

超长时长修正方案

目录

- 一、背景&现状
 - 1.1 背景
 - 1.2 双打duration
 - 1.2.1 duration
 - 1.2.2 cduration
- 二、修正方案
 - 2.1 方案概述
 - 2.2 阶段数据影响面评估
 - 2.3 修正操作
 - 2.3.1 修正端
 - 2.3.2 时间戳修改评估
- 三、附录
 - 3.1 参考文档
 - 3.2 双端数据
 - 3.2.1 系统时间跳变影响
 - 3.2.2 超长时长中正常时长pv占比
 - 3.2.3 超时时长
 - 3.2.4 阶段数据评估
 - 3.2.5 新版本双端修复数据评估

一、背景&现状

1.1 背景

os	原因排查	Count:cuid
- ANDROID	Total	61
	app在前台实际时长大于1天	24
	真实时长小于1天，系统时间跳变	1
	真实时长小于1天，进程冻结影响	36
- IOS	Total	39
	app在前台实际时长大于1天	27
	真实时长小于1天，系统时间跳变	12
Total		100

duration > 24h 超时时长归因

在超时时长Case的排查过程中，随机选取100名单条日志时长超过24h的用户进行日志分析。将超时时长的原因归为三类：

- 【Android, iOS】App在前台实际时长大于1天
- 【Android, iOS】真实时长小于1天，系统时间跳变
- 【Android】真实时长小于1天，进程冻结影响

部分用户(Android 1.64% , iOS 30.77%)的超时时长是由于系统时间跳变造成的，而真实时长并没有超过 24h 。

Android 端清除无尾流时机较多，部分 duration > 24h 在 SDK 内部就已经被过滤，导致 Android 归因于系统时间跳变的用例数量偏低。
在手百 v13.52 随版上车了减少清除无尾流时机需求，与 iOS 端保持一致。开启实验放量1%

1.2 双打duration

1.2.1 duration

计算方式: $duration = endTime - startTime$

其中 `endTime` 和 `startTime` 取自标准时钟(或挂钟)的时间戳。标准时钟的时间受用户设置和网络设置的影响, 会不可预测地向前或向后跳动。

双端 Api

- Android: `System.currentTimeMillis()`
- iOS: `floor([[NSDate date] TimeIntervalSince1970] * 1000)`

1.2.2 cduration

计算方式: $cduration = clockEndTime - clockStartTime$

- 其中 `clockEndTime` 和 `clockStartTime` 取自系统启动以来的时间。该时钟保证是单调的

追加条件:

- SDK 判断 `duration` 和 `cDuration` 相差超过1秒钟, 才会在该条日志中追加 `cduration`

双端Api

- Android: `SystemClock.elapsedRealtime()`
- iOS: `mach_continuous_time()` or `mach_absolute_time()`

`cduration` 【手百版本v13.39】

UBC SDK随版手百v13.39添加日志进行验证: [评审: 【Sprint13.39.0】【BAIDUSEARCH-BUG-260571】](#) 时长打点duration异常问题添加日志。

目的:

- 统计系统时间突变 (包括未联网时间不准、用户修改系统时间) 造成的duration超长占比
- 排除系统时间突变对时长统计和监控的影响

【手百v13.39+】若日志同时包含 `duration` 和 `cduration`, 那么表明这条日志受系统时间跳变影响 (SDK 1s的阈值)

