

# 相机启动慢、人脸解锁慢分析

软件系统集成部策略中心 柯锦玲 2020/01/03



# Agenda

## 背景

- 相机启动慢
- 人脸解锁慢

## 原因

- 相机占用内存大
- 内存碎片化导致ION内存分配慢

## ION内存优化方案

## ION内存优化效果

## ION内存优化patch

## 感悟：系统优化靠大家

## 附录

# Agenda

## 背景

- 相机启动慢
- 人脸解锁慢

## 原因

- 相机占用内存大
- 内存碎片化导致ION内存分配慢

## ION内存优化方案

## ION内存优化效果

## ION内存优化patch

## 感悟：系统优化靠大家

## 附录

# 相机启动慢-背景

- 相机测试中，并未测试出相机启动慢的问题
  - 都是在1.5秒以内启动
- 但是PD1938(三星平台)试用多个试用用户反馈相机启动黑屏3秒以上
- 测试出了问题吗？

# 人脸解锁相机慢-背景

- 试用机器人脸解锁速度比工程测试机器慢100ms以上
- 2种测试复现方法:
  - 1.按power键进行人脸解锁
    - 解锁的时候，如果单单是按密码解锁键的人脸解锁速度比按power键解锁快60ms左右
  - 2.打开相机

# 人脸解锁相机慢-背景

- 实验现象：
  - 试用机器刷掉user data分区以及cache分区后，速度恢复
  - 恢复后机器，重新安装应用，使用几小时后，速度变慢
  - 通过互传将软件传送到PD1938工程机器后，后台打开应用，人脸解锁速度变慢。
  - 解锁的时候，如果单单是按密码解锁键的人脸解锁速度比按power键解锁快60ms左右

# Agenda

## 背景

- 相机启动慢
- 人脸解锁慢

## 原因

- 相机占用内存大
- 内存碎片化导致ION内存分配慢

## ION内存优化方案

## ION内存优化效果

## ION内存优化patch

## 感悟：系统优化靠大家

## 附录

# 相机启动慢以及 人脸解锁慢原因

- 主要原因
  - 三星启动相机瞬间分配太多内存导致相机启动需要3秒以上(1.5GB以上ion内存)
  - 内存碎片化导致，ion内存分配比较慢(三星原生系统问题)
    - 需要优化相机内存分配



# 三星vs高通710 相机使用内存

- 三星启动相机瞬间分配太多内存导致相机启动需要3秒以上
  - 启动相机ion内存分配
    - 三星平台需要1.5GB左右内存
    - 高通710平台仅需要700MB左右内存，拍照时候，ion内存也会涨到1.4GB左右
  - 从log以及systrace上看看，大部分时间耗费在buddy分配内存以及ion mmap上
  - 整个三星相机内存分配都是三星架构的行为， vivo相机团队需要到三星总部跟三星沟通讨论内存优化事情
- 人脸解锁约需要480MB ion内存
  - 三星与高通平台应该差不多
  - 人脸解锁慢的问题在高通710平台同样存在

# PD1938相机占用内存



PD1938相机占用  
内存

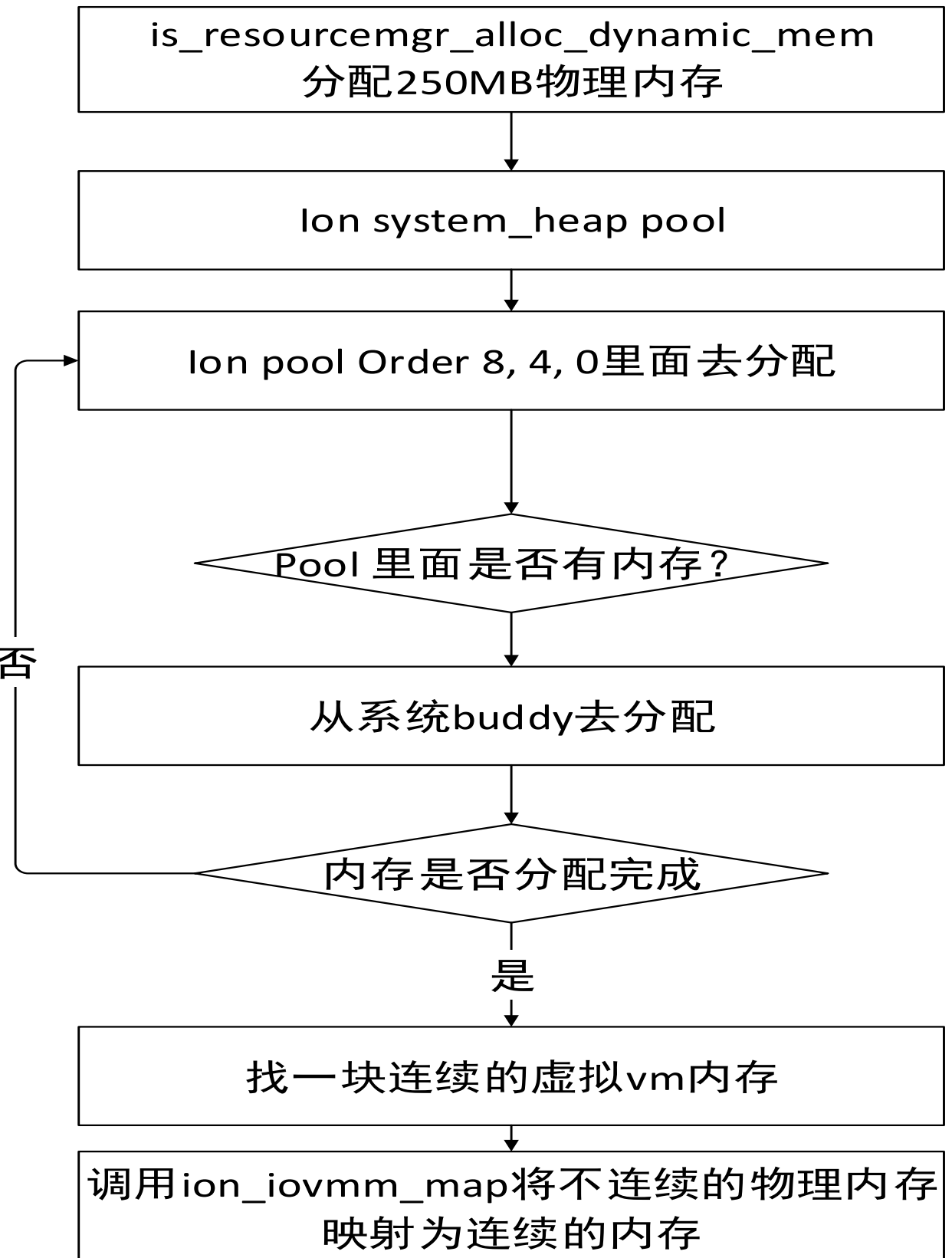


faceUnlock(PD1  
脸解锁相机使用的)

				App Heaps (KB)	CameraServer Heaps (KB)	Provider Heaps (KB)	Camera3rd Heaps (KB)	HEAP TOTAL (KB)	App ION (KB)	CameraSe rver ION (KB)	Provider ION (KB)	Camera3r d ION (KB)	ION TOTAL (KB )	
1														
2														
3	后置录像	720P (有人脸场景)	无滤镜，无美颜	92526	6351	153869	133760	386506	90256	4519	1152107	12477	1259359	1645865
4			开启滤镜	96655	6873	143851	133028	380407	135580	4519	1149995	10440	1300534	1680941
5			开启美颜	102223	7342	144630	177430	431625	103328	4067	1149543	78247	1335185	1766810
6			开启滤镜+美颜	111605	7262	150461	178391	447719	139960	4067	1149543	78902	1372472	1820191
7			1080P	82605	5737	151007	137408	376757	94907	9203	1262711	21543	1388364	1765121
8			4K	105977	5928	96011	139526	347442	96012	9203	794487	2270	901972	1249414
9			预览	105232	5693	148776	154986	414687	119441	13746	1420190	50461	1603838	2018525
10	后置照片模式	4: 3 (16M)	夜景拍照	105743	6950	158369	139467	410529	115783	20821	1804697	15016	1956317	2366846
11			HDR拍照	73713	7208	148078	156247	385246	115049	22347	1524447	62879	1724722	2109968
12			亮环境拍照	57245	6943	153801	156551	374540	106538	20821	1475937	62895	1666191	2040731
13														
13	连拍		后置	74103	5735	145475	154275	379588	147715	52946	1531554	50522	1782737	2162325
14			前置	67732	6202	94593	135722	304249	159560	89486	804694	2257	1055997	1360246
15	6400万	后置	拍照	94075	6974	107597	137911	346557	130617	43323	1965407	2257	2141604	2488161
16	3200万	后置	拍照	70764	6916	110515	136879	325074	118914	27059	929771	2257	1078001	1403075

# Ion内存分配流程

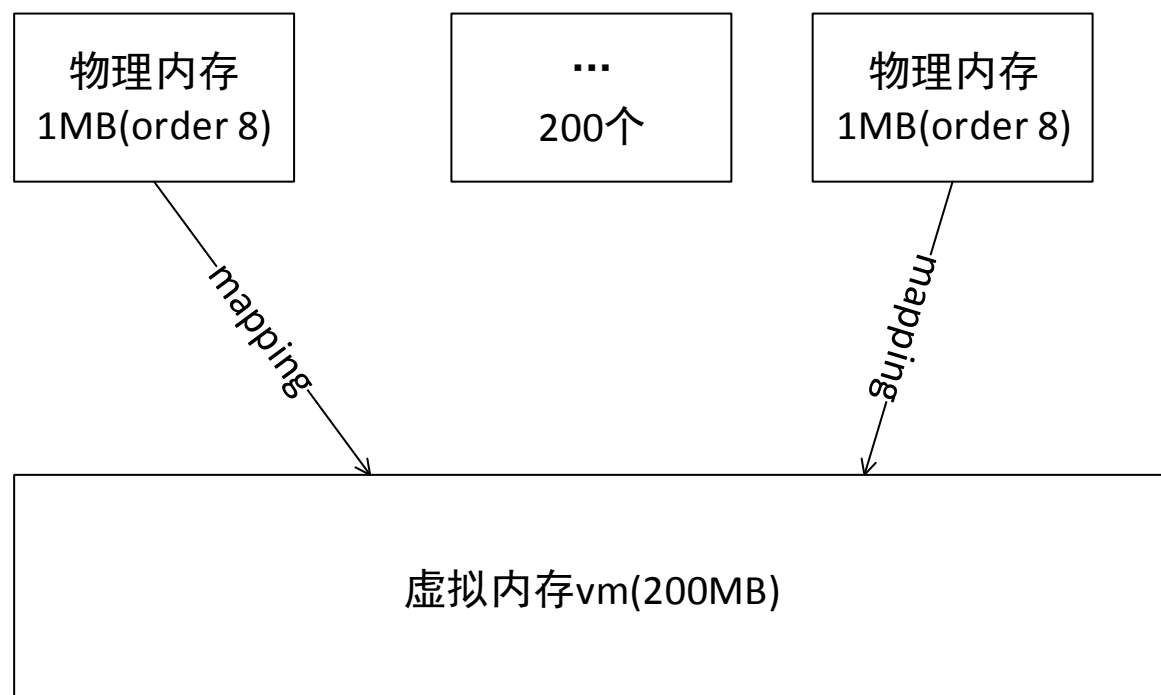
- Ion内存分配流程  
(Samsung9630为例子)
- 分配速度影响因素
  - 1. pool 是否有可用内存;
  - 2. Pool连续物理内存多少
  - 3. 分配线程是不是有时候跑大核, 有时候跑小核;
  - 4. 分配内存是不是有sleeping;
  - 5. 映射虚拟内存是不是比较慢?



