



第三届 eBPF开发者大会

www.ebpftravel.com

内存可观测：eBPF技术在Android系统内存剖析及调优实践

荣耀终端 | OS Kernel Lab

伊鹏翔 林琨力 王鑫

中国·西安



第三届 eBPF 开发者大会

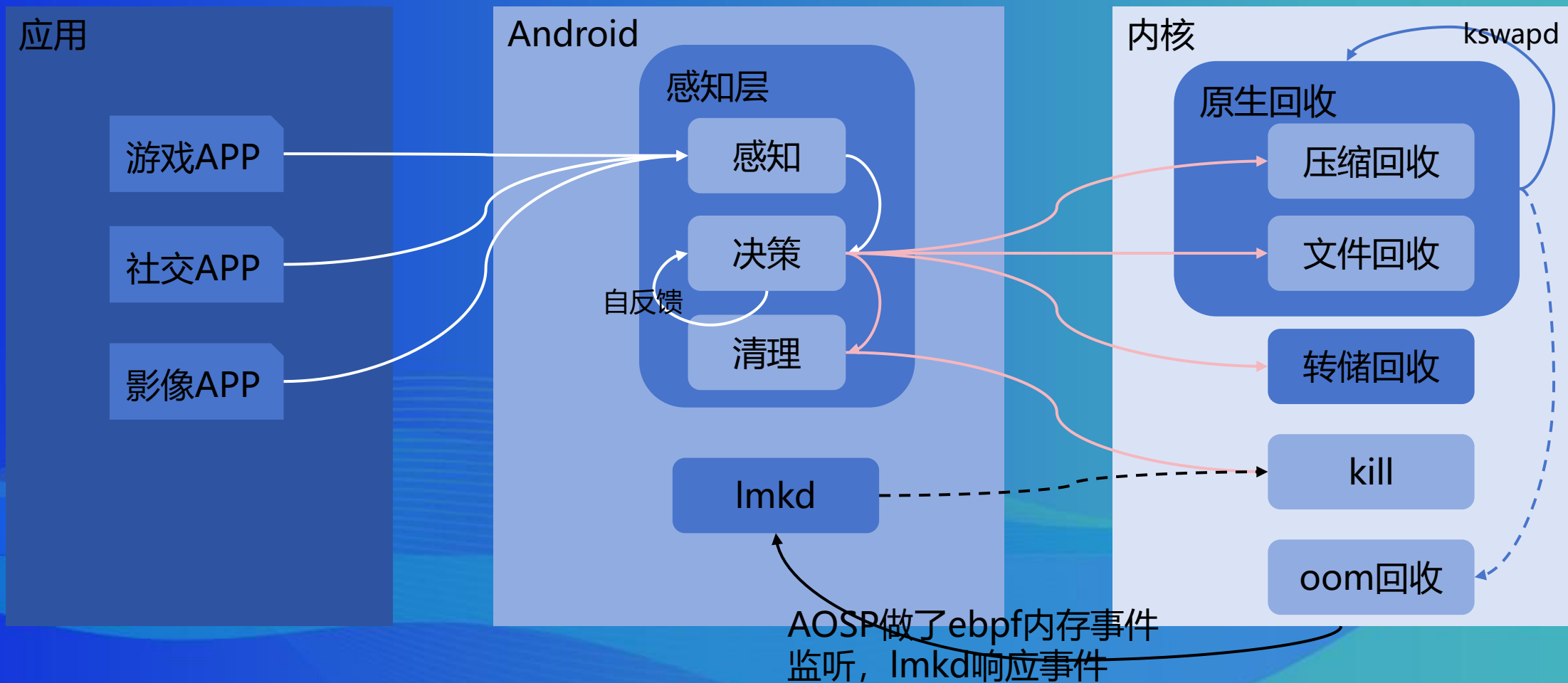
www.ebpftravel.com

作者介绍

- 伊鹏翔：荣耀内核开发专家（内存方向），负责内存方案开发及演进，主要在内存及性能方面工作。
- 林琨力：荣耀内核工程师，中科院博士，主导开发多个内存优化方案如：dmabuf压缩等方案。
- 王鑫：荣耀内核工程师，负责内存性能优化工作。

