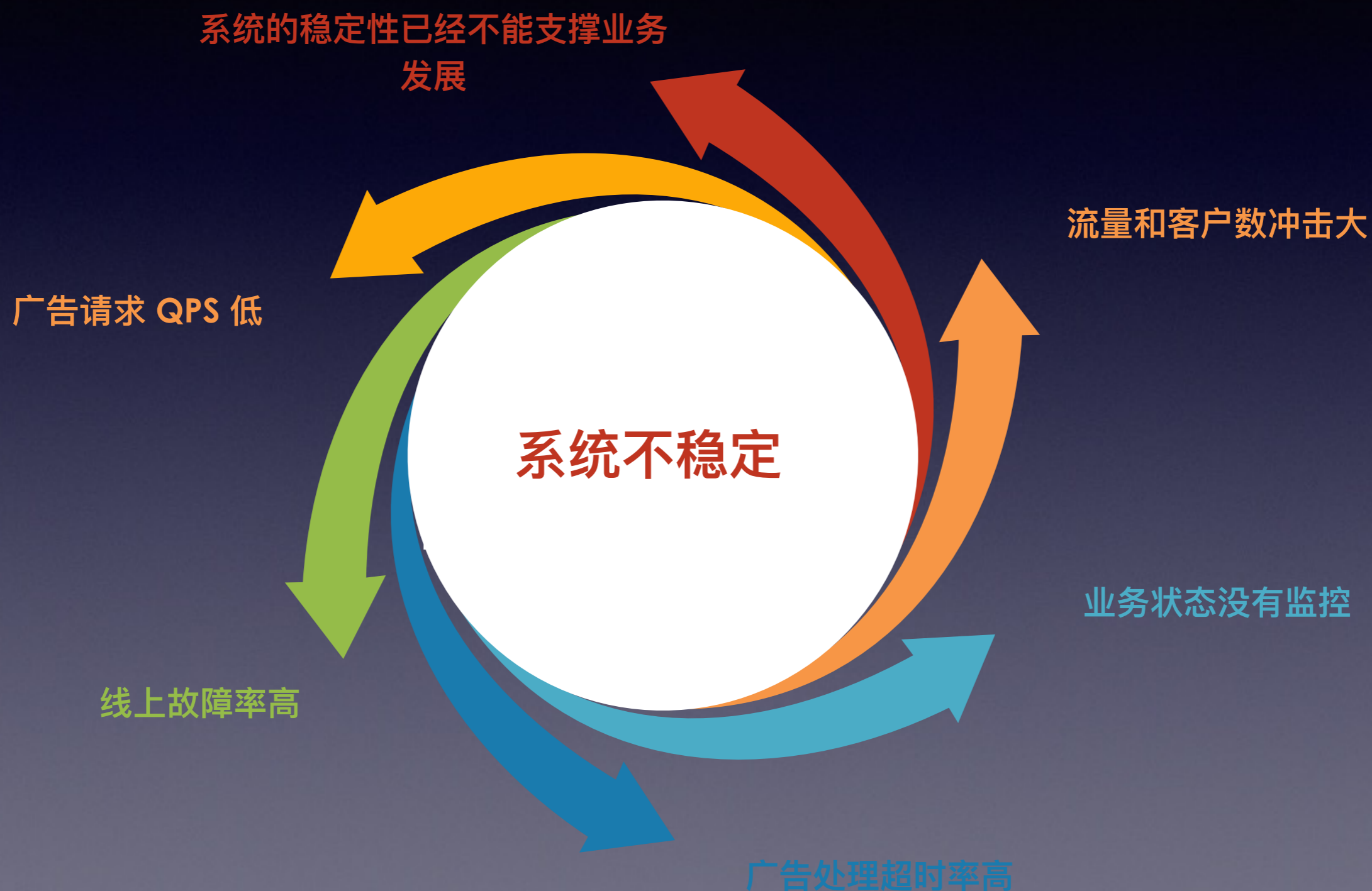
sinaX 旧架构的痛点





广告引擎服务化之前的架构

广告引擎旧架构的痛点



技术痛点的总结

- 功能单体化严重
- 模块间的功能交互是 plug-in 方式
- 编程模型不适应业务的发展
- 系统运行状态不透明

Agenda

- ~~新浪广告系统旧的技术架构和技术痛点~~
- 新浪广告系统解决痛点问题的技术分析与选择
- ~~新浪广告系统的服务化过程~~
 - ~~技术上的权衡取舍~~
 - ~~对高可靠性和可扩展性的架构设计~~
- ~~服务化过程中的技术收获和经验总结~~