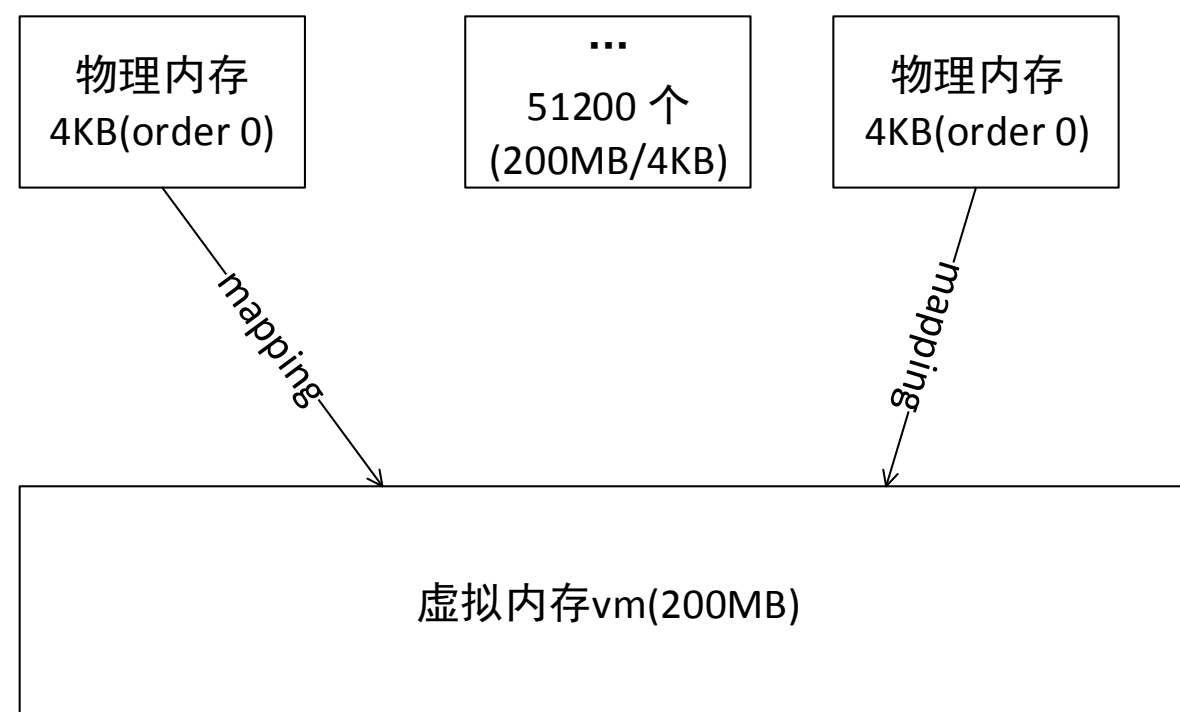
# Ion内存分配

- Ion内存分配（内存碎片导致ion内存分配慢）

系统刚刚开机时候



试用一段时间后



# Ion内存分配流程

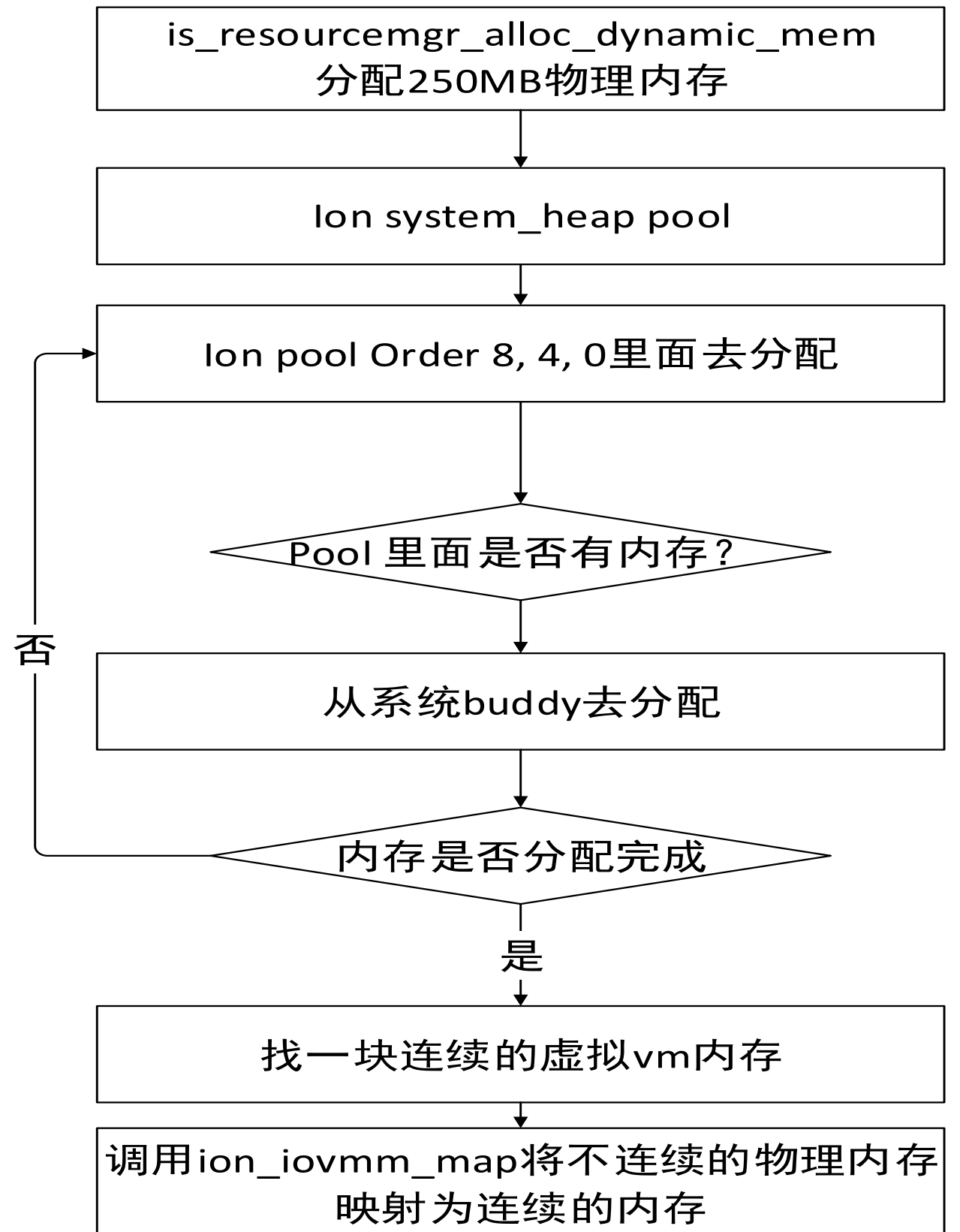
- Ion内存分配
  - 原生逻辑依次从order 8(1MB), order 4(64KB), order 0(4KB) 分配物理内存
    - 当没有order 8的连续物理内存的时候，会从order 4的分配，如果没有就从order 0分配。
  - 将分配到的不连续的物理内存通过 iommu map 成连续的内存给驱动使用

# Ion内存分配流程

- Ion内存分配
  - 假设分配200MB的ion内存，
    - 如果按照order 8(1MB)分配
      - 只需要向buddy system分配200次
      - 分配完成后，map成连续内存的时候，也是只需要map 200次就可以
    - 如果按照order 0(4KB)分配
      - 需要向buddy system分配51200次
      - 分配完成后，map成连续内存的时候，也需要map 51200次

# Ion内存分配影响因素

- 影响ion内存分配速度的因素
  - ion pool 里面是否有可用内存
  - 系统是否有连续的物理内存或者ion pool里面的内存是否是连续的大块物理内存



# Agenda

## 背景

- 相机启动慢
- 人脸解锁慢

## 原因

- 相机占用内存大
- 内存碎片化导致ION内存分配慢

## ION内存优化方案

## ION内存优化效果

## ION内存优化patch

## 感悟：系统优化靠大家

## 附录



# Ion内存分配优化方案

- ion内存分配优化
  - ion pool里面尽量保留内存
  - ion pool里面尽量保留连续的大片物理内存

# Ion内存分配优化方案

- ion内存分配优化
  - 修改分配逻辑
    - order 8 -> order 4 -> order 0 修改为
    - order 9 -> order 4 -> order 3 -> order 2 -> order 1 -> order 0
    - 当系统内存碎片化很严重的时候，仍然会有很多 order 3 ， order 2, order 1的连续物理内存

- ion内存分配优化

- ## 一 修改分配逻辑

- ## 一 优化

# oder\_to\_index函数

```

29: #define NUM_ORDERS ARRAY_SIZE(orders)
30:
31: static gfp_t high_order_gfp_flags = (GFP_HIGHUSER | __GFP_ZERO | __GFP_NOWARN |
32: » » » » » . . . . . __GFP_NORETRY) & ~__GFP_RECLAIM;
33: static gfp_t low_order_gfp_flags = GFP_HIGHUSER | __GFP_ZERO;
34: #ifdef CONFIG_RSC_ION_OPT
35: #if 1
36: static const unsigned int orders[] = {9, 4, 3, 2, 1, 0};
37: static const unsigned int orders_idx[MAX_ORDER+1] = {
38: » » » » » » » » » » » 5, 4, 3, 2,
39: » » » » » » » » » » » 1, 0, 0, 0,
40: » » » » » » » » » » » 0, 0, 0, 0
41: » » » » » } ;
42: #else
43: static const unsigned int orders[] = {8, 4, 3, 2, 1, 0};
44: static const unsigned int orders_idx[MAX_ORDER+1] = {
45: » » » » » » » » » » » 5, 4, 3, 2,
46: » » » » » » » » » » » 1, 0, 0, 0,
47: » » » » » » » » » » » 0, 0, 0, 0
48: » » » » » } ;
49: #endif
50: /*
51: static const unsigned int orders[] = {8, 4, 0};
52: static const unsigned int orders_idx[MAX_ORDER+1] = {
53: » » » » » » » » » » » 2, 1, 1, 1,
54: » » » » » » » » » » » 1, 0, 0, 0,
55: » » » » » » » » » » » 0, 0, 0, 0
56: » » » » » } ;
57: */
58: #define order_to_index(order) orders_idx[order]
59: #define ION_OPT_SHRINK_POOL_SIZE_MB 10
60: #else
61: static const unsigned int orders[] = {8, 4, 0};
62:
63: static int order_to_index(unsigned int order)
64: {
65:     int i;
66:
67:     for (i = 0; i < NUM_ORDERS; i++)
68:         if (order == orders[i])
69:             return i;
70:     BUG();
71:     return -1;
72: }
73: #endif
74:

```

# Ion内存分配优化方案

- ion内存分配优化

- ion pool里面尽量保留物理内存

- 开机保留1000MB ion连续物理内存

```
static void ion_sys_heap_reserve(struct work_struct *work)
{
    c1 = local_clock(); [ion_system_heap.c (kernel\...\ion) *]

    if (rsc_system_heap) {
        cpu = raw_smp_processor_id();
        for (i = 0; i <= order_to_index(4); i++) {
            if (global_zone_page_state(NR_ION) < reserve_max_pool_size) {
                cached_pool = rsc_system_heap->cached_pools[i];
                /*
                 * max_page_pool_size expend to max_page_pool_size * 13 / 10,
                 * max_page_pool_size 700MB expend to 700MB * 13 / 10,
                 */
                count_cached += ion_sys_heap_charge(cached_pool, max_page_pool_size * RSC_ION_POOL_CACHED_PERCENT / 100, true);
                uncached_pool = rsc_system_heap->uncached_pools[i];
                count_noncached += ion_sys_heap_charge(uncached_pool, max_page_pool_size * RSC_ION_POOL_NONE_CACHED_PERCENT / 100, false);
            }
        }
        c2 = local_clock();
        printk("[RSC] ion_sys_heap_reserve: ion: %lu pool: %u max_page_pool_size: %u "
               "cached: %u uncached: %u cost: %llu us cpu %d %d\n",
               global_zone_page_state(NR_ION), atomic_read(&rsc_ion_pool_pages),
               max_page_pool_size, count_cached, count_noncached, (c2 - c1) / 1000, cpu, raw_smp_processor_id());
    }
}

static DECLARE_DELAYED_WORK(reserve_ion_worker, ion_sys_heap_reserve);

static struct ion_heap * __ion_system_heap_create(void)
{
    if (ion_system_heap_create_pools(heap->cached_pools, true))
        goto destroy_uncached_pools;

    heap->heap.debug_show = ion_system_heap_debug_show;

#ifdef CONFIG_RSC_ION_OPT
    rsc_system_heap = heap;
    schedule_delayed_work(&reserve_ion_worker, 8 * HZ);
#endif

    return &heap->heap;
}
```

# Ion内存分配优化方案

- ion内存分配优化
  - ion pool里面尽量保留物理内存
    - 内存回收压力足够大的时候才进行回收
      - $pri \leq 1$ 时候回收700MB以内的ion内存
      - stressapptest -M 5000 可以进行回收
    - 通过节点可以控制
      - cat /sys/rsc/svp/ion\_pool
      - 850 800 750 700

```
291: #ifdef CONFIG_RSC_ION_OPT
292: #include <linux/vivo_rsc/rsc_internal.h>
293: #endif
294:
295: unsigned int rsc_ion_pool[RSC_ION_POOL_LEVEL] = {
296:     850,
297:     800,
298:     750,
299:     700,
300: };
301: /*
302: static unsigned int rsc_ion_pool[4] = {
303:     800,
304:     600,
305:     400,
306:     300,
307: };
308: */
309:
310: module_param_array_named(pool, rsc_ion_pool, uint, NULL,
311:     S_IRUGO | S_IWUSR);
312:
313: static unsigned int ion_priority[RSC_ION_POOL_LEVEL] = {
314:     3, // 9
315:     5, // 7
316:     9, // 3
317:     11, // 1
318: };
319:
320: module_param_array_named(priority, ion_priority, uint, NULL,
321:     S_IRUGO | S_IWUSR);
322:
323: static unsigned long ion_heap_shrink_count(struct shrinker *shrinker,
324:     struct shrink_control *sc)
325: {
326:     struct ion_heap *heap = container_of(shrinker, struct ion_heap,
327:     shrinker);
328:     int total = 0;
329:
330:     total = ion_heap_freelist_size(heap) / PAGE_SIZE;
331:     if (heap->ops->shrink)
332:         total += heap->ops->shrink(heap, sc->gfp_mask, 0);
333:
334:     BUILD_BUG_ON(DEF_PRIORITY < 11);
335:
336:     if (totalram_pages > RAM_PAGES_6G) {
337:         if (sc->priority >= (DEF_PRIORITY - ion_priority[0])) {
338:             if (total >= (SZ_1M * rsc_ion_pool[0] / PAGE_SIZE))
339:                 return total - (SZ_1M * rsc_ion_pool[0] / PAGE_SIZE);
340:         }
341:     }
342: }
```

# Ion内存分配优化方案

- ion内存分配优化
  - ion pool里面尽量保留物理内存

- 驱动或者应用调用ion\_system\_heap\_free -> free\_buffer\_page进行回收的时候，只有pool大于700MB时候才进行回收
  - 修改节点  
/sys/module/ion\_system\_heap/parameters/max\_page\_pool\_size的值

```
static void free_buffer_page(struct ion_system_heap *heap,
                             struct ion_buffer *buffer, struct page *page)
{
    struct ion_page_pool *pool;
    unsigned int order = compound_order(page);
    bool cached = ion_buffer_cached(buffer);

    /* go to system */
    if (buffer->private_flags & ION_PRIV_FLAG_SHRINKER_FREE) {
        free_pages(page, order);
    }
    #ifdef CONFIG_RSC_MEM_STAT
        mod_zone_page_state(page_zone(page), NR_ION, --((long)(1 << order)));
    #endif
    return;

    #ifdef CONFIG_RSC_ION_OPT
        if (atomic_read(&rsc_ion_pool_pages) >= max_page_pool_size) {
            /*
             * order 0 directly free to buddy system.
             */
            if (0 == order) {
                free_pages(page, order);
            }
            #ifdef CONFIG_RSC_MEM_STAT
                mod_zone_page_state(page_zone(page), NR_ION, --((long)(1 << order)));
            #endif
            return;
        } else {
            int count;

            /*
             * (10 << (20 - PAGE_SHIFT)) == 10 * SZ_1M / PAGE_SIZE
             * shrink 10MB every time.
             */
            count = ion_system_heap_shrink_fast(heap, (ION_OPT_SHRINK_POOL_SIZE_MB << (20 - PAGE_SHIFT)));
            printk("[RSC] count: %u max_page_pool_size: %u buffer->size: %lu shrink order: %d count: %d %d\n",
                   atomic_read(&rsc_ion_pool_pages), max_page_pool_size, (unsigned long)buffer->size,
                   order, count, (ION_OPT_SHRINK_POOL_SIZE_MB << (20 - PAGE_SHIFT)));
        }
    #endif
    if (cached && (buffer->flags & ION_FLAG_SYNC_FORCE)) {
        cached = !cached;
        flush_dcache_area(page_to_virt(page),
                           1 << (PAGE_SHIFT + order));
    }
}
```

ard_9.0/android_device_samsung_erd9630 / conf/init.exynos9630.rc	Patch Set Base 1	Patch Set 1
245	chmod 0666 /dev/sec-nfc	chmod 0666 /dev/sec-nfc
246	chmod 0666 /dev/sec-esepwr	chmod 0666 /dev/sec-esepwr
247		
248	# NFC storage	# NFC storage
249	mkdir /data/nfc 770 nfc nfc	mkdir /data/nfc 770 nfc nfc
250		
251	# NFC hal surfix	# NFC hal surfix
252	setprop ro.hardware.nfc_nci sec	setprop ro.hardware.nfc_nci sec
253		
254	# Limit to ION page pool (Three UHD 32bpp RGB)	# Limit to ION page pool (Three UHD 32bpp RGB)
255	write /sys/module/ion_system_heap/parameters/max_page_pool_size 24300	write /sys/module/ion_system_heap/parameters/max_page_pool_size 179200
256		
257	# Configuration to G2D blocking mode	# Configuration to G2D blocking mode
258	write /sys/module/g2d_task/parameters/block_on_contension 1	write /sys/module/g2d_task/parameters/block_on_contension 1

# Ion内存分配优化方案

- ion内存分配优化
  - ion pool里面尽量保留大的连续物理内存
    - 驱动或者应用调用ion\_system\_heap\_free -> free\_buffer\_page进行回收的时候，尽量保留大order的物理内存

```
static void free_buffer_page(struct ion_system_heap *heap,
>> struct ion_buffer *buffer, struct page *page)
{
>> struct ion_page_pool *pool;
>> unsigned int order = compound_order(page);
>> bool cached = ion_buffer_cached(buffer);
>>
>> /* go to system */
>> if (buffer->private_flags & ION_PRIV_FLAG_SHRINKER_FREE) {
>>     __free_pages(page, order);
>> #ifdef CONFIG_RSC_MEM_STAT
>>     mod_zone_page_state(page_zone(page), NR_ION, -((long)(1 << order)));
>> #endif
>>     return;
>> }
>>
>> #ifdef CONFIG_RSC_ION_OPT
>> if (atomic_read(&rsc_ion_pool_pages) > max_page_pool_size) {
>>     /*
>>     * order 0 directly free to buddy system.
>>     */
>>     if (0 == order) {
>>         __free_pages(page, order);
>> #ifdef CONFIG_RSC_MEM_STAT
>>         mod_zone_page_state(page_zone(page), NR_ION, -((long)(1 << order)));
>> #endif
>>         return;
>>     } else {
>>         int count;
>>
>>         /*
>>         * (10 << (20 - PAGE_SHIFT)) == 10 * SZ_1M / PAGE_SIZE
>>         * shrink 10MB every time.
>>         */
>>         count = ion_system_heap_shrink_fast(heap, (ION_OPT_SHRINK_POOL_SIZE_MB << (20 - PAGE_SHIFT)));
>>         /*
>>         printk("[RSC]: count: %u max_page_pool_size: %u buffer->size: %lu shring order: %d count: %d %d\n",
>>         atomic_read(&rsc_ion_pool_pages), max_page_pool_size, (unsigned long)buffer->size,
>>         order, count, (ION_OPT_SHRINK_POOL_SIZE_MB << (20 - PAGE_SHIFT)));
>>         */
>>     }
>> }
>> }
>> #endif
>>
>> if (cached && (buffer->flags & ION_FLAG_SYNC_FORCE)) {
>>     cached = !cached;
>>     __flush_dcache_area(page_to_virt(page),
>>     1 << (PAGE_SHIFT + order));
>> }
```