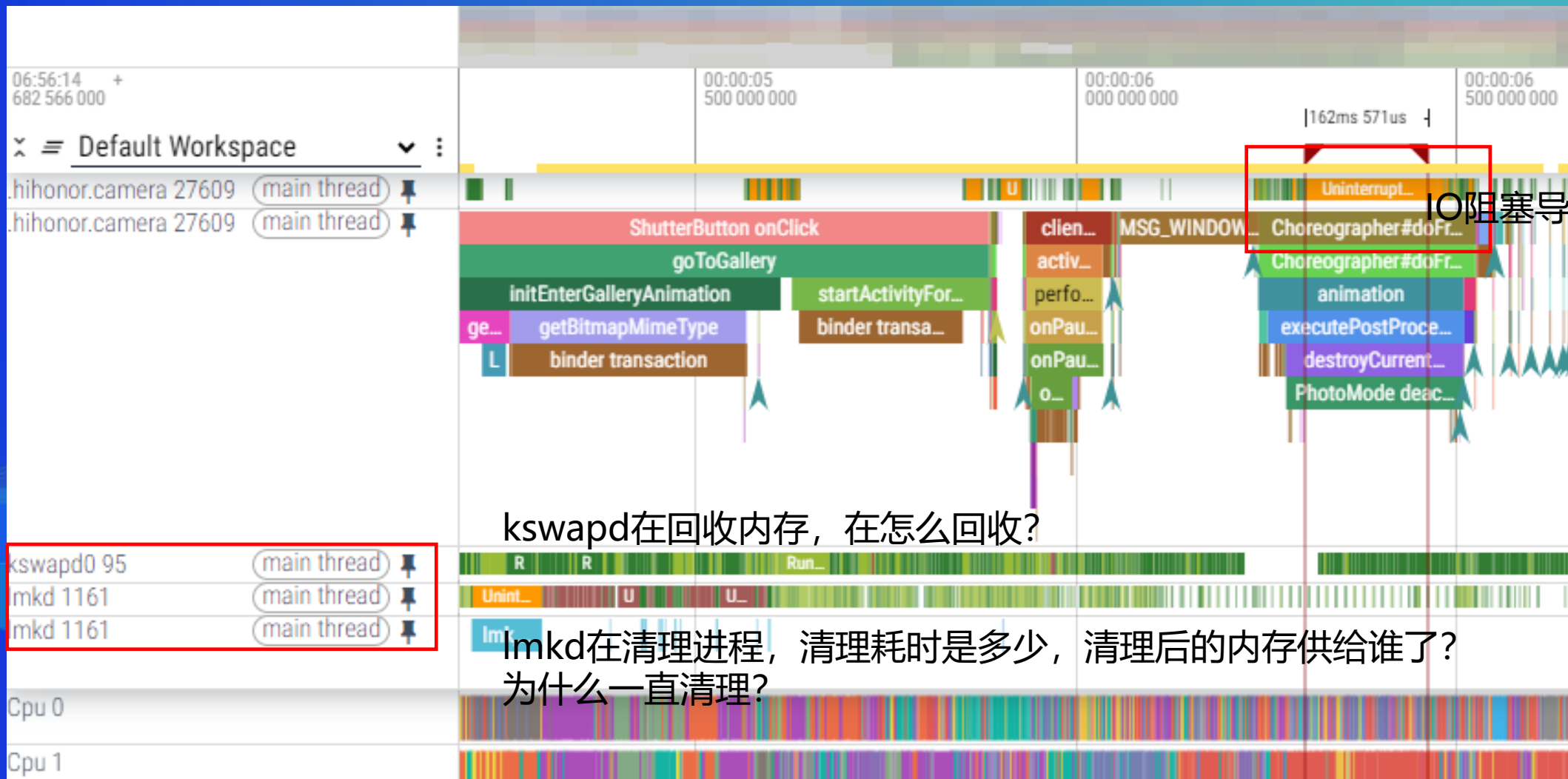
中国·西安

终端设备内存性能现状



内存管理基于事件触发，触发后决策清理/回收/转储等策略。而性能事件产生后，我们只能看到结果，无法评估策略的合理性。

问题：性能问题发生后只有内存事件，怎么性能调优？



性能问题采集到离散内存数据

dumpsys meminfo

```
289 Total PSS by OOM adjustment:
290 2,136,268K: Native (944,883K in swap)
291 691,003K: vendor.honor.hardware.camera.dme-V1-ndk-service (pid.4255) (10,304K in swap)
292 391,082K: vendor.qti.camera.provider-service_64 (pid.1994) (265,540K in swap)
293 302,895K: surfaceflinger (pid.2044) (41,428K in swap)
294 63,528K: aptouch_daemon (pid.2197) (52,376K in swap)
295 28,451K: vendor.qti.hardware.display.composer-service (pid.1997) (17,764K in swap)
296 24,890K: dubaid (pid.2653) (15,120K in swap)
297 24,719K: media.hwcodec (pid.2005) (23,852K in swap)
298 24,243K: android.hardware.bluetooth@aidl-service-qti (pid.1898) (23,552K in swap)
299 20,965K: audioservice.qti (pid.1911) (20,940K in swap)
300 19,569K: audioserver (pid.2030) (17,744K in swap)
301 18,473K: mediaserver (pid.2665) (15,652K in swap)
302 16,566K: vendor.honor.hardware.aoservice-service (pid.1927) (16,328K in swap)
303 15,363K: qcrilNrd (pid.2616) (12,272K in swap)
304 15,076K: hiview (pid.4770) (8,500K in swap)
305 14,327K: hiview (pid.2834) (5,652K in swap)
306 12,495K: vendor.honor.hardware.biometrics.hwfacerecognize.aidl-service (pid.2628) (12,148K in
```

/proc/vmstat

```
037 Number of blocks type Unmovable Movable Reclaimable CMA HighAtomic Isolate
