CHINA LINUX KERNEL
中国Linux内核开发者大会 | vivo

Uncached buffer IO 探索与在 f2fs 上的支持

韩棋

vivo 存储系统工程师

目录
CONTENTS

1 Uncached buffer I/O 背景简介

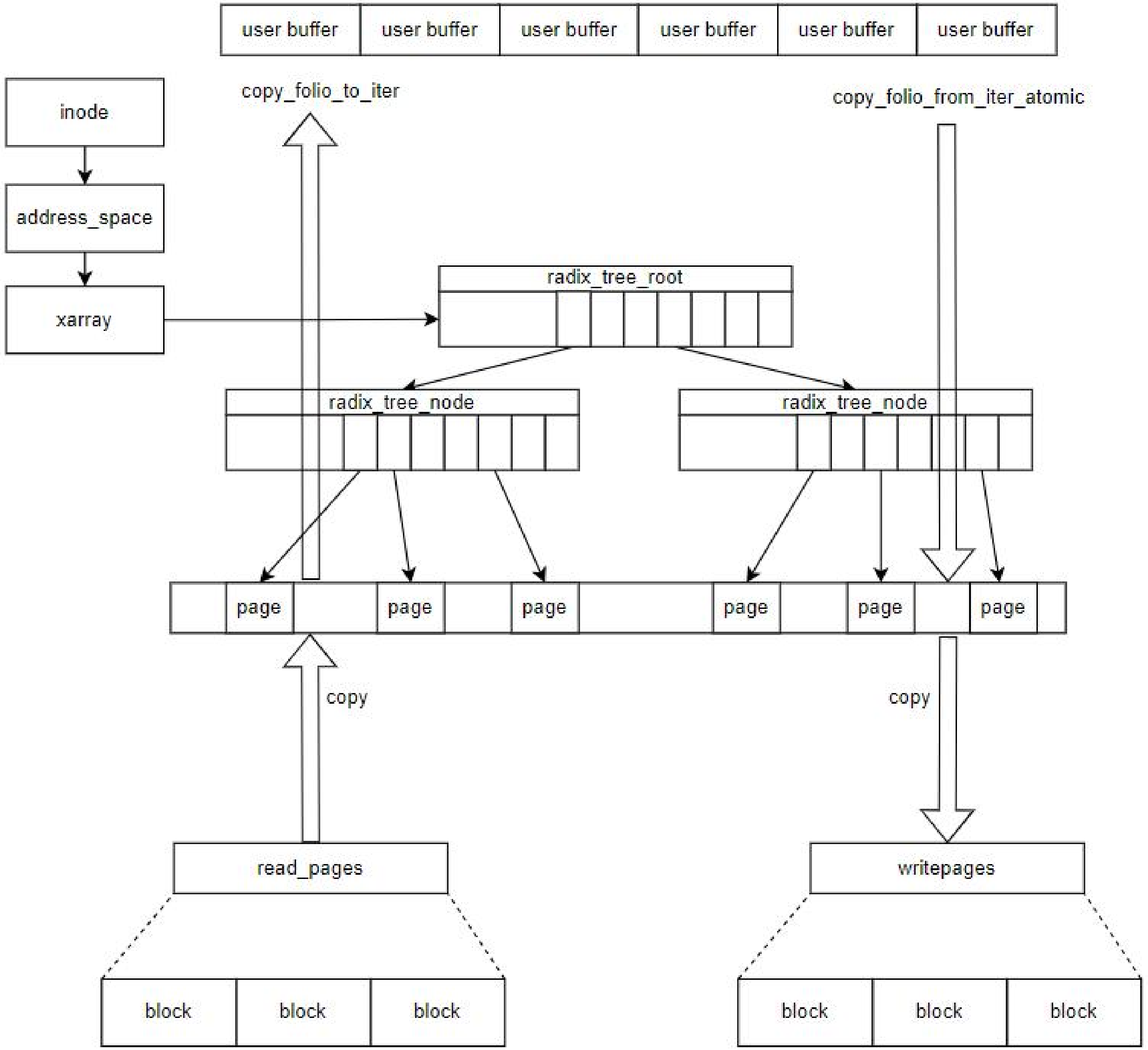
