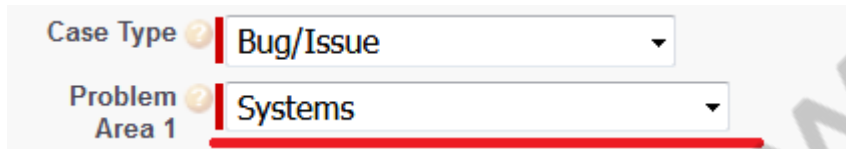


## 性能问题 Case 内容模板

本文主要描述的是 AP 侧性能相关常见问题 Case 描述的模板。对于每种问题需要提供的各种 Log，请参考文档。

性能问题更新了 Problem Area，请客户遵循更新过后的 PA 进行提交。新的 PA 下，Performance 问题全部放在 System 目录下。

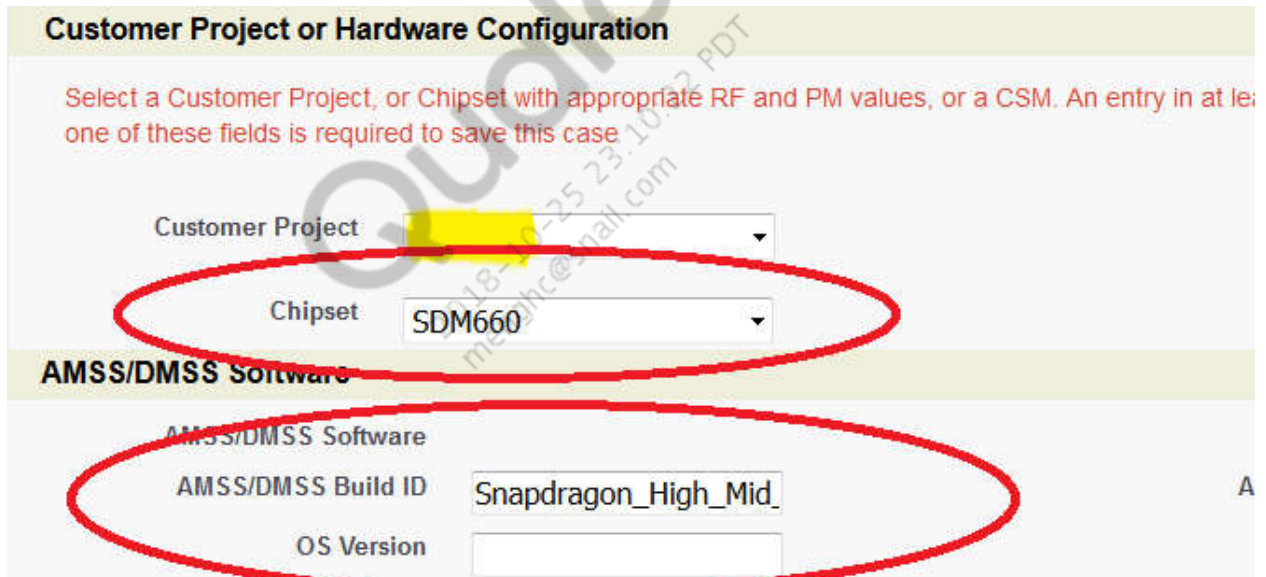


Case Type ? Bug/Issue

Problem Area 1 ? Systems

### 1. 常见性能问题和 Case 内容模板

前提：务必提交正确的平台信息，包括 Build ID，Android 版本，芯片型号等。



**Customer Project or Hardware Configuration**

Select a Customer Project, or Chipset with appropriate RF and PM values, or a CSM. An entry in at least one of these fields is required to save this case.

