

Figure - Bug #118827

测试 Test-ST # 113421 (New): V4.0功能与专项测试

测试 Test-ST # 113423 (New): V4.0专项--BSP专项--稳定性-各项功能长时间运行

【AF】【EVT】【稳定性】MTBF长时间跑测，出现一次异常重启

2023-06-17 14:02 - CDTs_Test 吴诗雨

Status:	New	Start date:	2023-06-17
Priority:	Normal	Due date:	
Assignee:	CDTS_Test 吴诗雨	% Done:	0%
Category:	CD-FW	Estimated time:	0.00 hour
Target version:			
Need_Info:	--	Found Version:	FlatBuild_HH_MCE_FSE.M.D.user.01.00.C104(X104)
Resolution:	--	Degrated:	--
Severity:	Normal	Verified Version:	
Reproducibility:	Every time	Fixed Version:	
Test Type:	ST	Root cause:	

Description			
【前提条件】			
1、设备成功启动			
2、确保MTBF脚本运行的前置条件已准备			
【测试步骤】			
1、使用python运行对应MTBF脚本			
2、运行三天			
3、查看设备运行结果			
【预期结果】			
3、设备正常运行			
【实际结果】			
3、出现一次异常重启			

History

- #1 - 2023-06-17 14:03 - CDTs_Test 吴诗雨
- Subject changed from 【AF】【EVT】【稳定性】MTBF长时间跑测，出现异常重启 to 【AF】【EVT】【稳定性】MTBF长时间跑测，出现一次异常重启

日志已私发

#2 - 2023-06-17 14:10 - CDTs_Test 吴诗雨

日志已上传：ftp://cdiot@192.168.87.46/Pre_figure/Test_Log/Bug_118827

- #3 - 2023-06-17 14:23 - CD FW 曹覃刚
- Category changed from CD-FW to SYSTEM
- Assignee changed from CD FW 曹覃刚 to CD SYSTEM-胡兵

■ 我的分析

06-17 08:48:27.555 1610 1720 W Watchdog: *** WATCHDOG KILLING SYSTEM PROCESS: Blocked in monitor com.android.server.am.ActivityManagerService on foreground thread (android.fg), Blocked in handler on main thread (main), Blocked in handler on display thread (android.display), Blocked in handler on animation thread (android.anim), Blocked in handler on ActivityManager (ActivityManager)

system_server挂了，导致的重启

"main" prio=5 tid=1 Blocked | group="main" sCount=1 ucsCount=0 flags=1 obj=0x72bd99d8 self=0xb4000073f204cbe0 | sysTid=1610 nice=-2 cgrp=default sched=0/0 handle=0x75bfcc24f8 | state=S schedstat=(277432259766 200961731556 1091359) utm=18605 stm=9137 core=0 HZ=100 | stack=0x7ff9e3e000-0x7ff9e40000 stackSize=8188KB | held mutexes=
at com.android.server.am.BroadcastDispatcher\$1.broadcastAlarmComplete(BroadcastDispatcher.java:186)
- waiting to lock <0x090fe42d> (a com.android.server.am.ActivityManagerService) held by thread 23
at com.android.server.alarm.AlarmManagerService.notifyBroadcastAlarmCompleteLocked(AlarmManagerService.java:1511)
at com.android.server.alarm.AlarmManagerService.access\$3300(AlarmManagerService.java:171)
at com.android.server.alarm.AlarmManagerService\$DeliveryTracker.removeLocked(AlarmManagerService.java:4750)
at com.android.server.alarm.AlarmManagerService\$DeliveryTracker.onSendFinished(AlarmManagerService.java:4865)
- locked <0x0bf60b94> (a java.lang.Object)
at android.app.PendingIntent\$FinishedDispatcher.run(PendingIntent.java:325)
at android.os.Handler.handleCallback(Handler.java:938)
at android.os.Handler.dispatchMessage(Handler.java:99)
at android.os.Looper.loopOnce(Looper.java:201)
at android.os.Looper.loop(Looper.java:288)
at com.android.server.SystemServer.run(SystemServer.java:909)
at com.android.server.SystemServer.main(SystemServer.java:616)
at java.lang.reflect.Method.invoke(Native method)
at com.android.internal.os.RuntimeInit\$MethodAndArgsCaller.run(RuntimeInit.java:548)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:987)

