



D 2 T E R M I N A L

基于OTel的移动端全链路 Trace建设思考和实践

高玉龙（元泊）

阿里巴巴-阿里云高级开发工程师

目录

01

1. 背景介绍

02

2. 解决思路

03

3. 方案介绍

04

4. 总结



第十七届

D2 终端技术大会

2022.12.17-12.18

前端 & 客户端

合作伙伴

阿里云开发者社区
ALIBABA CLOUD DEVELOPER COMMUNITY

大淘宝技术
TAO TECHNOLOGY

老司机技术
SWIFT OLDDRIVER



segmentfault 思否



稀土掘金

钉钉开发者

InfoQ 极客传媒

T Salon

YOUKU 优酷

* 排名不分先后顺序



扫码进入D2官网

背景介绍

研发人员

2个人

1000个人

代码行数

5000行

100万

➤ 研发过程变化大

架构规模

10个模块

100+模块

➤ 问题难以复现和排查

发布频率

一个月

一周



产品成熟度

修复时间

1个小时

5个小时

为什么很难复现

01

多端日志不统一

Android、iOS、微服务等多端日志协议

不统一，日志分析难

02

端侧数据采集难

调用链路复杂、模块多，技术框架不同；

代码隔离，模块间状态传递频繁；

设备碎片化，网络环境复杂

03

多系统数据关联难

不同框架、系统的数据获取难；

数据分析缺少上下文信息，数据关联难

04

确定问题难

模块、系统依赖多，复现问题难；

对应域的同学多人排查，人肉运维成本高

解决思路

统一数据采集标准



平台、框架无关，统一采集能力



数据上下文自动关联



领域经验沉淀，自动化经验分析

统一数据采集标准

业务现状

- 数据采集没有统一的规范
- 跨端/跨系统日志分析难

行业方案



- 协议/数据类型不统一
- 方案之间难以兼容/互通

解法

- 各个端基于OTel协议统一实现
- 基于OTel协议统一存储、处理、分析



- 跨语言规范
- API / SDK
- Exporters
- Collectors



统一采集

统一协议