Customer Project [Redacted]

Chipset SDM660

**AMSS/DMSS Software**

AMSS/DMSS Software

AMSS/DMSS Build ID Snapdragon\_High\_Mid\_


OS Version


模板当中标注星号（\*）的是必要信息，红色的是关键信息。


#### 1.1 开/关机速度


此类 Case 主要用于提交开/关机时间达不到标准的场景。

- Case 分类

Case Type  Bug/Issue

Problem Area 1  Systems

Problem Area 2  Performance-AP

Problem Area 3  UX/Boot Performance

- 模板

标题*	简单描述一下问题
测试机信息	测试机软硬件信息，内存大小，分辨率，Android 版本
问题场景*	详细描述问题，例如是开机时间和对比机相比偏慢；还是开机长时间卡在那里；或者就是对当前的结果不满意，希望能够继续优化，优化的目标是多少等。系统是否有安装第三方应用，如果有请提供数量。
复现概率*	是必现的问题还是概率性
开关机计时起始点*	是按下 power 键开始计时，或者是手机震动开始计时等
开关机计时终止点*	动画结束，或者锁屏界面，或者是 launcher 启动界面等
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的开机时间结果
必要的 log 信息（测试机和对比机）*	请参考文档2.1节
其他	如果有自己初步的分析，请提供分析结果供参考

- 例子

标题：开机时间比同平台 xxx 机器慢

问题描述： 我司 MMM 项目当前开机时间是 30 秒，无法达标

概率：100%

开机计时起始点：按下 power 键开始计时

开机计时终止点：锁屏界面出现时结束计时

对比机信息： xxx 和 MMM 同平台，硬件配置相同，Android N 版本，他们测试结果是 25 秒

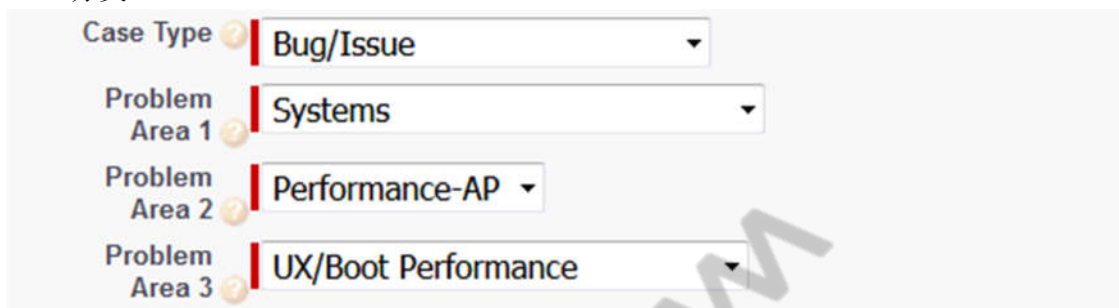
Log 信息： 上传到附件

其他： 我们分析了 logcat 的 event 信息，发现是 kernel 初始化和 PMS scan 比参考机慢是主要原因 etc。。。

## 1.2 系统唤醒速度

此类 Case 主要用于提交通过各种途径唤醒系统的速度不达标问题

- Case 分类



- 模板

标题*	简单描述一下问题
测试机信息*	测试机软硬件信息，总内存，分辨率，Android 版本
问题场景*	详细描述问题，什么场景下通过 <b>什么方式</b> （power key, fp, incoming call）来唤醒系统。测试的 <b>开始时间</b> 和 <b>结束时间</b> 分别是以什么作为标准。
复现概率*	是必现的问题还是概率性
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的测试结果和对比 log
必要的 log 信息（测试机和对比机）*	请参考文档 2.2 节
其他	如果有自己初步的分析，请提供分析结果供参考

- 例子

标题：系统通过 power 键唤醒的时间不达标

问题场景：确保系统进入休眠后一段时间，通过 power 键唤醒系统，以 power 键按下为起点，屏幕点亮并且显示第一帧内容作为终点。我们的设备测试结果是 300ms。

概率：100%

对比机信息：XXX 是 8953 平台，Android N 版本，测试结果是 250ms。

Log 信息：请查看附件

其他：无

## 1.3 App 启动速度

此类 Case 主要用于提交通过各种途径唤醒系统的速度不达标问题

- Case 分类

- 模板

标题*	简单描述一下问题
测试机信息*	测试机软硬件信息，总内存，分辨率，Android 版本
问题场景*	详细描述问题，app 名字，应用启动类型（冷启动，热启动）， <b>启动时间的起始点和终止点</b>
复现概率*	是必现的问题还是概率性
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的测试结果和对比 log
必要的 log 信息（测试机和对比机）*	请参考文档 2.3 节
其他	如果有自己初步的分析，请提供分析结果供参考