痛点问题的技术与选择

1

功能服务化

各服务相互独立，降低业务耦合度，提高产品需求开发的响应时间

痛点问题的技术与选择

2

RPC

服务间的通讯采用 RPC

对比各个 thrift 、 gRPC 、 dubbo
和 Finagle 等 RPC 框架的生态圈，最后采用 Twitter 开源的 finagle RPC 框架。

痛点问题的技术与选择

3

监控

系统状态实时监控与跟踪

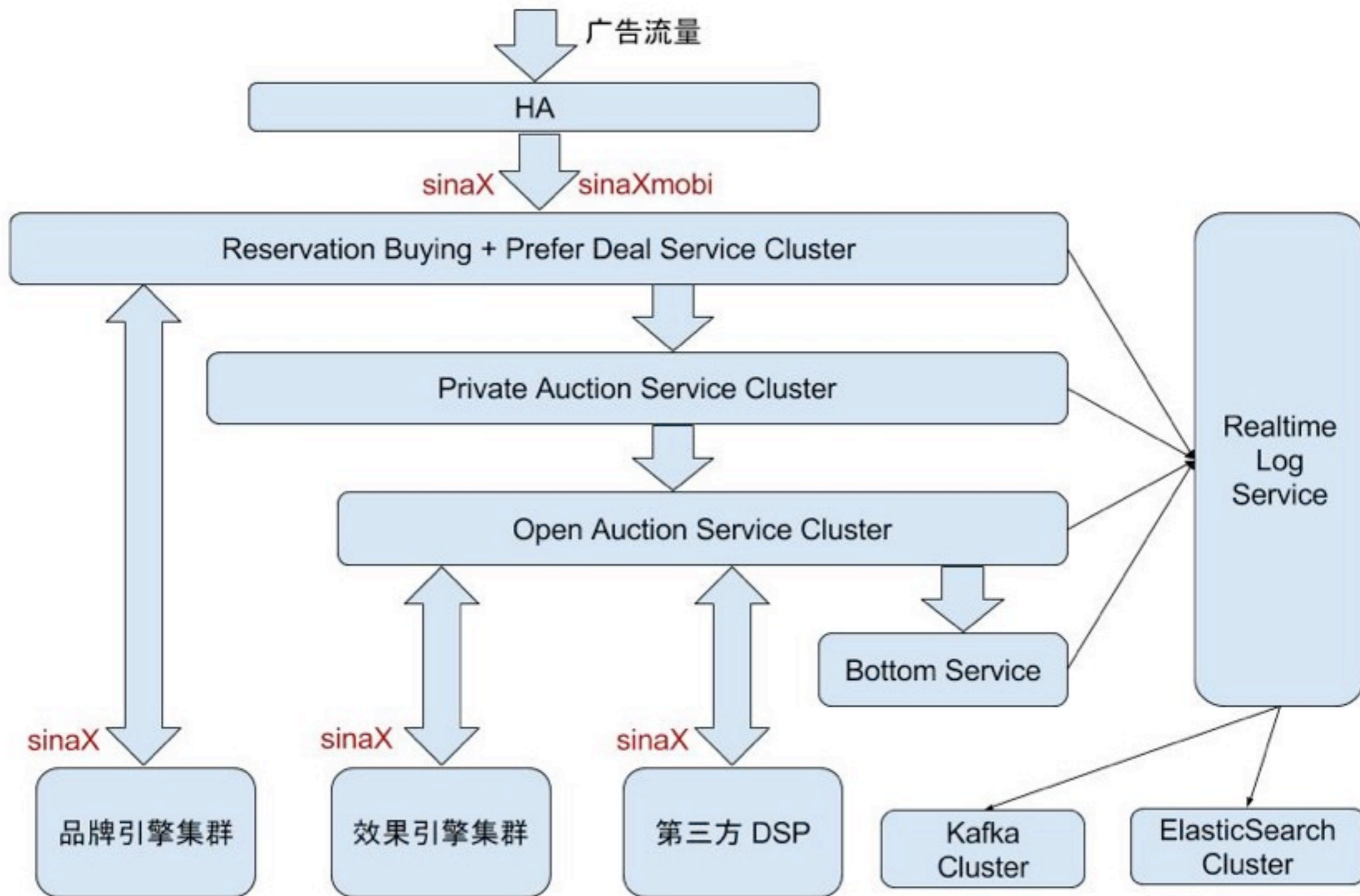
业务数据实时分析与统计

Agenda