038 Node 0, zone Normal 1563 1022 35 260 4 0
039 nr_free_pages 104887
040 nr_zone_inactive_anon 207291
041 nr_zone_active_anon 151740
042 nr_zone_inactive_file 269286
043 nr_zone_active_file 117924
044 nr_zone_unevictable 24026
045 nr_zone_write_pending 3077
046 nr_mlock 23682
047 nr_bounce 0
048 nr_zspages 404668
049 nr_free_cma 4034
050 nr_inactive_anon 207291
051 nr_active_anon 151740
052 nr_inactive_file 269286
053 nr_active_file 117924
054 nr_unevictable 24026
055 nr_slab_reclaimable 64892
056 nr_slab_unreclaimable 179828
```

/proc/meminfo

```
607 MemTotal: 11477792 kB
608 MemFree: 421112 kB
609 MemAvailable: 3871004 kB
610 Buffers: 1628 kB
611 Cached: 1682556 kB
612 SwapCached: 12388 kB
613 Active: 1077428 kB
614 Inactive: 1906328 kB
615 Active(anon): 606168 kB
616 Inactive(anon): 829296 kB
617 Active(file): 471260 kB
618 Inactive(file): 1077032 kB
619 Unevictable: 96104 kB
620 Mlocked: 94728 kB
621 SwapTotal: 12582908 kB
622 SwapFree: 7388212 kB
623 Dirty: 12128 kB
624 Writeback: 0 kB
625 AnonPages: 1393760 kB
626 Mapped: 714456 kB
627 Shmem: 43020 kB
628 KReclaimable: 2230944 kB
629 Slab: 978864 kB
630 SReclaimable: 259520 kB
631 SUnreclaim: 719344 kB
632 KernelStack: 77056 kB
633 ShadowCallStack: 0 kB
634 PageTables: 239560 kB
635 SecPageTables: 0 kB
636 NFS_Unstable: 0 kB
637 Bounce: 0 kB
638 WritebackTmp: 0 kB
639 CommitLimit: 18321804 kB
640 Committed_AS: 172949784 kB
```