2 Uncached buffer I/O 技术原理

3 F2FS 适配 Uncached buffer I/O

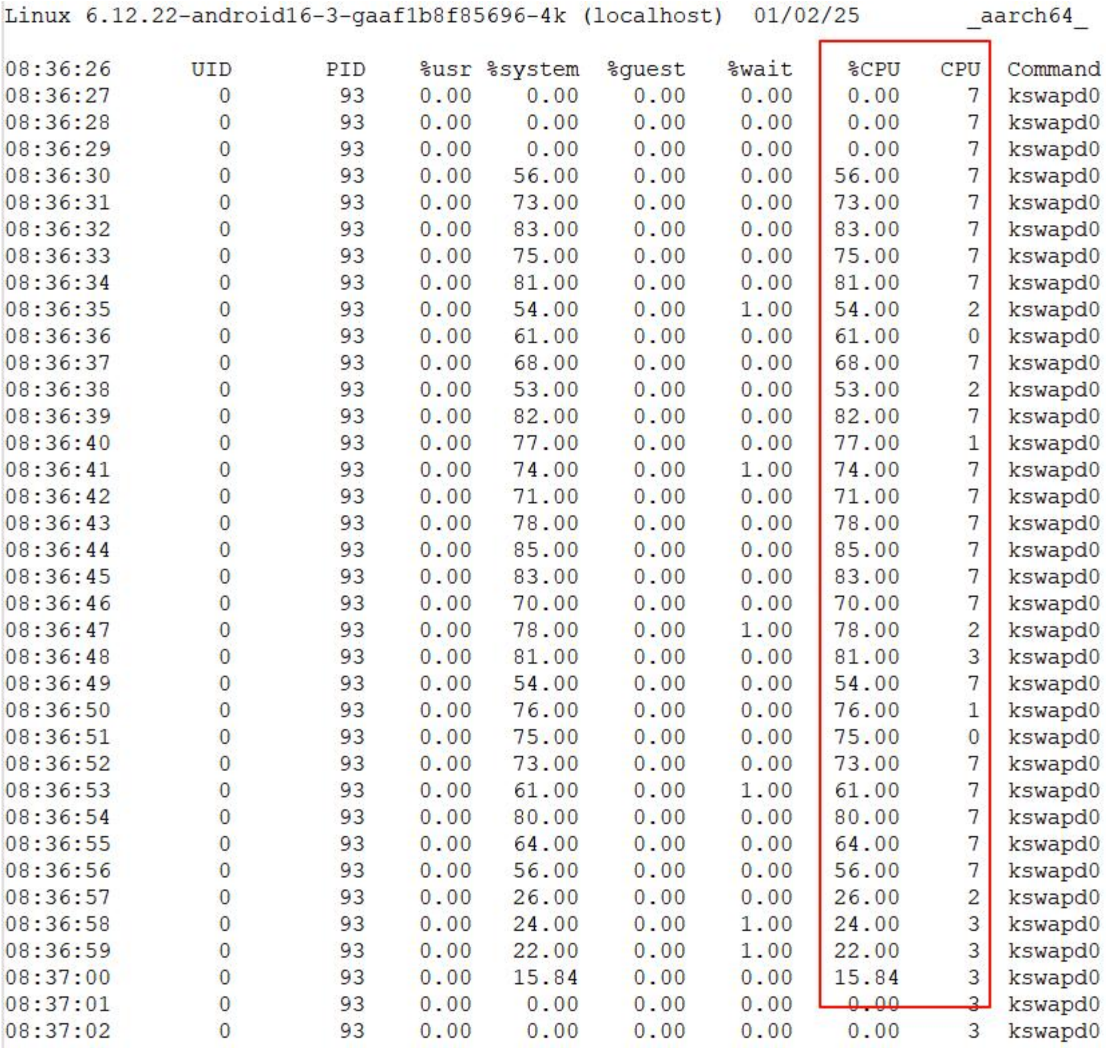
4 Uncached buffer I/O 收益

1、buffer I/O 会可能会消耗大量 CPU 资源进而影响系统性能和功耗

