

Figure - Bug #118113

【IOVDEV-29496】【内部】【VC1】【0095】【台架】【EE环境】【副屏】【第三方应用】打开多个4个以上第三方应用，系统卡顿

2023-05-19 18:17 - CD TEST-方永红

Status:	CLOSED	Start date:	2023-05-19
Priority:	Normal	Due date:	2023-06-05
Assignee:	CD System 雷祥	% Done:	10%
Category:	卡顿	Estimated time:	0.00 hour
Target version:			
Need_Info:	--	Found Version:	0095
Resolution:	--	Degrated:	--
Severity:	Major	Verified Version:	
Reproducibility:	Frequently	Fixed Version:	
Test Type:	ST	Root cause:	
Description			
【问题出现时间】：20230519 16:50 第40次开机 【前提条件】：电源ON 【操作步骤】：1、打开三个以上应用到后台 【实际结果】系统卡顿 【期望结果】系统流畅 【备注】：【应用版本】：【硬件版本号】：不涉及【副屏版本号】：01.00.0095.C103 【中控版本号】：03.02.0080.C101.03_userdebug_2305121121 【中控VIN】HRYTTESTVINMG403 【第三方应用版本号】企业微信4.1.3 喜马拉雅9.1.35.3 原神3.6.0_13833164-13951843 录屏&log:https://ofs.human-horizons.com/#!/download/index/7eL%2F66MjvvQ%3D			

History

- #1 - 2023-05-19 19:56 - CDTS_TEST 王成
- ftp://cdiot@192.168.87.46/Pre_figure/Test_Log/Bug_118113/%E6%80%A7%E8%83%BD%E5%8D%A1%E9%A1%BF.7z
- #2 - 2023-05-19 20:05 - CD FW 曹覃刚
- Category changed from CD-FW to SYSTEM
- Assignee changed from CD FW 曹覃刚 to CD SYSTEM-夏旭
- Hi 夏旭
- 系统性能卡顿问题，请系统组先帮忙分析，谢谢
- #3 - 2023-05-22 11:05 - CD SYSTEM-夏旭
- File mem.jpg added
- 1.从视频的时间进行确认（17：32）左右，发现是是上层的gc现象，
- 05-19 17:33:54.694 11835 11844 | .smile.gifmake: Background concurrent copying GC freed 386023(23MB) AllocSpace objects, 43(1656KB) LOS objects, 34% free, 44MB/68MB, paused 166us,145us total 192.045ms