二、修正方案

2.1 方案概述

【整体思路】

使用 `cduration` 校准 `duration`

【建议方案】

从修复范围上分三个阶段推进：

「阶段1」：对于18点位，针对异常时长日志(`duration` \geq 24h)，用 `cduration` 修正；

「阶段2」：对于18点位，使用 `cduration` 修正；

「阶段3」：对于其他点位，使用 `cduration` 修正。

先启动「阶段1」，逐步放量并观察 24h 数据情况，再决定推进「阶段2」、「阶段3」。

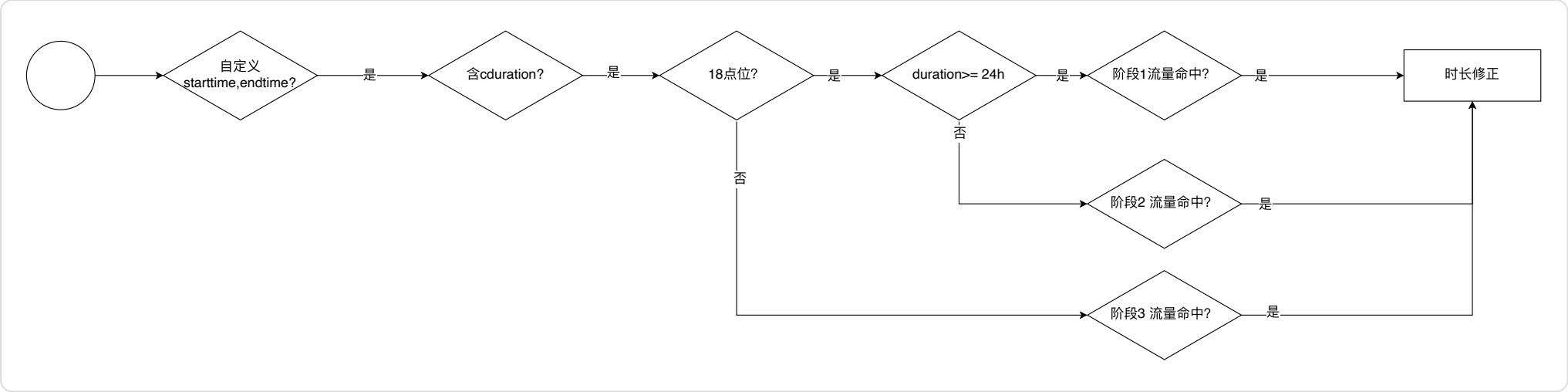
修正操作包括：

使用 `cduration` 替换 `duration` (swap)；以 `endTime` 为基准值重新计算 `startTime`。特别地，UBC支持自定义 `startTime` 和 `endTime`，这种场景下的日志不做修正处理。

