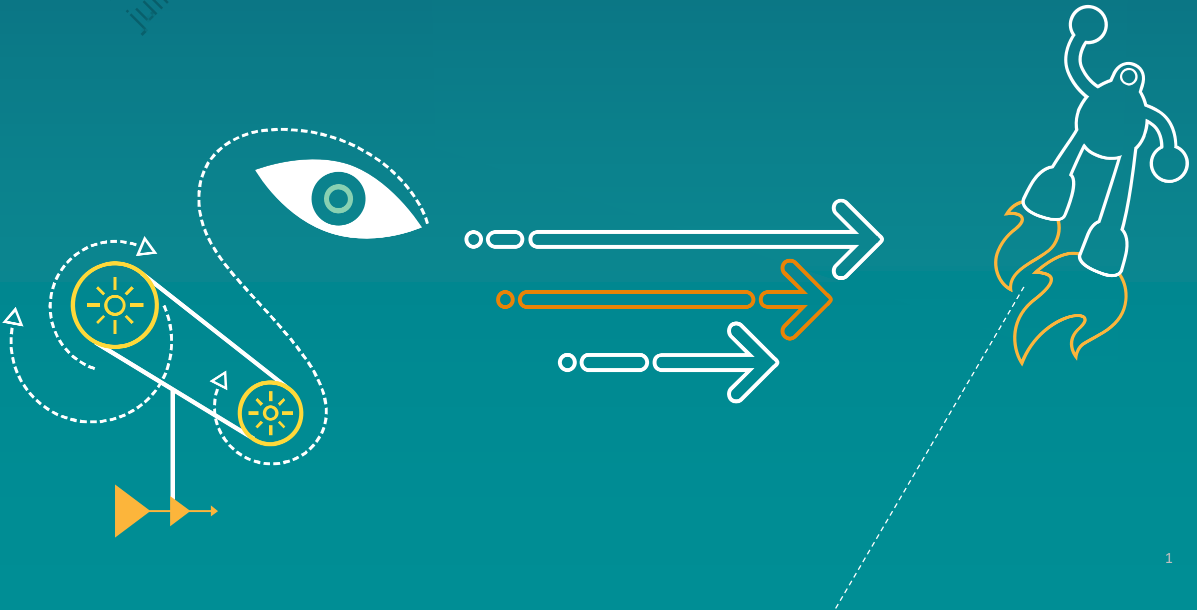
Game Performance Issue Debugging

QUALCOMM®

Oct, 2018



Confidential and Proprietary – Qualcomm Technologies, Inc.

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision History

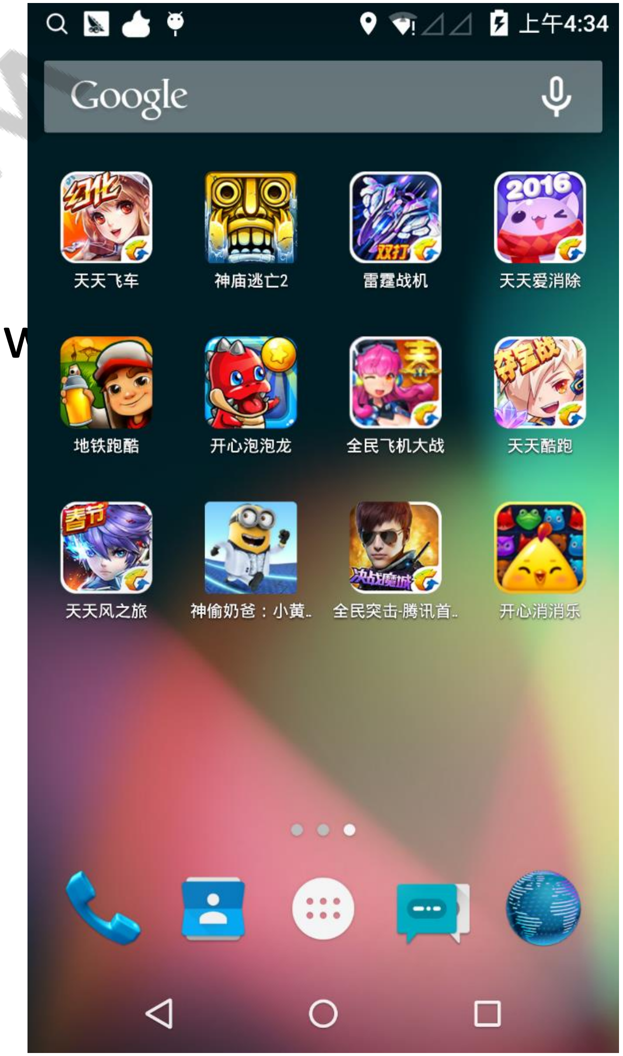
Revision	Date	Description
A	Oct 2018	Initial release

Contents

- Typical issue and cause
- Metrics and test procedure
- Debugging steps
- Analysis tool and Log capturing
- Report issue
- Questions?

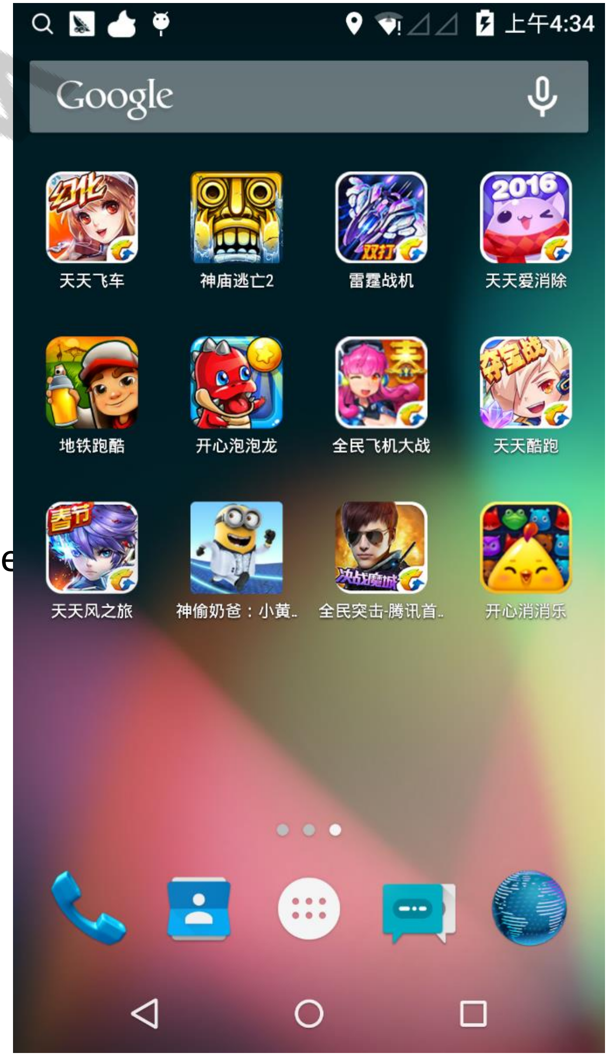
Typical issue

- Game launch time is long
- Game is janky, frame drop
- Game display shake, stuck
- Switch response between scenario is slow
- Audio flick/noise



Typical Cause

- Tunning:
  - scheduling latency,
  - cpufreq governor
  - GPU dvfs
  - Thermal mitigation
  - Low latency audio 4ms vs normal 20ms
- System level performance
  - low power model impact
  - In-efficiency of support module: mem allocation, gfx drive display
- Game design issue
  - bad design:
    - Inefficiency leads to cpu/gpu bound
    - Synchronization: less thread or too many thread
    - security mechanism side effect
    - Logic error: touch protocol



Contents

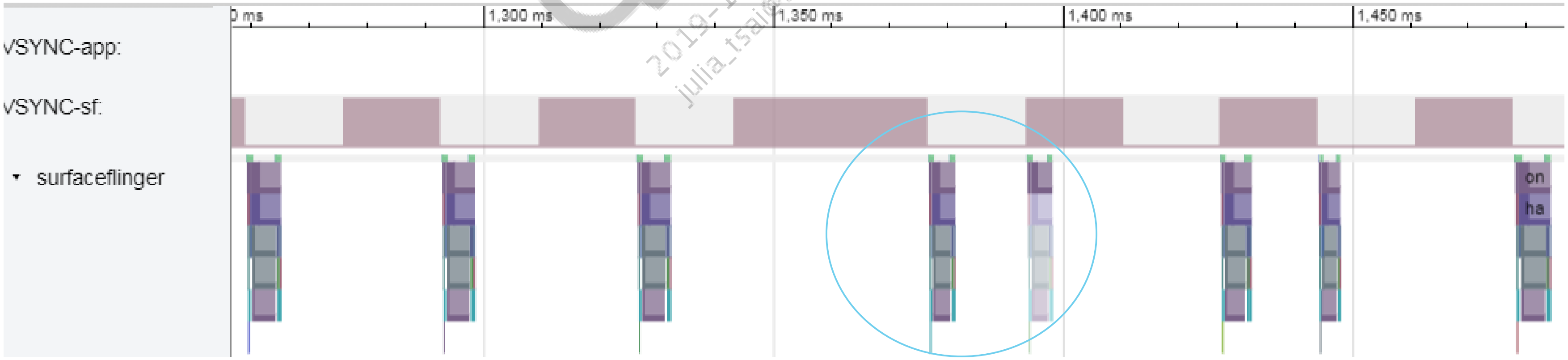
- Typical issue and cause
- Metrics and test procedure
- Debugging steps
- Analysis tool and Log capturing
- Report issue
- Questions?

