UBC.onMutilProcessEvent()打点方法支持pv抽样方案

目录

- 背景
- 1.1 UBC.onMutilProcessEvent()
- 1.2 影响点位
- · 方案
 - 2.1 子进程读取SQLite点位配置表
 - 2.3 日志协议字段补充

1.1 UBC.onMutilProcessEvent()

● UBC初始化仅在主进程执行。一般的,子进程中业务方调用UBCManager暴露的打点API会通过进程间通信(IPC)在主进程执行UBC逻辑;

UBC.onMutilProcessEvent()方法用于多进程数据文件存储。子进程调用该方法会将日志直接持久化到本地文件,不通过 IPC ,避免了某些 业务场景下打点唤起主进程的问题。

但该**方法在子进程调用缺少点位抽样逻辑**,导致网络库94,850点位(均1%抽样)调用该API后流量增长。

```
* <code>@param eventId event id</code>
                                      * @param content content
                                      * @param fileName 文件名
                                     public static void onMutilProcessEvent(String eventId, String content, String fileName) {
                                         String name = UbcUtils.processFileName(fileName);
                                             Log.d(TAG, msg: "on onMultiProcessEvent id:" + eventId + " value:" + content + " name: " + name);
                                             if (AppProcessManager.isServerProcess()) {
222 2021/6/16 litianhua02
                                                 BehaviorProcessor.getInstance().multiProcessEvent(eventId, content, name, UBG.FLAG_EVENT_FILE);
                                                 BehaviorProcessor.multiProcessSaveEvent(eventId, content, name, UBC.FLAG_EVENT_FILE);
                                         }catch (Exception e){
                                             YalogHelper.saveRuntimeLogWithMsg( msg: "id:" + eventId + ";content:" + content + "fileName:" + fileName;
                                                     EnumConstants.RunTime.ON_MULTI_PROCESS_EVENT_ERROR);
```

apk反编译查询调用代码,手百包中有5处代码调用:

1.2 影响点位

app-debug.apk **I** Inputs Scripts

pv抽

样

图片

```
Source code
                                                                                               Usage for: n com.baidu.ubc.UBC.onMutilProcessEvent(String, String, String) void
                                                                                                                                               Node
    □ bolts
b cn
co.touchlab.stately
com
a a
a a
a a
a airbnb.lottie
a alipay
a android
a sus.msa
b async.download.manager.sdk
b b.a.a
b baidu
b abtest
b abtestv2
b abymg
b account
c a ad.magic.flute
a ai
a android
b annotation.sqlite
b antibot.ymgbehaviorverify
apssearch
a spssaras
a ar
                                                                                                 Found 5 (complete)
                                                                                                                                                                                                                                                                                  Open Cancel
                                                                                             220
222
               apsaras ar ar
```

id

点位

影响点位有5个:

备注

```
无
  765
      多进程ipc异常问题统计
  94
      https连接监控
                      1%
  850
      手百网络库错误日志收集
                      1%
      华为地理围栏付费拉活百
                      无
  6973
      度进程以触发push
  7546
      Ach拉取漏斗建设点位
                      无
                           push做了封装,没有直接关联到点位;和负责人沟通,目前只有7546点位在用
2. 方案
```

过 UBCDatabaseAdapter/UBCDatabaseHelper 初始化 BehaviorRuleItems 将点位配置从 SQLite 数据库读取到内存中 </>

1

2.1 子进程读取SQLite点位配置表

EventData data = new EventData(eventId, value, option);

* 子进程里打点,直接写入文件(和原逻辑相同,绕过Processor初始化,避免注册各种监听)