在性能故障时，如何将离散的内存数据变成连续数据，优化内存管理策略？

mmtrace主要工作

mmtrace实现了低开销的内存资源实时采集方案并能关联性能故障，极大方便了定位内存供给/回收机制，系统低内存高IO等问题。

【主要内容】

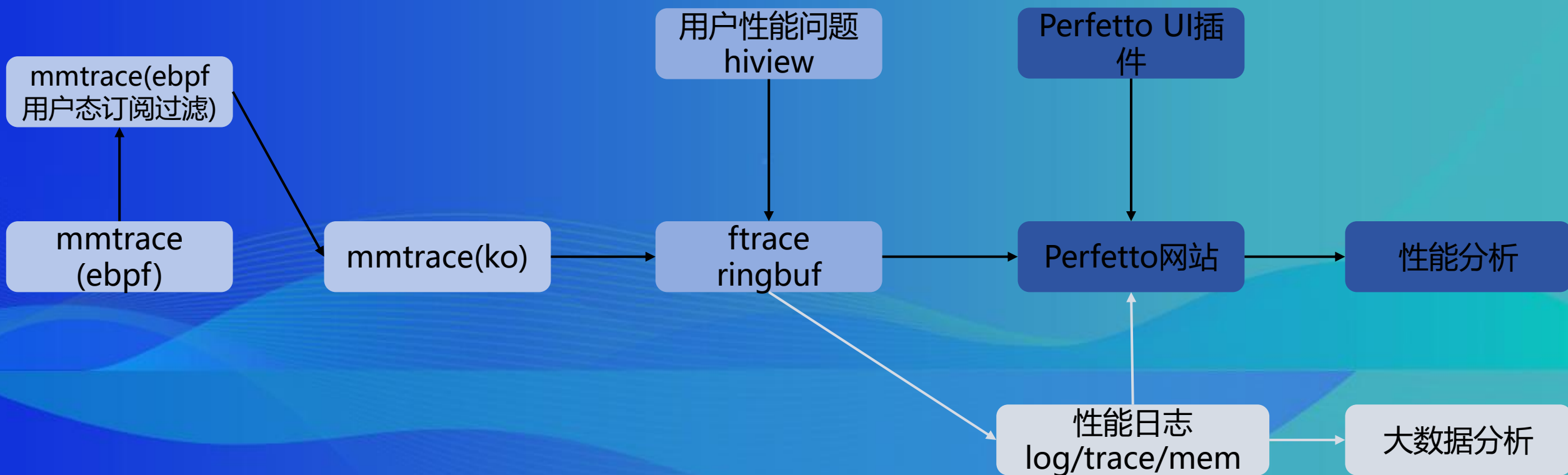
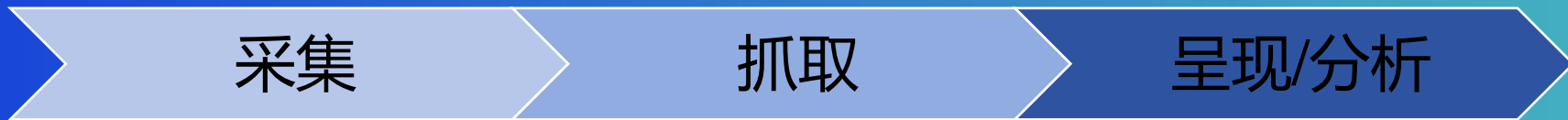
在本次分享中，我们将重点介绍：