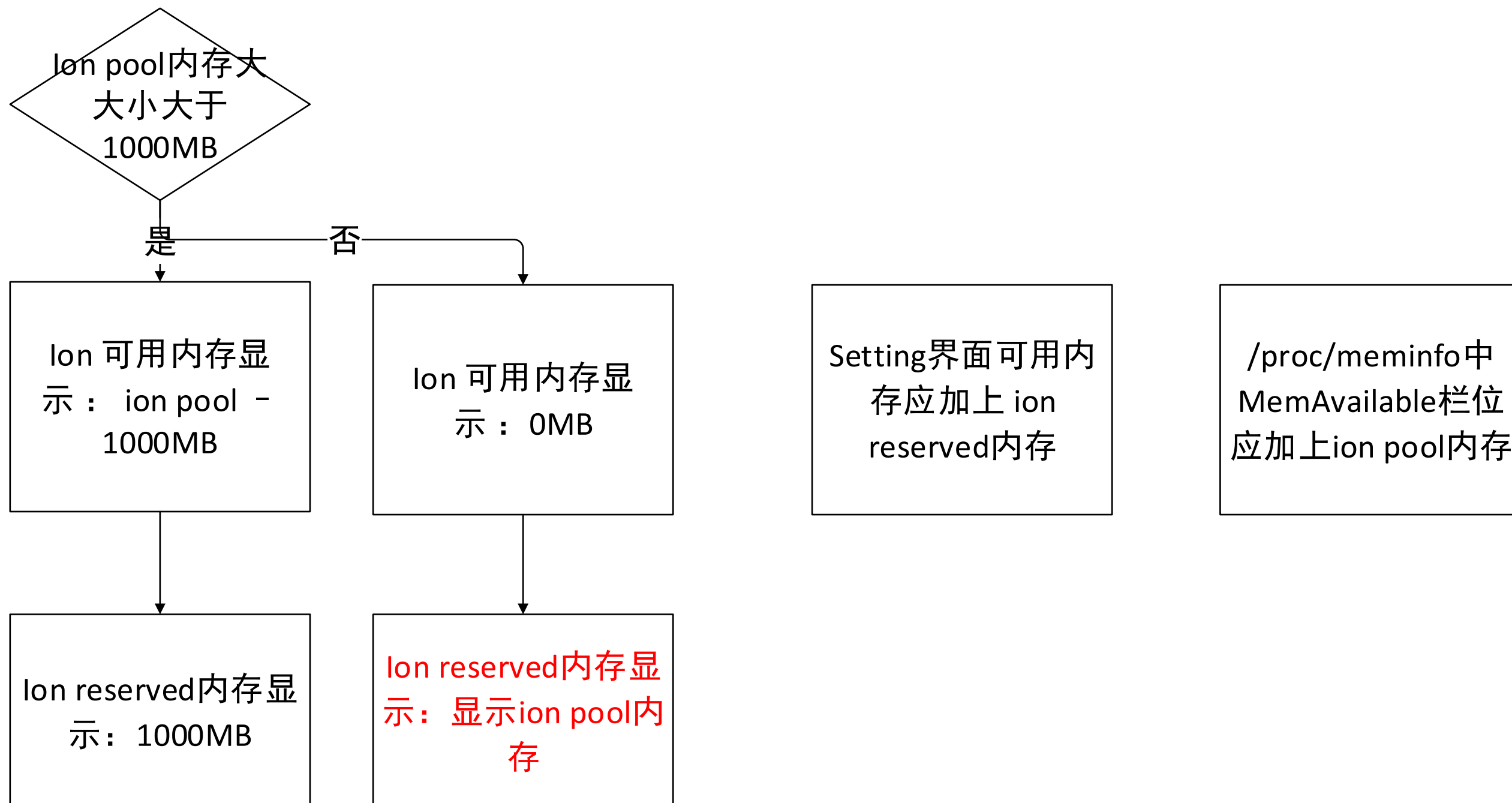
# Ion内存分配优化方案

- ion内存分配优化
  - ion pool里面尽量保留大的连续物理内存
    - 驱动或者应用调用ion\_system\_heap\_free -> free\_buffer\_page进行回收的时候，尽量保留大order的物理内存
    - shrinker回收的时候，从order 0开始回收

```
459: static int ion_system_heap_shrink(struct ion_heap *heap, gfp_t gfp_mask,
460: >> >> int nr_to_scan)
461: {
462: >> struct ion_page_pool *uncached_pool;
463: >> struct ion_page_pool *cached_pool;
464: >> struct ion_system_heap *sys_heap;
465: >> int nr_total = 0;
466: >> int i, nr_freed;
467: >> int only_scan = 0; [ion_system_heap.c (kernel\...\ion)]
468: >>
469: >> sys_heap = container_of(heap, struct ion_system_heap, heap);
470: >>
471: >> if (!nr_to_scan)
472: >> >> only_scan = 1;
473: >>
474: >> /*-shrink low order page first!!!*/
475: >> #ifdef CONFIG_RSC_ION_OPT
476: >> >> for (i = NUM_ORDERS - 1; i >= 0; i--)
477: >> #else
478: >> >> for (i = 0; i < NUM_ORDERS; i++)
479: >> #endif
480: >> {
481: >> >> uncached_pool = sys_heap->uncached_pools[i];
482: >> >> cached_pool = sys_heap->cached_pools[i];
483: >>
484: >> >> if (only_scan) {
485: >> >> >> nr_total += ion_page_pool_shrink(uncached_pool,
486: >> >> >> >> gfp_mask,
487: >> >> >> >> nr_to_scan);
488: >> }
```



# 内存显示调整



# Agenda

## 背景

- 相机启动慢
- 人脸解锁慢

## 原因

- 相机占用内存大
- 内存碎片化导致ION内存分配慢

## ION内存优化方案

## ION内存优化效果

## ION内存优化patch

## 感悟：系统优化靠大家

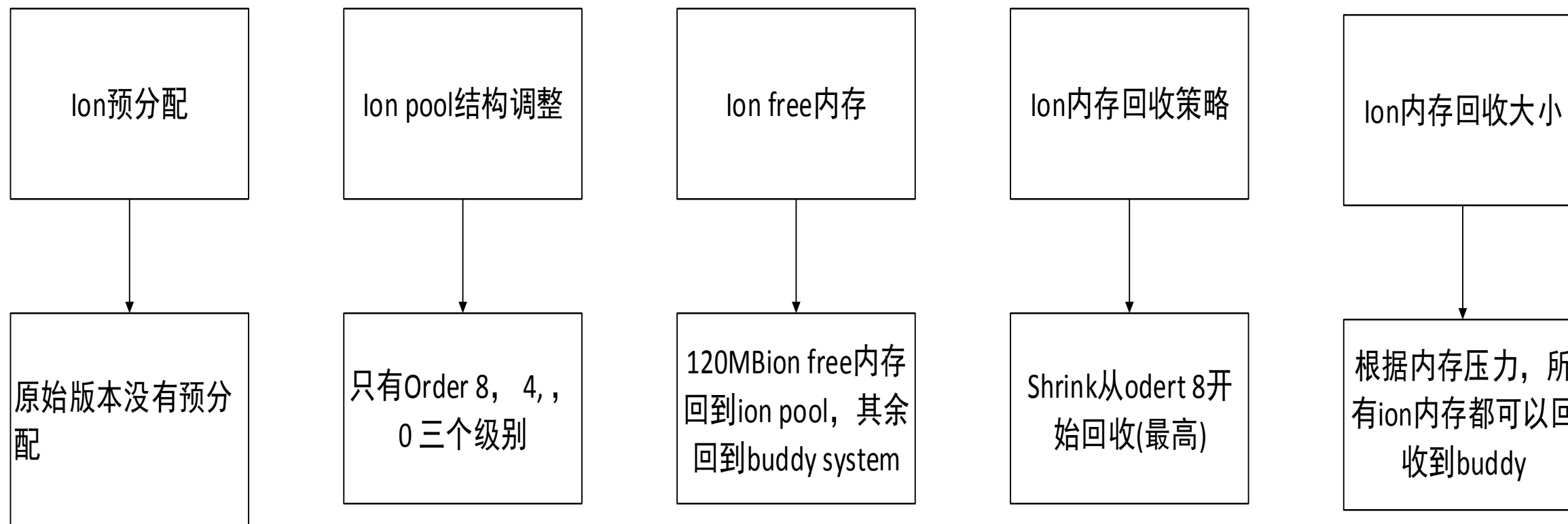
## 附录

# Ion内存优化方案v1

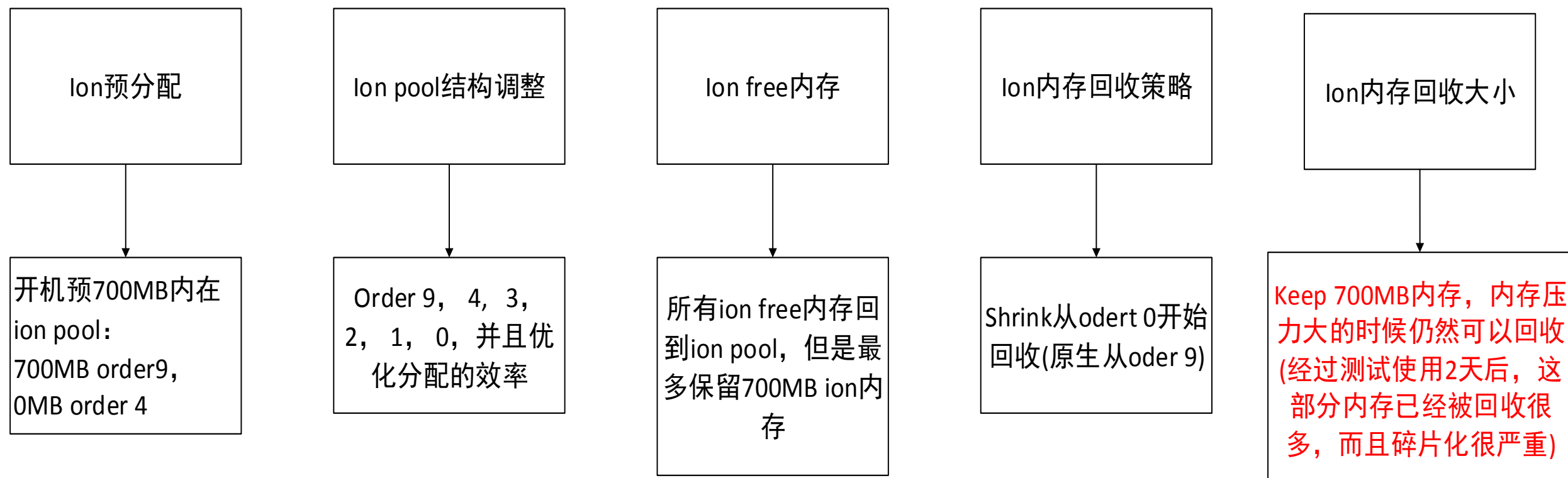
- 预留700MB内存
- 这个优化方案大大减低相机打开黑屏问题，但是仍然有用户反馈偶尔打开相机需要3秒以上的问题
- 需要进一步加大预留内存优化