- 例子

标题：xyz 应用冷启动速度不达标

问题场景：安装完 xyz 应用之后手动启动一次，并且完成必要的初始化步骤（登陆信息，联网等）。通过系统的清理功能把 xyz kill 掉，并且没有相关线程在后台运行。

以手指触摸到 icon 为起始点，应用第一帧内容显示出来作为终止点，xyz 的启动时间是 5 秒。


概率：100%


对比机信息：对比机测试结果是 4.3 秒


#### 1.4 卡顿问题


此类 Case 主要用于提交各种系统卡顿问题

- Case 分类

Case Type  Bug/Issue

Problem Area 1  Systems

Problem Area 2  Performance-AP

Problem Area 3  UX/Boot Performance

- 模板

标题*	简单描述一下问题
测试机信息*	测试机软硬件信息，总内存，分辨率，Android 版本
问题场景*	详细描述问题，app 名字，具体界面，卡顿现象（短时间的卡顿，抖动，UI 明显卡住（卡住的时间长度）），系统状态等。短暂的卡顿，丢帧问题请提供 FPS
复现概率*	是必现的问题还是概率性
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的测试结果和对比 log
必要的 log 信息（测试机和对比机）*	请参考文档 2.4 节
其他	如果有自己初步的分析，请提供分析结果供参考

- 例子

标题：系统状态栏下滑卡顿（或丢帧，抖动，卡住 1 秒）

问题场景：系统安装 50 个 apk，并且轮流启动 50 个 apk 放在后台运行，下拉状态栏过程出现丢帧现象。

概率：60%


对比机信息：参考机软硬件信息，参考机结果：参考机没有这个现象


其他：


## 1.5 Benchmark 跑分


此类 Case 主要用于提交各种跑分软件不达标问题

- Case 分类

Case Type  Bug/Issue

Problem Area 1  Systems

Problem Area 2  Performance-AP

Problem Area 3  Benchmark

- 模板

标题*	简单描述一下问题
测试机信息*	测试机软硬件信息，总内存，分辨率，Android 版本
问题场景*	跑分软件名字，版本号，总分，各个子项分数
复现概率*	是必现的问题还是概率性
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的测试结果和对比 log
必要的 log 信息（测试机和对比机）*	请参考文档的 2.5 节
其他	如果有自己初步的分析，请提供分析结果供参考

- 例子

标题：安兔兔跑分对比机低

问题场景：安兔兔 6.2.6 版本跑分对比机低，总分和具体的各子项分数如下。通过和参考机对比，我们发现主要是 UX IO、UX image process 比参考机低。

Antutu	
Total Score	114618
3D	30339
3D [Marooned]	18852
3D [Garden]	11487
UX	41555
UX Data Secure	7609
UX Data Process	7103
UX Strategy Games	10748

UX Image Process	11035
UX I/O Performance	5060
CPU	34634
CPU Mathematics	9021
CPU Common Use	9999
CPU Multi-Core	15614
Ram Score	8090

概率：100%

对比机信息：参考机软硬件信息，参考机结果：参考机的总分和子项分数如下。

Antutu	
Total Score	114618
3D	30339
3D [Marooned]	18852
3D [Garden]	11487
UX	41555
UX Data Secure	7609
UX Data Process	7103
UX Strategy Games	10748
UX Image Process	11035
UX I/O Performance	5060
CPU	34634
CPU Mathematics	9021
CPU Common Use	9999
CPU Multi-Core	15614
Ram Score	8090