- 1.通过eBPF等技术实现进程内存，整机内存，内存事件的抓取及聚合，单次采集小于1KB，大幅减少了采集量。
- 2.如何实现与性能故障时序关联，可视化内存资源与性能关系。
- 3.通过实际案例展示如何通过本项目对内存回收策略优化，实现性能调优。

【实践成效】

1. 事件触发，故障数据快速关联
2. 开销低，能部署到Android用户环境，实现规模商用

mmtrace设计思路



mmtrace采集：低开销连续采集

阈值（如：时间窗内超过x次、内存压力上报等）

类型（如：vip线程、render线程、cameraprovider等）

重点名单（如：systemui、launcher等）

内存tracepoint:

mm_vmscan_direct_reclaim_begin

mm_vmscan_direct_reclaim_end

mm_vmscan_kswapd_wake

kprobe & kretprobe:

dma_heap_buffer_alloc

kgsl_allocate_user

_alloc_pages 等

mmtrace
(ebpf)

mmtrace(ebpf
用户态订阅过滤)

ko模块连续读取内核信息:

系统状态/关键进程状态/memcg状态

file/anon/dmabuf/kgsl/mlock等

读取时机: 1.定时读取 2.ebpf触发

3.vendor_hook事件 4.主动触发

mmtrace(ko)

示例:

监测事件: 用户使用某应用,
产生了多次direct_reclaim

流量控制: 应用的vip线程达到了
x次direct_reclaim, 往mmtrace
写入事件

信息采集: mmtrace采集预先设
定内存信息, 如果多次发生会采
集多次, 也有定时采集。

mmtrace抓取：用户性能事件触发

用户性能问题
hiview

mmtrace(ko)

ftrace ringbuf

25 > ebpf > 内存观测 > 拍照查看缩略图慢 > logCached

名称	修改日期
20250317-230223-440_jankexce.log	2025/4/3 10:56
20250317-230230-492_activity.log	2025/4/3 10:56
20250317-230230-564_broadcastsmini.log	2025/4/3 10:56
20250317-230230-613_surfaceflinger.log	2025/4/3 10:56
20250317-230230-652_dmabuf.log	2025/4/3 10:56
20250317-230232-404_dumpsysprocmem.log	2025/4/3 10:56
trace.systtrace	2025/4/3 17:33

```
...kworker/u16:3-1264... (.1264) . [000] ... 160.988526: sched_switch: prev_comm=kworker/u16:3 prev_pid=1264 prev_prio=120 prev_state=D ==> next_co
...kworker/4:3-1018... (.1018) . [004] ... 160.988551: tracing_mark_write: C|1018|type: 0, snapshot_mark: 159804087282, ts_snapshot:15980
6291455, swapfree: 6291455, vmalloc: 102935, dmabuf_used: 99374, dmabuf_cached: 0, dmabuf_resv: 0, kgs1_used:
0, inactive_anon: 0, active_anon: 864985, active_file: 1463585, inactive_file: 2045262, unevictable: 23254, slab_reclaimable:
0, anon_mapped: 859218, file_mapped: 445803, file_pages: 3537824, file_dirty: 1197, writeback: 0, writeback_temp: 0,
0, kernel_misc_reclaimable: 223116, kernel_stack_kb: 61516
|1
...kworker/0:5-1021... (.1021) . [000] ... 160.988589: sched_switch: prev_comm=kworker/0:5 prev_pid=1021 prev_prio=120 prev_state=I ==> next_comm
...kworker/1:14-1514... (.1514) . [001] ... 160.988604: sched_switch: prev_comm=kworker/1:14 prev_pid=1514 prev_prio=120 prev_state=I ==> next_com
...kworker/4:3-1018... (.1018) . [004] ... 160.988604: tracing_mark_write: C|1018|type: 1, snapshot_mark: 159804087282, ts_snapshot:15980
6291454, swapfree: 6291454, vmalloc: 102938, dmabuf_used: 99342, dmabuf_cached: 0, dmabuf_resv: 0, kgs1_used:
0, inactive_anon: 0, active_anon: 865780, active_file: 1463357, inactive_file: 2044675, unevictable: 23253, slab_reclaimable:
0, anon_mapped: 860091, file_mapped: 445602, file_pages: 3536969, file_dirty: 1015, writeback: 3, writeback_temp: 0,
0, kernel_misc_reclaimable: 223261, kernel_stack_kb: 61503
|1
...kworker/u16:3-1075... (.1075) . [000] ... 160.988616: sched_switch: prev_comm=kworker/u16:3 prev_pid=1075 prev_prio=120 prev_state=I ==> next_co
```