# Ion内存优化方案v1

原始版本



优化版本

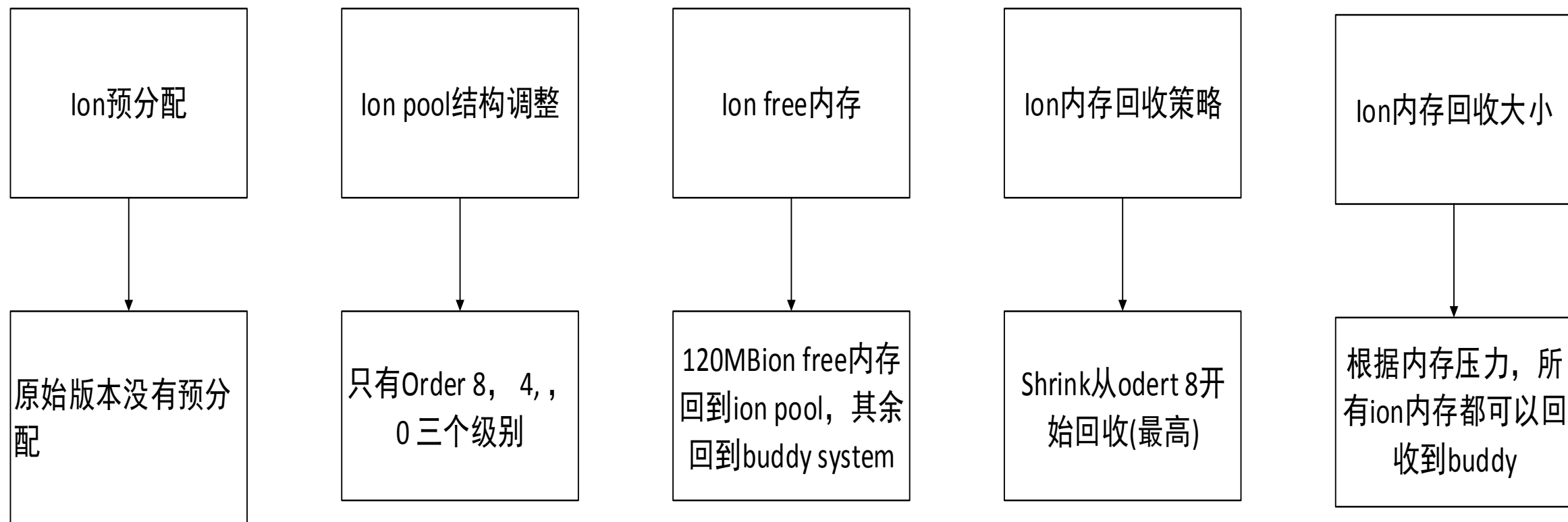


# Ion内存优化方案v2

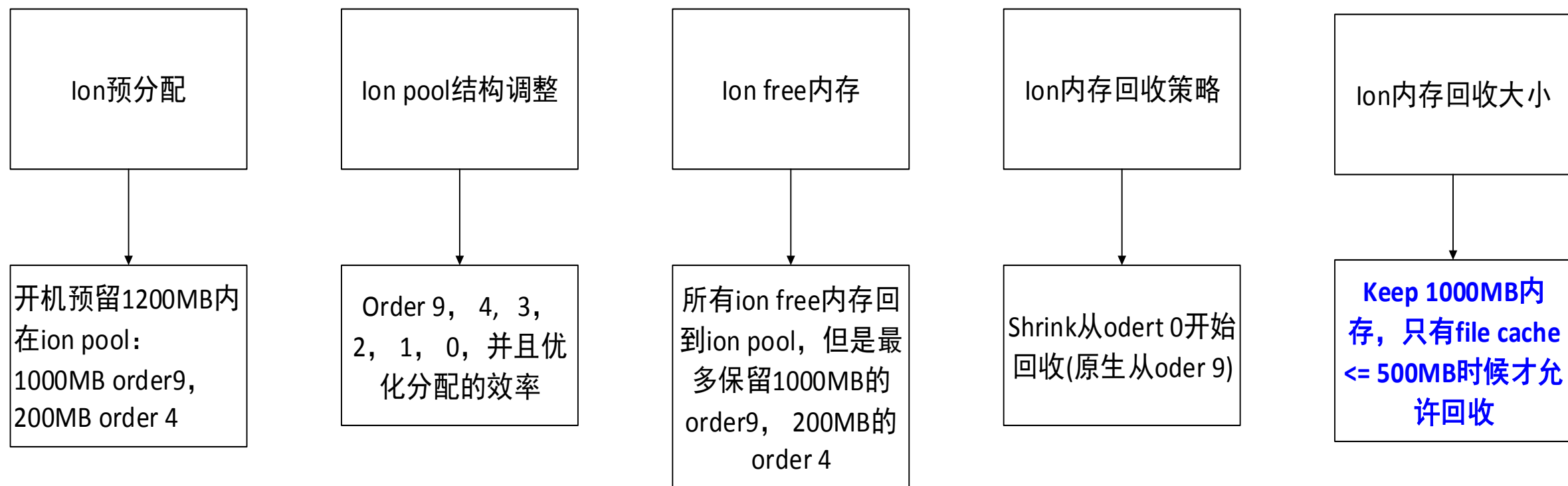
- 预留1000MB内存给相机使用，并且锁定ION内存存在ION pool

# Ion内存优化方案v2

原始版本



优化版本



# Ion内存优化方案v2

- 这个方案从系统版本1.10.0上线(12月17日)到现在，相机未反馈启动相机慢的问题。
- 升级第一次启动相机时间减少**706ms**
  - 由2071ms降低到1365ms，减少34%时间



相机第一次启动优化

1		log	视频					
2	测试版本--系统版本	系统版本1	IMG_8645	104	495	1.629		
3	测试版本--系统版本	系统版本2	IMG_8647	389	851	1.925		
4	测试版本--系统版本	系统版本3	IMG_8649	197	628	1.796		
5	测试版本--系统版本	系统版本4	IMG_8651	122	737	2.563		
6	测试版本--系统版本	系统版本5	IMG_8653	225	730	2.104		
7	测试版本--系统版本	系统版本6	IMG_8655	408	1050	2.675		
8	测试版本--系统版本	系统版本7	IMG_8657	513	1023	2.125		
9	测试版本--系统版本	系统版本8	IMG_8659	196	707	2.129		
10	测试版本--系统版本	系统版本9	IMG_8661	146	610	1.933		
11	测试版本--系统版本	系统版本10	IMG_8663	99	539	1.833	2.071	
12								提升(秒)
13								0.706
14	系统版本--测试版本	测试版本1	IMG_8646	118	465	1.446		
15	系统版本--测试版本	测试版本2	IMG_8648	228	545	1.321		
16	系统版本--测试版本	测试版本3	IMG_8650	228	545	1.321		
17	系统版本--测试版本	测试版本4	IMG_8652	213	503	1.208		
18	系统版本--测试版本	测试版本5	IMG_8654	215	497	1.175		
19	系统版本--测试版本	测试版本6	IMG_8656	150	473	1.346		
20	系统版本--测试版本	测试版本7	IMG_8658	81	406	1.354		
21	系统版本--测试版本	测试版本8	IMG_8660	156	456	1.250		
22	系统版本--测试版本	测试版本9	IMG_8662	100	464	1.517		
23	系统版本--测试版本	测试版本10	IMG_8664	95	507	1.717	1.365	

# 人脸解锁优化

- 人脸解锁时间降低了将近80ms
- 人脸解锁打开相机的时间稳定在280-350ms之间，比之前版本总是飙到400多要稳定很多。



# 人脸解锁优化

## • 优化后

次数	对应视频	按键的第一帧	亮屏那一帧	解锁成功那一帧	亮屏至解锁成功时间	人脸解锁时间
1	IMG_2580	1304	1420	1420	0.000	0.483
2		2160	2274	2274	0.000	0.475
3		2950	3068	3068	0.000	0.492
4		3804	3908	3908	0.000	0.433
5		4566	4698	4698	0.000	0.550
6		5296	5412	5412	0.000	0.483
7		6115	6222	6222	0.000	0.446
平均解锁时间					0.000	0.476

## • 优化前

次数	对应视频	按键的第一帧	亮屏那一帧	解锁成功那一帧	亮屏至解锁成功时间	人脸解锁时间
1	IMG_2462	338	503	503	0.000	0.688
2		1421	1559	1559	0.000	0.575
3		2987	3122	3122	0.000	0.563
4		4161	4291	4291	0.000	0.542
5		5488	5619	5619	0.000	0.546
6		7013	7138	7138	0.000	0.521
7		8324	8460	8460	0.000	0.567
平均解锁时间					0.000	0.558

# Agenda

## 背景

- 相机启动慢
- 人脸解锁慢

## 原因

- 相机占用内存大
- 内存碎片化导致ION内存分配慢

## ION内存优化方案

## ION内存优化效果

## ION内存优化patch

## 感悟：系统优化靠大家

## 附录

# Ion分配优化的patch

- Ion分配优化的patch如下：
  - **init.exynos9630.rc**
    - <http://smartgit/gerrit/#/c/2322602/>
    - <http://smartgit/gerrit/#/c/2326849/>
    - <http://smartgit/gerrit/#/c/2328454/>
  - kernel
    - <http://smartgit/gerrit/#/c/2322509/>
    - <http://smartgit/gerrit/#/c/2326847/>
    - <http://smartgit/gerrit/#/c/2327879/>
    - <http://smartgit/gerrit/#/c/2328449/>
    - <http://smartgit/gerrit/#/c/2339949/>
    - <http://smartgit/gerrit/#/c/2341574/>

# Agenda

## 背景

- 相机启动慢
- 人脸解锁慢

## 原因

- 相机占用内存大
- 内存碎片化导致ION内存分配慢

## ION内存优化方案

## ION内存优化效果

## ION内存优化patch

## 感悟：系统优化靠大家

- 我们需要更开放去接受新的技术

## 附录



# 系统优化靠大家

- 开放是一种手段，不是目的
- 开放才有活力，开放才能进步

# 加拉帕戈斯综合症

- 加拉帕戈斯群岛的地方特有物种——按照理想中的进化，并逐渐与其他大陆上的同类物种区分开来，自成一个体系。
- 它只反映当时、当地的条件，进化出只属于自己的肢体、翅膀和牙齿。如果你把他们放在别的海岛或者大陆上的话，不到一个月它就死定了。
- 我们需要更open地与世界接轨，小心水土不服。





# 开放才有活力

- 杀人鲸的鱼鳍在野外是直的，但是在动物园里面是弯的。



Left: A Sea World trainer measuring the progressive bending of the dorsal fin of a captive orca (Kanduke, deceased 9/20/1990). Right: The straight dorsal of a wild killer whale (T20). Note: 100% of captive adult male orcas have collapsed dorsal fins, versus less than 1% in the wild.

# Agenda

## 背景

- 相机启动慢
- 人脸解锁慢

## 原因

- 相机占用内存大
- 内存碎片化导致ION内存分配慢

## ION内存优化方案

## ION内存优化效果

## ION内存优化patch

## 感悟：系统优化靠大家

## 附录

- 人脸解锁慢分析
- 预留内存不生效问题
- ION内存观察



# 人脸解锁相机慢-分析

- 抓取systrace前
  - adb vivoroot
  - adb shell setprop persist.facedetect.debug.level 1
  - 然后重启手机
- 抓取systrace同时，需要抓取kernel log以及人脸时间的log
  - cat /dev/kmsg | grep RSC]
  - logcat | grep -iE FaceCamera2Speed

# 人脸解锁相机慢-分析

- 查看systrace分析观察FaceCamera2线程解锁
- 解锁过程关键进程
  - system\_server的FaceCamera2线程
  - 相机的provider进程android.hardware.camera.provider@2.4-service (provider@2.4-se)
    - 第一个FrameFactoryCre线程，负责主要工作
      - 设置为
    - 第二个FrameFactoryCre线程
    - setBuffersThrea线程
    - FrameFactorySta线程
    - HwBinder线程

# 人脸解锁相机慢-分析

- setBuffersThread、FrameFactoryCreateThread、dualFrameFactoryCreateThread、FrameFactoryStartThread、DualFrameFactoryStartThread
- 将这几个线程优先级放到104  
PRIORITY\_AUDIO

# 人脸解锁相机慢-分析



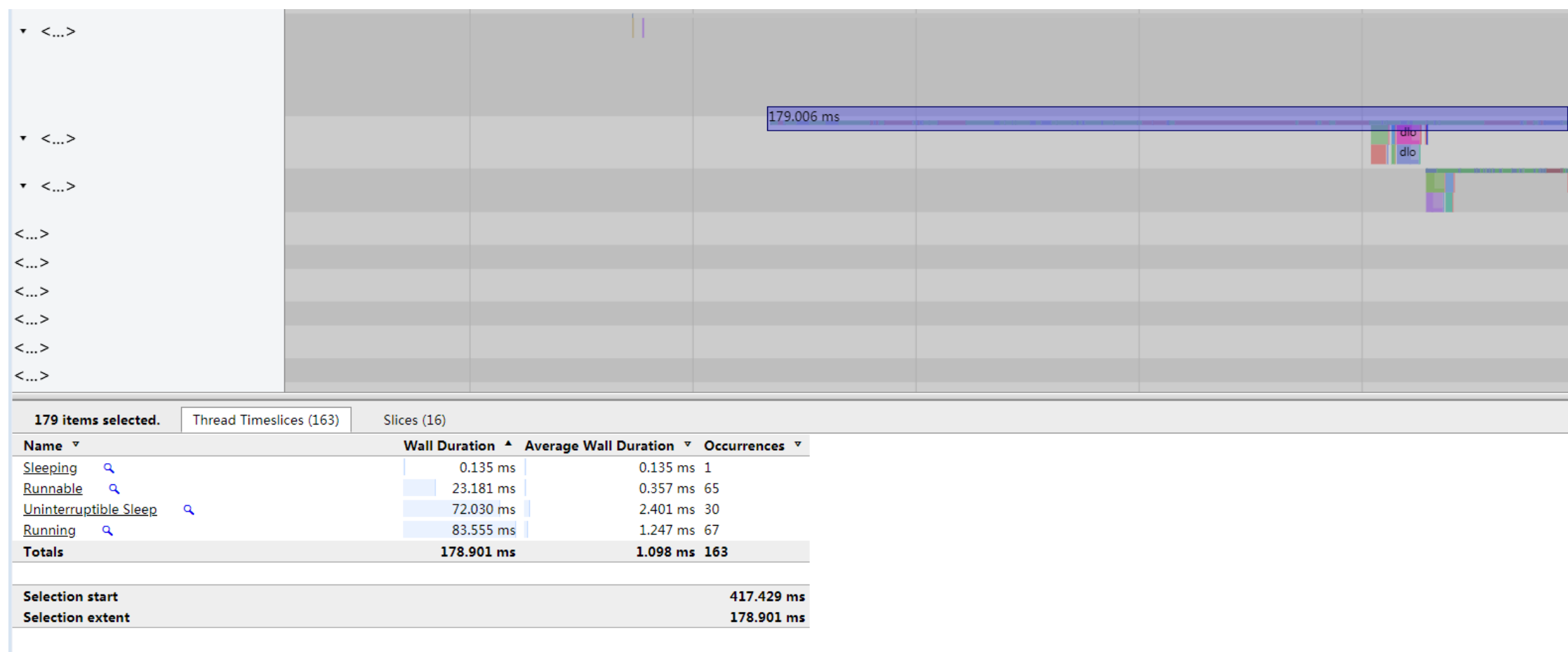
atrace.out.3\_add\_kthreadd 14034.rar

- FrameFactoryCreate后面还有factorystart, 里面会初始化 pipeline、streamon等操作, 再拿到第一帧后报给framework