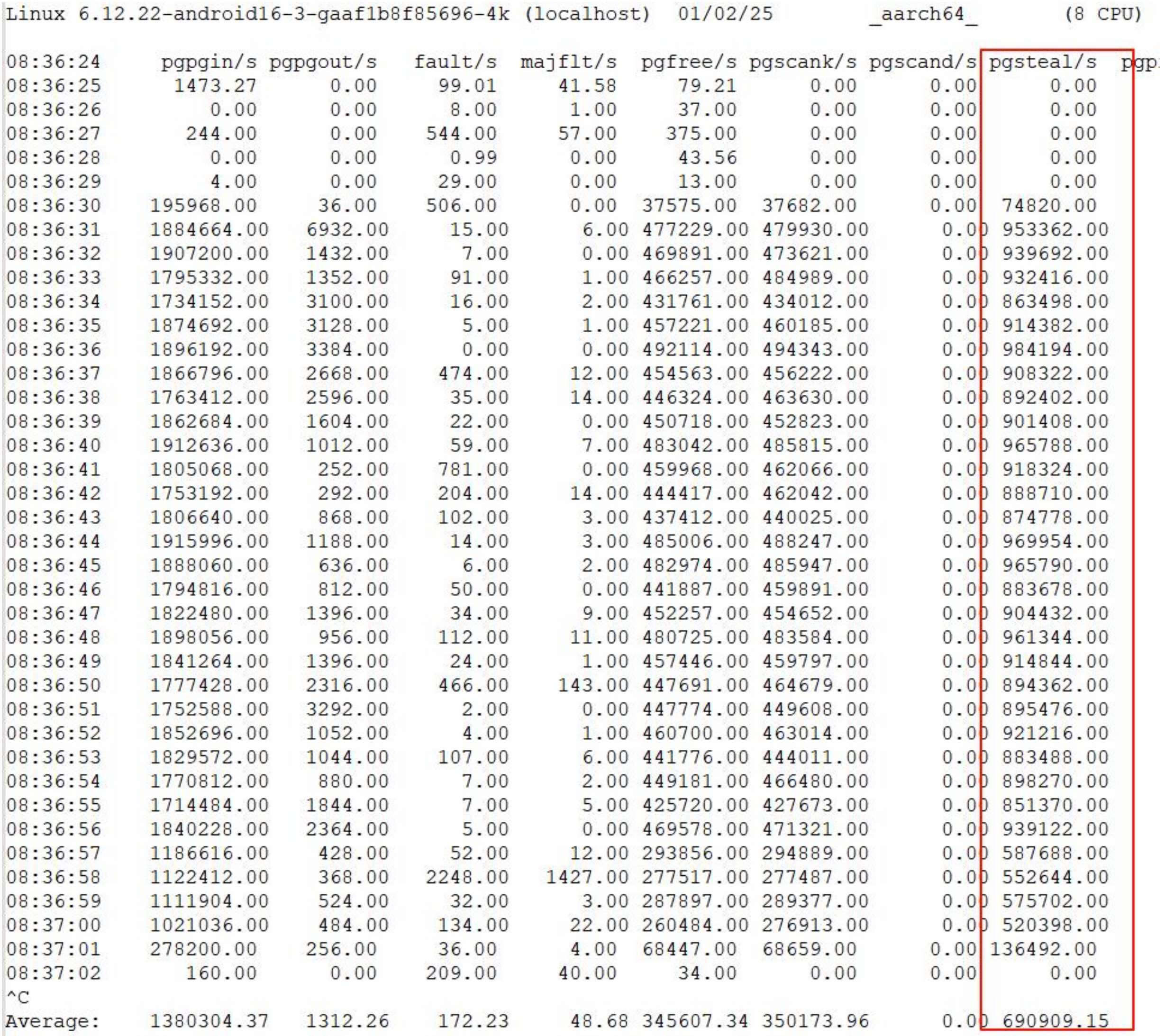
在传统的 buffer I/O 模式中，数据会被存储在内核的页缓存中，以加速未来的访问。然而，当读写大量数据时，页缓存可能会迅速填满，导致内存回收线程（如 kswapd）频繁运行，从而消耗大量 CPU 资源，影响系统性能和功耗。