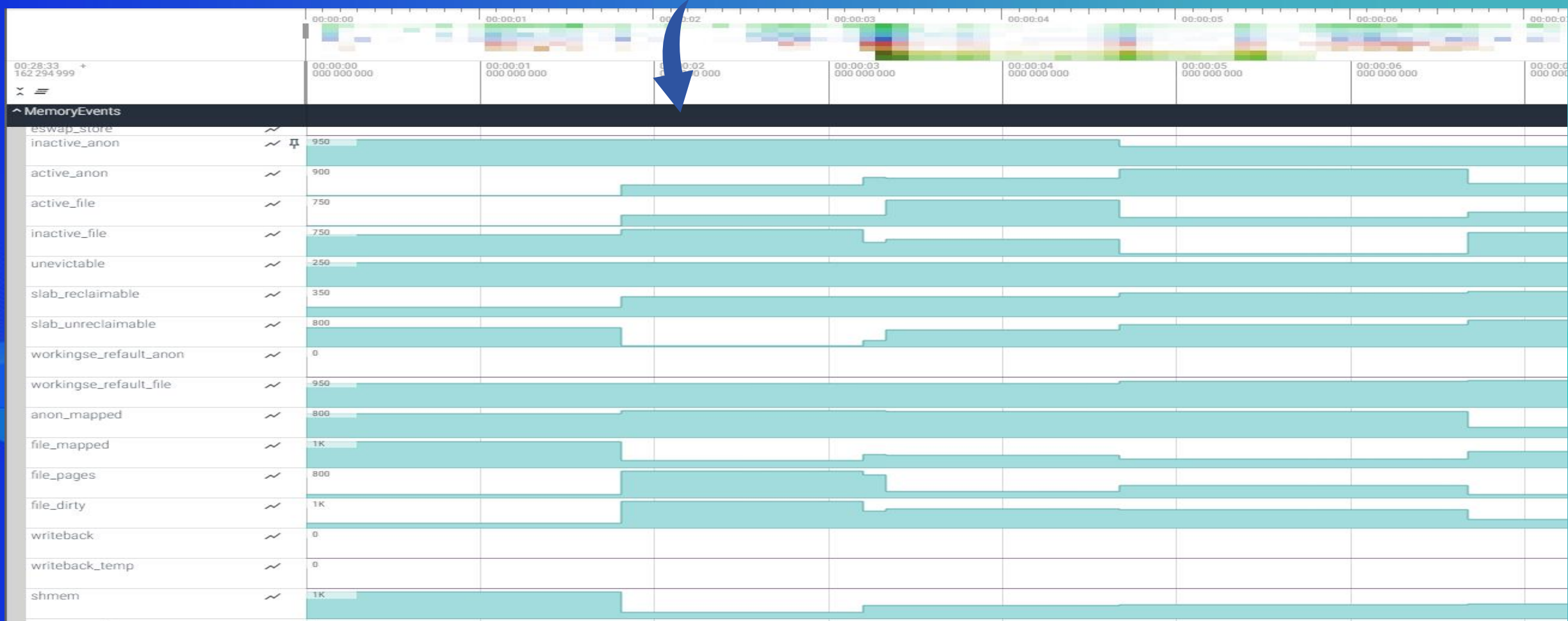
mmtrace呈现&分析：可视化分析问题

ftrace ringbuf

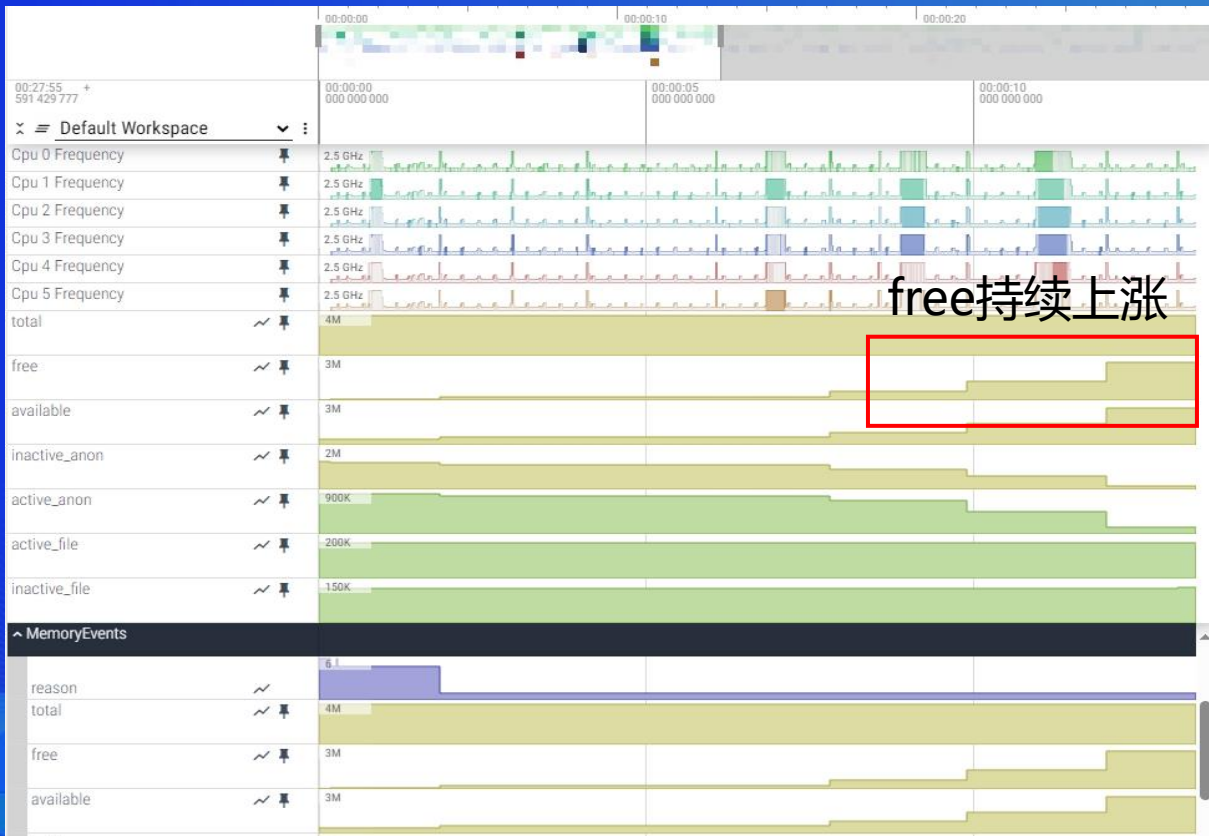
Perfetto UI插
件

Perfetto网站

性能分析



mmtrace实践1：内存过度回收导致的性能问题



```
/* don't abort memcg reclaim to ensure fairness */
if (!root_reclaim(sc) && !bypass)
    return false;

if (sc->nr_reclaimed >= max(sc->nr_to_reclaim, compact_gap(sc->order)))
    return true;

trace_android_vh_mglru_should_abort_scan_order(sc->order, &bypass);