"ActivityManager" prio=5 tid=23 Blocked | group="main" sCount=1 ucsCount=0 flags=1 obj=0x154c0cb8 self=0xb4000073f2089950 | sysTid=1732 nice=-2 cgrp=default sched=0/0 handle=0x71ea9b4cb0 | state=S schedstat=(126222961418 61316928766 313526) utm=5758 stm=6863 core=3 HZ=100 | stack=0x71ea8b1000-0x71ea8b3000 stackSize=1039KB | held mutexes=
at com.android.server.wm.ActivityTaskManagerService\$LocalService.compatibilityInfoForPackage(ActivityTaskManagerService.java:5622)
- waiting to lock <0x071f13e7> (a com.android.server.wm.WindowManagerGlobalLock) held by thread 44
at com.android.server.am.ActivityManagerService.compatibilityInfoForPackage(ActivityManagerService.java:2807)
at com.android.server.am.BroadcastQueue.processCurBroadcastLocked(BroadcastQueue.java:326)
at com.android.server.am.BroadcastQueue.processNextBroadcastLocked(BroadcastQueue.java:1683)
at com.android.server.am.BroadcastQueue.processNextBroadcast(BroadcastQueue.java:987)
- locked <0x090fe42d> (a com.android.server.am.ActivityManagerService)
at com.android.server.am.BroadcastQueue.access\$000(BroadcastQueue.java:79)
at com.android.server.am.BroadcastQueue\$BroadcastHandler.handleMessage(BroadcastQueue.java:197)
at android.os.Handler.dispatchMessage(Handler.java:106)
at android.os.Looper.loopOnce(Looper.java:201)
at android.os.Looper.loop(Looper.java:288)
at android.os.HandlerThread.run(HandlerThread.java:67)
at com.android.server.ServiceThread.run(ServiceThread.java:44)

"Binder:1610_10" prio=5 tid=44 Native | group="main" sCount=1 ucsCount=0 flags=1 obj=0x154cd0c8 self=0xb4000073f2198c10 | sysTid=8335 nice=-10 cgrp=default sched=0/0 handle=0x71ed6dbcb0 | state=S schedstat=(144453678611 44679018577 358209) utm=10676 stm=3768 core=7 HZ=100 | stack=0x71ed5e4000-0x71ed5e6000 stackSize=991KB | held mutexes=
native: #00 pc 000000000004def0 /apex/com.android.runtime/lib64/bionic/libc.so (syscall+32)
native: #01 pc 000000000028dc74 /apex/com.android.art/lib64/libart.so (art::ConditionVariable::WaitHoldingLocks(art::Thread*)+152)
native: #02 pc 000000000074cd0c /apex/com.android.art/lib64/libart.so (art::GoToRunnable(art::Thread*)+412)
native: #03 pc 000000000074cb38 /apex/com.android.art/lib64/libart.so (art::JniMethodEnd(unsigned int, art::Thread*)+28)
at android.util.MemoryIntArray.nativeSize(Native method)
at android.util.MemoryIntArray.size(MemoryIntArray.java:136)