- ~~新浪广告系统旧的技术架构和技术痛点~~
- ~~新浪广告系统解决痛点问题的技术分析与选择~~
- **新浪广告系统的服务化过程**
 - 技术上的权衡取舍
 - 对高可靠性和可扩展性的架构设计
- ~~服务化过程中的技术收获和经验总结~~

新浪广告系统的服务化过程

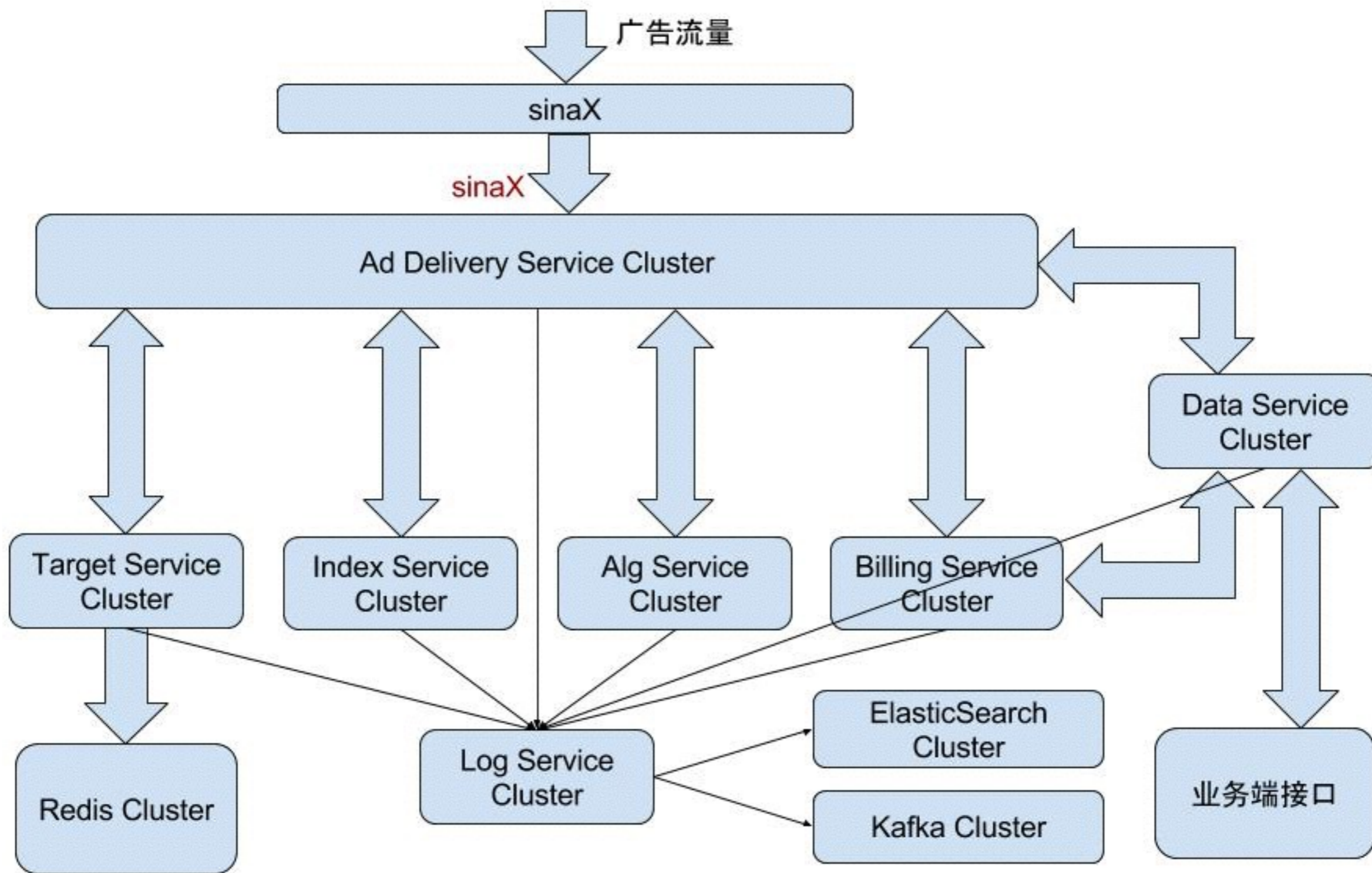
- 技术的权衡取舍
 - sinaX 的服务化过程中服务化分的问题
 - 部分服务的上下游调用过于频繁
 - 带宽占用过大，处理时间的 I / O 占比高
 - 解决：部分服务合并