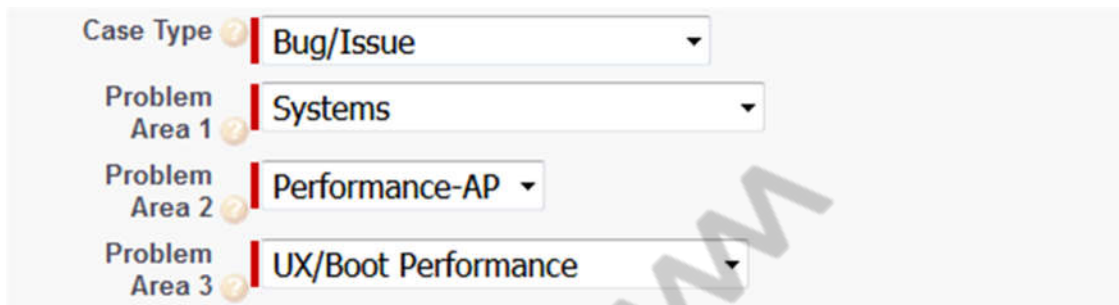
其他：我们尝试过提高 CPU、DDR、EMMC 频率，对这个问题有锁帮助等等



## 1.6 长时间使用后卡顿问题

此类 Case 主要用于提交机器经过长时间使用/monkey test 之后才出现的卡顿问题，包括高概率和低概率卡顿问题。

- Case 分类



- 模板

标题*	简单描述一下问题
测试机信息*	测试机软硬件信息，总内存，分辨率，Android 版本
问题场景*	详细描述问题，具体是在什么场景下出现的卡顿，例如是日常使用，monkey test，后台有多少 app 在运行，当前网络状况，内存状况等信息。因为这类问题通常都是很多因素综合引起的，因此越详细的信息对分析问题越有帮助。
复现概率*	是必现的问题还是概率性
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的测试结果和对比 log
必要的 log 信息（测试机和对比机）*	请参考文档的 2.6 节
其他	如果有自己初步的分析，请提供分析结果供参考

- 例子

标题：系统 monkey test 72 小时以后，整体运行都很卡顿

问题场景：系统运行 monkey test 72 小时以后，整机反应很慢，启动 app 很慢，上下滑动列表很卡，当时查看了后台有很多的 app 在运行，可用内存比较低。

概率：80%

Log 信息：请查看附件

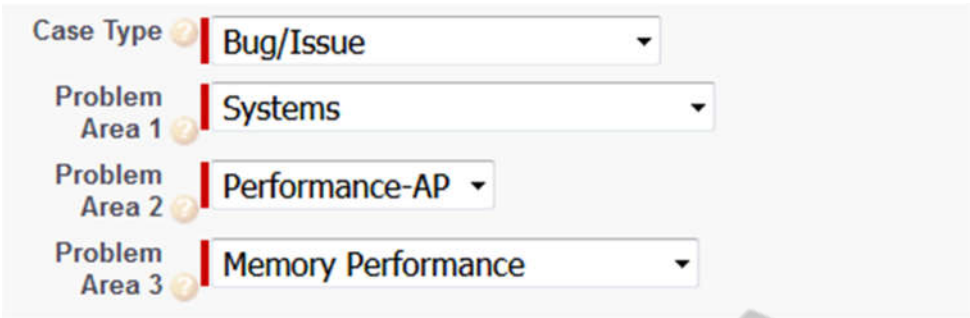
其他：无


## 1.7 内存优化


此类 Case 主要用于提交内存优化相关的问题，例如希望通过优化内存来满足小内存设备的正常运行。





• Case 分类



Case Type  Bug/Issue

Problem Area 1  Systems

Problem Area 2  Performance-AP

Problem Area 3  Memory Performance

• 模板

标题*	简单描述一下问题
测试机信息*	测试机软硬件信息，总内存，分辨率，Android 版本
问题场景*	详细描述问题，正常开机之后的内存状态，启动若干个 app 之后的内存状态，memory map 信息等
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的测试结果和对比 log
必要的 log 信息（测试机和对比机）*	请参考文档的 2.7 节
其他	如果有自己初步的分析，请提供分析结果供参考