at android.util.MemoryIntArray.enforceValidIndex(MemoryIntArray.java:214)
 at android.util.MemoryIntArray.get(MemoryIntArray.java:109)
 at android.provider.Settings\$GenerationTracker.readCurrentGeneration(Settings.java:2714)
 at android.provider.Settings\$GenerationTracker.isGenerationChanged(Settings.java:2698)
 at android.provider.Settings\$NameValueCache.getStringsForPrefix(Settings.java:3101)
 - locked <0x082ee7d6> (a android.provider.Settings\$NameValueCache)
 at android.provider.Settings\$Config.getStrings(Settings.java:16341)
 at android.provider.DeviceConfig.getProperties(DeviceConfig.java:661)
 at android.provider.DeviceConfig.getProperty(DeviceConfig.java:633)
 at android.provider.DeviceConfig.getBoolean(DeviceConfig.java:696)
 at android.content.pm.ConstrainDisplayApisConfig.neverConstrainDisplayApis(ConstrainDisplayApisConfig.java:65)
 at android.content.pm.ActivityInfo.neverSandboxDisplayApis(ActivityInfo.java:1356)
 at com.android.server.wm.ActivityRecord.providesMaxBounds(ActivityRecord.java:7879)
 at com.android.server.wm.ActivityRecord.resolveOverrideConfiguration(ActivityRecord.java:7422)
 at com.android.server.wm.ConfigurationContainer.onConfigurationChanged(ConfigurationContainer.java:124)
 at com.android.server.wm.WindowContainer.onConfigurationChanged(WindowContainer.java:356)
 at com.android.server.wm.WindowToken.onConfigurationChanged(WindowToken.java:378)
 at com.android.server.wm.ActivityRecord.onConfigurationChanged(ActivityRecord.java:7972)
 at com.android.server.wm.ConfigurationContainer.dispatchConfigurationToChild(ConfigurationContainer.java:154)
 at com.android.server.wm.ConfigurationContainer.onConfigurationChanged(ConfigurationContainer.java:145)
 at com.android.server.wm.WindowContainer.onConfigurationChanged(WindowContainer.java:356)
 at com.android.server.wm.Task.onConfigurationChangedInner(Task.java:2275)
 at com.android.server.wm.Task.onConfigurationChanged(Task.java:2375)
 at com.android.server.wm.Task.onDescendantOrientationChanged(Task.java:3359)
 at com.android.server.wm.WindowContainer.onDescendantOrientationChanged(WindowContainer.java:1176)
 at com.android.server.wm.ActivityRecord.reportDescendantOrientationChangeIfNeeded(ActivityRecord.java:7153)
 at com.android.server.wm.RootWindowContainer.ensureVisibilityAndConfig(RootWindowContainer.java:1819)
 at com.android.server.wm.ActivityTaskSupervisor.realStartActivityLocked(ActivityTaskSupervisor.java:835)
 at com.android.server.wm.ActivityTaskSupervisor.startSpecificActivity(ActivityTaskSupervisor.java:1066)
 at com.android.server.wm.EnsureActivitiesVisibleHelper.makeVisibleAndRestartIfNeeded(EnsureActivitiesVisibleHelper.java:222)
 at com.android.server.wm.EnsureActivitiesVisibleHelper.setActivityVisibilityState(EnsureActivitiesVisibleHelper.java:149)
 at com.android.server.wm.EnsureActivitiesVisibleHelper.lambda\$process\$0\$EnsureActivitiesVisibleHelper(EnsureActivitiesVisibleHelper.java:100)
 at com.android.server.wm.EnsureActivitiesVisibleHelper\$\$ExternalSyntheticLambda0.accept(unavailable:-1)
 at com.android.server.wm.ActivityRecord.forAllActivities(ActivityRecord.java:4095)
 at com.android.server.wm.WindowContainer.forAllActivities(WindowContainer.java:1457)
 at com.android.server.wm.WindowContainer.forAllActivities(WindowContainer.java:1451)
 at com.android.server.wm.EnsureActivitiesVisibleHelper.process(EnsureActivitiesVisibleHelper.java:99)
 at com.android.server.wm.Task.lambda\$ensureActivitiesVisible\$20(Task.java:6073)
 at com.android.server.wm.Task\$\$ExternalSyntheticLambda23.accept(unavailable:-1)
 at com.android.server.wm.Task.forAllLeafTasks(Task.java:3920)
 at com.android.server.wm.Task.ensureActivitiesVisible(Task.java:6073)
 at com.android.server.wm.DisplayContent.lambda\$ensureActivitiesVisible\$44(DisplayContent.java:5768)
 at com.android.server.wm.DisplayContent\$\$ExternalSyntheticLambda17.accept(unavailable:-1)
 at com.android.server.wm.Task.forAllRootTasks(Task.java:3932)
 at com.android.server.wm.WindowContainer.forAllRootTasks(WindowContainer.java:1770)
 at com.android.server.wm.WindowContainer.forAllRootTasks(WindowContainer.java:1770)
 at com.android.server.wm.WindowContainer.forAllRootTasks(WindowContainer.java:1770)
 ... repeated 2 times
 at com.android.server.wm.WindowContainer.forAllRootTasks(WindowContainer.java:1763)
 at com.android.server.wm.DisplayContent.ensureActivitiesVisible(DisplayContent.java:5767)
 at com.android.server.wm.RootWindowContainer.ensureActivitiesVisible(RootWindowContainer.java:1995)
 at com.android.server.wm.RootWindowContainer.ensureVisibilityAndConfig(RootWindowContainer.java:1799)
 at com.android.server.wm.ActivityTaskSupervisor.realStartActivityLocked(ActivityTaskSupervisor.java:835)

```
at com.android.server.wm.ActivityTaskSupervisor.startSpecificActivity(ActivityTaskSupervisor.java:1066)
at com.android.server.wm.Task.resumeTopActivityInnerLocked(Task.java:6670)
at com.android.server.wm.Task.resumeTopActivityUncheckedLocked(Task.java:6212)
at com.android.server.wm.RootWindowContainer.resumeFocusedTasksTopActivities(RootWindowContainer.java:2436)
at com.android.server.wm.RootWindowContainer.resumeFocusedTasksTopActivities(RootWindowContainer.java:2422)
at com.android.server.wm.Task.completePauseLocked(Task.java:5991)
at com.android.server.wm.ActivityRecord.activityPaused(ActivityRecord.java:5602)
at com.android.server.wm.ActivityClientController.activityPaused(ActivityClientController.java:201)
- locked <0x071f13e7> (a com.android.server.wm.WindowManagerGlobalLock)
at android.app.IActivityClientController$Stub.onTransact(IActivityClientController.java:571)
at com.android.server.wm.ActivityClientController.onTransact(ActivityClientController.java:134)
at android.os.Binder.execTransactInternal(Binder.java:1179)
at android.os.Binder.execTransact(Binder.java:1143)
    主线程被block了，nativeSize一直没返回
```

