

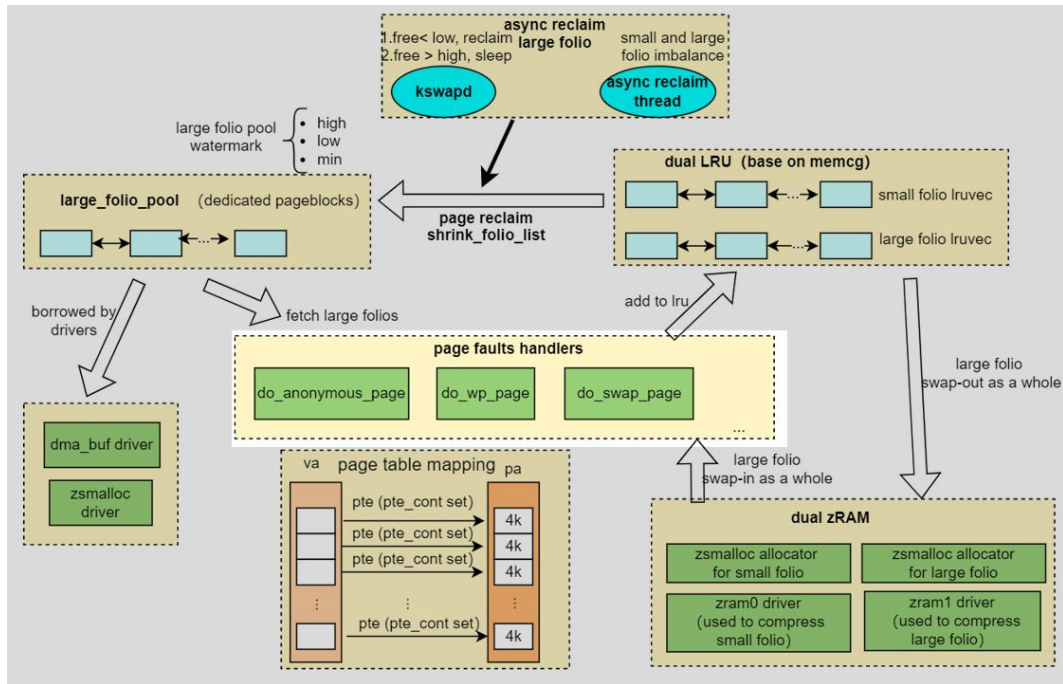
oppo

oppo

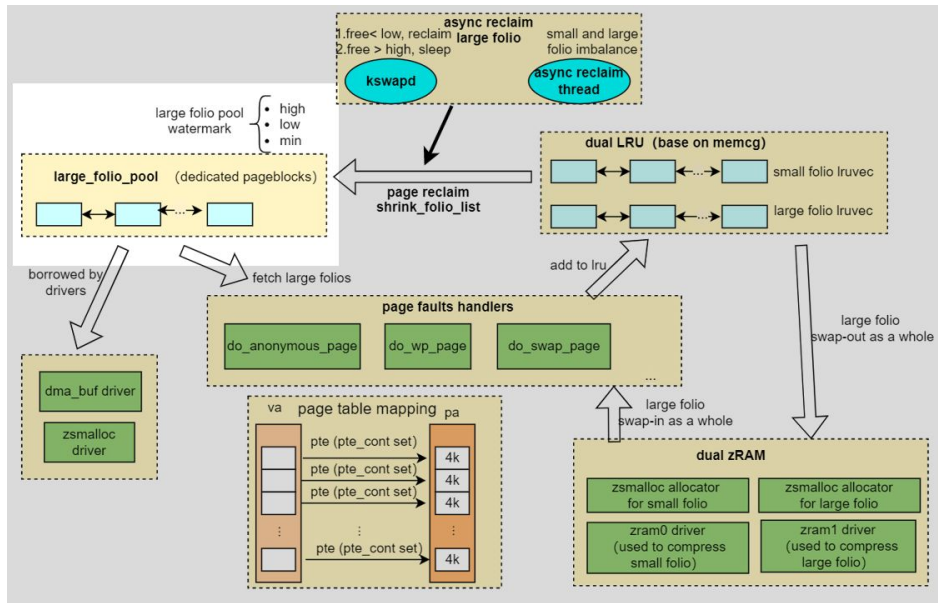
Product Practices of Large Folios on Millions of OPPO Android Phones