UBC目前已经封装了 BehaviorRuleManager ,包含点位开关、打点限流和点位抽样逻辑,判断逻辑依赖从数据库读取的点位配置;通

```
public static void multiProcessSaveEvent(String eventId, final String value
   5
                                                    final String fileName, int option) {
              /**** start ****/
              if (!UBCMultiProcessHelper.needRecordEvent(eventId, option)) {
   9
   10
   11
              /**** end ****/
   12
   13
              data.setFileName(fileName);你
   14
              BehaviorFileAdapter mFileAdapter = new BehaviorFileAdapter(UBCHelper.getAppContext());
   15
              mFileAdapter.saveEvent(data,
   16
                      BehaviorRuleManager.getInstance().checkRealTimeUpload(data.getId()));
   17
   18
   19
   20 ### UBCMultiProcessHelper.java
  21 public class UBCMultiProcessHelper {
         private static UBCDatabaseAdapter ubcDatabaseAdapter;
   22
   23
   24
   25
           * 是否需要记录本次事件打点
   26
           * @param eventId 事件ID
   27
           * @param option 事件配置项
   28
           * @return boolean true: 记录,记录打点流程; false:不记录,中止打点流程
   29
          */
   30
          public static boolean needRecordEvent(String eventId, int option) {
   31
              BehaviorRuleManager mBehaviorRule = BehaviorRuleManager.getInstance();
   32
              // 懒加载 实例化ubcDatabaseAdapter的同时初始化ubc规则
   33
              if (ubcDatabaseAdapter == null) {
   34
                  ubcDatabaseAdapter = new UBCDatabaseAdapter(UBCHelper.getAppContext());
   35
                  ubcDatabaseAdapter.initRuleCache(mBehaviorRule.getRuleItems());
   36
   37
   38
              // 避免BehaviorProcessor实例化,代码迁移自com.baidu.ubc.BehaviorProcessor.EventRunnable#needRecordEvent
   39
              if ((option & BehaviorRule.OPTION_SAMPLE) != 0) {
   40
                  if (!UBCHelper.getUBCContext().isSampled(eventId)) {
   41
                      return false;
   42
                  3
   43
   44
   45
              // 检查id 是否关闭
             if (mBehaviorRule != null && !mBehaviorRule.checkRecord(eventId, option)) {
   46
                  return false;
   47
   48
              // 打点达到一定频率后限流
   49
   50
              if (mBehaviorRule != null && mBehaviorRule.isControl(eventId)) {
                  return false;
   51
   52
              // 检查id 是否命中UBC 抽样
   53
              if (mBehaviorRule != null && mBehaviorRule.checkSample(eventId)) {
   55
                  return false;
   56
              // 被动点位不处理
   57
              if (mBehaviorRule != null && mBehaviorRule.checkPassiveId(eventId)) {
   58
   59
   60
   61
              return true;
   62
   63 }
实例化/初始化类图
                                                                                                   + onEvent()
+ beiginFlow(): Flow
+ flowEnd()
+ flowAddEvent()
                                              UBCDataUploade
         LifeCycleListener
```

```
onActiivtyStopped(): void
                                                                                                                                                                                       mUploadTimingManager: UBCUploadTimingManagmUploader: IUBCUploader
                                                                                                                                                                                      mBehaviorRuleManager: BehaviorRuleManage
                                      onLogTooMany(): void
                                                                                                                                    databaseHelper: UBCDatabaseHelpe
                             未在子进程实例化
                             当前已经在子进程实例化
通过dump验证子进程实例化的对象
```

20 }

```
2.3 日志协议字段补充
UBC.onMultiProcessEvent() 方法子进程调用会缺少日志字段,如 app_version, uuid 等,示例:
 </> 765日志详情
   1 {
       "content": {
         "useContentResolverCall": {
           "process_name": "com.baidu.searchbox:widgetProvider",
           "stack_trace": "java.lang.IllegalArgumentException: Unknown authority com.baidu.searchbox.provider.ipc.server\n\
           "process_info": 400,
   6
           "report_time": 1746870475152
   8
         "useContentResolverQuery": {
   9
           "process_name": "com.baidu.searchbox:widgetProvider",
  10
           "stack_trace": "java.lang.NullPointerException: Attempt to invoke interface method 'android.os.Bundle android.da
   11
   12
           "process_info": 400,
           "report_time": 1746870475160
  13
   14
         }
   15
       },
       "id": "765",
   16
       "idtype": "0",
   17
       "timestamp": 1746870475260,
  18
       "type": "0"
  19
```