存储、处理、分析

端侧数据采集的难点



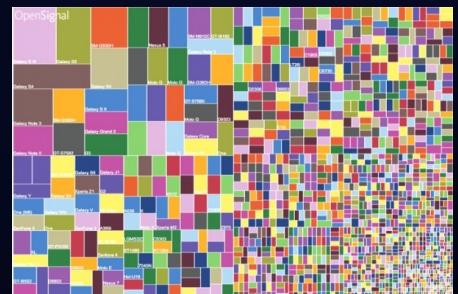
数据串联难

- 框架多，模块多，业务差异大
- 线程/协程，完全异步调用
- 可观测数据源多



性能保障难

- 40+ 系统版本
- 10000+ 机型



不丢数据难

- App崩溃、OOM、非优雅退出
- 全球 150 + 电信运营商
- 跨洋数据传输，链路质量波动，劫持

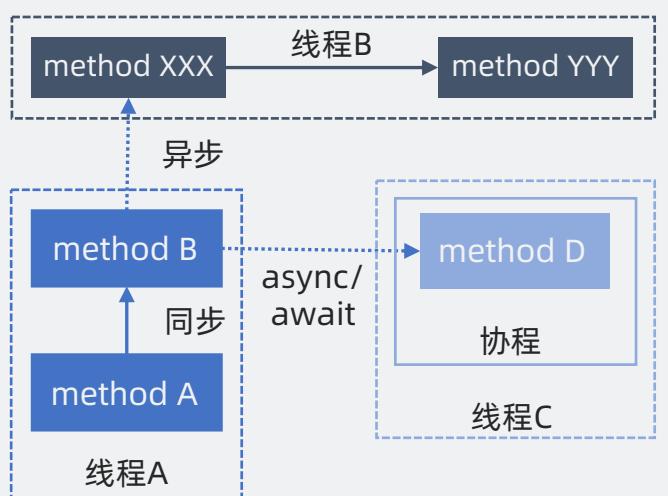


端侧数据串联的难点



可观测数据来源

- ◆ 可观测数据来源多
- ◆ 研发框架技术能力对外黑盒



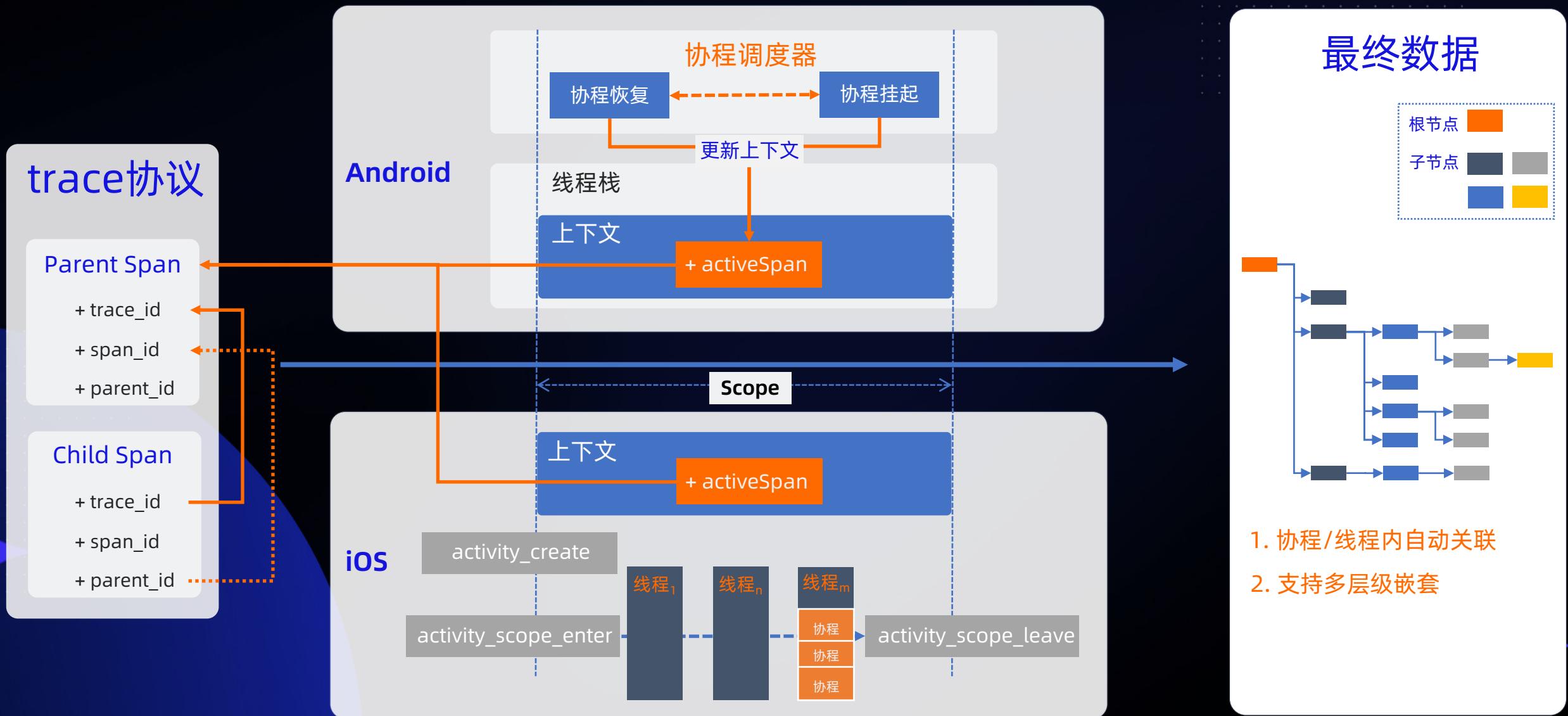
同步/异步调用形式

- ◆ 端侧业务链路完全异步调用
- ◆ 异步调用方式多样化

共性问题

1. 三方框架数据如何采集、串联？
2. 不同可观测数据如何串联？
3. 线程/协程如何自动串联？

端侧数据自动串联方案



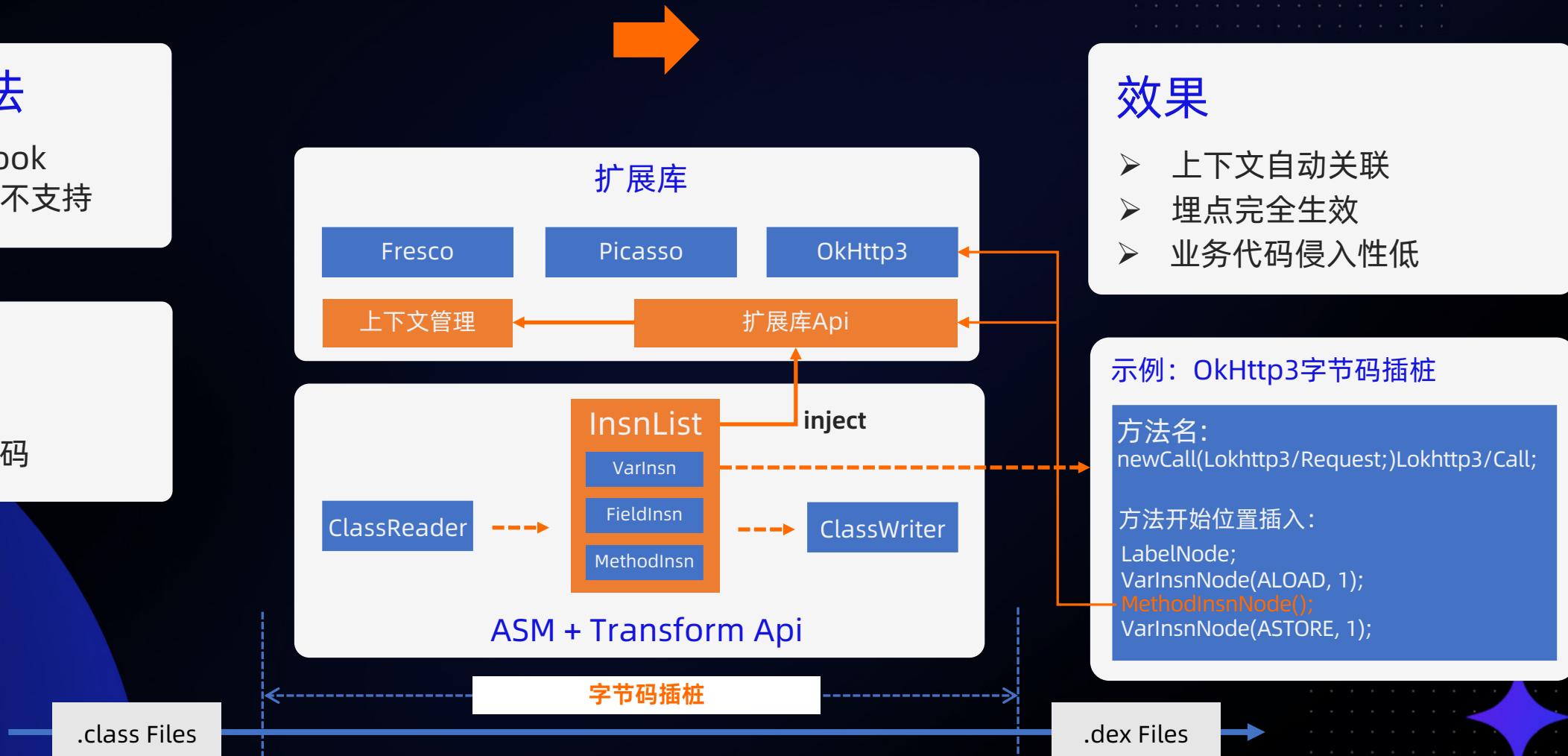