buffer I/O 读场景下 kswapd 回收线程负载高，消耗 CPU 资源

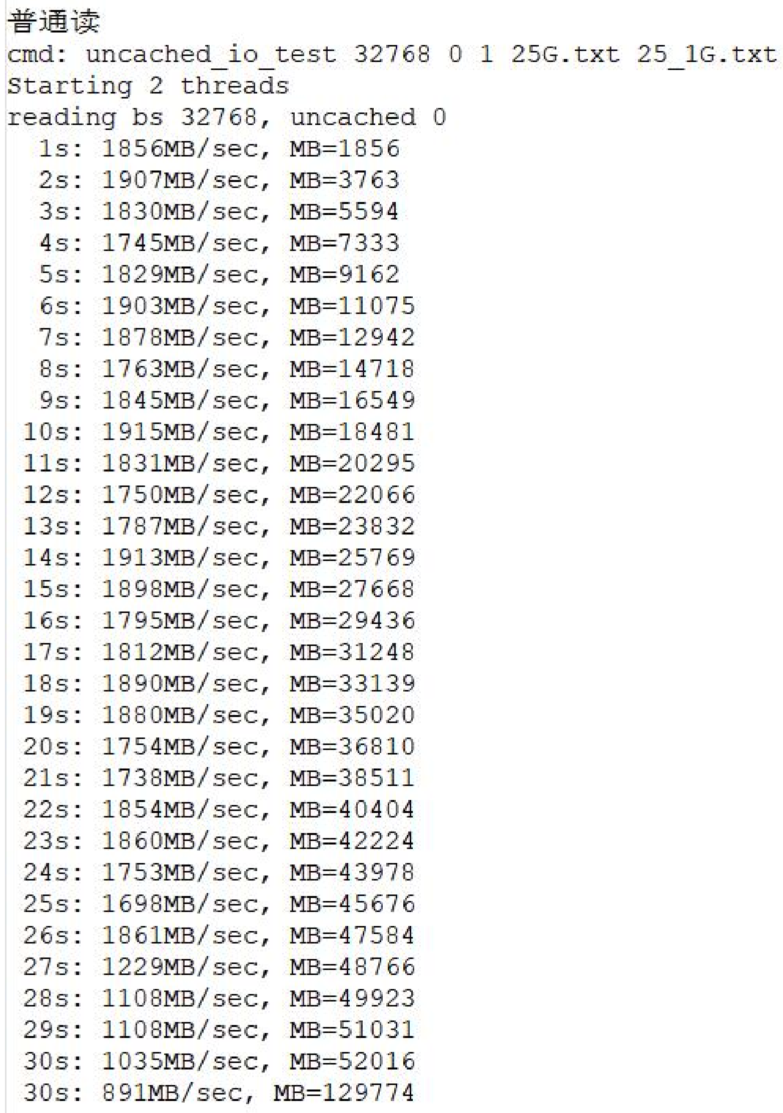


```
Linux 6.12.22-android16-3-gaaf1b8f85696-4k (localhost)  01/02/25        _aarch64_

08:36:26      UID       PID    %usr %system  %guest   %wait    %CPU   CPU  Command
08:36:27        0        93    0.00    0.00    0.00    0.00    0.00     7  kswapd0
08:36:28        0        93    0.00    0.00    0.00    0.00    0.00     7  kswapd0
08:36:29        0        93    0.00    0.00    0.00    0.00    0.00     7  kswapd0
08:36:30        0        93    0.00   56.00    0.00    0.00   56.00     7  kswapd0
08:36:31        0        93    0.00   73.00    0.00    0.00   73.00     7  kswapd0
08:36:32        0        93    0.00   83.00    0.00    0.00   83.00     7  kswapd0
08:36:33        0        93    0.00   75.00    0.00    0.00   75.00     7  kswapd0
08:36:34        0        93    0.00   81.00    0.00    0.00   81.00     7  kswapd0
08:36:35        0        93    0.00   54.00    0.00    1.00   54.00     2  kswapd0
08:36:36        0        93    0.00   61.00    0.00    0.00   61.00     0  kswapd0
08:36:37        0        93    0.00   68.00    0.00    0.00   68.00     7  kswapd0
08:36:38        0        93    0.00   53.00    0.00    0.00   53.00     2  kswapd0
08:36:39        0        93    0.00   82.00    0.00    0.00   82.00     7  kswapd0
08:36:40        0        93    0.00   77.00    0.00    0.00   77.00     1  kswapd0
08:36:41        0        93    0.00   74.00    0.00    1.00   74.00     7  kswapd0
08:36:42        0        93    0.00   71.00    0.00    0.00   71.00     7  kswapd0
08:36:43        0        93    0.00   78.00    0.00    0.00   78.00     7  kswapd0
08:36:44        0        93    0.00   85.00    0.00    0.00   85.00     7  kswapd0
08:36:45        0        93    0.00   83.00    0.00    0.00   83.00     7  kswapd0
08:36:46        0        93    0.00   70.00    0.00    0.00   70.00     7  kswapd0
08:36:47        0        93    0.00   78.00    0.00    1.00   78.00     2  kswapd0
08:36:48        0        93    0.00   81.00    0.00    0.00   81.00     3  kswapd0
08:36:49        0        93    0.00   54.00    0.00    0.00   54.00     7  kswapd0
08:36:50        0        93    0.00   76.00    0.00    0.00   76.00     1  kswapd0
08:36:51        0        93    0.00   75.00    0.00    0.00   75.00     0  kswapd0
08:36:52        0        93    0.00   73.00    0.00    0.00   73.00     7  kswapd0
08:36:53        0        93    0.00   61.00    0.00    1.00   61.00     7  kswapd0
08:36:54        0        93    0.00   80.00    0.00    0.00   80.00     7  kswapd0
08:36:55        0        93    0.00   64.00    0.00    0.00   64.00     7  kswapd0
08:36:56        0        93    0.00   56.00    0.00    0.00   56.00     7  kswapd0
08:36:57        0        93    0.00   26.00    0.00    0.00   26.00     2  kswapd0
08:36:58        0        93    0.00   24.00    0.00    1.00   24.00     3  kswapd0
08:36:59        0        93    0.00   22.00    0.00    1.00   22.00     3  kswapd0
08:37:00        0        93    0.00   15.84    0.00    0.00   15.84     3  kswapd0
08:37:01        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
08:37:02        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
```