• 例子

标题：系统可用内存较低，希望优化

问题场景：当前系统开机之后只有 200MB 的可用内存，启动 camera 之后很容易引起 lowmemorykiller 导致很多后台应用被杀

概率：100%

Log 信息：请查看附件

其他：无

1.8 IO 性能

此类 Case 主要用于提交 IO 性能相关的问题，例如文件拷贝速度，MTP 传输速度等。

• Case 分类

Case Type: Bug/Issue

Problem Area 1: Systems

Problem Area 2: Performance-AP

Problem Area 3: IO Performance

- 模板

标题*	简单描述一下问题
测试机信息*	测试机软硬件信息，总内存，分辨率，Android 版本
问题场景*	详细描述问题，测试过程的机器状态，例如存储器占用比例，剩余内存大小，是否连接 USB，如果是 T 卡相关的还要提供 T 卡的容量、级别、型号等
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的测试结果和对比 log
必要的 log 信息（测试机和对比机）*	请参考文档的 2.8 节
其他	如果有自己初步的分析，请提供分析结果供参考

- 例子

标题：系统拷贝文件到 T 卡比参考机慢

问题场景：手机插入一张 Sandisk Class 10 的 T 卡，容量 16GB。格式化完 T 卡之后，从手机拷贝一个 3GB 的大文件到 T 卡并且测量拷贝时间，测试机花 30s，参考机 26s。

概率：100%

Log 信息：请查看附件


其他：无


### 1.9 Data throughput


此类 Case 主要用于提交由于 **AP 侧的性能引起的数据业务吞吐量**相关的问题，如果是 modem 侧的问题请提交相应的 case 到 modem team。


判断modem侧吞吐量是否有问题，请参考文档80-NV852-1

- Case 分类

Case Type  Bug/Issue

Problem Area 1  Systems

Problem Area 2  Performance-AP

Problem Area 3  IO Performance

- 模板

标题*	简单描述一下问题
测试机信息*	测试机软硬件信息，总内存，分辨率，Android 版本
问题场景*	详细描述问题，是否使用 USB tethering 测试，是否连接 callbox 通过 ftp 来上传下载进行测试，通过调节 CPU/DDR/Storage 的频率是否有锁帮助
对比机器在同样测试条件下的结果*	如果有对比机信息，请务必提供对比机的详细信息（厂商，机器型号，芯片型号，机器硬件配置，Android 版本）并且提供对比机的测试结果和对比 log
必要的 log 信息（测试机和对比机）*	请参考文档的 2.8 节
其他	如果有自己初步的分析，请提供分析结果供参考

- 例子

标题：测试机在 callbox 条件下测试吞吐量不达标

问题场景：使用 USB tethering 的模式，连接 Callbox，通过 ftp 协议进行数据业务下载测试，测试结果和芯片理论支持的结果差距比较大。通过调节 CPU 工作模式发现对问题有所改善。

概率：100%

Log 信息：请查看附件

其他：无

### 1.10 \*新增的 PA 介绍

#### i. CPU Scheduler/DCVS

当客户碰到和 CPU 调度器/DCVS 相关问题时请提交到这里，并提供必要的 systrace/log。必要的 systrace 信息建议参考文档 KBA-161001235906。

Case Type Bug/Issue

Problem Area 1 Systems

Problem Area 2 Performance-AP

Problem Area 3 CPU Scheduler/DCVS

ii. DSP(Compute)

当客户碰到 DSP 计算相关性能问题时，请提交到此 PA。

Case Type Bug/Issue

Problem Area 1 Systems

Problem Area 2 Performance-AP

Problem Area 3 DSP (Compute)

iii. Initial performance tuning

当客户需要对手机软件进行初步的性能调优时，请提交 case 到此 PA。提交 case 时请详细描述需要 Qualcomm 帮忙优化哪些场景的性能，并且附上手机对应的 kernel config 和 Antutu 跑分结果供参考。

Kernel config 的获取请参考[这里](#)。

Case Type Bug/Issue

Problem Area 1 Systems

Problem Area 2 Performance-AP

Problem Area 3 Initial Performance Tuning

iv. Performance framework

当客户提交 BoostFramework/QPerformance/perfd/IOP 等模块相关问题时，请提交到这个 PA。当提交 log/systrace 时，请参考[这里](#)来打开相关 log 信息。