三方框架数据采集和串联

业内常见做法

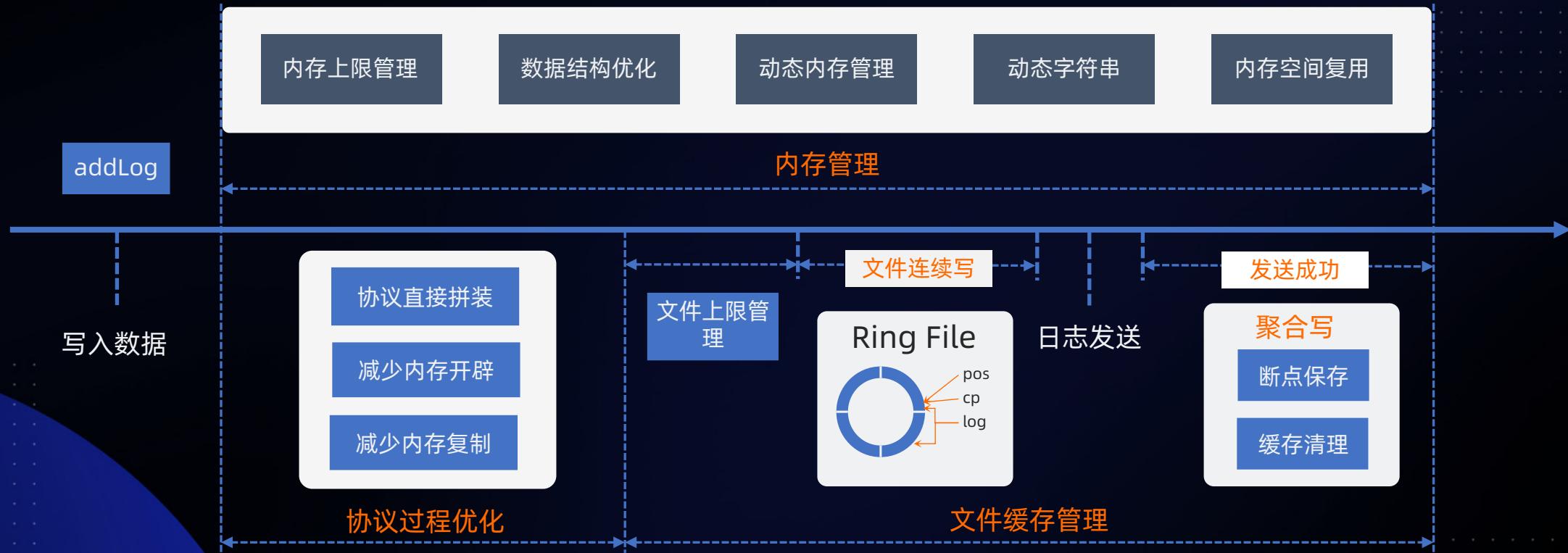
1. 拦截过滤器、Hook
2. 其他扩展方式或不支持

共性问题

1. 埋点不完全
2. 需要侵入业务代码



如何确保性能



性能影响因素

1. 内存受限, 频繁GC, 易OOM
2. 频繁协议化, 系统占用高
3. 数据离线缓存 I/O 负载
4. 端碎片化, 多端性能不一致

核心收益

- 吞吐量提升2倍
- 内存最高降低50%

CPU最高降低60%

C内核实现, 多平台性能一致

如何确保日志不丢失

应用生命周期不可控



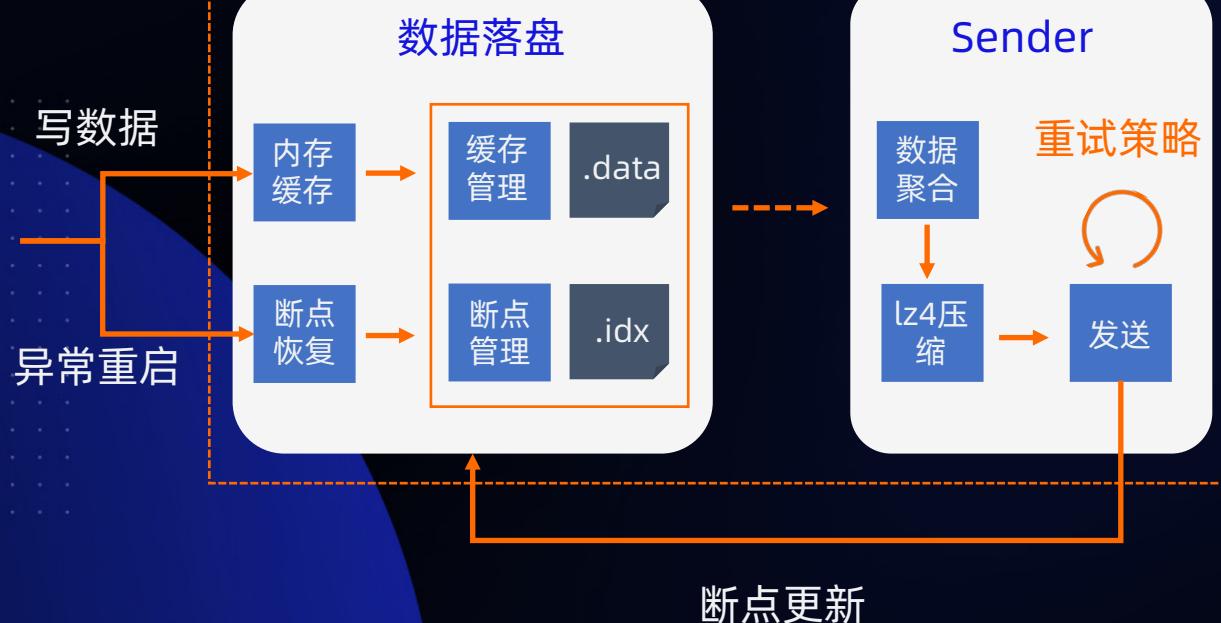
- 应用非优雅退出
- 设备掉电，异常重启

网络延时高，抖动大



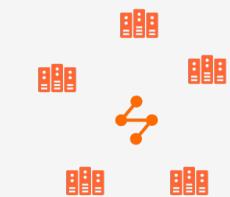
- 运营商网络环境复杂，网络劫持问题
- 偏远地区、跨洋数据传输，链路质量波动

WAL(预写日志)