Software Architecture - Page Faults

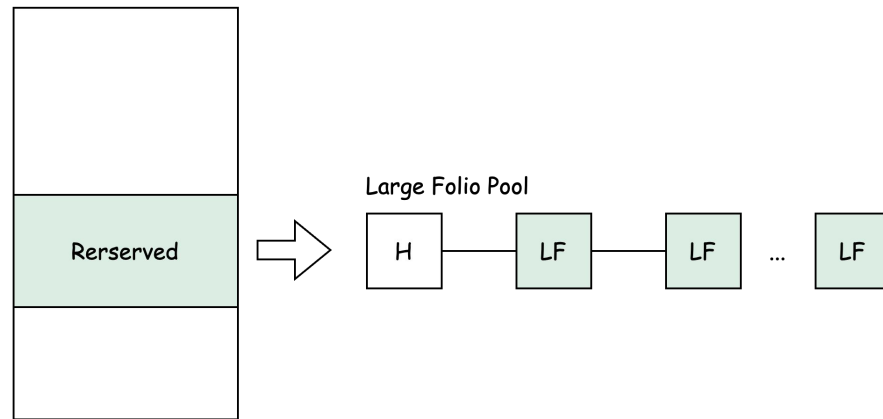
Try large folio at first, if it fails, use small folio



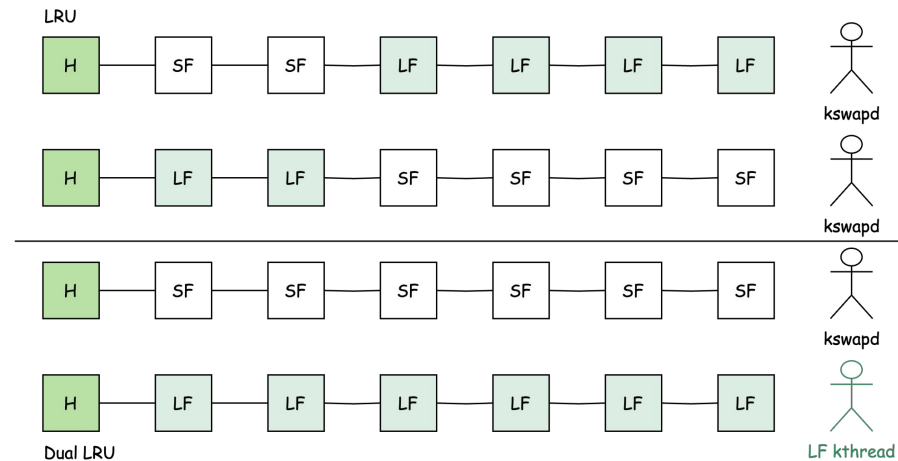
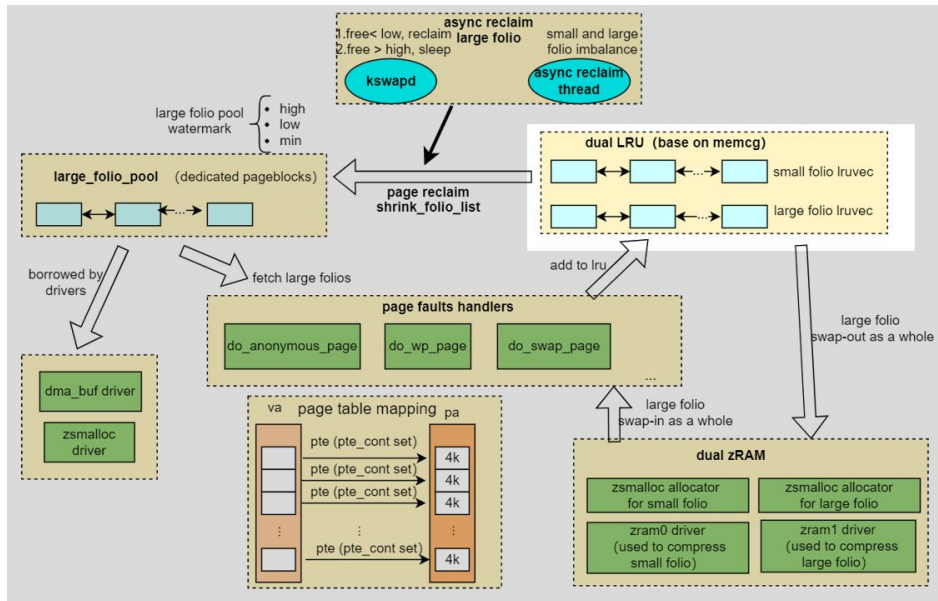
Software Architecture - Large Folio Allocator