# 人脸解锁相机慢-分析

- 分析第一个FrameFactoryCre线程
  - 总共耗时178.9ms vs 正常情况下应该是80ms左右
  - Uninterruptible sleeping时间达到72ms
  - runnable时间达到23ms



# 人脸解锁相机慢-分析

- PD1938试用用户人脸解锁时间统计
- 统计数据

PD1938试用用户  
人脸解锁时间统计\_2019

A	罗斌手机			FrameFactoryCre时间			
	启动时间	内存分配时间	总时间	running	runnable	sleeping	un_sleeping
1	529	62	258.676	184.307	51.612	0	22.757
2	378	27	91.9	72.443	9.538	0	9.921
3	396	43	141.2	94.517	36.98	0	9.703
4	414	44	155	90.326	46.934	0	18.563
5	417	50	162.01	84.666	51.266	0	26.078
平均	426.8	45.2	161.7572	105.2518	39.266	0	17.4044
B	罗桂钊手机			FrameFactoryCre时间			
	启动时间	内存分配时间	总时间	running	runnable	sleeping	un_sleeping
1	369	66	135.697	119.418	10.522	0	5.757
2	410	39	110.504	85.22	7.559	0	17.725
3	414	91	158.511	132.565	18.575	0	7.371
4	427	80	139.614	121.102	8.482	0	10.03
5	508	114	163.493	134.241	20.846	0	8.406
平均	425.6	78	141.5638	118.5092	13.1968	0	9.8578
C	刘香君手机			FrameFactoryCre时间			
	启动时间	内存分配时间	总时间	running	runnable	sleeping	un_sleeping
1	399	73	158.036	122.878	11.063	0	24.095
2	416	63	145.04	115.297	12.305	0	17.438
3	386	63	137.238	118.983	3.797	0	14.458
4	406	69	138.137	125.729	3.924	0	8.484
5	442	67	157.749	139.249	4.726	0	13.774
平均	409.8	67	147.24	124.4272	7.163	0	15.6498
D	齐双成手机			FrameFactoryCre时间			
	启动时间	内存分配时间	总时间	running	runnable	sleeping	un_sleeping
1	514	52	178.725	106.222	28.782	3.508	40.213
2	459	69	165.564	117.201	15.546	0	32.817
3	436	68	120.309	89.198	17.041	0	13.706
4	425	55	118.343	92.378	6.809	0	19.156
5	438	78	178.403	134.943	17.546	1.59	24.324
平均	454.4	64.4	152.2688	107.9884	17.1448	1.0196	26.0432
E	卿智勇手机			FrameFactoryCre时间			
	启动时间	内存分配时间	总时间	running	runnable	sleeping	un_sleeping
1	482	115	189.503	119.671	57.402	0	12.43
2	447	88	149.514	101.13	36.432	0	11.952
3	406	30	114.369	81.396	2.442	0	30.531
4	536	112	208.65	127.574	52.446	0	28.63
5	434	68	141.923	110.942	13.432	0	17.549
平均	461	82.6	160.7918	108.1426	32.4308	0	20.2184
总平均	435.52	67.44	152.7243	112.8638	21.84028	0.20392	17.83472
最大优化	129	47	82	42	21	0.2	17
优化结果	306.52	20.44	70.72432	70.86384	0.84028	0.00392	0.83472
目标	300						

# 预留内存不生效问题

- OTA或者T card升级后，ion reserve 1000MB内存不生效
  - 原因为OTA升级的时候applypatch做了drop\_caches的动作
  - 解决方法
    - 识别出drop\_cache，让系统不要drop ion pool

<http://smartgit/gerrit/#/c/2339949/>

```
xref: /bootable/recovery/applypatch/applypatch.cpp (revision cb6163ec)
Home Project | History | Annotate | Line# | Scopes# | Navigate# | Raw | Download
218 int WriteToPartition(const unsigned char* data, size_t len, const std::string&
219   std::vector<std::string> pieces = android::base::Split(target, ":");
220   if (pieces.size() < 2 || pieces[0] != "EMMC") {
221     printf("WriteToPartition called with bad target (%s)\n", target.c_str());
222     return -1;
223   }
224
225   const char* partition = pieces[1].c_str();
226   unique_fd fd(ota_open(partition, O_RDWR));
227   if (fd == -1) {
228     printf("failed to open %s: %s\n", partition, strerror(errno));
229     return -1;
230   }
231
232   fd.reset(ota_open(partition, O_RDONLY));
233   if (fd == -1) {
234     printf("failed to reopen %s for verify: %s\n", partition, strerror(errno));
235     return -1;
236   }
237
238   // Drop caches so our subsequent verification read won't just be reading the cache.
239   sync();
240   unique_fd dc(ota_open("/proc/sys/vm/drop_caches", O_WRONLY));
241   if (TEMP_FAILURE_RETRY(ota_write(dc, "3\n", 2)) == -1) {
242     printf("write to /proc/sys/vm/drop_caches failed: %s\n", strerror(errno));
243   } else {
244     printf("  caches dropped\n");
245   }
246 }
```

# ION内存观察

- 原生版本，观察手机使用一天后，内存分布
  - 内存碎片化很严重



# 手机使用一天后 内存分布(原生版本)

- 试用用户使用一天后，内存碎片化严重

```
TOTAL: 393008 kb
0 order 8 highmem pages uncached 0 total
0 order 8 lowmem pages uncached 0 total
0 order 4 highmem pages uncached 0 total
0 order 4 lowmem pages uncached 0 total
0 order 0 highmem pages uncached 0 total
21142 order 0 lowmem pages uncached 86597632 total
0 order 8 highmem pages cached 0 total
0 order 8 lowmem pages cached 0 total
0 order 4 highmem pages cached 0 total
0 order 4 lowmem pages cached 0 total
0 order 0 highmem pages cached 0 total
7948 order 0 lowmem pages cached 32555008 total
MemTotal: 7630440 kB
MemFree: 1643300 kB
MemAvailable: 3539716 kB
Buffers: 2752 kB
Cached: 1989220 kB
SwapCached: 15212 kB
Active: 2511576 kB
Inactive: 1307548 kB
Active(anon): 1503572 kB
Inactive(anon): 322452 kB
Active(file): 1008004 kB
Inactive(file): 985096 kB
NR_ION: 509368 kB
free_ion: 116360 kB
free_ion_pool: 116360 kB
free_ion_heap: 0 kB
NR_GPU: 485968 kB
free_gpu: 238968 kB
zram_size: 248988 kB
zcache_size: 159812 kB
pcppages: 3232 kB
ALL_MEM: 7655268 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 48410 52049 20180 3429 649 54 5 1 0 0 0
Node 0, zone Normal, type Movable 1 9142 24866 1875 53 0 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 830 355 203 85 9 0 0 0 0 0 0
Node 0, zone Normal, type HighAtomic 32 23 12 16 14 5 2 1 0 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 703 1284 23 1 37 0
7630440 1643052 2752 1989220 2084 2097148 1471068 1823212 988512 120484 263488 77988 93832 116360 238968 159812 0 0 0

free_gpu: 238968 kB
zram_size: 248988 kB
zcache_size: 159812 kB
pcppages: 3232 kB
ALL_MEM: 7655268 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 48410 52049 20180 3429 649 54 5 1 0 0 0
Node 0, zone Normal, type Movable 1 9142 24866 1875 53 0 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 830 355 203 85 9 0 0 0 0 0 0
Node 0, zone Normal, type HighAtomic 32 23 12 16 14 5 2 1 0 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0
```

# ION内存观察

- 预留700MB内存给相机使用，观察下面请看的内存分布
  - 刚刚开机
  - T card或者OTA第一次升级后
  - Monkey压力测试9小时后
  - Monkey压力测试15小时后
- 大大减少相机启动慢的问题
- 预留的700MB内存仍然被回收，需要进一步优化

# 预留700MB内存

- 刚刚开机，系统内存存在大片可用的连续物理内存。因此，相机启动速度很快

```
TOTAL: 422464 kb
0 order 9 highmem pages uncached 0 total
0 order 9 lowmem pages uncached 0 total
0 order 4 highmem pages uncached 0 total
0 order 4 lowmem pages uncached 0 total
0 order 3 highmem pages uncached 0 total
1 order 3 lowmem pages uncached 32768 total
0 order 2 highmem pages uncached 0 total
1 order 2 lowmem pages uncached 16384 total
0 order 1 highmem pages uncached 0 total
1 order 1 lowmem pages uncached 8192 total
0 order 0 highmem pages uncached 0 total
1 order 0 lowmem pages uncached 4096 total
0 order 9 highmem pages cached 0 total
345 order 9 lowmem pages cached 723517440 total
0 order 4 highmem pages cached 0 total
0 order 4 lowmem pages cached 0 total
0 order 3 highmem pages cached 0 total
0 order 3 lowmem pages cached 0 total
0 order 2 highmem pages cached 0 total
0 order 2 lowmem pages cached 0 total
0 order 1 highmem pages cached 0 total
0 order 1 lowmem pages cached 0 total
0 order 0 highmem pages cached 0 total
0 order 0 lowmem pages cached 0 total
MemTotal: 7630408 kB
MemFree: 1197772 kB
MemAvailable: 4128424 kB
Buffers: 6536 kB
Cached: 3046700 kB
SwapCached: 15676 kB
Active: 1705596 kB
Inactive: 2672876 kB
Active(anon): 983660 kB
Inactive(anon): 350024 kB
Active(file): 721936 kB
Inactive(file): 2322852 kB
Unevictable: 3444 kB
NR_DMA_CMA: 40208 kB
NR_ION: 1129084 kB
free_ion: 706620 kB
free_ion_pool: 706620 kB
free_ion_heap: 0 kB
NR_GPU: 300404 kB
free_gpu: 189148 kB
zram_size: 25648 kB
zcache_size: 0 kB
pcppages: 2660 kB
ALL_MEM: 7632988 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 546 439 285 156 131 84 71 42 28 34 156
Node 0, zone Normal, type Movable 8604 5214 3110 1241 422 128 34 9 0 0 39
Node 0, zone Normal, type Reclaimable 1 10 23 13 2 1 1 0 1 1 0
Node 0, zone Normal, type HighAtomic 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 714 1275 22 0 37 0
7630408 1198020 6536 3046700 6104 2097148 1951228 1321856 998836 85660 195260 55360 77828 0 189204 0 0 0 0
15:34:15 up 3 min, 0 users, load average: 6.56, 3.94, 1.62
```