Case Type Bug/Issue

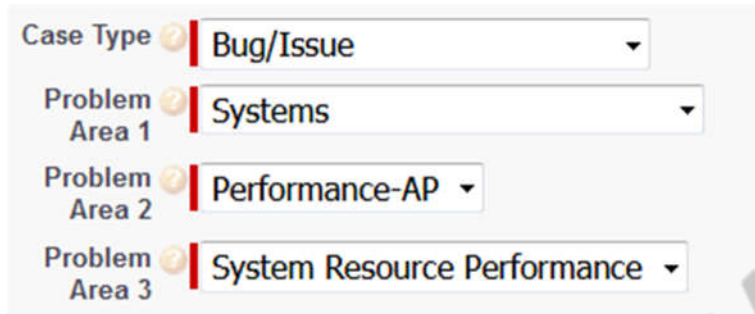
Problem Area 1 Systems

Problem Area 2 Performance-AP

Problem Area 3 Performance Framework

v. System resource performance

当客户需要 QC 帮助分析解决 Bus 相关性能问题时，请提交到这里。如果需要进行 Bus profiling，请参考文档 KBA-170815024731 做好准备工作。



## 2. 必要的 log 信息

所有问题都需要提供 **kernel configuration**，可以从 `/proc/config.gz` 获取。如果 `config.gz` 不存在，则需提供和提交问题相同版本的 `out/target/product/msmxxxx/obj/KERNEL_OBJ/.config`

### 2.1 开关机速度

Kernel log, events log, logcat log

#### 2.1.1 抓取方法

```
adb wait-for-device root
adb wait-for-device
adb shell dmesg > dmesg.txt
adb logcat -v threadtime -b events -d > logcat_events.txt
adb logcat -v threadtime -d *:V > logcat.txt
```

### 2.2 系统唤醒速度

Kernel log, systrace

#### 2.2.1 抓取方法

- Kernel log

```
kernel config: CONFIG_PM_SLEEP_DEBUG=y
device\qcom\msmxxx\BoardConfig.mk:
BOARD_KERNEL_CMDLINE += initcall_debug log_buf_len=16M
```

- Systrace

```
adb shell "echo mdss:* >> /d/tracing/set_event"
adb shell "echo 1 > /d/tracing/events/mdss/enable"
adb shell "echo 1 > /d/tracing/events/mdss/tracing_mark_write/enable"
```

- 连接 usb 线，按下 power 键，关闭屏幕，等 1 分钟
- 开始抓 kernel 和 adb log
- 开始抓 systrace，稍等 2s 后，按下 power 键，点亮屏幕
- 如果连接 USB 后，不能重现该问题，我们就不能用连接 USB 的方法抓取 Systrace log，可以采用不用连接 USB 来 catch systrace log. 请参考 kba-161001235906 怎样用离线的方式获取 systrace log.

## 2.3 APP 启动速度

Kernel, logcat, systrace

### 2.3.1 抓取方法

Systrace 抓取之前需要 enable perfd 的 log。方法如下：

```
adb pull /system/build.prop
```

在 build.prop 中增加 debug.trace.perf=1

```
adb push build.prop /system/
```

```
adb shell chmod 0644 /system/build.prop
```

```
adb shell sync
```

```
adb shell reboot
```

或者

```
adb shell setenforce 0
```

```
adb shell setprop debug.trace.perf 1
```

```
adb shell stop perfd
```

```
adb shell start perfd
```

perfd 的 log 就会显示在 logcat 和 systrace 中。

## 2.4 卡顿问题

Systrace

### 2.4.1 抓取方法

Systrace 需要使能以下 event:

```
adb shell "echo sched:sched_migrate_task >> /sys/kernel/debug/tracing/set_event"
```

```
adb shell "echo sched:sched_switch >> /sys/kernel/debug/tracing/set_event"
```

```
adb shell "echo sched:sched_wakeup >> /sys/kernel/debug/tracing/set_event"
```

```
adb shell "echo sched:sched_cpu_load >> /sys/kernel/debug/tracing/set_event"
```

```
adb shell "echo kgsi:* >> /d/tracing/set_event"
```

```
adb shell "echo mdss:* >> /d/tracing/set_event"
```

```
adb shell "echo 1 > /d/tracing/events/mdss/enable"
```

```
adb shell "echo 1 > /d/tracing/events/mdss/tracing_mark_write/enable"
```

```
adb shell "echo 1 > /d/tracing/events/binder/enable"
```