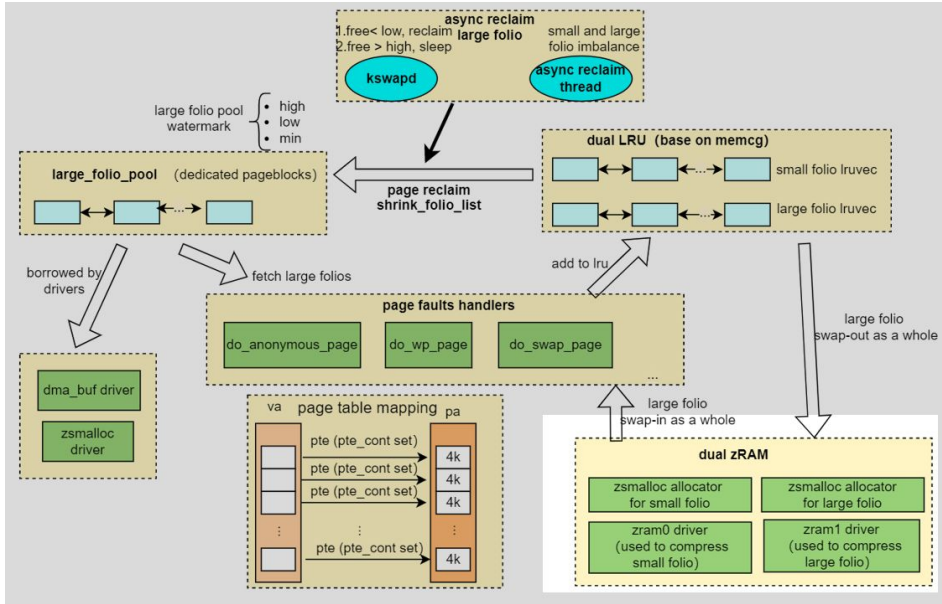
Memory



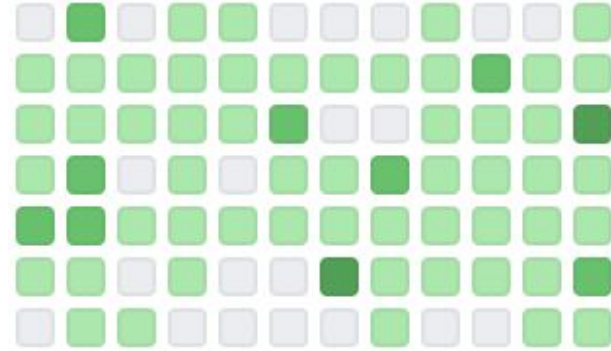
Software Architecture - Dual LRU



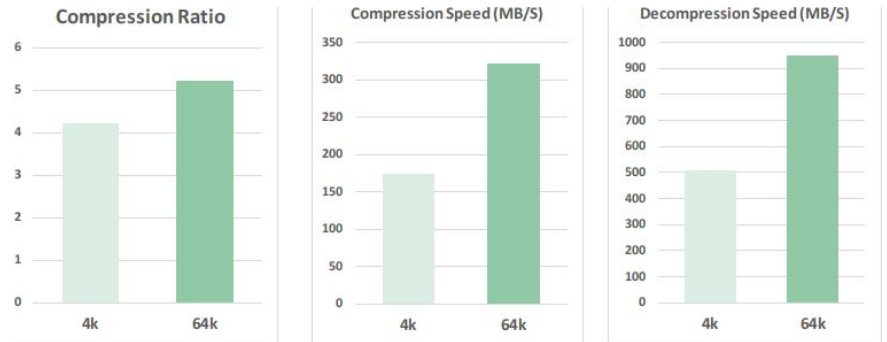
Software Architecture - Dual Zram



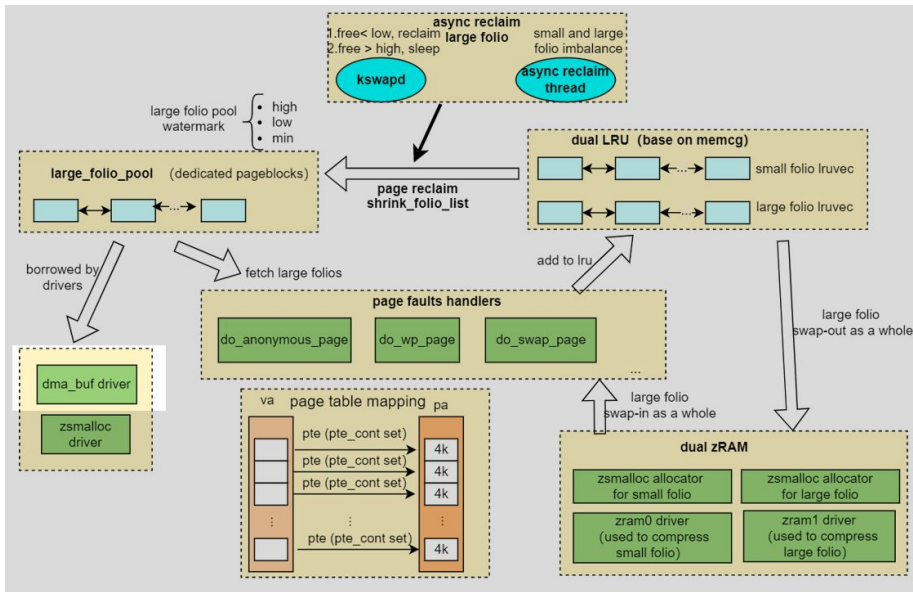
Swap Fragmentation



Performance Testing of 64k and 4k Folio with zstd Compression Algorithm



Software Architecture - System Heap



android-mainline > > common/drivers/dma-buf/heaps/system_heap.c

system_heap.c

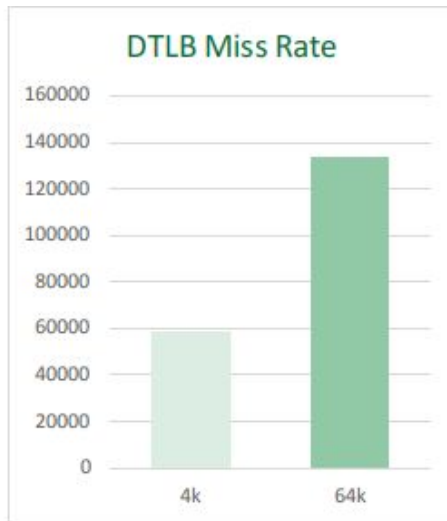
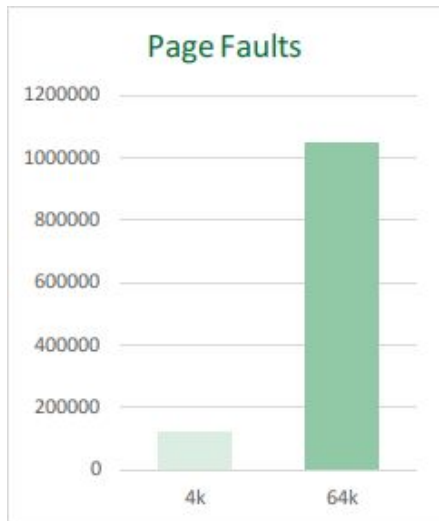
```

51 #define LOW_ORDER_GFP (GFP_HIGHUSER | __GFP_ZERO)
52 #define HIGH_ORDER_GFP (((GFP_HIGHUSER | __GFP_ZERO | __GFP_NOWARN \
53                          | __GFP_NORETRY) & ~__GFP_RECLAIM) \
54                          | __GFP_COMP)
55 static gfp_t order_flags[] = {HIGH_ORDER_GFP, HIGH_ORDER_GFP, LOW_ORDER_GFP};
56 /*
57  * The selection of the orders used for allocation (1MB, 64K, 4K) is designed
58  * to match with the sizes often found in IOMMUs. Using order 4 pages instead
59  * of order 0 pages can significantly improve the performance of many IOMMUs
60  * by reducing TLB pressure and time spent updating page tables.
61  */
62 static const unsigned int orders[] = {8, 4, 0};
63 #define NUM_ORDERS ARRAY_SIZE(orders)
    
```