android/util/MemoryIntArray.java
nativeSize方法是用于获取共享内存区域大小
这个方法是调用JNI接口实现的，它会调用底层共享内存库中的相应方法获取内存区域的大小

■ 下一步计划
请SYSTEM组的同学帮忙调查nativeSize一直没返回的原因

#4 - 2023-06-28 18:47 - CDTS-TEST 周婷
- Assignee changed from CD SYSTEM-胡兵 to CD SYSTEM-夏旭

#5 - 2023-07-03 10:03 - CD SYSTEM-夏旭

【当前分析】

两次堆栈不一致 非卡死在native 获取内存上

从堆栈上看 其实是在执行的时候卡死

watchdog发生前一分钟 可以看到wm_restart_activity在疯狂的重新启动这
个com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity

```
行 372676: 06-17 08:47:11.064 1610 12866 I wm_pause_activity: [0,241614778,com.android.launcher3/ui/overrides/QuickstepLauncher,userLeaving=true,pauseBackTasks]
行 372709: 06-17 08:47:11.173 1610 8335 I wm_restart_activity:
[0,37528006,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
.....
```

```
行 374449: 06-17 08:47:23.779 1610 8335 I wm_restart_activity:
[0,180886340,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
    行 374456: 06-17 08:47:23.831 1610 8335 I wm_restart_activity:
[0,150254013,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
    行 374462: 06-17 08:47:23.894 1610 8335 I wm_restart_activity:
[0,140336969,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
    行 374468: 06-17 08:47:23.943 1610 8335 I wm_restart_activity:
[0,109147038,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
    行 374474: 06-17 08:47:23.992 1610 8335 I wm_restart_activity:
[0,114874765,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
```

行 374480: 06-17 08:47:24.041 1610 8335 I wm_restart_activity:
[0,155637138,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
行 374486: 06-17 08:47:24.090 1610 8335 I wm_restart_activity:
[0,181317602,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
行 374493: 06-17 08:47:24.138 1610 8335 I wm_restart_activity:
[0,32251779,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
行 374499: 06-17 08:47:24.199 1610 8335 I wm_restart_activity:
[0,221825154,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
行 374505: 06-17 08:47:24.253 1610 8335 I wm_restart_activity:
[0,31724451,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
行 374511: 06-17 08:47:24.302 1610 8335 I wm_restart_activity:
[0,90447214,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
行 374517: 06-17 08:47:24.350 1610 8335 I wm_restart_activity:
[0,57658655,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
行 374523: 06-17 08:47:24.400 1610 8335 I wm_restart_activity:
[0,195037244,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
行 374529: 06-17 08:47:24.449 1610 8335 I wm_restart_activity:
[0,157618517,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
at com.android.server.wm.ActivityRecord.activityPaused(ActivityRecord.java