Metrics and test procedure

- Basic metrics
  - Launch latency (LL), switch latency
  - Fps
- Measurement:
  - With high speed camera:
    - LL: From touch event to first UI update.
    - Fps: count the number of fps in a second
  - From systrace:
    - LL: from deliverInputEvent to doComposition
    - Fps: number of queueBuffer in game threads

Metrics and test procedure

- Jank definition:
  - Average fps is bad
  - Average fps good, miss vsync
  - queueBuffer is fast and slow
  - Dramatically change of UI, for example spirit position



Metrics and test procedure

- Precondition: Different configuration, different workload
  - Rendering quality, visual effect(AA in PUBG, character stoke in KOH)
  - Target fps: 20/30/40/60
  - Player Role(levels, looks) Engine specific(Multithread, vulkan)
  - Others: battery %, charging or not(thermal impact) , Network

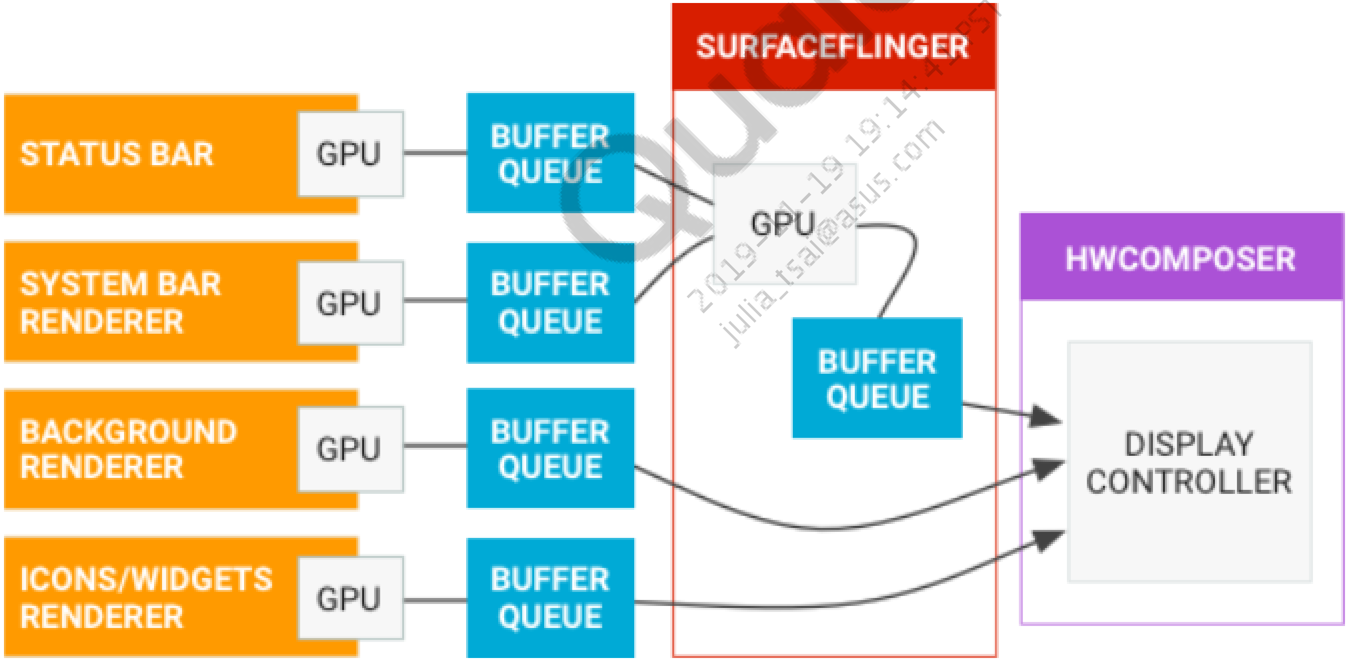
- Comparison test
  - Playback model
  - Watching model

Contents

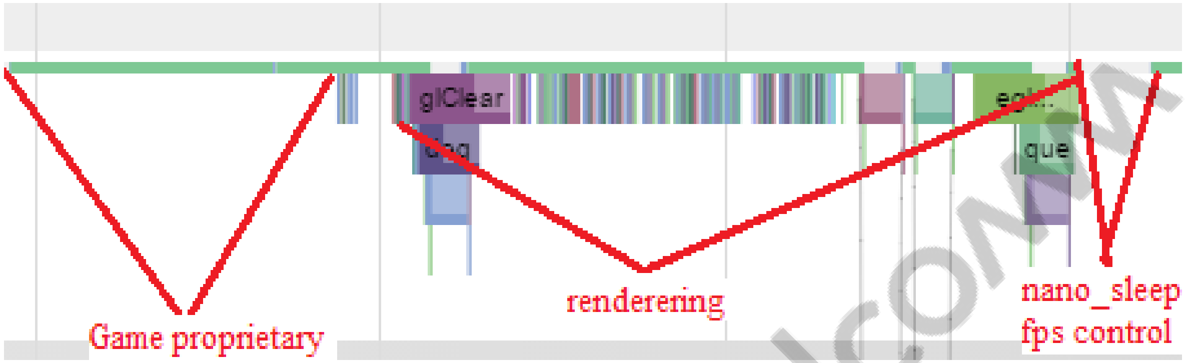
- Typical issue and cause
- Metrics and test procedure
- Debugging steps
- Analysis tool and Log capturing
- Report issue
- Questions?

Components that impact game perf

- Graphics pipeline of Android
  - Game content(surface) is rendered with GPU
  - Game surface is sent to SurfaceFlinger
  - SurfaceFlinger compose different Surface with GPU or HWC
- In bad case, SF will consume some GPU

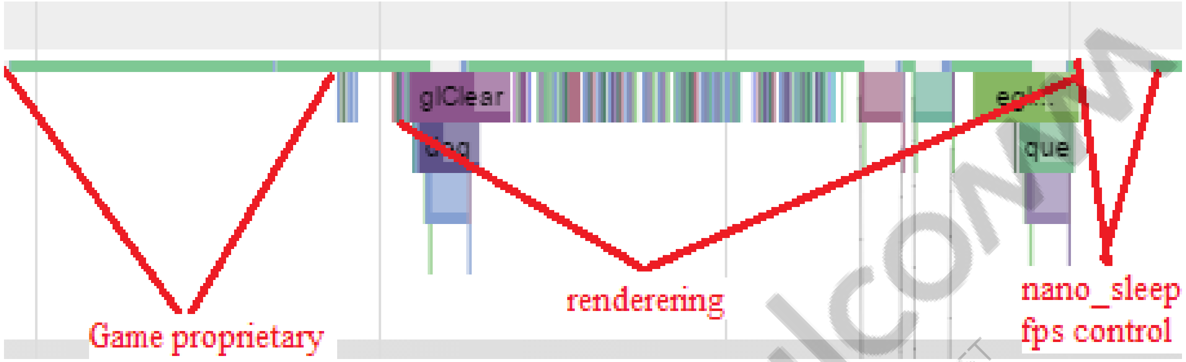


Understanding the work mode of game



- Most Game renderer its content via SurfaceView
  - 1)Game proprietary code: read data from disk/network,…
  - 2)dequeuer buffer and draw
  - 3)queue buffer
  - 4)sleep to control to the fps (only happen with rendering too fast)
- Game is very sensitive to cpu execution time
- 1~2ms unexpected delay will cause a frame drop.

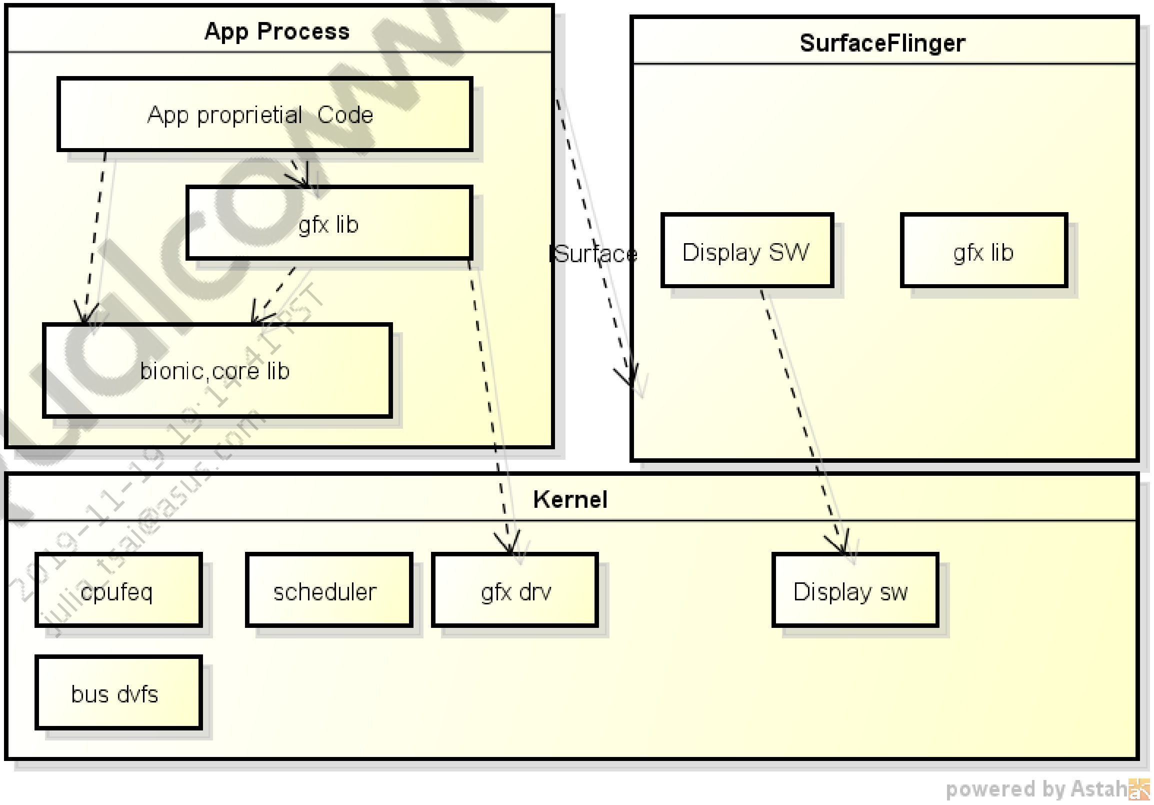
Understanding the work mode of game(cont.)