就近接入

边缘节点



自建网络加速通道

SLS



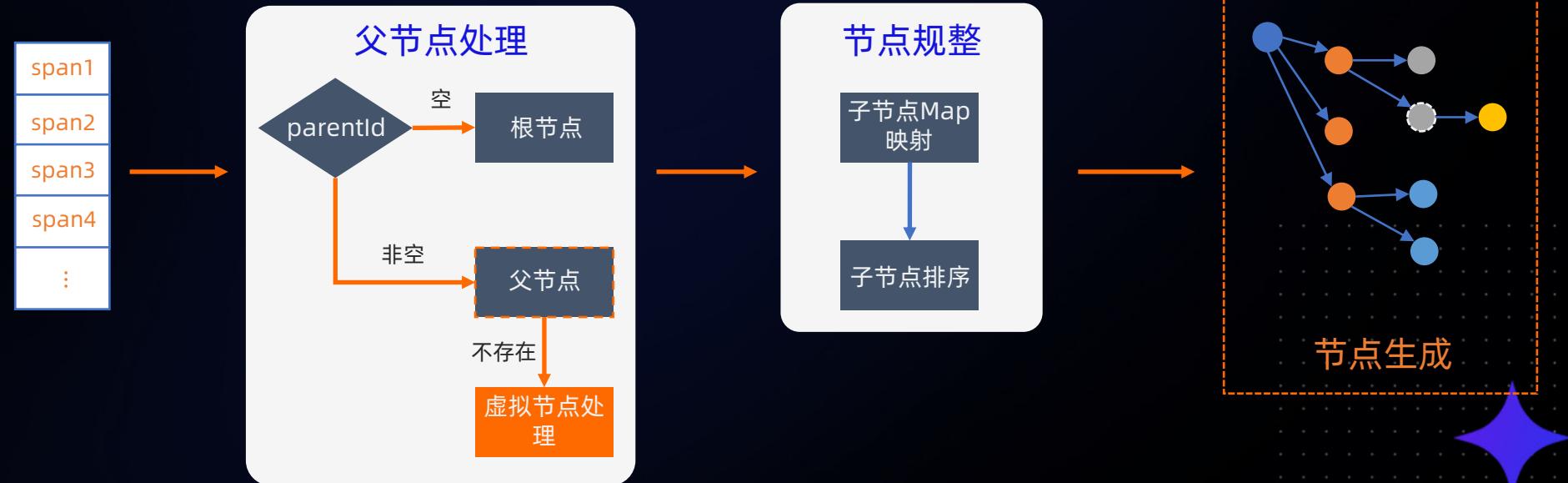
效果

- ❑ 数据包体积下降2.1倍
- ✓ 发送成功率99.3%

- ↖ QPS提升13倍
- ▣ 网络延时下降50%

加速前 加速后

多系统数据关联处理



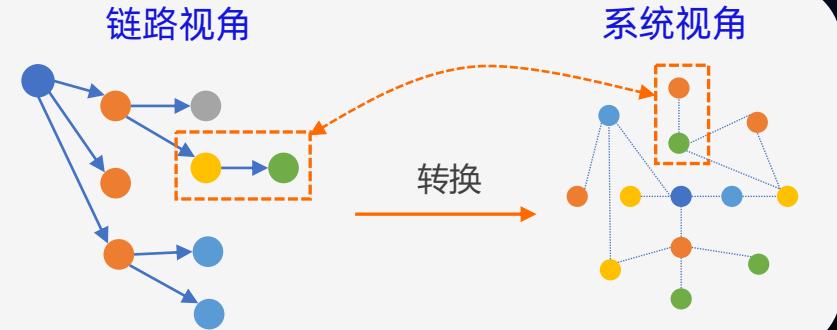
多系统数据拓扑生成

挑战

Trace数据规模往往比较大，时效性要求高，怎么做？