----- pid 1610 at 2023-06-17 08:47:46.498513281+0800 -----

Cmd line: system_server

Binder:1610_10" prio=5 tid=44 Runnable | group="main" sCount=0 ucsCount=0 flags=0 obj=0x154cd0c8 self=0xb4000073f2198c10 | sysTid=8335 nice=-10 cgrp=default
sched=0/0 handle=0x71ed6dbcb0 | state=R schedstat=(112121591701 44545176123 356375) utm=8140 stm=3071 core=7 HZ=100 | stack=0x71ed5e4000-0x71ed5e6000
stackSize=991KB | held mutexes= "mutator lock"(shared held)
at com.android.server.wm.WindowContainer.getRequestConfigurationOrientation(WindowContainer.java:1220)
at com.android.server.wm.WindowContainer.getRequestConfigurationOrientation(WindowContainer.java:1203)
at com.android.server.wm.ActivityRecord.resolveFixedOrientationConfiguration(ActivityRecord.java:7570)
at com.android.server.wm.ActivityRecord.resolveOverrideConfiguration(ActivityRecord.java:7383)
at com.android.server.wm.ConfigurationContainer.onConfigurationChanged(ConfigurationContainer.java:124)
at com.android.server.wm.WindowContainer.onConfigurationChanged(WindowContainer.java:356)
at com.android.server.wm.WindowToken.onConfigurationChanged(WindowToken.java:378)
at com.android.server.wm.ActivityRecord.onConfigurationChanged(ActivityRecord.java:7972)
at com.android.server.wm.ConfigurationContainer.dispatchConfigurationToChild(ConfigurationContainer.java:154)
at com.android.server.wm.ConfigurationContainer.onConfigurationChanged(ConfigurationContainer.java:145)
at com.android.server.wm.WindowContainer.onConfigurationChanged(WindowContainer.java:356)
at com.android.server.wm.Task.onConfigurationChangedInner(Task.java:2275)

----- pid 1610 at 2023-06-17 08:48:19.649293373+0800 -----

Cmd line: system_server

"Binder:1610_10" prio=5 tid=44 Native

| group="main" sCount=1 ucsCount=0 flags=1 obj=0x154cd0c8 self=0xb4000073f2198c10

| sysTid=8335 nice=-10 cgrp=default sched=0/0 handle=0x71ed6dbcb0

| state=S schedstat=(144453678611 44679018577 358209) utm=10676 stm=3768 core=7 HZ=100

| stack=0x71ed5e4000-0x71ed5e6000 stackSize=991KB

| held mutexes=

native: #00 pc 000000000004def0 /apex/com.android.runtime/lib64/bionic/libc.so (syscall+32)

native: #01 pc 000000000028dc74 /apex/com.android.art/lib64/libart.so (art::ConditionVariable::WaitHoldingLocks(art::Thread*)+152)

native: #02 pc 000000000074cd0c /apex/com.android.art/lib64/libart.so (art::GoToRunnable(art::Thread*)+412)

native: #03 pc 000000000074cb38 /apex/com.android.art/lib64/libart.so (art::JniMethodEnd(unsigned int, art::Thread*)+28)

at android.util.MemoryIntArray.nativeSize(Native method)

at android.util.MemoryIntArray.size(MemoryIntArray.java:136)

at android.util.MemoryIntArray.enforceValidIndex(MemoryIntArray.java:214)

at android.util.MemoryIntArray.get(MemoryIntArray.java:109)

```
at android.provider.Settings$GenerationTracker.readCurrentGeneration(Settings.java:2714)
at android.provider.Settings$GenerationTracker.isGenerationChanged(Settings.java:2698)
at android.provider.Settings$NameValueCache.getStringsForPrefix(Settings.java:3101)
- locked <0x082ee7d6> (a android.provider.Settings$NameValueCache)
at android.provider.Settings$Config.getStrings(Settings.java:16341)
at android.provider.DeviceConfig.getProperties(DeviceConfig.java:661)
```