- Game have it’s own fps control, unexpected delay will introduce frame drop
  - NR 162 is syscall of nano_sleep , used to sleep for desired time
  - UnityMain-6837 (6798) [000] ...1 14592.364816: sys_enter: NR 162 (dfb5e418, 0, fffff720, de6bc374, de73fde8, 0)
  - UnityMain-6837 (6798) [000] d..3 14592.364827: sched_enq_deq_task: cpu=0 dequeue comm=UnityMain pid=6837 prio=120 nr_running=0 cpu_load=60 rt_nr_running=0 affine=f demand=14073829
  - UnityMain-6837 (6798) [000] d..3 14592.364830: sched_switch: prev_comm=UnityMain prev_pid=6837 prev_prio=120 prev_state=S ==> next_comm=swapper/0 next_pid=0 next_prio=120
  - UnityMain-6837 (6798) [000] ...1 14592.365840: sys_exit: NR 162 = 0

Components that impact game perf

- HW
  - cpu/gpu capability
  - ddr/storage/bus
- SW
  - App proprietial lib
  - game engine
  - gfx lib, bionic
  - display SW
  - Scheduler/cpufreq/
  - bw dcvs
  - Lpm
  - thermal



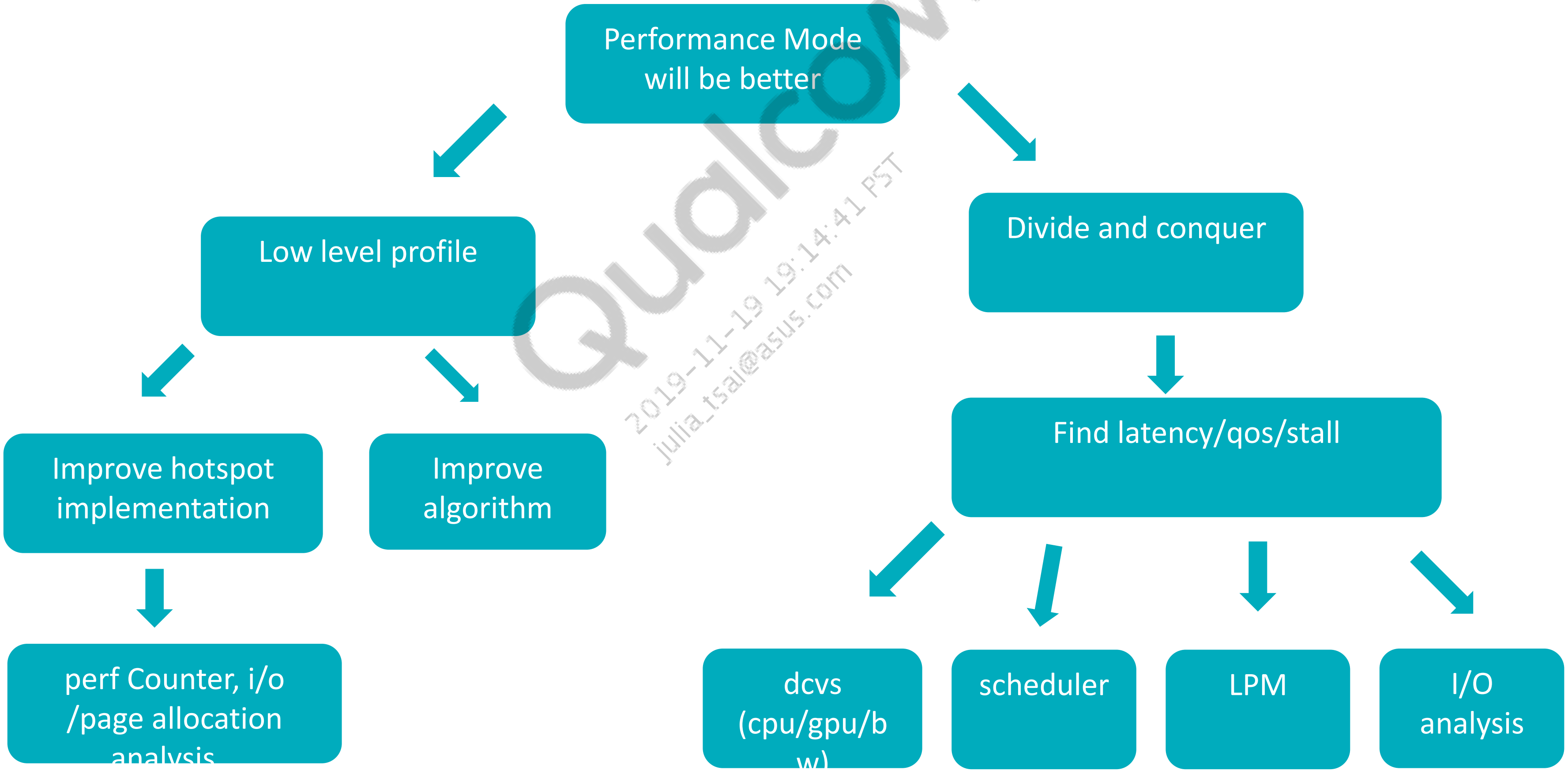
Difficulty

- Game performance issues are hard to debug
  - Many factors impact Game performance
  - No Source code available
  - Game design varies one from one
- Need to balance performance and power
- Hot games keeps changing

Solution

- Try to understand the game logic from different Level
  - opengGL/framework Api level
  - Core Library
  - System call
  - Kernel driver level
- Profile with different tool from different perspective
  - Traceview
  - gfxTracer
  - ftrace: framework/syscall/gfx/display/block
  - atrace
- Log analysis
  - Gles call sequence
  - Shader program
  - syscall
  - Mem allocation, gpu rendering context metrics

Approach


Performance Mode will be better
Low level profile
Divide and conquer
Improve hotspot implementation
Improve algorithm
Find latency/qos/stall
perf Counter, i/o /page allocation analysis
dcvs (cpu/gpu/b w)
scheduler
LPM
I/O analysis

Find direction

- Game design issue/hw capability
  - Performance mode to examine
- Cpufreq
  - Workload analysis: put into fix cpufreq to verify
- Scheduler
  - SMP: online 1/2/3/4 cpu to find difference. Task_pack/affinity/load_balance
  - HMP: energy aware/power table/CFS tuning
- Gfx lib
  - Draw call statistics
  - Texture Upload/shader compiling
  - Rendering quality control
- LPM
  - Power mode residence statistics

Game design issue/hw capability

- Performance mode to examine(refer to tuning guide for different target)
  - #disable perflock and thermal mitigation

    ```
    adb shell stop thermal-engine
    adb shell stop perf-hal-1-0
    adb shell stop perfd
    ```

  - #disable low power model

    ```
    adb shell "echo Y > /sys/module/lpm_levels/parameters/sleep_disabled“
    ```

  - #online all cpus

    ```
    adb shell "echo 4 > /sys/devices/system/cpu/cpu0/core_ctl/min_cpus“
    adb shell "echo 4 > /sys/devices/system/cpu/cpu0/core_ctl/max_cpus“
    adb shell "echo 4 > /sys/devices/system/cpu/cpu4/core_ctl/min_cpus"
    adb shell "echo 4 > /sys/devices/system/cpu/cpu4/core_ctl/min_cpus"
    ```

  - #cpufreq performance

    ```
    adb shell "echo performance > /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor"
    ```

  - #gpu performance

    ```
    adb shell "echo 0 > /sys/class/kgsl/kgsl-3d0/min_pwrlevel“
    adb shell "echo 0 > /sys/class/kgsl/kgsl-3d0/max_pwrlevel“
    ```