修正端：

采用 `Server` 端修正——在上线灵活性和数据收敛上有明显优势，方便策略调整。

【核心流程】



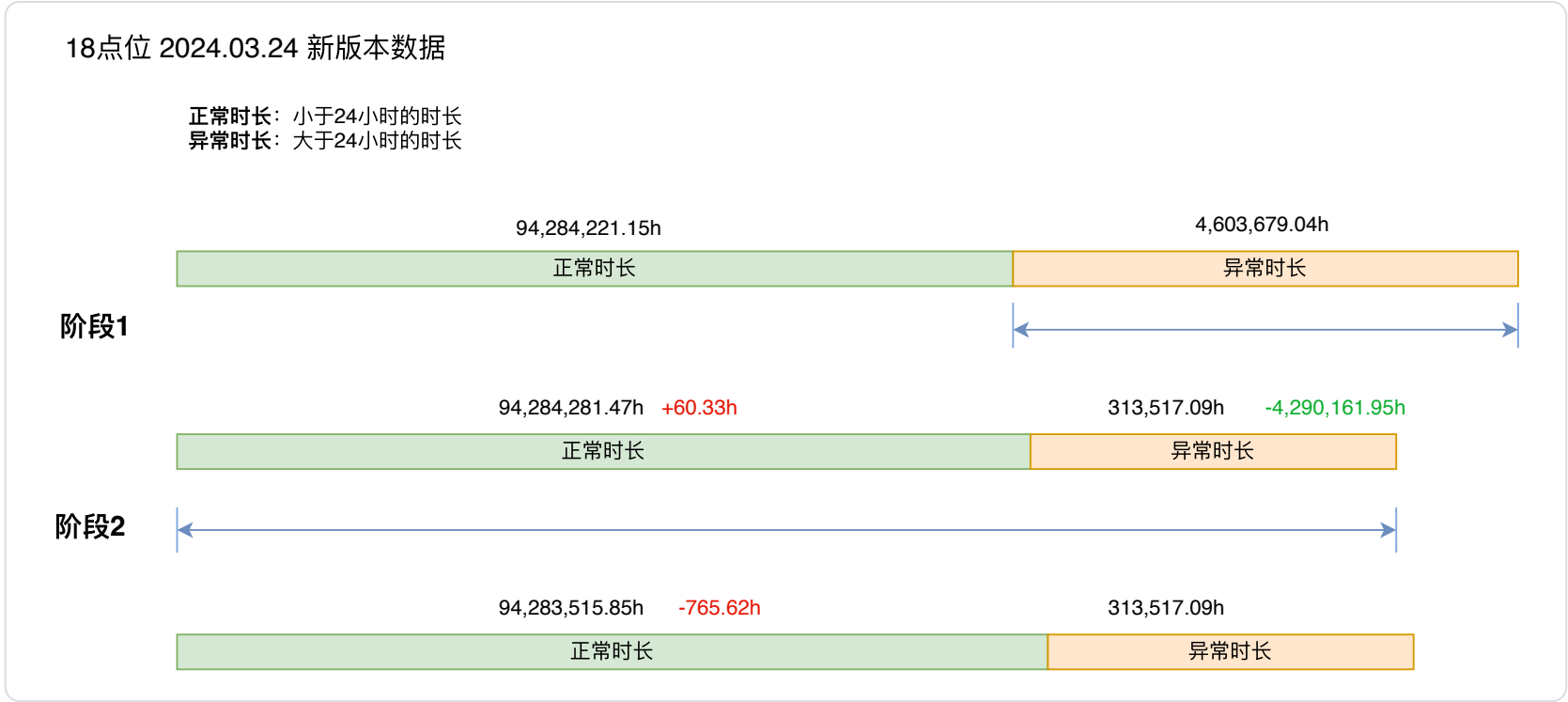
☰ 超长时长修复技术方案- Server

【放量计划】

在「阶段1」、「阶段2」过程中，计划每个阶段在两周内完成放量：

1	工作日	Day-0	Day-1	Day-3	Day-5	Day-7	Day-8
2	放量	5‰	1%	5%	10%	50%	100%
3	收敛天数	1	2	2	2	1	-

2.2 阶段数据影响面评估



「阶段1」: 针对异常时长(`duration` \geq 24h), 用 `cduration` 进行修正 (18 点位)

影响面评估 (2024.03.24)

- 异常时长从 **4,603,679.04h**下降到 **313,517.09h** (**-4,290,161.95h**); 异常时长修复比例 **93.19%**
- 异常时长占比从~~6.70%~~ **4.66%**下降到~~2.66%~~**0.33%**; 全版本异常时长占比从~~27.58%~~下降到**25.31%**
- 正常时长**增加 60.33h**

「阶段2」: 使用无条件的 `cduraiton` 替换 `duration` (18 点位)

影响面评估 (2024.03.24)

- 在「阶段1」基础上, 正常时长**减少 765.62h** (其中 **iOS** 正常时长**减少 955.66h**)

「阶段3」: 修正全部时长点位

影响面评估 (2024.03.24) 包含**18**点位

- 异常时长从~~21,943,669.90h~~下降到 **7,474,212.48h** (**-14,469,457.43h**) 异常时长修复比例**65.93%**

- 异常时长占比从**7.75%**下降到**2.78%**；全点位、全版本异常时长占比从**19.10%**下降到**14.87%**
- 正常时长减少**2,516.50h**(占正常时长比例**0.001%**)

见3.2.5 新版本双端修复数据评估

2.3 修正操作

2.3.1 修正端

使用 `cDuration` 校准 `duration` 的操作可以在 `SDK` 或 `Server` 执行，保证业务方无感知。

1		SDK	Server
2	上线灵活性	SDK发布新版本	不依赖于组件、产品版本
3	版本收敛	收敛缓慢、新老版本数据表义不一致的问题	收敛快速，方便策略调整
4	修改风险	-	清源或实时流需要做修改
5	操作步骤	<p>前置条件：含<code>content.realstarttime</code>和 <code>content.reallendtime</code>字段的日志不处理；</p> <p>Step1: 使用 <code>cduration</code> 校准 <code>duration</code></p> <p>【方式】将 <code>duration</code> 和 <code>cduration</code> 的值互换</p> <p>Step2: 修改 <code>startTime</code></p> <p>【方式】 $startTime = endTime - duration$</p>	

2.3.2 时间戳修改评估

【收益】

- 保持当前 `duration = endTime - startTime` 的数据一致性
- 若业务方使用 `endTime - startTime` 计算时长，结果会是校准后的时长

- endTime更贴近于上报时间

【影响&风险】

- 无法确认 startTime 和 endTime 哪个是正确的基准值
- 使用 endTime - startTime 计算时长的业务方，可切换为使用 duration
- 业务方可能会使用 startTime endTime 计算时长或者用于其他数据指标

【使用 endTime 作为基准值】

startTime 和 endTime 都有可能会不准，但是目前没办法做区别。

- 若只有一个时间不准
时间戳受用户设置和网络设置影响，从直觉性的使用习惯来看，如果用户要修改系统时间，需要进入设置页，这样会导致应用处于后台而结束流式打点。所以以 endTime 为基准。另外 endTime 更可能会贴近于上报时间
- 若两个时间都不准

三、附录

3.1 参考文档

 [Android时长超长问题排查](#)

[Android系统时钟](#)

[iOS timeintervalsince1970](#)

[iOS mach_continuous_time](#)