# 预留700MB内存

- T card升级第一次开机内存分布，连续大片可用物理内存比开机少很多

```
TOTAL: 399720 kb
0 order 9 highmem pages uncached 0 total
19 order 9 lowmem pages uncached 39845888 total
0 order 4 highmem pages uncached 0 total
105 order 4 lowmem pages uncached 6881280 total
0 order 3 highmem pages uncached 0 total
0 order 3 lowmem pages uncached 0 total
0 order 2 highmem pages uncached 0 total
1 order 2 lowmem pages uncached 16384 total
0 order 1 highmem pages uncached 0 total
3 order 1 lowmem pages uncached 24576 total
0 order 0 highmem pages uncached 0 total
8 order 0 lowmem pages uncached 32768 total
0 order 9 highmem pages cached 0 total
151 order 9 lowmem pages cached 316669952 total
0 order 4 highmem pages cached 0 total
31 order 4 lowmem pages cached 2031616 total
0 order 3 highmem pages cached 0 total
1 order 3 lowmem pages cached 32768 total
0 order 2 highmem pages cached 0 total
0 order 2 lowmem pages cached 0 total
0 order 1 highmem pages cached 0 total
2 order 1 lowmem pages cached 16384 total
0 order 0 highmem pages cached 0 total
1 order 0 lowmem pages cached 4096 total
MemTotal: 7656072 kB
MemFree: 81960 kB
MemAvailable: 4533248 kB
Buffers: 6124 kB
Cached: 4567328 kB
SwapCached: 0 kB
Active: 2401740 kB
Inactive: 3599040 kB
Active(anon): 1325764 kB
Inactive(anon): 110204 kB
Active(file): 1075976 kB
Inactive(file): 3488836 kB
NR_DMA_CMA: 40208 kB
NR_ION: 756708 kB
free_ion: 356988 kB
free_ion_pool: 356988 kB
free_ion_heap: 0 kB
NR_GPU: 162076 kB
free_gpu: 75928 kB
zram_size: 12 kB
zcache_size: 99128 kB
pcppages: 3100 kB
ALL_MEM: 7678108 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 1 34 17 23 37 17 12 7 2 2 0
Node 0, zone Normal, type Movable 5 5 20 441 634 76 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 30 24 6 3 2 1 0 1 0 1 0
Node 0, zone Normal, type HighAtomic 68 26 7 1 1 1 1 0 1 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 575 1415 20 1 37 0
7656072 88784 6124 4557296 6292 2097148 2097148 1430928 992608 87876 182324 55792 74944 0 75928 101184 0 0 0
10:24:26 up 3 min, 0 users, load average: 15.11, 5.55, 2.06
```

# 预留700MB内存

- Monkey压力测试9小时后
  - 分配150MB左右的ion内存只需17.6ms(monkey测试数小时后出现), 相比发生问题时候, 分配需要60+ms时间, 时间降低了40+ms

```
<7>[34067.495242][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] TAAISP_DMA memory size (aligned) : 00a15860
<7>[34067.495760][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] ME_DRC memory size (aligned) : 01040000
<7>[34067.507401][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] TNR_DMA memory size: 07a73000
<7>[34067.511767][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] ORBMCH_DMA memory size: 00997540
<7>[34067.512824][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] CLAE_DMA memory size: 00870000
<7>[34067.514141][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] TAAISP_DMA memory kva:0xffffffff805a400000, dva: 0x0000000017800000
<7>[34067.517078][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] ME_DRC memory kva:0xffffffff805bc00000, dva: 0x0000000018300000
<7>[34067.519480][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] TNR_DMA memory kva:0xffffffff805cc41000, dva: 0x0000000019400000
<7>[34067.523574][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] ORBMCH_DMA memory kva:0xffffffff80646b5000, dva: 0x0000000020f00000
<7>[34067.525177][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] CLAE_DMA memory kva:0xffffffff8067000000, dva: 0x0000000021900000
<7>[34067.525185][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] is_resourcemanr_init_dynamic_mem done
<7>[34067.525535][12-10 09:00:34][7:dualFrameFactor:10521][@][RSC] rsctype: 0, rsccount: device[1], core[1]
<7>[34067.589213][12-10 09:00:34][2:dualFrameFactor:10521][@][RSC] is_ischain_runtime_resume: QoS LOCK [INT_CAM(444000)] [TNR(444000)] [INT(666000), MIF(1539000), CAM(533000), HPG(1)]
<7>[34067.589609][12-10 09:00:34][2:dualFrameFactor:10521][@][RSC] rsctype: 6, rsccount: device[1], core[2]
<7>[34067.594908][12-10 09:00:34][2:FrameFactoryCre:10520][@][RSC] rsctype: 3, rsccount: device[1], core[3]
<7>[34067.628134][12-10 09:00:34][6:FrameFactoryCre:10520][@][RSC] rsctype: 6, rsccount: device[2], core[4]
<7>[34067.631464][12-10 09:00:34][4:dualFrameFactor:10521][@][RSC] rsctype: 5, rsccount: device[1], core[5]
<7>[34067.660594][12-10 09:00:34][6:dualFrameFactor:10521][@][RSC] rsctype: 6, rsccount: device[3], core[6]
<7>[34067.662832][12-10 09:00:34][6:dualFrameFactor:10521][@][RSC] rsctype: 6, rsccount: device[4], core[7]
<7>[34067.666384][12-10 09:00:34][4:FrameFactoryCre:10520][@][RSC] rsctype: 6, rsccount: device[5], core[8]
<7>[34067.669108][12-10 09:00:34][6:dualFrameFactor:10521][@][RSC] rsctype: 6, rsccount: device[6], core[9]
<7>[34067.916069][12-10 09:00:34][7:FrameFactorySta:10581][@][RSC] is_dvfs_sel_table(0):0
```

512.8 - 495.2 =  
17.6ms

# 预留700MB内存

- Monkey测试9小时后，手机内存碎片化严重
  - 已经找不到order 4(64KB)的连续物理内存了
  - adb shell cat /d/ion/heaps/ion\_system\_heap;
  - cat /proc/meminfo;cat /proc/pagetypeinfo;
  - cat /proc/meminfo\_quick
- ○

```
TOTAL: 680388 kb
0 order 9 highmem pages uncached 0 total
0 order 9 lowmem pages uncached 0 total
0 order 4 highmem pages uncached 0 total
1198 order 4 lowmem pages uncached 78512128 total
0 order 3 highmem pages uncached 0 total
1632 order 3 lowmem pages uncached 53477376 total
0 order 2 highmem pages uncached 0 total
8 order 2 lowmem pages uncached 131072 total
0 order 1 highmem pages uncached 0 total
10 order 1 lowmem pages uncached 81920 total
0 order 0 highmem pages uncached 0 total
13 order 0 lowmem pages uncached 53248 total
0 order 9 highmem pages cached 0 total
8 order 9 lowmem pages cached 16777216 total
0 order 4 highmem pages cached 0 total
1160 order 4 lowmem pages cached 76021760 total
0 order 3 highmem pages cached 0 total
13163 order 3 lowmem pages cached 431325184 total
0 order 2 highmem pages cached 0 total
2564 order 2 lowmem pages cached 42008576 total
0 order 1 highmem pages cached 0 total
0 order 1 lowmem pages cached 0 total
0 order 0 highmem pages cached 0 total
0 order 0 lowmem pages cached 0 total
MemTotal: 7656072 kB
MemFree: 1139752 kB
MemAvailable: 1939836 kB
Buffers: 1912 kB
Cached: 912320 kB
SwapCached: 12116 kB
Active: 1824148 kB
Inactive: 1044012 kB
Active(anon): 1425588 kB
Inactive(anon): 537300 kB
Active(file): 398560 kB
Inactive(file): 506712 kB
NR_KMALLOC: 31552 kB
NR_VMALLOC: 199712 kB
NR_DMA_NOR: 0 kB
NR_DMA_CMA: 40208 kB
NR_ION: 1362408 kB
free_ion: 682020 kB
free_ion_pool: 682020 kB
free_ion_heap: 0 kB
NR_GPU: 736040 kB
free_gpu: 244640 kB
zram_size: 435876 kB
zcache_size: 237324 kB
pcppages: 2364 kB
ALL_MEM: 7710768 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 20875 30115 6926 171 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Movable 89108 36573 2551 18 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 825 266 93 7 0 0 0 0 0 0 0 0
Node 0, zone Normal, type HighAtomic 34 31 28 5 0 0 0 0 0 0 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 936 1050 24 1 37 0
7656072 1139752 1912 912320 6496 2097148 664112 1949852 692204 104604 302208 109200 132220 0 244640 237324 0 0 0
```

# 预留700MB内存

- Monkey测试15小时后
  - 700MB的ion free内存分布
    - $\leq$  order 1
      - 1MB左右
    - $=$  order 2
      - 212MB
    - order 3 and order 4
      - 400MB
    - order 9
      - 78MB

```
TOTAL: 595136 kb
0 order 9 highmem pages uncached 0 total
34 order 9 lowmem pages uncached 71303168 total
0 order 4 highmem pages uncached 0 total
2037 order 4 lowmem pages uncached 133496832 total
0 order 3 highmem pages uncached 0 total
2413 order 3 lowmem pages uncached 79069184 total
0 order 2 highmem pages uncached 0 total
82 order 2 lowmem pages uncached 1343488 total
0 order 1 highmem pages uncached 0 total
82 order 1 lowmem pages uncached 671744 total
0 order 0 highmem pages uncached 0 total
33 order 0 lowmem pages uncached 135168 total
0 order 9 highmem pages cached 0 total
5 order 9 lowmem pages cached 10485760 total
0 order 4 highmem pages cached 0 total
537 order 4 lowmem pages cached 35192832 total
0 order 3 highmem pages cached 0 total
4978 order 3 lowmem pages cached 163119104 total
0 order 2 highmem pages cached 0 total
13478 order 2 lowmem pages cached 220823552 total
0 order 1 highmem pages cached 0 total
2 order 1 lowmem pages cached 16384 total
0 order 0 highmem pages cached 0 total
11 order 0 lowmem pages cached 45056 total
MemTotal: 7656072 kb
MemFree: 180980 kb
MemAvailable: 1265356 kb
Buffers: 904 kb
Cached: 1210820 kb
SwapCached: 11680 kb
Active: 2389980 kb
Inactive: 1171864 kb
Active(anon): 1877816 kb
Inactive(anon): 504564 kb
Active(file): 512164 kb
Inactive(file): 667300 kb
NR_DMA_CMA: 40208 kb
NR_ION: 1294064 kb
free_ion: 698928 kb
free_ion_pool: 698928 kb
free_ion_heap: 0 kb
NR_GPU: 855800 kb
free_gpu: 23592 kb
zram_size: 482760 kb
zcache_size: 372584 kb
pcppages: 2952 kb
ALL_MEM: 7727956 kb
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 14052 11392 1837 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Movable 251 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 0 37 11 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type HighAtomic 28 23 30 15 13 4 0 0 0 0 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 983 1010 17 1 37 0
7656072 180492 904 1210820 29760 2097148 382328 2347552 836240 124472 303692 111744 144956 0 23592 372584 0 0 0
14:42:33 up 15:09, 0 users, load average: 5.50, 8.05, 9.16
```