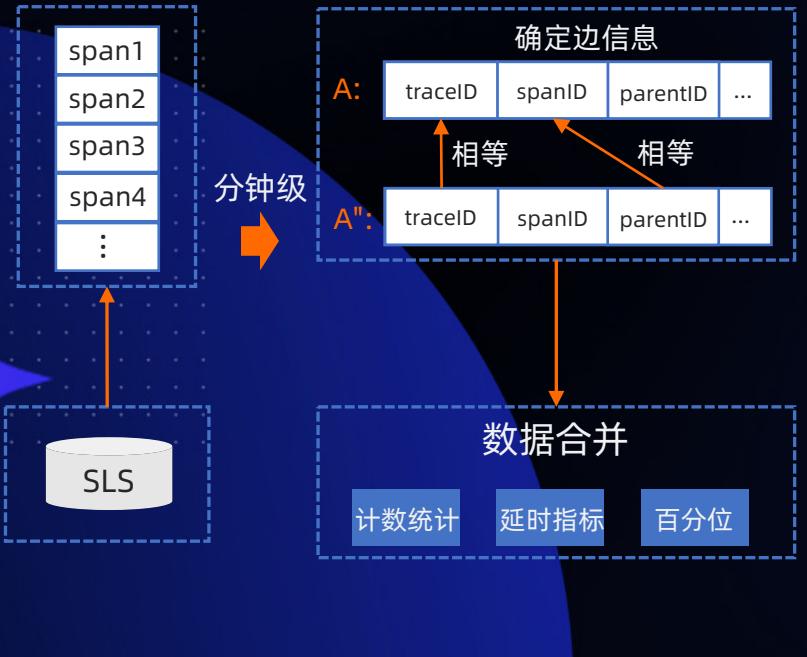
解法

1. 流处理 join 耗时高，把流处理问题转为批处理问题
2. 中间态最大化价值



数据源

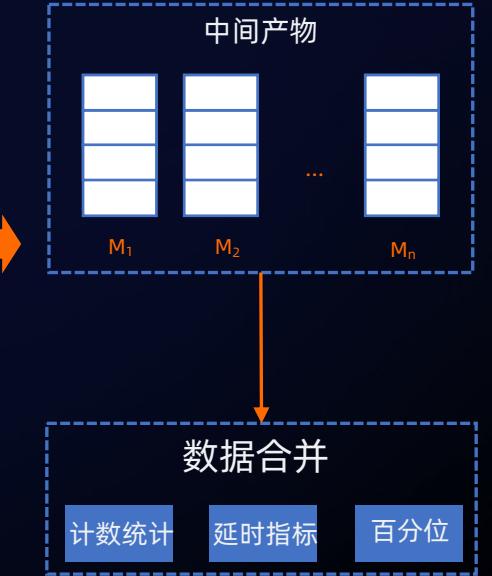
Map阶段



中间产物



Combine阶段



最终产物



自动化问题根因定位探索

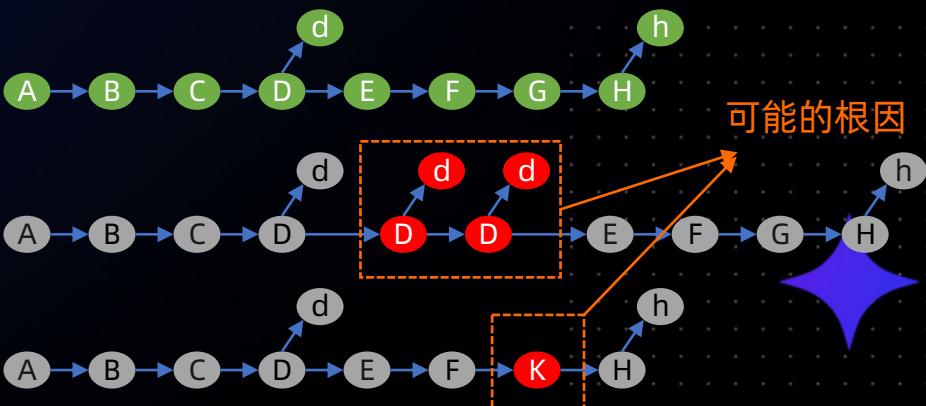


问题

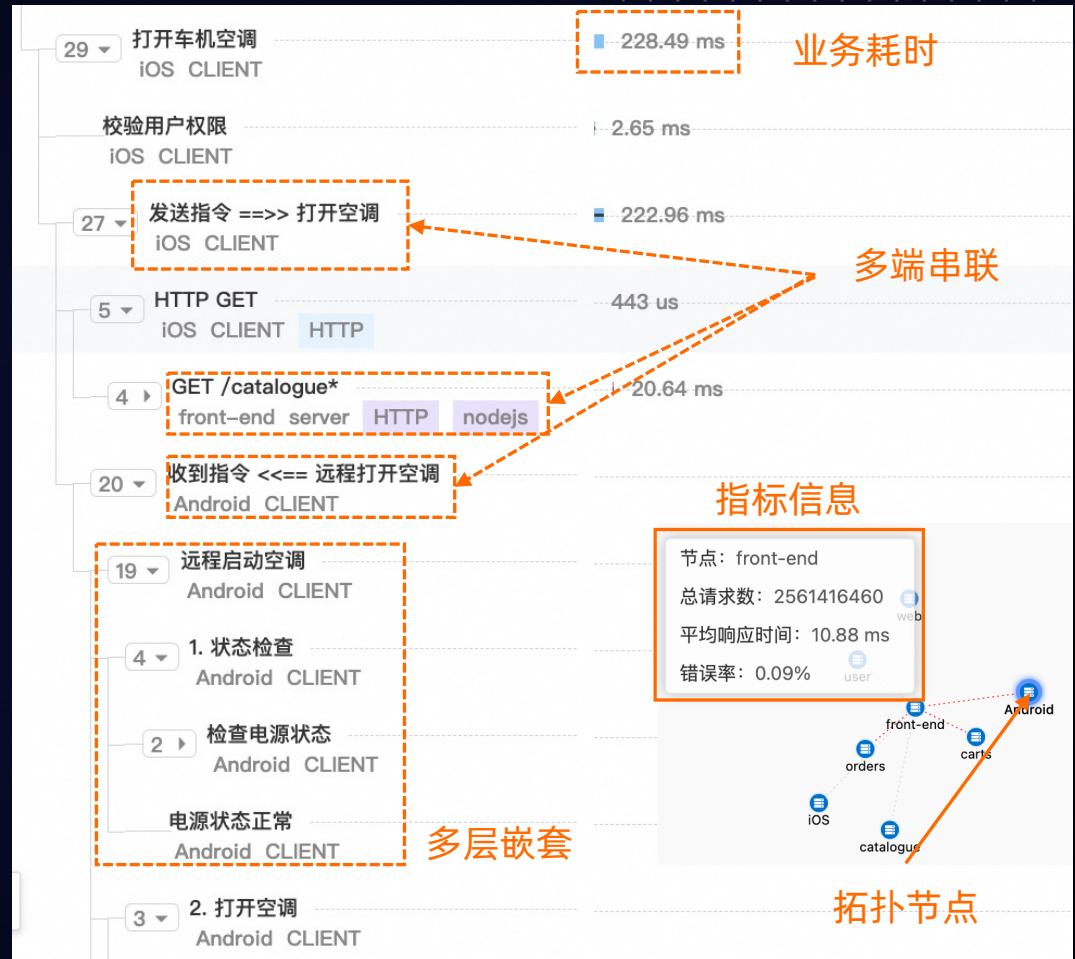