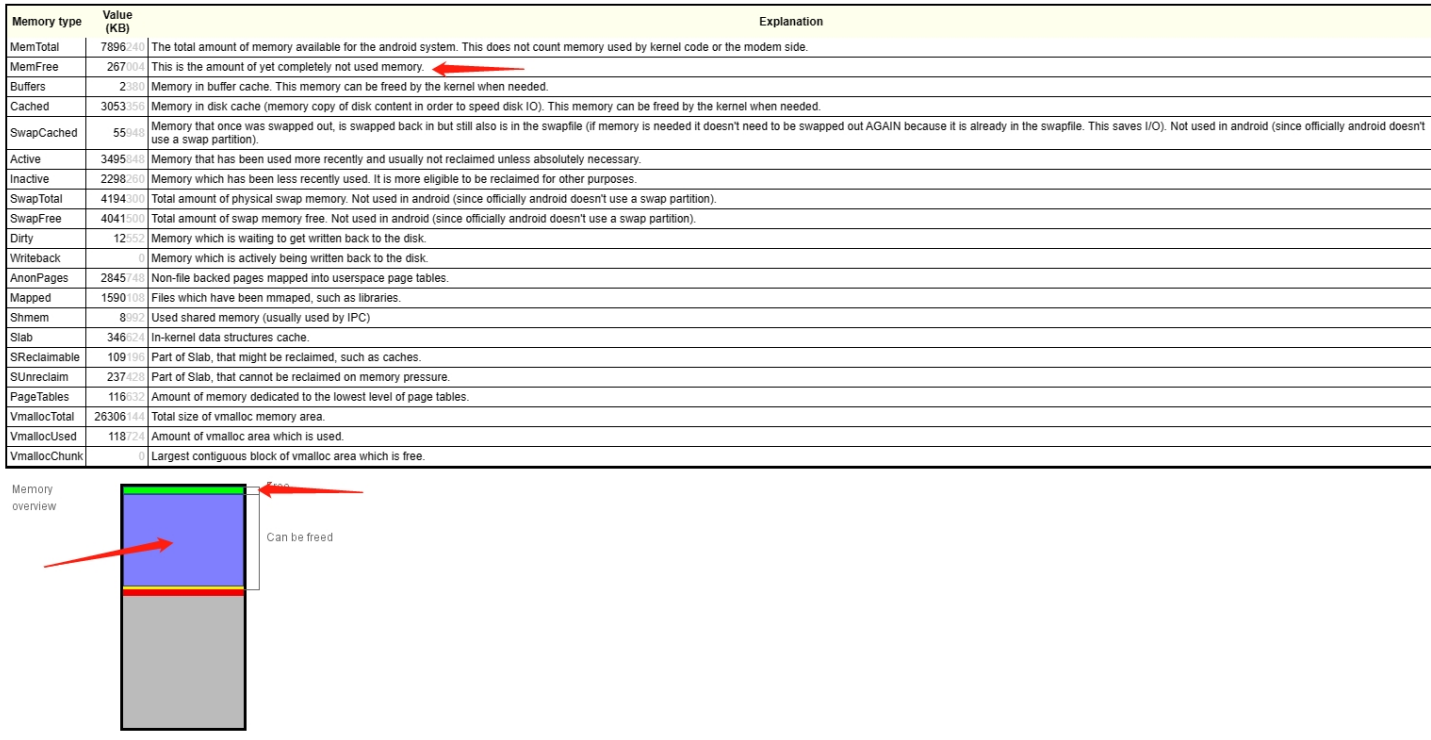
2.内存存在压力，但是情况处理正常情况下

05-19 17:33:54.713 lowmemorykiller: Ignoring pressure since per-zone watermarks ok

3.测试场景说明是：20230519 16:50 第40次开机，但是从日志的文件名看，此log是开机42次的log，没有抓到现场

000042_manual_00000_230519_173234

4. 从bugReport看(bugreport-figure_0519173805.zip)，此时能用的内容内存应该很少了



5.从提供的视频看他们打开了

飞书，QQ音乐HD，QQ音乐,酷狗音乐，百度手机，bilibili,网易新闻,优酷视频，咪咕视频，王者荣耀，爱奇艺,芒果TV

且这此APK全是在后台运行，内存占用量具大，从第4点看内存已经很少，有上卡顿现象是正常的

#4 - 2023-05-24 23:26 - CDTs-TEST 周婷

- Category changed from SYSTEM to 卡顿

#5 - 2023-05-25 17:09 - CD System 雷祥

- Status changed from New to ASSIGNED

- Assignee changed from CD SYSTEM-夏旭 to CD System 雷祥

- % Done changed from 0 to 10

[当前状态]

1. 根据bugreport可知，MemFree剩余在267M,设备开机就只剩余1.7G左右，过滤日志，发现打开的应用中有微信，豌豆荚，QQ，这些应用每一个都大概占用200M左右，所以很快剩余内存就被使用到剩余很少。

2. 同时根据视频的发生时间，搜索关键字，可以看到当时发生了80多次内存回收机制，因为GC操作会暂停应用程序的执行，直到垃圾回收完成。频繁执行GC操作，系统缺失会卡顿，属于正常现象。

```
logcat/000042_manual_00009_230519_173238.log:1885:05-19 17:32:38.035 7869 7877 I .ServiceProces: Background young concurrent copying GC freed 12402(965KB) AllocSpace objects, 0(0B) LOS objects, 50% free, 3798KB/7672KB, paused 157us,98us total 690.686ms
```

...