# 预留700MB内存

- 右边为系统版本1.8.0试用机使用1天的手机
- 左边为预留700MB内存，monkey压力测试 9小时的手机





# ION内存观察

- 预留1000MB内存给相机使用，并且锁定ION内存存在ION pool，观察下面请看的内存分布
  - Monkey压力测试8小时后
  - Monkey压力测试9小时后
  - 试用用户，使用一天手机

# 预留1000MB内存

- monkey测试8小时后，仍然有1000MB ION内存保留在ION pool里面

```
TOTAL: 553752 kb
0 order 9 highmem pages uncached 0 total
17 order 9 lowmem pages uncached 35851584 total
0 order 4 highmem pages uncached 0 total
1336 order 4 lowmem pages uncached 87556096 total
0 order 3 highmem pages uncached 0 total
6 order 3 lowmem pages uncached 196608 total
0 order 2 highmem pages uncached 0 total
7 order 2 lowmem pages uncached 114688 total
0 order 1 highmem pages uncached 0 total
6 order 1 lowmem pages uncached 49152 total
0 order 0 highmem pages uncached 0 total
10 order 0 lowmem pages uncached 40960 total
0 order 9 highmem pages cached 0 total
424 order 9 lowmem pages cached 889192448 total
0 order 4 highmem pages cached 0 total
1645 order 4 lowmem pages cached 107806720 total
0 order 3 highmem pages cached 0 total
3 order 3 lowmem pages cached 98304 total
0 order 2 highmem pages cached 0 total
0 order 2 lowmem pages cached 0 total
0 order 1 highmem pages cached 0 total
0 order 1 lowmem pages cached 0 total
0 order 0 highmem pages cached 0 total
3 order 0 lowmem pages cached 12288 total
MemTotal: 7656072 kB
MemFree: 84488 kB
MemAvailable: 2422760 kB
Buffers: 2012 kB
Cached: 1353652 kB
SwapCached: 12488 kB
Active: 2478616 kB
Inactive: 1211736 kB
Active(anon): 1850844 kB
Inactive(anon): 499060 kB
Active(file): 627772 kB
Inactive(file): 712676 kB
Unevictable: 3540 kB
Mlocked: 3540 kB
SwapTotal: 2097148 kB
SwapFree: 482916 kB
Dirty: 612 kB
Writeback: 0 kB
AnonPages: 2331628 kB
Mapped: 906752 kB
NR_ION: 1648204 kB
free_ion: 1094452 kB
free_ion_pool: 1094452 kB
free_ion_heap: 0 kB
NR_GPU: 636992 kB
free_gpu: 66848 kB
zram_size: 505904 kB
zcache_size: 212820 kB
pcppages: 4048 kB
ALL_MEM: 7724184 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 132 1736 1424 1285 53 0 0 0 0 0 0
Node 0, zone Normal, type Movable 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 0 13 4 1 1 0 0 0 0 0 0
Node 0, zone Normal, type HighAtomic 76 67 40 17 2 1 0 0 0 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 1012 977 21 1 37 0
08:14:48 up 15:00, 0 users, load average: 9.59, 10.32, 11.58
```

# 预留1000MB内存

- Monkey压力测试9小时后(左边为测试前，右边为测试后)
- 仍然有1000MB ION内存保留在ION pool里面

```
[ 8] ion_system_heap system 0x1 68 swapper/0( 1) swapper/0( 1)
TOTAL: 390092 kb
0 order 9 highmem pages uncached 0 total
65 order 3 lowmem pages uncached 136314880 total
0 order 4 highmem pages uncached 0 total
5045 order 4 lowmem pages uncached 330629120 total
0 order 3 highmem pages uncached 0 total
3145 order 3 lowmem pages uncached 103055360 total
0 order 2 highmem pages uncached 0 total
974 order 2 lowmem pages uncached 15958016 total
0 order 1 highmem pages uncached 0 total
3 order 1 lowmem pages uncached 24576 total
0 order 0 highmem pages uncached 0 total
7 order 0 lowmem pages uncached 28672 total
0 order 9 highmem pages cached 0 total
423 order 9 lowmem pages cached 887095296 total
0 order 4 highmem pages cached 0 total
4030 order 4 lowmem pages cached 264110080 total
0 order 3 highmem pages cached 0 total
115 order 3 lowmem pages cached 3768320 total
0 order 2 highmem pages cached 0 total
83 order 2 lowmem pages cached 1359872 total
0 order 1 highmem pages cached 0 total
65 order 1 lowmem pages cached 532480 total
0 order 0 highmem pages cached 0 total
3 order 0 lowmem pages cached 12288 total
MemTotal: 7656040 kB
MemFree: 152172 kB
MemAvailable: 4200164 kB
Buffers: 4660 kB
Cached: 2427804 kB
SwapCached: 1996 kB
Active: 2951240 kB
Inactive: 1374248 kB
Active(anon): 1629792 kB
Inactive(anon): 272320 kB
Active(file): 1321448 kB
Inactive(file): 1101928 kB
Unevictable: 3284 kB
Mlocked: 3284 kB
SwapTotal: 2097148 kB
SwapFree: 2066428 kB
NR_ION: 2092132 kB
free_ion: 1702040 kB
free_ion_pool: 1702040 kB
free_ion_heap: 0 kB
NR_GPU: 289092 kB
free_gpu: 115900 kB
zram_size: 5436 kB
zcache_size: 102296 kB
pcppages: 2432 kB
ALL_MEM: 7683136 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 1 1 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Movable 175 1378 4729 1422 201 0 0 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 210 95 86 31 0 0 0 0 0 0 0 0
Node 0, zone Normal, type HighAtomic 7 2 0 2 1 0 0 1 1 0 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 694 1273 43 1 37 0
22:29:49 up 2 min, 0 users, load average: 9.69, 5.01, 1.95
```

```
[ 8] ion_system_heap system 0x1 68 swapper/0( 1) swapper/0( 1)
TOTAL: 588060 kb
0 order 9 highmem pages uncached 0 total
2 order 9 lowmem pages uncached 4194304 total
0 order 4 highmem pages uncached 0 total
0 order 4 lowmem pages uncached 0 total
0 order 3 highmem pages uncached 0 total
418 order 3 lowmem pages uncached 13697024 total
0 order 2 highmem pages uncached 0 total
4223 order 2 lowmem pages uncached 89189632 total
0 order 1 highmem pages uncached 0 total
3 order 1 lowmem pages uncached 24576 total
0 order 0 highmem pages uncached 0 total
6 order 0 lowmem pages uncached 24576 total
0 order 9 highmem pages cached 0 total
423 order 9 lowmem pages cached 887095296 total
0 order 4 highmem pages cached 0 total
673 order 4 lowmem pages cached 44105728 total
0 order 3 highmem pages cached 0 total
90 order 3 lowmem pages cached 2948120 total
0 order 2 highmem pages cached 0 total
3 order 2 lowmem pages cached 49152 total
0 order 1 highmem pages cached 0 total
0 order 1 lowmem pages cached 0 total
0 order 0 highmem pages cached 0 total
3 order 0 lowmem pages cached 12288 total
MemTotal: 7656040 kB
MemFree: 72500 kB
MemAvailable: 1976536 kB
Buffers: 1136 kB
Cached: 1018808 kB
SwapCached: 10528 kB
Active: 2720168 kB
Inactive: 1238160 kB
Active(anon): 2305824 kB
Inactive(anon): 648032 kB
Active(file): 414344 kB
Inactive(file): 590128 kB
Unevictable: 3592 kB
Mlocked: 3592 kB
SwapTotal: 2097148 kB
SwapFree: 427388 kB
NR_DMA_CMA: 40208 kB
NR_ION: 1585464 kB
free_ion: 997404 kB
free_ion_pool: 997404 kB
free_ion_heap: 0 kB
NR_GPU: 286164 kB
free_gpu: 1064 kB
zram_size: 545188 kB
zcache_size: 303540 kB
pcppages: 1652 kB
ALL_MEM: 7737580 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 1571 5244 928 1 4 0 0 0 0 0 0 0
Node 0, zone Normal, type Movable 2575 539 33 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 115 10 4 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type HighAtomic 33 29 12 13 0 0 0 0 0 0 0 0
Node 0, zone Normal, type CMA 24 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 843 1143 24 1 37 0
07:55:45 up 9:28, 0 users, load average: 38.09, 43.95, 37.95
```

# 预留1000MB内存

- 手机使用一天后，仍然有1000MB ION内存保留在ION pool里面。
- 使用下面命令获取：
  - adb shell cat  
/d/ion/heaps/ion\_system\_heap;cat  
/proc/meminfo;cat  
/proc/pagetypeinfo;uptime

```
TOTAL: 503036 kb
0 order 9 highmem pages uncached 0 total
10 order 9 lowmem pages uncached 20971520 total
0 order 4 highmem pages uncached 0 total
362 order 4 lowmem pages uncached 23724032 total
0 order 3 highmem pages uncached 0 total
127 order 3 lowmem pages uncached 4161536 total
0 order 2 highmem pages uncached 0 total
0 order 2 lowmem pages uncached 0 total
0 order 1 highmem pages uncached 0 total
0 order 1 lowmem pages uncached 0 total
0 order 0 highmem pages uncached 0 total
0 order 0 lowmem pages uncached 0 total
0 order 9 highmem pages cached 0 total
425 order 9 lowmem pages cached 891289600 total
0 order 4 highmem pages cached 0 total
1729 order 4 lowmem pages cached 113311744 total
0 order 3 highmem pages cached 0 total
0 order 3 lowmem pages cached 0 total
0 order 2 highmem pages cached 0 total
0 order 2 lowmem pages cached 0 total
0 order 1 highmem pages cached 0 total
0 order 1 lowmem pages cached 0 total
0 order 0 highmem pages cached 0 total
0 order 0 lowmem pages cached 0 total
MemTotal: 7656000 kB
MemFree: 1056440 kB
MemAvailable: 2830668 kB
Buffers: 1032 kB
Cached: 848104 kB
SwapCached: 12896 kB
Active: 1802816 kB
Inactive: 1108436 kB
Active(anon): 1426956 kB
Inactive(anon): 638348 kB
Active(file): 375860 kB
Inactive(file): 468088 kB
Unevictable: 3440 kB
Mlocked: 3440 kB
SwapTotal: 2097148 kB
SwapFree: 394288 kB
NR_ION: 1531804 kB
free_ion: 1028768 kB
free_ion_pool: 1028768 kB
free_ion_heap: 0 kB
NR_GPU: 418084 kB
free_gpu: 3608 kB
zram_size: 548616 kB
zcache_size: 370364 kB
pcppages: 1996 kB
ALL_MEM: 7738376 kB
Page block order: 10
Pages per block: 1024

Free pages count per migrate type at order
Node 0, zone Normal, type Unmovable 51167 11075 1695 395 91 0 0 0 0 0 0 0
Node 0, zone Normal, type Movable 126213 22067 852 34 8 0 0 0 0 0 0 0
Node 0, zone Normal, type Reclaimable 1920 518 242 96 5 0 0 0 0 0 0 0
Node 0, zone Normal, type HighAtomic 22 1 17 16 4 0 0 0 0 0 0 0
Node 0, zone Normal, type CMA 0 0 0 0 0 0 0 0 0 0 0 0
Node 0, zone Normal, type Isolate 0 0 0 0 0 0 0 0 0 0 0 0

Number of blocks type Unmovable Movable Reclaimable HighAtomic CMA Isolate
Node 0, zone Normal 893 1082 35 1 37 0
5:20:53 up 1 day, 5:01, 0 users, load average: 6.29, 5.05, 4.81
```

谢谢聆听  
欢迎加入策略中心