/* check the order to exclude compaction-induced reclaim */
if (!current_is_kswapd() || sc->order) && !bypass)
    return false;

mark = sysctl_numa_balancing_mode & NUMA_BALANCING_MEMORY_TIERING ?
        WMARK_PROMO : WMARK_HIGH;

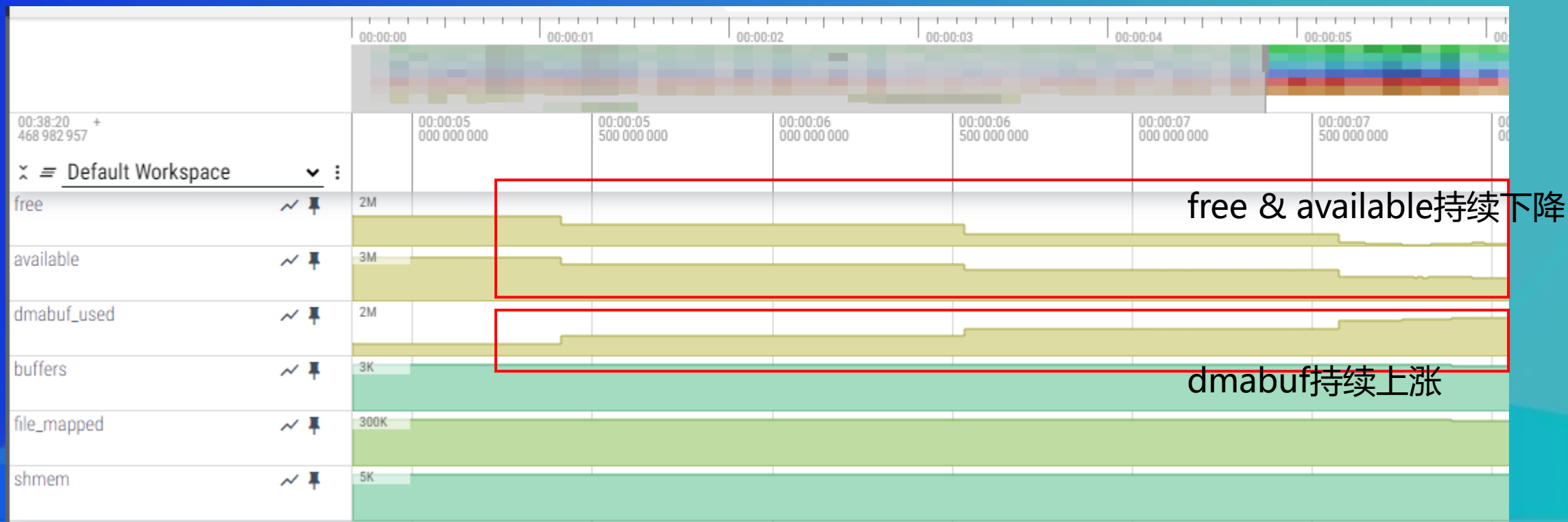
for (i = 0; i <= sc->reclaim_idx; i++) {
    struct zone *zone = lruvec_pgdat(lruvec)->node_zones + i;
    unsigned long size = wmark_pages(zone, mark) + MIN_LRU_BATCH;

    if (managed_zone(zone) && !zone_watermark_ok(zone, 0, size, sc->reclaim_idx, 0))
```

内存持续被回收，free上涨，kswapd占用cpu资源导致性能卡顿。

修改链接：[.../vmscan.c · Gerrit Code Review \(googlesource.com\)](https://go.dev/dotcom/vmscan.c)

mmtrace实践2: dmabuf异常占用导致的性能问题



前台应用滑动申请内存卡顿， perfetto显示free & available持续下降。
而dmabuf占用又持续升高，说明dmabuf内存有明显异常。再查看应用内存占用，可以快速定界到应用。

总结

mmtrace实现了一个低开销内存采集工具，适合部署在beta/商用手机。

通过mmtrace可以关联性能和内存问题，呈现出内存变化过程，变离散数据为连续数据，大幅提升性能开发人员的工作效率。



Thank you