  - #ddr performance

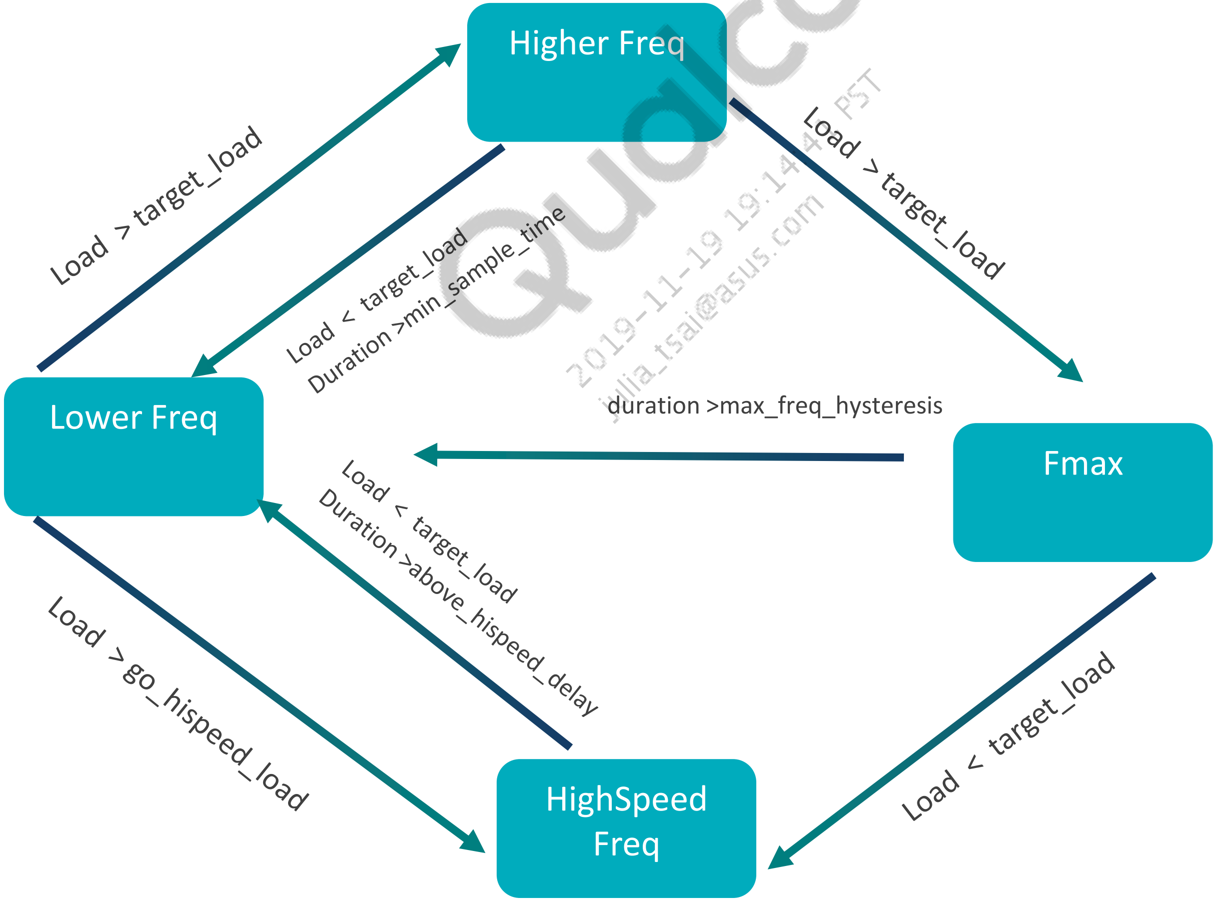
    ```
    adb shell "echo performance >/sys/class/devfreq/*/governor”
    ```

CpuFreq governor

- Governor: scaling cpu freq with load
  - history Load is used to predict
  - Cpufreq is update either by timer(20ms) or notified by scheduler
- Fix cpufreq to identify governor issue
  - adb shell "echo xx > sys/devices/system/cpu/cpu0/cpufreq/scaling_minfreq”
  - adb shell "echo xx > sys/devices/system/cpu/cpu0/cpufreq/scaling_maxfreq”
  - …
- Interactive governor(kernel-4.4 and previous)
  - target_load, high_speed_load
  - min_sample_time
- Sched_util(kernel 4.9 and above)
  - highfreq_load

CpuFreq governor

- Interactive governor(kernel-4.4 and previous)
  - target_load, high_speed_load
  - min_sample_time
- Sched_util(kernel 4.9 and above)
  - highfreq_load



Higher Freq

Lower Freq

Fmax

HighSpeed Freq

Load > target_load

Load < target_load
Duration >min_sample_time

Load > target_load

duration >max_freq_hysteresis

Load < target_load
Duration >above_hispeed_delay

Load > go_hispeed_load

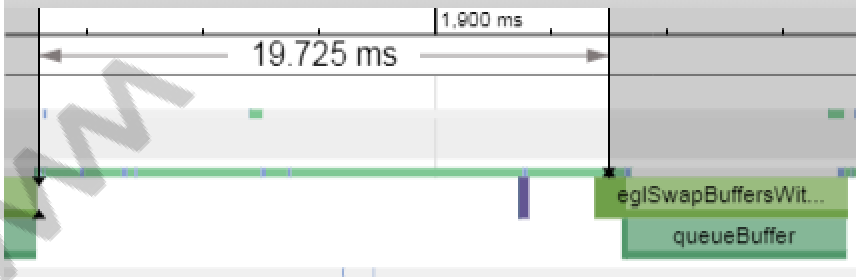
Load < target_load

Scheduler

- Scheduler
  - There is one run queue for each cpu:
    - Normal task(100~139): CFS base on priority
    - RT task(<100): priority based
  - Task placement goal: save power as much as possible, minimize hurting performance
    - Big task -> eligible to perf cpu : task demand higher than sched_upmigrate
    - Small task -> eligible to power cpu: task demand lower than sched_downmigrate
    - Task packing: put new eligible running task in to run queue of cpu to avoid wake up cpus
    - Load balance: keep load on each cpu is “balanced”
    - Affinity: task is pinned to certain cpu
    - Sched_boost: force move task to perf cpu to improve performance
  - Load reporting:
    - Report the current load, let cpufreq governor to decide next cpufreq

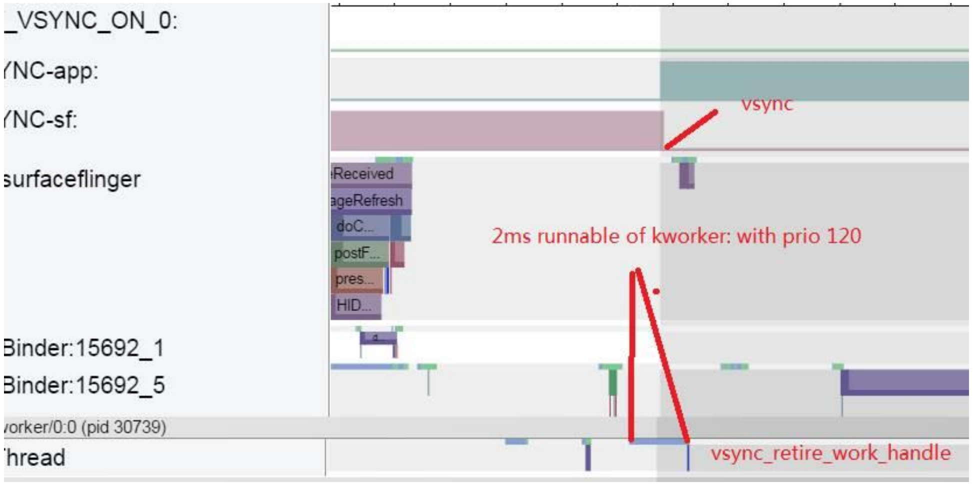
Scheduler latency check steps

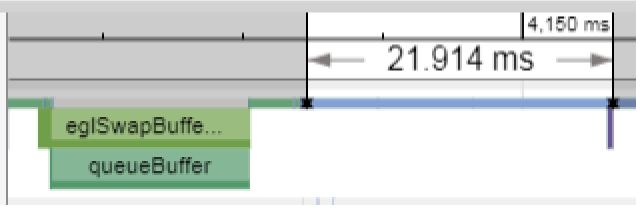
- Long **Running**: green bar in systrace
  - In which cpu the interested thread is running
  - What is the cpufreq
  - What's the upmigrate/down migrate configs



```
sched_enq_deq_task: cpu=0 dequeue comm=rcu_sched pid=8 prio=120 nr_running=0 cpu_load=0 rt_nr_running=0 affine=ff
sched_update_history: 2876 (sensors.qcom): runtime 98674 samples 1 event TASK_WAKE demand 102223 (hist: 98674 112007 95990 0 0)
cpu 5 nr_big 0 nr_small 2
```

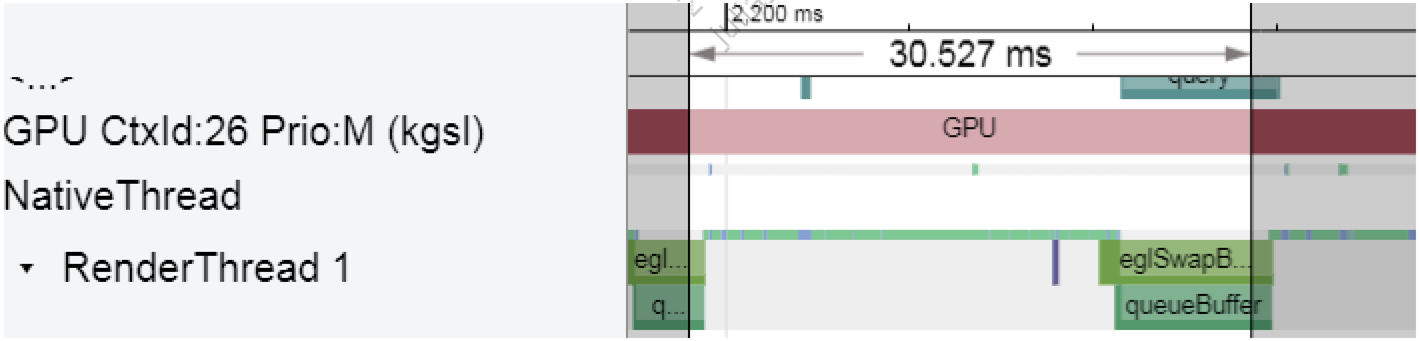
- Long **Runnable**: blue bar in systrace
  - How many task in the cpu run queue
  - What is the priority of interested thread
  - What's the affinity of interested thread