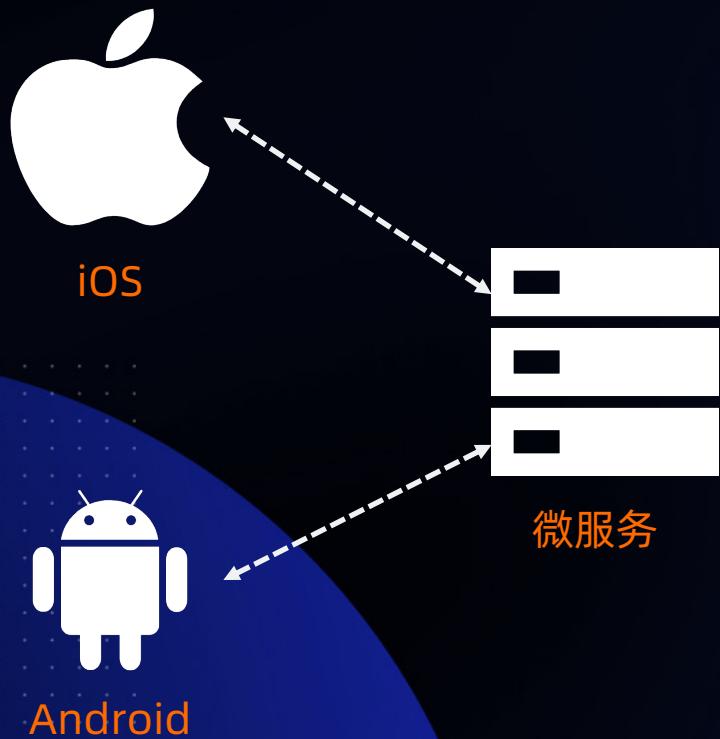
1. 业务模式多，数据量大，不确定性大
2. 业务系统复杂多变，请求链路复杂多变
3. 多人参与排查，人肉运维成本高

思路

- 业务链路特征实时生成
- 模式识别，业务自动聚类
- 图算法，根因自动定位



案例：多端链路追踪



整体架构方案

上层应用

链路分析

拓扑查询

指标查询

日志查询

根因定位

数据处理

指标预计算

Trace链路生成

依赖关系生成

拓扑结构

特征处理中心

数据存储

统一存储

数据源



总结



01

- 完善插件、注解等方式采集支持
- 更多三方框架的支持



02

- 丰富可观测数据源
- 网络质量采集
 - 性能数据采集等



03

扩展端侧应用场景

- 用户访问监测
- 性能 Profiling 等



04

能力开源，贡献社区



扫码回复「D2」
获取第十七届 D2 演讲 PDF 材料

后续也将推送 D2 会后技术文章，敬请关注！！





THANK FOR YOUR WATCH

感谢大家观看