sinaX 服务化的架构

新浪广告系统的服务化过程

- 技术的权衡取舍
 - 广告检索服务
 - 调用传输数据量太大，影响系统扩展性
 - 通过 rsync 周期性推数据集到投放服务
 - 解决：直接 local 查询



广告引擎服务化的架构

新浪广告系统的服务化过程

- 高可靠性和可扩展性的架构
 - 系统容错性
 - 面向错误进行系统设计
 - 服务的无状态设计
 - 故障服务的自动替换

新浪广告系统的服务化过程

- 高可靠性和可扩展性的架构
 - 系统可靠性
 - 跟踪监控程序运行状态和热点
 - 实时监控系统异常，直接隔离异常服务
 - 服务分等级做多重备份

新浪广告系统的服务化过程

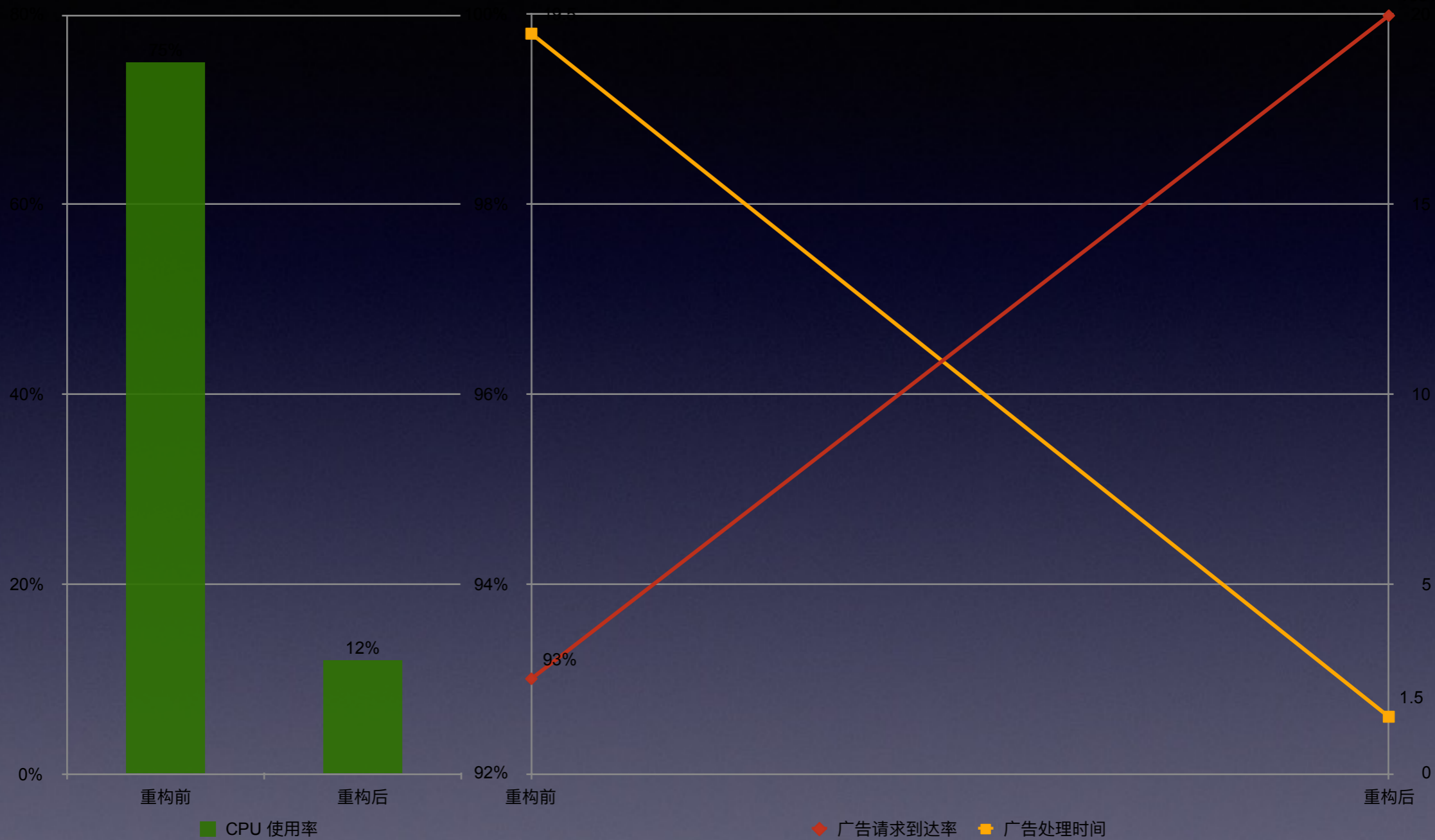
- 高可靠性和可扩展性的架构
 - 系统处理能力的可扩展
 - 单机服务处理能力的线性化能力
 - 系统的服务处理能力扩展关键在资源分配
 - 重点关注系统的数据热点

新浪广告系统的服务化过程

- 高可靠性和可扩展性的架构
 - 系统架构的可扩展
 - lambda 架构的模式
 - 数据一致性是核心，分等级给予保证
 - 一致性 hash 的问题

新浪广告系统的服务化过程

- 高可靠性和可扩展性的架构