## 2.5 Benchmark 跑分

Systrace, thermal log

### 2.5.1 抓取方法

Thermal log 需要通过特定的 thermal 工具来获取，请先联系 Qualcomm 获取 thermal 工具，然后按照下面方法抓 thermal log:

- Push tool 到手机里

```
adb push <source_to_tsens_logger> /data
```

```
adb shell chmod 777 /data/msm_tsens_logging
```

- 运行  
adb shell./data/msm\_tsens\_logging 250 3600000 &  
这里 250 是采样频率， 3600000 是获取 log 的时间，两者的单位都使 ms。  
上面的 command 其含义是每隔 250ms 读一次 tsonsensor log，持续 1 个小时。
- 获取 log  
其 log 存放在/data/tsens\_logger.csv。当测试完毕，pull 其 log 即可。  
adb pull /data/tsens\_logger.csv <path\_to\_pull\_to>

## 2.6 长时间使用后卡顿问题

有多种原因可能导致此问题，例如程序本身的健壮性，有时候是程序本身 crash 导致机器反应慢，网络问题，thermal 等问题。所以当发生此问题时，提供充足的 log 是必要的。

Kernel, logcat, systrace, thermal log, memory information

### 2.6.1 抓取方法

- Memory information:

```
adb shell dumpsys meminfo
adb shell cat /proc/meminfo
adb shell dumpsys procstats --hours 3
adb shell dumpsys usagestats
adb shell cat /proc/vmstat
```

如果需要获取长时间的 memory 信息，请写一个脚本来实现，例如

```
while true
do
adb -s $1 shell date >> dumpsys_meminfo_$1_MLP.txt
adb -s $1 shell dumpsys meminfo >> dumpsys_meminfo_$1_MLP.txt
adb -s $1 shell date >> meminfo_$1_MLP.txt
adb -s $1 shell 'cat /proc/meminfo' >> meminfo_$1_MLP.txt
adb -s $1 shell date >> vmstat_$1_MLP.txt
adb -s $1 shell 'cat /proc/vmstat' >> vmstat_$1_MLP.txt
echo "One round done"
sleep 1
done
```

- Systrace

如果是低概率的问题，除了通过上述方式循环获取 memory 信息之外，还需要提供 systrace。低概率问题的 systrace 获取，请参考 KBA-161001235906:

- 连接 USB
- adb shell atrace --async\_start -z -b 20960 gfx input audio view webview wm am hal app res dalvik rs bionic power sched freq idle load sync workq memreclaim binder\_lock binder\_driver
- 断开 USB，复现问题
- 当问题复现，链接 USB 后立即运行下面命令



- adb shell atrace --async\_dump -z -b 20960 gfx input audio view webview wm am hal app res dalvik rs bionic power sched freq idle load sync workq memreclaim binder\_lock binder\_driver >atrace.out

## 2.7 内存优化

Kernel log, memory

### 2.7.1 抓取方法

- Kernel log  
开机过程的 kernel log，log 信息是需要从 0 秒开始的，用来计算 memory map
- Memory 信息

```
adb shell dumpsys meminfo
adb shell cat /proc/meminfo
adb shell cat /proc/vmstat
adb shell cat /proc/iomem
```

## 2.8 IO 性能/Data throughput

Systrace, kernel log, logcat

### 2.8.1 抓取方法

- Systrace 必要的 category: sched, irq, freq, idle, disk, mmc, load, power, workq, memreclaim, pagecache  
其中红色字体的需要 root 权限
- 还需要打开下面的 events:  
adb shell "echo 1 > /d/tracing/events/block/enable"  
adb shell "echo 1 > /d/tracing/events/writeback/enable"  
adb shell "echo 1 > /d/tracing/events/ext4/enable" ///这是 ext4 文件系统需要打开  
adb shell "echo 1 > /d/tracing/events/f2fs/enable" ///这是 f2fs 文件系统需要打开

如果是 USB 相关的 IO 性能，还需要

```
adb shell "echo 1 > /d/tracing/events/dwc3/enable"
```