80次左右GC操作

...

logcat/000042_manual_00039_230519_173435.log.ing:175:05-19 17:34:15.981 19124 19137 I oid:pushservic: Background young concurrent copying GC freed 736346(26MB) AllocSpace objects, 23(1748KB) LOS objects, 40% free, 21MB/35MB, paused 11.740ms,33.442ms total 193.027ms

3. 使用stressapptest工具对同平台（qcs8250）的设备进程测试，当时memory Free的值剩余500M以下时，点击几次app后，系统反应同样卡顿，再次查看memory Free剩余值变为200M左右。

4. figure 开机之后的内存占用

Total RAM: 7,844,416K (status normal)
Free RAM: 4,739,177K (473,861K cached pss + 2,523,516K cached kernel + 1,741,800K free)
ION: 106,148K (16,940K mapped + 89,208K unmapped + 0K pools)
Used RAM: 2,723,216K (1,646,124K used pss + 1,077,092K kernel)
Lost RAM: 383,287K
ZRAM: 1,604K physical used for 3,204K in swap (4,194,300K total swap)
Tuning: 512 (large 512), oom 322,560K, restore limit 107,520K (high-end-gfx)

同平台（QCS8550）开机之后内存占用

Total RAM: 7,885,484K (status normal)
Free RAM: 4,914,959K (614,199K cached pss + 842,700K cached kernel + 3,458,060K free)
Used RAM: 2,199,409K (1,746,613K used pss + 452,796K kernel)
Lost RAM: 771,104K
ZRAM: 12K physical used for 0K in swap (4,194,300K total swap)
Tuning: 512 (large 512), oom 322,560K, restore limit 107,520K (high-end-gfx)

关于cached kernel，2者的使用差异巨大，需要调查是否之前的开发过程中有修改此值来提升内核的运行性能。

5. 在进行bug中的操作时，通过可以看到memory free只有300M，但是cached kernel还有1.8G，有释放一些出来缓解内存压力，但是感觉不够，这里也需要调查是否异常。

Total RAM: 7,896,240K (status normal)
Free RAM: 4,155,237K (1,924,369K cached pss + 1,924,644K cached kernel + 306,224K free)
ION: 299,116K (9,912K mapped + 289,204K unmapped + 0K pools)
Used RAM: 3,242,218K (2,471,106K used pss + 771,112K kernel)
Lost RAM: 530,956K
ZRAM: 21,664K physical used for 87,996K in swap (4,194,300K total swap)
Tuning: 512 (large 512), oom 322,560K, restore limit 107,520K (high-end-gfx)
----- 10.545s was the duration of dumpsys meminfo, ending at: 2023-05-19 17:39:10

[下一步动作]

内部需要讨论沟通：

1. 调查为什么figure项目开机后cached kernel占用的memory为2.5G左右，是否为正常值。
2. 当memory有压力时，cached kernel没有释放足够的memory来缓解压力，是否正常。
3. 目前问题的规避方案。

#6 - 2023-05-26 17:05 - CD System 雷祥

[当前状态]

1. 之前的AI：

a) 当memory有压力时，cached kernel没有释放足够的memory来缓解压力，是否正常。

---> 目前内部多次测试，在free memory存在压力时，cached kernel有释放memory 给到free使用，之前抓取的日志大概率是没有抓取到现场，cached kernel是后面又升上去的（内部测试结果如此）

b) 调查为什么figure项目开机后cached kernel占用的memory为2.5G左右，是否为正常值。

这个依赖高通Memory management - AP case（06644550）

2. 内部测试，设备跑完monkey后，在快速点击多个app，的确存在系统不流畅的问题，但是客户的测试现场没有跑monkey（日志也没有monkey相关记录）

3. 如果不跑menkey，快速点击多个app,系统有轻微的卡顿，感觉比较正常，且有查看真正能用的memory（available memory）确实很少，但是还能比较及时响应。

4. 同步有提一个高通case给到UX performace（06645981），看上层是否有什么发现。

5. 客户现场memory RSS 占用情况（前十），看起来APP占用memory较多，加上当时频繁发生GC操作，的确会导致卡顿。

Total RSS by process:

561,960K: com.cmcc.cmvideo:remote_web (pid 5395)
520,100K: system (pid 1511)
481,008K: com.cmcc.cmvideo (pid 4548)
437,208K: com.smile.gifmaker (pid 5049)
344,480K: com.tencent.mobileqq (pid 3511)
332,000K: com.tencent.android.qqdownloader:daemon (pid 3239)
328,604K: com.qiyi.video.pad (pid 6236)
293,312K: com.android.systemui (pid 1855)
285,740K: com.cmcc.cmvideo:pushservice (pid 7949)
285,200K: com.android.settings (pid 2347 / activities)