Use Large Folio

Performance Improvements

Significantly improves **page faults** and **dtlb miss rate**

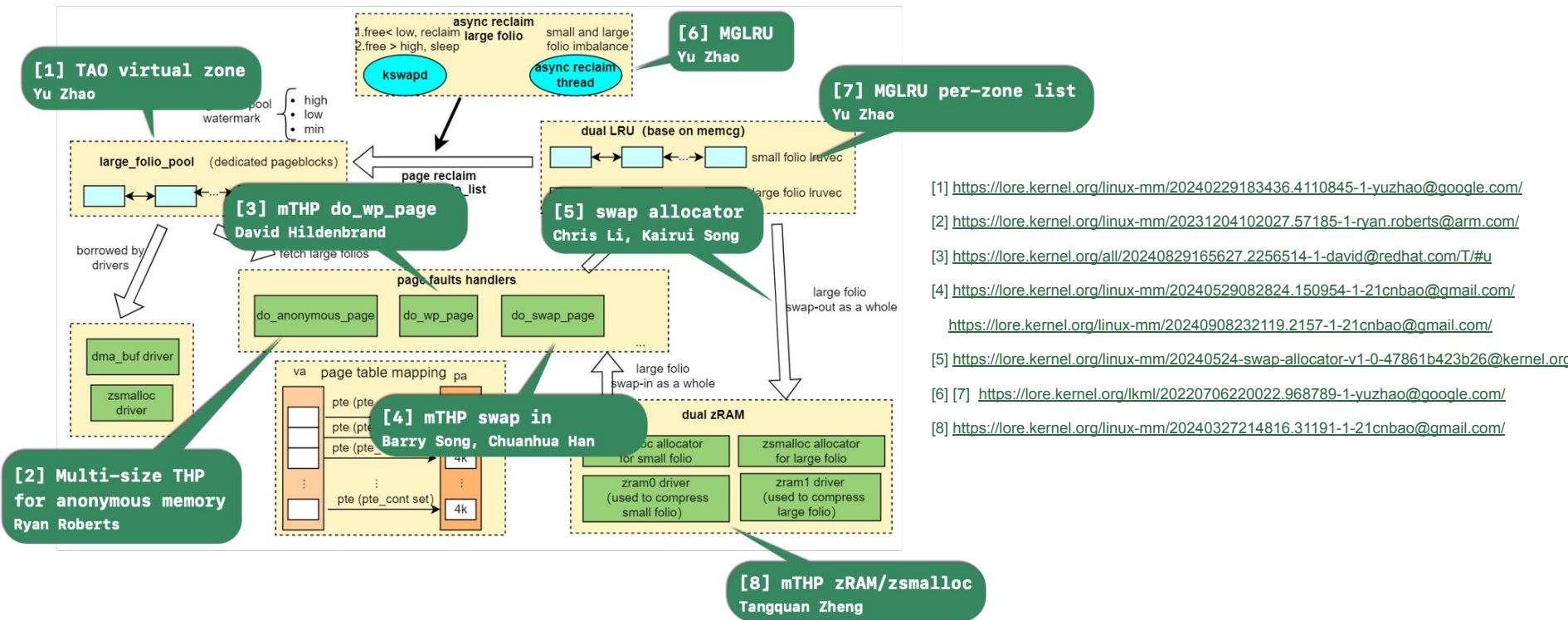


Benchmark

- **RamBench**
Improved memory access performance by an average of **10%**
- **AndroBench**
Database update performance improved by **16%**
- **AnTuTu Benchmark**
Total score increased by **22,000+**

