mTHP swap-out and swap-in

Barry Song
Chuanhua Han
Tangquan Zheng

THP_SWPOUT

oppo

✓ Previously, only PMD-mapped THP was supported, where the swap cluster size equaled the THP size.

✓ Ryan Roberts extends this in the patchset to support multiple sizes.
"Swap-out mTHP without splitting" [landed in 6.10]
https://lore.kernel.org/all/20240408183946.2991168-1-ryan.roberts@arm.com/

There is a fragmentation issue reported by Barry Song:
we need to find an empty cluster to place an mTHP that is smaller than the cluster size.

✓ Chris Li and Kairui Song partially addressed the fragmentation issue with their patchset:
"mm: swap: mTHP swap allocator based on swap cluster order."
https://lore.kernel.org/all/20240730-swap-allocator-v5-0-cb9c148b9297@kernel.org/

THP SWAPIN

oppo

- ✓ Once an mTHP was swapped out, we could no longer restore it as an mTHP because the swap-in granularity was at the small folio level.

- ✓ Re-mapping mTHP within the swap cache in do_swap_page() as a whole was implemented in Barry Song and Chuanhua Han's patchset [landed in 6.11]:

"Large folios swap-in: handle refault cases first."
https://lore.kernel.org/all/20240529082824.150954-1-21cnbao@gmail.com/

- ✓ Allocating and mapping mTHP in do_swap_page() from sync I/O devices (such as mobile phones using zRAM) is addressed in Barry Song and Chuanhua Han's patchset:

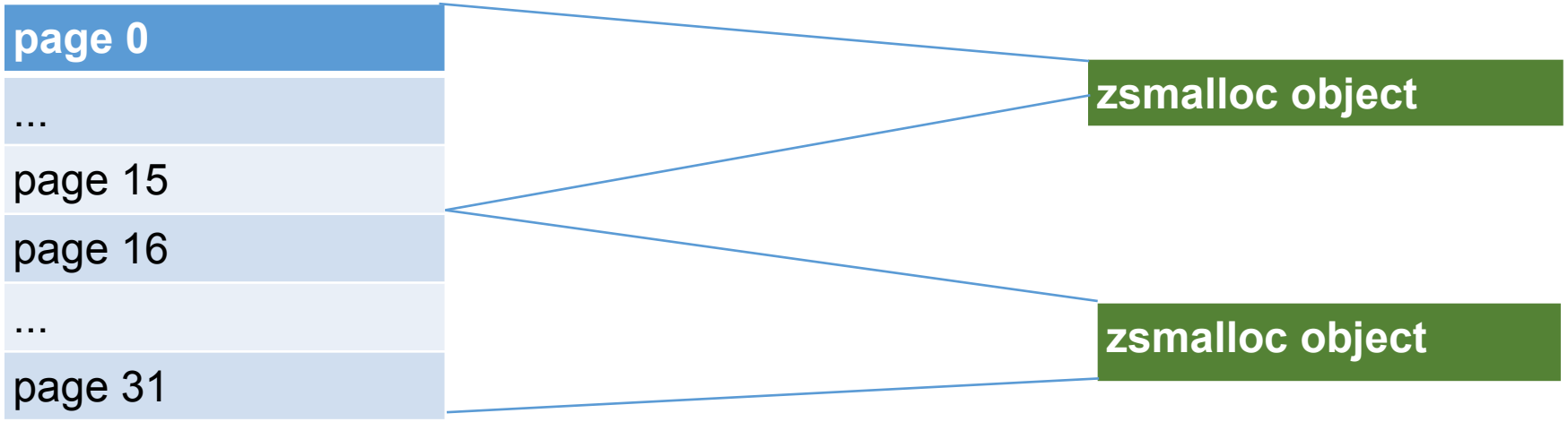
"mm: enable large folios swap-in support"
https://lore.kernel.org/all/20240908232119.2157-1-21cnbao@gmail.com/

It can only function if zswap is not enabled, which is typically the case for phones.

zRAM/zsmalloc optimization with mTHP swap

✓ Greater granularity means higher compression ratio and faster compression and decompression speeds even by CPU

“mTHP-friendly compression in zsmalloc and zram based on multi-pages” by Tangquan Zheng：
https://lore.kernel.org/all/20240327214816.31191-1-21cnbao@gmail.com/
ZSMALLOC_MULTI_PAGES_ORDER default =4
for mTHPs whose orders=5



✓ “by_n compression and decompression with Intel IAA” by Andre Glover:
Further 10x improvement in zram write latency and 7x improvement in zram read latency
https://lore.kernel.org/all/cover.1714581792.git.andre.glover@linux.intel.com/

zswap mTHP swap support

oppo

✓ “mm: ZSWAP swap-out of mTHP folios” by Kanchana P Sridhar
https://lore.kernel.org/all/20240829212705.6714-1-kanchana.p.sridhar@intel.com/

“This patch provides a sequential implementation of storing an mTHP in zswap_store() by iterating through each page in the folio to compress and store it in the zswap zpool.”

Options
Compress large folios directly instead of iterating through each subpage?

✓ TODO: ZSWAP swap-in of mTHP folios ?

Discussion(1)

✓ TODO: mTHP swap-in for non-sync IO swapfiles

What's the proper size for mTHP swap-in?

- proposed adding per-subpage readahead flags to determine the swap-in size vs. Always consider swapping in mTHP
- An additional swap-in-enabled knob is suggested but is disliked by Willy and Christoph.

/sys/kernel/mm/transparent_hugepage/hugepages-<size>kB/swapin_enabled

- reuse readahead knob?

  /proc/sys/vm/page-cluster (default 3)

  SWAP_RA_ORDER_CEILING (default 5)
- use a unified per-size enable knob, or separate enables for page cache, anon and swap?

- Should we always use `/sys/kernel/mm/transparent_hugepage/hugepages-<size>kB/enabled` for anonymous memory, swapping, and even the page cache? Also, shmem?
- This would ensure consistent folio sizes, potentially reducing TLB misses by utilizing contiguous PTEs and also minimizing memory fragmentation in systems under high memory pressure.

Ryan's "Control folio sizes used for page cache memory" suggested

`/sys/kernel/mm/transparent_hugepage/hugepages-*kB/file_enable`

https://lore.kernel.org/linux-mm/20240717071257.4141363-1-ryan.roberts@arm.com/

Discussion(2)

✓ Currently, mTHP swap-in is not permitted from multiple hybrid swap backends.

swap_read_folio() performs the read operation in the following sequence:

- swap_read_folio_zeromap()
- zswap_load()
- swap_read_folio_fs/bddev()

✓ We need an efficient method to filter swap entries coming from different sources.

- zeromap (a bitmap operation to filter partially zeromap has been there)

https://lore.kernel.org/linux-mm/20240908232119.2157-2-21cnbao@gmail.com/

- zswap (need a cheap way to filter partially zswap)
- swapcache
- swapfile