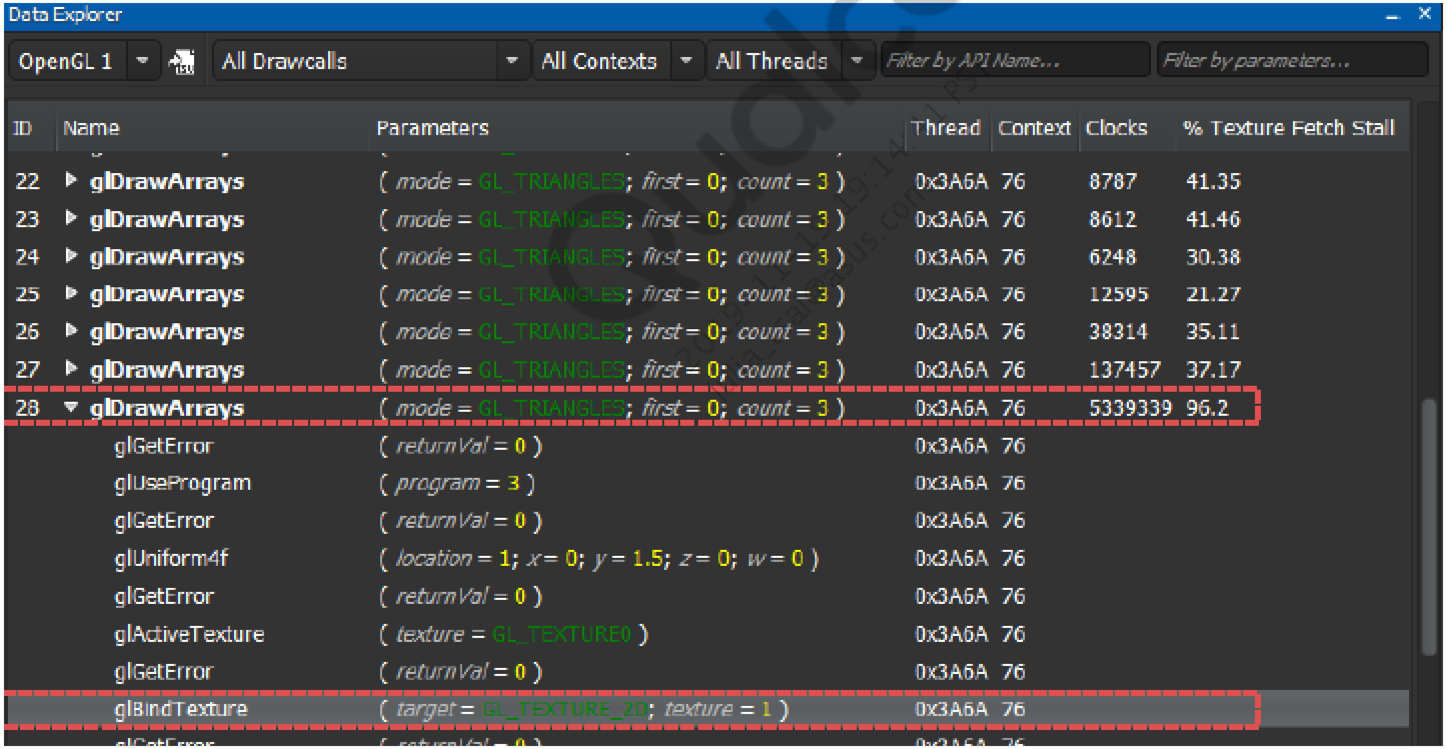
Gfx lib

- EGL Draw call:
  - Is there any long eglXX function call or frequently call ?
  - Pre android-N: settings -> dev options -> enable elg draw call
  - After android-N: contact qcom for patch on difference android version
- Per context rendering time checking
  - Understand how many rendering task in the system, time cost of each one
  - Refer to KBA-160606233316
- Gpu clock and power state checking:



Gfx lib

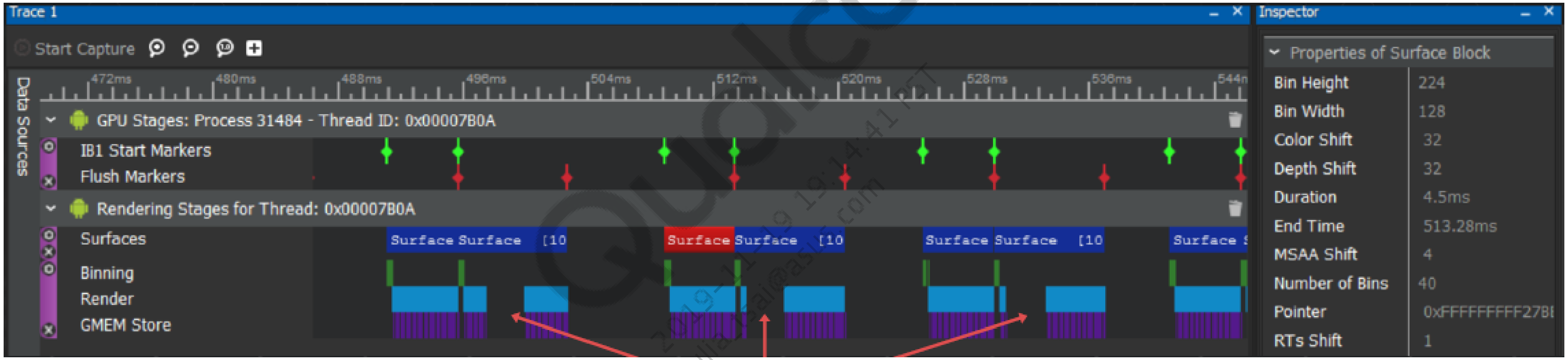
- Draw call analysis:
  - Follow up KBA to capture dcap for QCOM to analyze KBA-170116005123
- Low level profiling with SnapdragonProfiler
  - cycles for each drawCall



ID	Name	Parameters	Thread	Context	Clocks	% Texture Fetch Stall
22	▶ glDrawArrays	(*mode* = GL_TRIANGLES; *first* = 0; *count* = 3)	0x3A6A	76	8787	41.35
23	▶ glDrawArrays	(*mode* = GL_TRIANGLES; *first* = 0; *count* = 3)	0x3A6A	76	8612	41.46
24	▶ glDrawArrays	(*mode* = GL_TRIANGLES; *first* = 0; *count* = 3)	0x3A6A	76	6248	30.38
25	▶ glDrawArrays	(*mode* = GL_TRIANGLES; *first* = 0; *count* = 3)	0x3A6A	76	12595	21.27
26	▶ glDrawArrays	(*mode* = GL_TRIANGLES; *first* = 0; *count* = 3)	0x3A6A	76	38314	35.11
27	▶ glDrawArrays	(*mode* = GL_TRIANGLES; *first* = 0; *count* = 3)	0x3A6A	76	137457	37.17
28	▼ glDrawArrays	(*mode* = GL_TRIANGLES; *first* = 0; *count* = 3)	0x3A6A	76	5339339	96.2
	glGetError	(*returnVal* = 0)	0x3A6A	76		
	glUseProgram	(*program* = 3)	0x3A6A	76		
	glGetError	(*returnVal* = 0)	0x3A6A	76		
	glUniform4f	(*location* = 1; *x* = 0; *y* = 1.5; *z* = 0; *w* = 0)	0x3A6A	76		
	glGetError	(*returnVal* = 0)	0x3A6A	76		
	glActiveTexture	(*texture* = GL_TEXTURE0)	0x3A6A	76		
	glGetError	(*returnVal* = 0)	0x3A6A	76		
	glBindTexture	(*target* = GL_TEXTURE_2D; *texture* = 1)	0x3A6A	76		
	glGetError	(*returnVal* = 0)	0x3A6A	76		

Gfx lib

- Low level profiling with SnapdragonProfiler
  - Surface Drawing, binning,
  - Gmem store

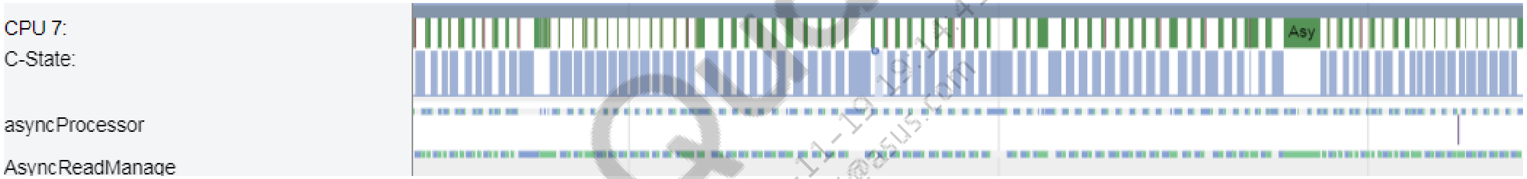


Properties of Surface Block

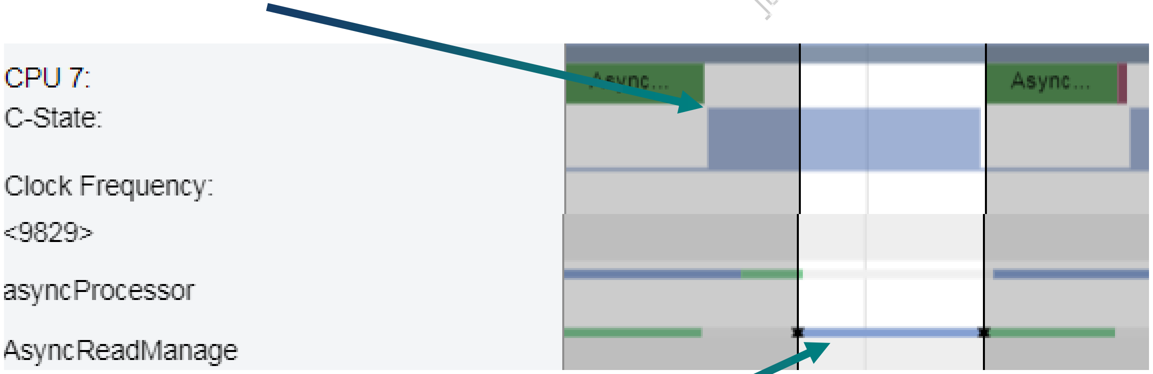
Bin Height	224
Bin Width	128
Color Shift	32
Depth Shift	32
Duration	4.5ms
End Time	513.28ms
MSAA Shift	4
Number of Bins	40
Pointer	0xFFFFFFFFF27BI
RTs Shift	1