#6 - 2023-07-04 19:12 - CD FW 曹覃刚

- Category changed from SYSTEM to CD-FW

- Assignee changed from CD SYSTEM-夏旭 to CD FW 曹覃刚

#7 - 2023-07-05 19:04 - CD FW 曹覃刚

- Assignee changed from CD FW 曹覃刚 to CD FW王武军

#8 - 2023-07-07 17:24 - CD FW王武军

【】当前状态

1) 重启原因

正如前面同事分析的信息，系统发生重启的原因是：system_server被watchdog的超时机制杀除导致。

而触发watchdog的超时原因是：system_server发生了anr，

anr的原因是：binder线程因为等待WindowManagerGlobalLock锁，触发watchdog超时。

而持有了0x071f13e7(WindowManagerGlobalLock)锁的方法，在执行MemoryIntArray.nativeSize方法时时间太长或者未正常执行完，导致WindowManagerGlobalLock全局锁一直未被释放，其他binder线程一直在等待这个锁，最终导致watchdog超时触发。

2) 结合MTBF脚本和日志深入分析

首先从MTBF日志文件：202306151137.log

可以知道异常是在脚本的第53轮测试中。

参考第52轮的测试日志：正确情况应该如下：

2023-06-17 07:24:50,733 - main - recent_APP_Longtime_switch_test - 打开APP

2023-06-17 07:25:33,042 - main - recent_APP_Longtime_switch_test - 等待30min

2023-06-17 07:55:40,919 - main - recent_APP_Longtime_switch_test - 打开成功——Pass

2023-06-17 07:55:57,044 - main - clearapp_test - 清除完毕

53轮发生异常：

2023-06-17 08:17:05,831 - main - recent_APP_Longtime_switch_test - 打开APP

2023-06-17 08:17:48,195 - main - recent_APP_Longtime_switch_test - 等待30min

2023-06-17 08:49:13,452 - main - run - 异常地点：

正常情况，应该在2023-06-17 08:47:48 - 2023-06-17 08:49:13这个时间段输出：打开成功——Pass

打开成功——Pass的相关测试是：

```
self.device.press("home")
self.device.shell("input keyevent 187")
self.device.click(800, 420)
time.sleep(3)
self.device.shell("input keyevent 187")
self.device.click(1400, 420)
time.sleep(3)
if self.device.xpath('//*[@@resource-id="com.android.launcher3:id/workspace"]/android.view.ViewGroup['
    '2]/android.view.ViewGroup1').exists:
    logger.info("打开失败——Error")
self.save_picture_report()
```

```
else:
    logger.error("打开成功——Pass")
self.clearapp_test()
```

翻译过来就是：进入recent，点击触发启动第一个task，和第二个task所对应的应用

常规情况下，存在于recent的第一个task应该是可以被正常启动的，且走的生命周期是：

```
07-02 06:04:21.388 1002 2364 I wm_set_resumed_activity: [0,com.tencent.tmgp.sgame/.SGameActivity,moveTaskToFront findTaskToMoveToFront]
07-02 06:04:21.392 1002 2364 I wm_pause_activity: [0,140182676,com.android.launcher3/.uioverrides.QuickstepLauncher,userLeaving=true,pauseBackTasks]
07-02 06:04:21.401 2693 2693 I wm_on_top_resumed_lost_called: [140182676,com.android.launcher3/.uioverrides.QuickstepLauncher,topStateChangedWhenResumed]
07-02 06:04:21.404 2693 2693 I wm_on_paused_called: [140182676,com.android.launcher3/.uioverrides.QuickstepLauncher,performPause]
07-02 06:04:21.425 1002 2364 I wm_resume_activity: [0,8325696,139,com.tencent.tmgp.sgame/.SGameActivity]
```