[下一步动作]

1. 跟踪高通的2个case：UX performace（06645981）和Memory management - AP（06644550）

2. 根据dumpsys meminfo的数据，是不是第三方app没有适配，导致占用过多memory，这个可以需要依赖上层从app的角度分析。

3. 内部沟通接下来的规避方案

#7 - 2023-05-29 19:58 - CD System 雷祥

[当前状态]

1. 和高通这边沟通，目前初步认为memory虽然比较少，但是影响不太大。

2. 目前使用同平台的QCS8250跑完monkey后也概率出现卡顿现象，现象比figure会好一点，使用memory压力工具进行测试，同样也会出现卡顿。

[下一步动作]

1. 目前认为经过monkey后，系统已经出现错误，再加上memory较少，频繁发生GC等事件同步发生，出现卡顿现象为正常行为。
2. 但是客户的现场没有进行monkey测试，我们内部没有复现过这种场景（开机直接点击多个app出现明显卡顿的问题），所以需要客户这边帮忙复现，同时复现问题时的抓取systrace。

#8 - 2023-05-31 11:50 - CDTS_TEST 王成

http://cdiot@192.168.87.46/Pre_figure/Test_Log/Bug_118113/trace-kona-SKQ1.220201.001-2023-05-31-09-16-26.perfetto-trace

#9 - 2023-05-31 19:15 - CDTS-TEST 周婷

- Due date set to 2023-06-05
- Severity changed from Normal to Major

#10 - 2023-05-31 20:35 - CD System 雷祥

- Severity changed from Major to Normal

[当前状态]

1. 查看客户提供的复现视频，发现打开喜马拉雅app时，系统反应有些滞后，查看客户提供的system数据，发现从点击app到绘制app第一帧，大概耗费了500ms。
而对比正常使用时抓取的systrace，大概在200ms左右。
2. 同时有查看对应的CPU，发现使用率很高，几乎快占满了。
3. 同时由于free memory较少，内核内存相关的进程kswapd占用CPU的大核一直在工作。

[下一步动作]

查看卡顿现场，为什么cpu使用率如此之高，以至于app点击响应滞后。

#11 - 2023-05-31 20:35 - CD System 雷祥

- Severity changed from Normal to Major

#12 - 2023-06-01 17:52 - CD System 雷祥

[当前状态]

1. 通过（tlog_jov0201017500006162025411a221229000000771_000176_0531142229\logcat）日志查看，在“05-31 14:26:10.455~14:27:06.211”1分钟内发生了34次GC操作，因为GC操作会暂停应用程序的执行，直到垃圾回收完成。频繁执行GC操作，系统缺失会卡顿，属于正常行为。
2. 以喜马拉雅为例，查看systrace中发生系统卡顿的点，发现cpu使用率几乎达到了100%，除开系统进程，喜马拉雅app自带的几个进程，以及其他app少量的进程，发现还有2个进程占比较多：
 - kswapd进程,内存不足时进程内存交换，占用cpu资源，正常现象
 - cn.moduoketech.recorder 卡顿期间，后台一直在运行，且占用的cpu比前台的喜马拉雅还多，

快速同步打开多个app，CPU使用率几乎占满，导致app进程不能及时分配到cpu资源，同样系统会卡顿，属于正常行为。

3. 查看后续的meminfo信息，可以发现swap也差不多使用完了。

Mem: 7896244K total, 7685616K used, 210628K free, 896K buffers
Swap: 4194300K total, 3838304K used, 355996K free, 736408K cached

以上分析的结论，卡顿主要因为如下原因：

- CPU资源不够
- 系统内存不足（又启动kswapd继续占用CPU资源，同时启动android回收机制）

[下一步动作]

规避方案：

1. 在快速打开多个app压力测试时，不启动cn.moduoketech.recorder进程（FM有声调频收音机），不清楚当时客户在使用这个app做什么操作，可以大大减轻CPU的负担。
2. 在1的基础上，看是否还会复现系统点击app卡顿的问题，可以实际情况看是否需要优化memory。

#13 - 2023-06-02 17:20 - CD System 雷祥

[当前状态]