Low power mode(LPM)

- LPM may impact scheduling latency
  - The deeper sleep state, the longer wakeup latency
  - Short burst, periodically task is easy to be impacted: I/O, network, etc
- Rule out:
  - adb shell "echo Y > /sys/module/lpm_levels/parameters/sleep_disabled“

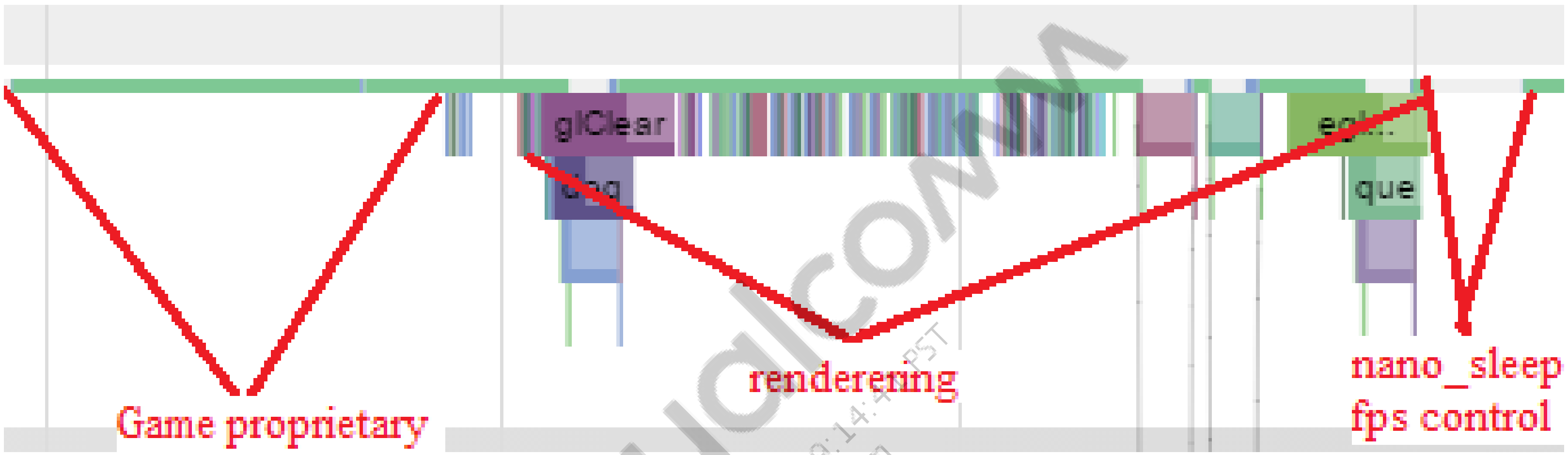


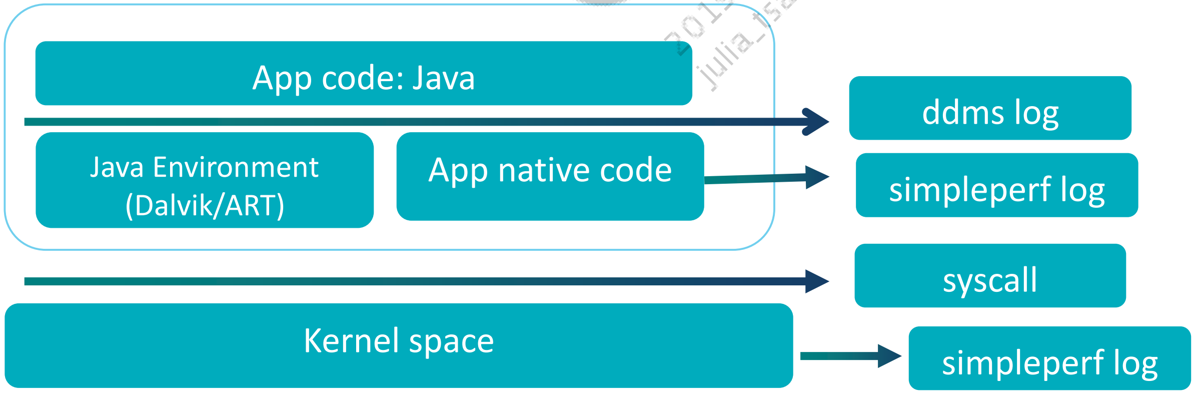
Cpu in State4 (PC4), wakeup latency is longer than execution time!



wakeup latency, displayed as “runnable”

Profiling long running: green bar in systrace





App code: Java

Java Environment (Dalvik/ART)

App native code

Kernel space

ddms log

simpleperf log

syscall

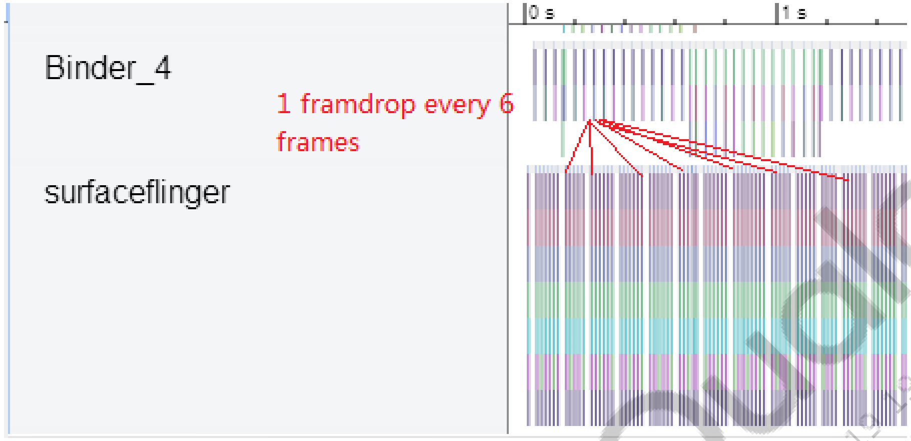
simpleperf log

Case study: Tiantianfeiche fps drop:

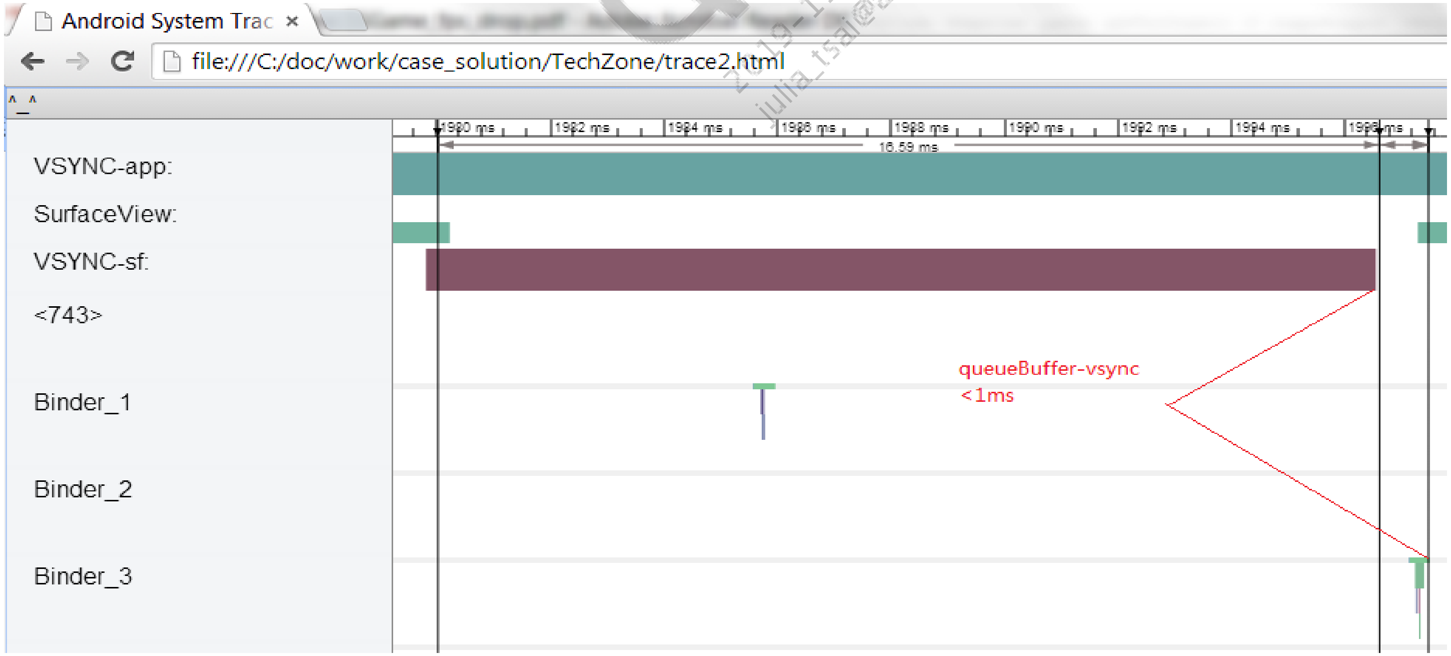
- Issue description: Game is janky
- Triage Steps:
  - frame drop confirmation, behavior analysis
  - put into performance mode: game is smooth
  - rule out: thermal/gpu/display sw/bus dcvs
  - Suspicious: LPM and cpufreq

Case study: Tiantianfeiche fps drop:

- confirmation, behavior analysis



1 framdrop every 6 frames



Android System Trac

file:///C:/doc/work/case_solution/TechZone/trace2.html

queueBuffer-vsync
<1ms

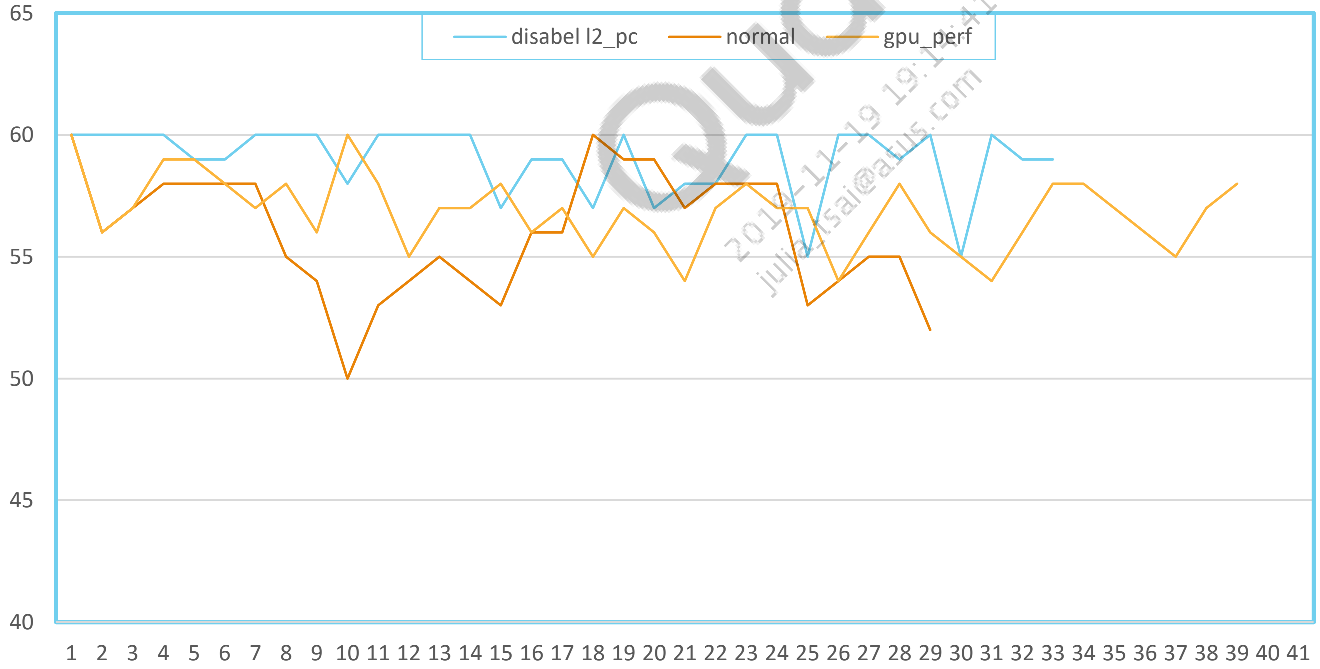
Case study: Tiantianfeiche fps drop:

- put into perf mode, 59fps

```
adb shell stop perfd
adb shell stop thermal-engine
adb shell "echo Y > /sys/module/lpm_levels/parameters/sleep_disabled"
adb shell "echo 0 > /sys/module/msm_thermal/parameters/enabled“
adb shell "echo performance >/sys/class/devfreq/X/governor“
adb shell "echo performance > sys/devices/system/cpu/cpuX/cpufreq/scaling_governor"
```

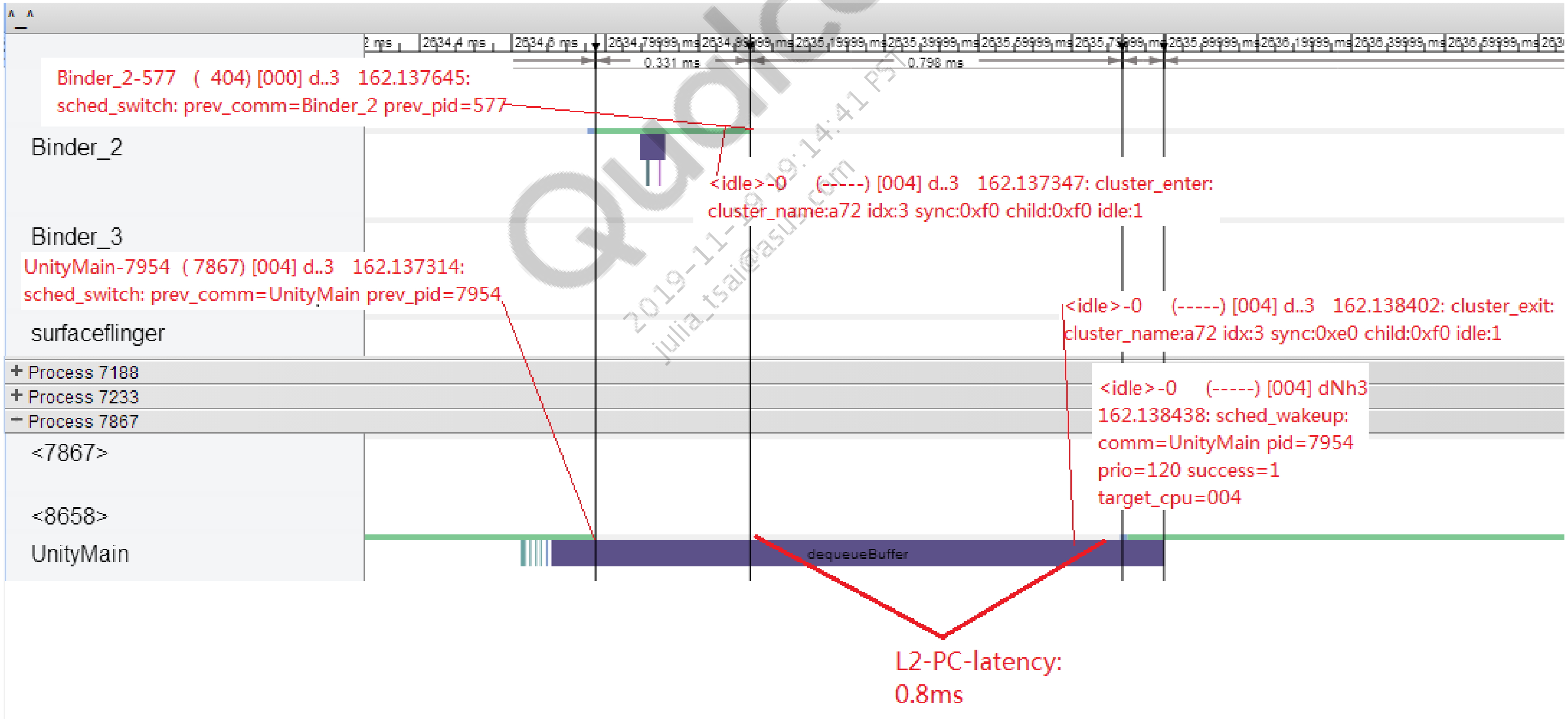
Case study: Tiantianfeiche fps drop:

- Rule out, find that cpufreq and lpm are suspicious
  - Load analysis show that cpufreq works well: very low latency in this case
- lpm
  - disable cpu /cluster/cci low power mode separately to check
  - confirm L2pc has great impact



Case study: Tiantianfeiche fps drop

- Find out latency
  - If queue/dequeuer Buffer is done 1ms earlier, jank will be eliminated
  - Examine the game thread, wake up latency is high in each binder call



```
Binder_2-577  ( 404) [000] d..3  162.137645:
sched_switch: prev_comm=Binder_2 prev_pid=577
```

```
<idle>-0    (-----) [004] d..3   162.137347: cluster_enter:
cluster_name:a72 idx:3 sync:0xf0 child:0xf0 idle:1
```

```
UnityMain-7954  ( 7867) [004] d..3   162.137314:
sched_switch: prev_comm=UnityMain prev_pid=7954
```

```
<idle>-0    (-----) [004] d..3   162.138402: cluster_exit:
cluster_name:a72 idx:3 sync:0xe0 child:0xf0 idle:1
```

```
<idle>-0    (-----) [004] dNh3
162.138438: sched_wakeup:
comm=UnityMain pid=7954
prio=120 success=1
target_cpu=004
```

L2-PC-latency:
0.8ms

Case study: Tiantianfeiche fps drop

- Latency Elimination
  - A72 wake up latency is 0.899ms
  - In this case, L2-pc residence time is very short
- Potential solution
  - disable l2-pc in this case: hurt power, hard to detect the scene
  - Tune pm-qos: little power impact with careful design
  - Make cpufreq aggressive to hide the latency: hurt power in all scene