User Scenarios

- **App Launch Speed**
Increased by **10%**
- **Frame Drop Rate**
Decreased by **35%**



[1] <https://lore.kernel.org/linux-mm/20240229183436.4110845-1-yuzhao@google.com/>
[2] <https://lore.kernel.org/linux-mm/20231204102027.57185-1-ryan.roberts@arm.com/>
[3] <https://lore.kernel.org/all/20240829165627.2256514-1-david@redhat.com/T/#u>
[4] <https://lore.kernel.org/linux-mm/20240529082824.150954-1-21cnbao@gmail.com/>
<https://lore.kernel.org/linux-mm/20240908232119.2157-1-21cnbao@gmail.com/>
[5] <https://lore.kernel.org/linux-mm/20240524-swap-allocator-v1-0-47861b423b26@kernel.org/>
[6] [7] <https://lore.kernel.org/lkml/20220706220022.968789-1-yuzhao@google.com/>
[8] <https://lore.kernel.org/linux-mm/20240327214816.31191-1-21cnbao@gmail.com/>

Merged 3132837 FROMLIST: BACKPORT: mm: swap: mTHP allocate swap entries from nonfull list

Change info

SHOW ALL ▾

SIGN IN

Submitted Jun 17

Owner Kalesh Singh

Uploader Carlos Llamas

Author Chris Li

Reviewers Suren Baghda... +2 Yu Zhao

Carlos Llamas T.J. Mercier

Chris Li Lint

Treehugger R... Performance ...

CC Treehugger R...

Repo | Branch [kernel/common](#) | [android15-6.6](#)

Hashtags [KMI-changes-for-2024-06-14](#)

Submit Requirements

- Code-Review +2
- Presubmit-Verified +1
- Lint +2 -1
- Performance +2
- Code-Owners Approved ?
- Open-Source-Licensing No votes

Links [builds](#)
[automerger](#)

FROMLIST: BACKPORT: mm: swap: mTHP allocate swap entries from nonfull list

Track the nonfull cluster as well as the empty cluster on lists. Each order has one nonfull cluster list.

The cluster will remember which order it was used during new cluster allocation.

When the cluster has free entry, add to the nonfull[order] list. When the free cluster list is empty, also allocate from the nonempty list of that order.

This improves the mTHP swap allocation success rate.

There are limitations if the distribution of numbers of

▾ SHOW ALL

Comments 1 resolved

Checks No results

Relation chain

SHOW ALL (48) ▾

- ^ [ANDROID: 2024/06/14 KMI update](#) ✓ (Merged)
- [ANDROID: Enable CONFIG_STACKTRACE_BUILD_ID=y](#) ✓ (Merged)
- [FROMLIST: BACKPORT: THP shattering: the reverse of collapsing](#) ✓ (Merged)
- [FROMLIST: BACKPORT: THP zones: the use cases of policy zones](#) ✓ (Merged)
- [ANDROID: ABI: mm: swap: reserve cluster according to mount option.](#) ✓ (Merged)
- [FROMLIST: BACKPORT: mm: swap: mTHP allocate swap entries from nonfull list](#) ✓ (Merged)
- [FROMLIST: BACKPORT: mm: swap: swap cluster switch to double link list](#) ✓ (Merged)
- [BACKPORT: FROMGIT: dm: optimize flushes](#) ✓ (Merged)
- [ANDROID: vendor_hooks: add inode as param to android_rvh_ctl_dirty_rate](#) ✓ (Merged)
- [ANDROID: vendor_hooks: modify vendor hook for page protect](#) ✓ (Merged)
- ▾ [ANDROID: GKI: Add cgroup ABI padding](#) ✓ (Merged)

Submitted together

SHOW ALL (47) ▾

- ^ [ANDROID: ABI: mm: swap: reserve cluster according to mount option.](#) kernel/common | android15-6.6
- [FROMLIST: BACKPORT: mm: swap: mTHP allocate swap entries from](#) kernel/common | android15-6.6
- ▾ [FROMLIST: BACKPORT: mm: swap: swap cluster switch to double link](#) kernel/common | android15-6.6

User Space:

- Native/Dalvik Allocator (mTHP awareness)

Kernel Space:

- FileSystem: F2FS/EROFS