[UBC SDK - iOS BBAUBCUtills](#)

 [cduration修正duration数据分析](#)

3.2 双端数据

3.2.1 系统时间跳变影响

2024.02.24~2024.02.26三日数据：

1		date	pv	pv: 含cduration	pv占比	uv	uv: 含duration	uv占比
2	Android(v13.51+)	2024.02.24	880,372,917	7,465,129	0.848%	118,690,018	4,776,471	4.024%
3		2024.02.25	866,932,532	7,409,516	0.855%	117,964,500	4,796,861	4.066%
4		2024.02.26	849,867,982	6,871,746	0.809%	117,238,075	4,633,881	3.953%
5		avg	0.837%			4.014%		
6	iOS(v13.51+)	2024.02.24	254,562,513	106,560	0.042%	36,851,463	96,461	0.262%
7		2024.02.25	250,923,004	109,706	0.044%	36,817,784	99,041	0.269%
8		2024.02.26	253,641,904	95,590	0.038%	37,481,517	87,611	0.234%
9		avg	0.041%			0.255%		

3.2.2 超长时长中正常时长pv占比

2024.02.24~2024.02.26三日数据：

1		date	pv : duration > 24h	pv : duration > 24h & cduration < 24h	pv占比
2	Android(v13.51+)	2024.02.24	467	12	2.570%
3		2024.02.25	468	19	4.060%
4		2024.02.26	575	10	1.739%

5		avg	2.790%		
6	iOS(v13.51+)	2024.02.24	251	83	33.068%
7		2024.02.25	260	94	36.154%
8		2024.02.26	259	68	26.255%
9		avg	31.825%		

3.2.3 超时时长

2024.02.24~2024.02.26三日数据:

1		date	总时长	总真实时长	总时长 : duration > 24h	总真实时长: duration > 24h & cduration < 24h
2	Android(v13.51+)	2024.02.24	257,876,972,356	257,119,689,469	901,503,557	5,332
3		2024.02.25	260,147,915,266	255,876,362,992	6,124,255,937	3,668
4		2024.02.26	233,238,227,808	231,135,697,825	2,261,777,674	1,447
5	iOS(v13.51+)	2024.02.24	58,007,321,211	57,952,926,970	92,240,011	66,795
6		2024.02.25	59,646,765,182	57,865,894,240	1,816,026,363	39,699
7		2024.02.26	51,958,949,860	51,896,082,037	99,693,974	30,001

3.2.4 阶段数据评估

2024.3.10 号 v13.40+ 双端数据评估 (时长单位是小时)

1				阶段1	阶段2
	Android	正常时长	78,424,220.39	78,424,220.76	78,428,571.59

2				+0.38	+4,350.83
		异常时长	3,802,812.85	660,022.55	660,052.86
3				-3,142,790.30	+30.30
		异常时长占比	4.62%	0.83%	0.83%
4	iOS	正常时长	18,289,074.61	18,289,121.20	18,288,212.89
5				+46.59	-908.31
		异常时长	782,811.06	167,139.94	167,139.94
6				-613561.68	-
		异常时长占比	4.10%	0.91%	0.93%
7	Android & iOS	正常时长	96,713,294.99	96,713,341.96	96,716,784.48
8				+46.97	+3,442.52
		异常时长	4,585,623.92	827,162.49	827,192.79
9				-3,756,352.04	+30.30
		异常时长占比	4.53%	0.85%	0.85%
10		异常时长修复比例	-	81.96%	-

2024.3.10 号 v13.52+ Android 实验组1%量级数据评估 (时长单位是小时)

1			时长	阶段1	阶段2
2	Android 1% EXP	正常时长	571,898.66	571,898.75	571,929.24
				+0.09	+30.48
3		异常时长	128,834.43	450.77	450.77
				-128,383.66	-0
4		异常时长占比	18.39%	0.08%	0.08%

2024.3.10 号全版本数据评估 (时长单位是小时)


1			时长	阶段1	阶段2
2	Android&iOS	正常时长	106,180,147.98	106,180,195.03 +47.05	106,183,651.23 +3,456.20
3		异常时长	38,120,155.81	34,205,659.04 -3,914,496.77	34,205,689.34 +30.30
4		异常时长占比	26.42%	24.37%	24.36%

对于全版本异常时长占比 26.42% -> 24.36%

3.2.5 新版本双端修复数据评估

2024.3.24 号 新版双端数据评估 (时长单位是小时)

cduration修正duration数据分析



20240323-0325 分点位数据评估.xlsx

683.0KB