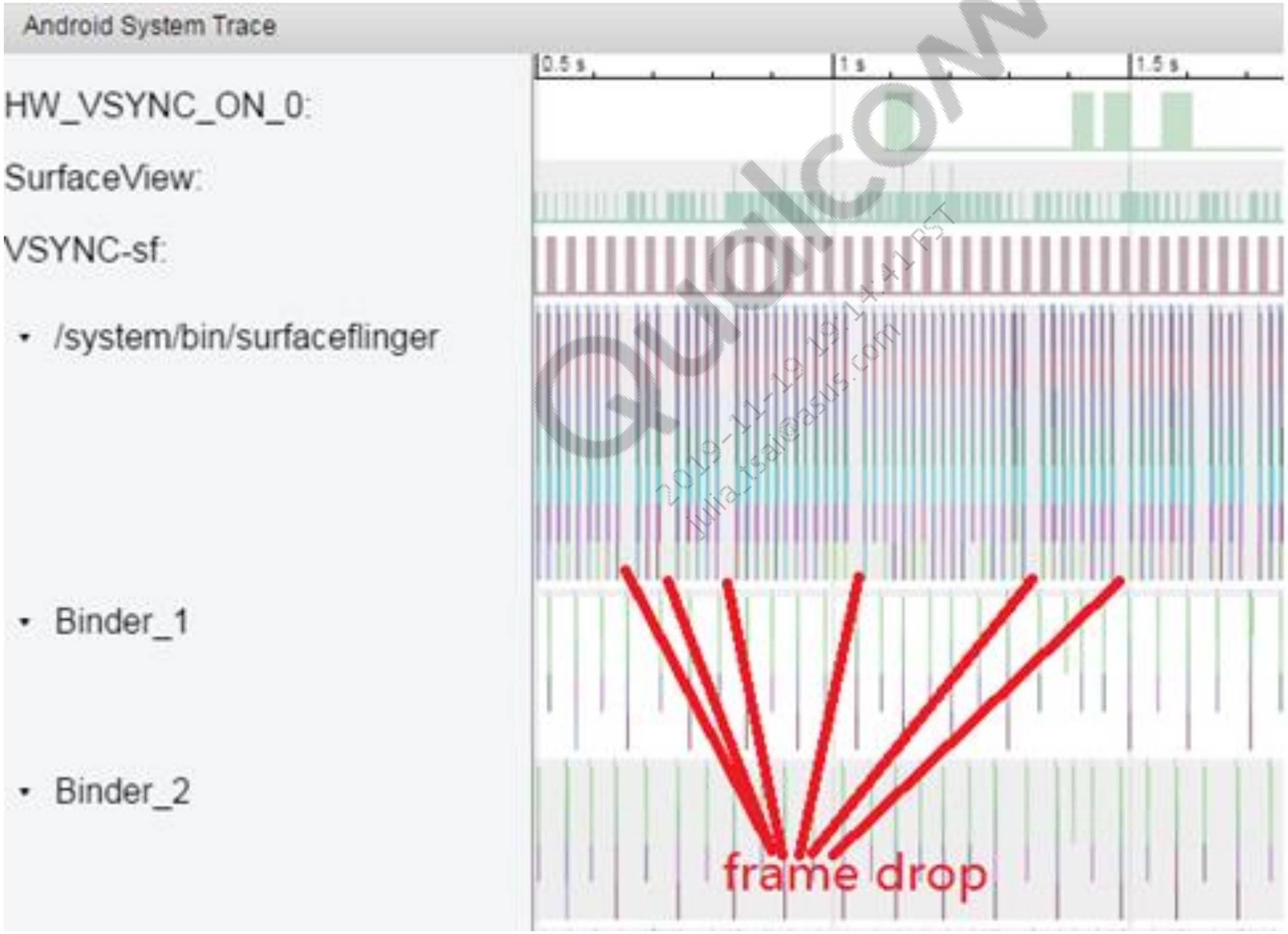
```
adb shell cat /d/lpm_stats/a72/stats
[a72] a72-l2-pc:
  success count:    7774
  total success time: 15.738975002
          <    0.000062500:       0 (0-0)
          <    0.000250000:     479 (87500-245938)
          <    0.001000000:    1729 (260781-999948)
          <    0.004000000:    5434 (1000052-3858907)
```

Case study: Templerun fps drop

- Issue description: Game is janky in 8996 devices
- Triage Steps:
  - frame drop confirmation, behavior analysis
  - put into performance mode: game is smooth
  - rule out: thermal/gpu/display sw/bus dcvs/lpm
  - Suspicious: cpufreq and scheduler

Case study: Templerun fps drop

- Issue description: Game is janky in 8996 devices

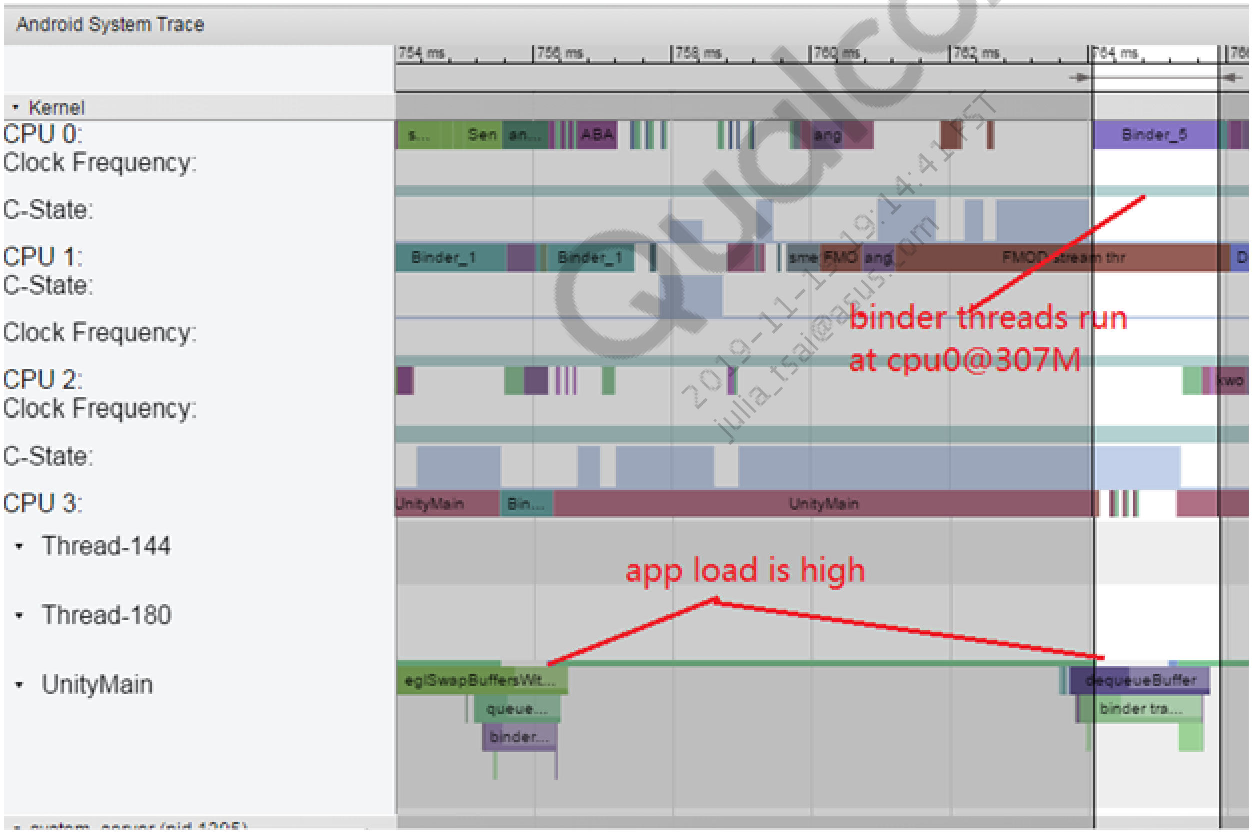


Case study: Templerun fps drop

- Find the delay
  - Binder call time is long
  - 5 binder calls in each game frame
- Check cpufreq
  - Load on cpu0/1 and cpu2/3 is stable
  - Set minfreq of cpu0 to 1000M, 59 fps
- check HMP scheduler:
  - Offline power cluster(cpu0/1): 59 fps
  - Offline perf cluster(cpu2/3): 59fps

Case study: Templerun fps drop

- Binder call time is long: bottleneck is callee
  - Caller(client) runs on cpu2 with cpufreq 729M
  - Callee(stub) run on cpu0 with cpufreq 307M



binder threads run at cpu0@307M

app load is high

Case study: Templerun fps drop

- Final solution
  - put callee task on caller’s cpu in sync binder call
  - Change the select_best_cpu logic of scheduler: 59fps

Contents

- Typical issue and cause
- Metrics and test procedure
- Debugging steps
- Analysis tool and Log capturing
- Report issue
- Questions?

Analysis tool and Log capturing

- Systrace:
  - Online/offline: refer to KBA-160803171155
  - Async logs: useful for capture logs for some random issue, refer to KBA-160803171155
  - Extract ftrace from Ram dump: KBA-180331192102
- Simple perf :
  - Debug long running in native space: KBA-170821011435
- TraceView
  - Understanding call stack, long running in Java layer: KBA-170810015921
- Syscall inspection:
  - adb shell "echo 1 > /sys/kernel/debug/tracing/events/raw_syscalls/enable“
  - Capture systrace as normal

Tuning document

- scheduler
  - 80-P9301-86 (kernel 4.9 and above): ENERGY AWARE SCHEDULING AND SCHEDUTIL
  - 80-P2484-47(kernel 4.4 and previous): MSM8998 LINUX SCHEDULER OVERVIEW
  - KBA-170606232259: New tunable on SDM660 MSM8998
- governor :
  - 80-NR256-3(kernel 4.4 and previous) : APPLICATION NOTE: CPU GOVERNOR
  -

Report issue

- File case to Qcom with bellow Problem area::
  - Detail description of app version, game config, replication steps
  - Replication video, or playback files
  - Experiment for “perf mode”:
  - Corresponding logs

Graphics:	
PA1	Multimedia
PA2	Adreno(Graphics)
PA3	Game Performance(GPU)
Perf:	
PA1	BSP/HLOS
PA2	Performance
PA3	Game performance
Power:	
PA1	BSP/HLOS
PA2	Power/Thermal(BSP/HLOS)
PA3	Power-Game

Thank You!

For more information on Qualcomm, visit us at:
www.qualcomm.com & www.qualcomm.com/blog