 - 系统架构的可扩展
 - 业务功能和日志功能的隔离
 - 服务接口的选择：RPC 和 Kafka
 - 系统的自动快速部署上线



系统优化效果

Agenda

- ~~新浪广告系统旧的技术架构和技术痛点~~
- ~~新浪广告系统解决痛点问题的技术与选择~~
- **新浪广告系统的服务化过程**
 - ~~技术上的权衡取舍~~
 - ~~对高可靠性和可扩展性的架构设计~~
- 服务化过程中的技术收获和经验总结

技术收获和经验总结

- 服务划分的粗细粒度把握
 - 切忌简单按照业务逻辑进行服务逻辑
 - 核心是找出真正的业务中心和业务支撑
 - 没有统一标准，关键的是 trade-off

技术收获和经验总结

- 服务的优化重点在业务逻辑的优化
 - 系统的热点不会随着服务化自然消失
 - 服务优化的核心是对业务逻辑的深刻理解
 - 技术的实现只是手段

技术收获和经验总结

- 服务需要平滑地切换上线
 - 系统的服务化过程就像飞行过程中换引擎
 - 灰度上线阶段时需要新旧系统并行相互兼容
 - 单个功能的服务化上线对系统整体的影响

技术收获和经验总结

- 服务治理
 - 服务处理能力的自动降级和恢复
 - 服务的保护：负载均衡和熔断
 - 服务的透明化运行

Any Question?