1. 快速打开过个app时，以喜马拉雅为例，打开时刻基本都会占用200%~300%CPU资源，再加上其他app处于后台，但任然处于active状态，继续占用cpu资源，导致如果打开多个app时，memory资源也紧张，启动kswapd进一步占用CPU资源，导致CPU资源不够，系统运行卡顿。

[下一步动作]

1. 准备限制后台app的数量，来减少后台app的占用cpu资源的问题，但是有个问题，后台app的数量达到限制数量时，系统也不是立马释放资源，所以还是可能导致cpu资源不够，需要验证测试后台保留app的数量为多少时，用户体验最佳。
正在出VB进行验证。

#14 - 2023-06-03 18:16 - CD System 雷祥

1. 在开发者选项中，可以找到设置后台的app限制数量的选项，设置对应的值后（比如说设置为4），通过覃刚提供的命令：

```
adb shell dumpsys activity settings | grep -Ei "max"
mCustomizedMaxCachedProcesses=32
CUR_MAX_CACHED_PROCESSES=4
CUR_MAX_EMPTY_PROCESSES=2
```

可以看到，对应的变量有生效。

2. 开始准备以打开喜马拉雅为例，对比默认和限制后台进程数量后后，通过systrace查看cpu的使用率的变化。
 - 不保留后台进程
 - 保留1个后台进程
 - 保留2个后台进程
 - 保留3个后台进程
 - 保留4个后台进程

但是发现至少“今天头条”和“抖音”，“腾讯视频”，“快手”，“百度网络相关进程”和“喜马拉雅的网络推送进程”后台使用都没有受到限制，即使不对这几个app进行操作，在测试手顺一样的情况，由于链接网络，这几个app时不时的就会占用cpu，导致cpu使用率没法量化。

3. 后面准备快速打开app，测试1分钟，统计cpu的使用率，多测试几次求平均值，发现2者相差不大，同时可能存在其他因素影响，所以可能并没有区别。
 - 不保留后台进程 40%~65%
 - 保留3个后台进程 cpu占用率35%-65%

和胡兵进行沟通后，修改CUR_MAX_CACHED_PROCESSES这种方式，主要适用于如下场景：

应用切换到后台时，其后台进程属于非缓存进程，如果此进程没有更多的业务或者资源的需求，才有可能转换成缓存进程，而DEFAULT_MAX_CACHED_PROCESSES是控制缓存进程数量，缓存进程的优先级较低，当系统内存不足时，会被系统回收以释放内存资源。根据设置的最大缓存进程数，系统会尽量保留指定数量的缓存进程，而其他进程则可能会被回收。

某些app切换到后台时，还会占用系统资源（网络请求等），就不会转换为缓存进程，导致即使设置了DEFAULT_MAX_CACHED_PROCESSES，也没法被释放。（比如“今天头条”和“抖音”，“腾讯视频”，“快手”，“百度网络相关进程”和“喜马拉雅的网络推送进程”）

4. 继续使用相同的测试手法，通过dumpsys meminfo查看上层app的cache大小，

- 不保留后台进程 cached pss：800M左右
- 保留1个后台进程 cached pss：300-400M
- 保留3个后台进程 cached pss：200-300M

从上面的测试来看，修改限制后台数量后，进行快速打开app时，CPU的占用率很难量化，没法对比cpu的使用率。
但是设置后台数量后，上层cached占用的memory确实会减少，这样总的free memory会增加，在这个测试环境，内存不足的条件下，可以减少kswap进程的工作频率，同样会降低cpu的占用率。

所以，从目前的测试结果来看，设置config_customizedMaxCachedProcesses是有一定优化作用。

#15 - 2023-06-05 10:32 - 物联网项目组-RD3_CDTs 周飞

客户攻关会决议只保留3个应用，拉专项评估工作量

#16 - 2023-08-01 15:35 - CDTs_TEST 王成
- Status changed from ASSIGNED to RESOLVED

客户已关闭BUG

#17 - 2023-08-01 15:36 - CDTs_TEST 王成
- Status changed from RESOLVED to VERIFIED

#18 - 2023-08-01 15:38 - CDTs_TEST 王成
- Status changed from VERIFIED to CLOSED

Files			
mem.jpg	276 KB	2023-05-22	CD SYSTEM-夏旭