但是异常发生时的日志却是：

```
06-17 08:47:11.051 1610 12866 I wm_set_resumed_activity:
[0,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity,moveTaskToFront findTaskToMoveToFront]
06-17 08:47:11.064 1610 12866 I wm_pause_activity: [0,241614778,com.android.launcher3/.uioverrides.QuickstepLauncher,userLeaving=true,pauseBackTasks]
06-17 08:47:11.071 25908 25908 I wm_on_top_resumed_lost_called:
[241614778,com.android.launcher3/.uioverrides.QuickstepLauncher,topStateChangedWhenResumed]
06-17 08:47:11.076 25908 25908 I wm_on_paused_called: [241614778,com.android.launcher3/.uioverrides.QuickstepLauncher,performPause]
06-17 08:47:11.173 1610 8335 I wm_restart_activity:
[0,37528006,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
```

```
wm_restart_activity: [0,37528006,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
```

这个条日志一直从recent这个操作开始一直循环的被输出，而触发这日志的逻辑是：ActivityTaskSupervisor.realStartActivityLocked，大致的逻辑就是，启动应用，如果应用进程存在那么执行热启动，就会执行realStartActivityLocked的逻辑，进而输出wm_restart_activity。

飞书启动的异常日志：

```
06-17 08:47:11.173 1610 8335 I wm_restart_activity:
[0,37528006,8241,com.ss.android.lark/com.ss.lark.android.passport.biz.feature.session.TransparentBringAppToFrontActivity]
06-17 08:47:11.174 1610 8335 I ActivityTaskManager: The Process com.ss.android.lark Already Exists in BG. So sending its PID: 21949
06-17 08:47:11.175 1610 8335 W ActivityTaskManager: Tried to set launchTime (0) < mLastActivityLaunchTime (166149975)
```

可以看出飞书(com.ss.android.lark)的这个界面启动时间为0，说明应用并没有启动起来，但是task的id是8241，一直都存在没有被销毁。

后面就是飞书的这个应用被循环的启动，并且一直没启动起来，而且这个启动操作非常的密集，非常密集的调用系统启动应用的流程方法，系统资源被大量占用，最后导致binder线程被某一个

一个耗时方法调用锁住了(WindowManagerGlobalLock)锁，导致watchdog超时。

【】 下一步

飞书从recent中启动时，为什么会存在无限重启？目前还不能定位这个异常的原因，目前的怀疑点有3个：

- 1) 是飞书应用自己启动的问题；
- 2) 系统杀除应用进程的原因，导致飞书应用不能被正常启动；
- 3) 系统启动应用的流程出现了异常，导致应用不能被正常启动。

计划从两个方向展开：

1) 系统层面

策略是打开更多的系统日志，编译vb版本，尝试复现，以获取更多的日志信息，进一步分析问题。

2) 应用层面

针对飞书这个应用做针对性测试，可以结合MTBF脚本来完成。

【】当前状态

1) 放开了日志，编译了Jenkins版本

<https://dev.thundercomm.com/gerrit/c/general/platform/frameworks/base/+213217>

版本：

/Pre_figure/VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230709/202307091014-4933

2) 尝试复现并跟进日志分析流程

自己将MTBF脚本中针对 “ 大量启动应用并从recent中启动的测试方式 ” 在本地进行了复现尝试。
未能复现票中的问题，从未复现的日志中分析了从recent中启动应用的流程，未找到切入的方式。

3) 单独测试了飞书这个应用

不论是从冷启动还是从recent中启动，都未发现异常。

【】下一步

1) @测试同事，请使用放开日志的Jenkins版本进行复现，以便获取更多的系统日志信息。

版本地址：

/Pre_figure/VerifyBuild/Pre_figure_turbox-c2130c-la1.1-qssi12-dev/20230709/202307091014-4933

(Tips：复现的方式按照票中测试进行方式进行，尽量还原当前的测试场景，如：相同的应用，飞书在最后一个启动等)

2) 研发这边根据日志版本，继续做本地的复现和原理分析

#10 - 2023-07-11 09:50 - CDTS_Test 吴诗雨

VB跑测20H，暂未复现

#11 - 2023-07-28 14:57 - CDTS_Test 吴诗雨

已复现，日志已发

#12 - 2023-08-29 10:02 - CD FW王武军

- Assignee changed from CD FW王武军 to CDTS_Test 吴诗雨