

UBC Android 时长不自洽修复方案 - HandleID持久化失败

目录

- 背景&原因
- 解决方案
 - 方案1: 冷启动阶段无条件清除无尾流日志
 - 方案2: endFlow添加uuid作为update条件

时长不自洽问题整体现状

1. 背景&原因

HandleID 在端上持久化在 SP 中，进程退出时有可能出现 SP 持久化失败或没有来得及持久化的情况，下一次生命周期的时长点位获取到重复的 handleID，导致旧日志被修改上报。

目前通过查询日志数据表没有很好的方式去筛选出满足这种条件的日志



result_20240506_142816_369.xlsx

28.7MB

2. 解决方案

方案1: 冷启动阶段无条件清除无尾流日志

这部分不自洽的时长点位，在原本的逻辑中属于无尾流的日志点位，可以在冷启时无条件清除无尾流数据；

历史提交记录中，冷启时原本保留了清除无尾流数据的逻辑，但是后来被注释了。现行逻辑中，冷启动阶段会判断日志开始时间作为是否要清除无尾流日志的条件

v13.57 固化了减少清除无尾流时机的实验

【优势】无条件清理无尾流日志，对数据库日志条数和磁盘空间有正向影响

【风险】历史版本冷启动清除无尾流条件的变动原因

方案2: endFlow添加uuid作为update条件

endFlow 在UBC SDK内部补充 uuid 作为 update 条件

【优势】更保险，能够覆盖更多场景

【缺点】修改成本更高，API方法需要添加对应 uuid 参数

评审意见：采用方案2

CR时发现方案2级联的改动很大，临时改用了冷启时即更新 SP 中的全局 handleID，以保证 handleID 及时持久化。修复效果后续在时长不自洽日志回捞分析中确定