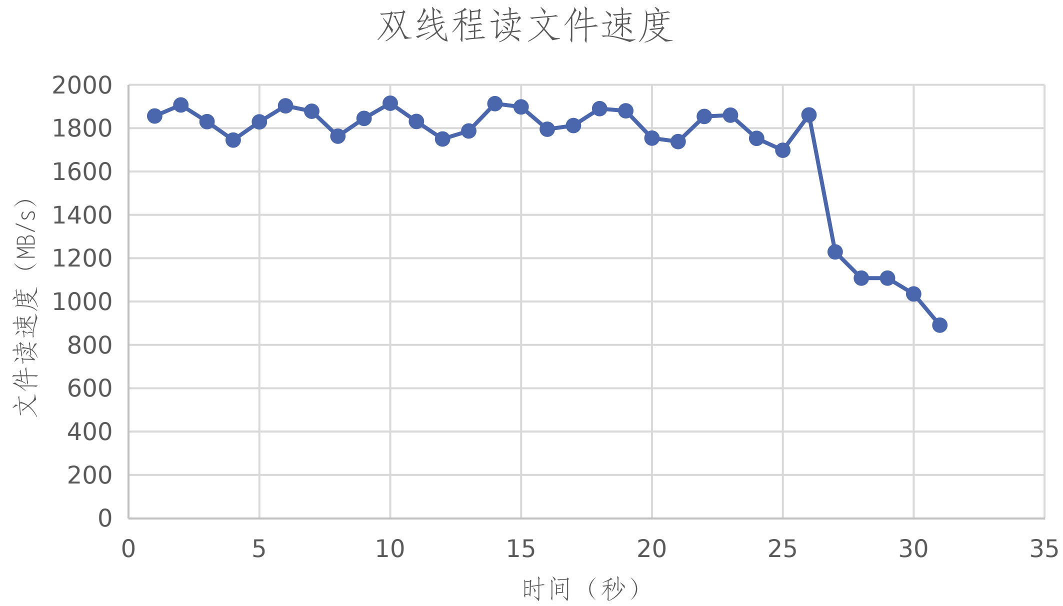


```
Linux 6.12.22-android16-3-gaaf1b8f85696-4k (localhost)  01/02/25        _aarch64_       (8 CPU)

08:36:24     pgpgin/s pgpgout/s   fault/s  majflt/s  pgfree/s pgscank/s pgscand/s pgsteal/s   pgp
08:36:25      1473.27      0.00     99.01     41.58     79.21      0.00      0.00      0.00
08:36:26         0.00      0.00      8.00      1.00     37.00      0.00      0.00      0.00
08:36:27       244.00      0.00    544.00     57.00    375.00      0.00      0.00      0.00
08:36:28         0.00      0.00      0.99      0.00     43.56      0.00      0.00      0.00
08:36:29         4.00      0.00     29.00      0.00     13.00      0.00      0.00      0.00
08:36:30    195968.00     36.00    506.00      0.00  37575.00  37682.00      0.00  74820.00
08:36:31   1884664.00   6932.00     15.00      6.00 477229.00 479930.00      0.00 953362.00
08:36:32   1907200.00   1432.00      7.00      0.00 469891.00 473621.00      0.00 939692.00
08:36:33   1795332.00   1352.00     91.00      1.00 466257.00 484989.00      0.00 932416.00
08:36:34   1734152.00   3100.00     16.00      2.00 431761.00 434012.00      0.00 863498.00
08:36:35   1874692.00   3128.00      5.00      1.00 457221.00 460185.00      0.00 914382.00
08:36:36   1896192.00   3384.00      0.00      0.00 492114.00 494343.00      0.00 984194.00
08:36:37   1866796.00   2668.00    474.00     12.00 454563.00 456222.00      0.00 908322.00
08:36:38   1763412.00   2596.00     35.00     14.00 446324.00 463630.00      0.00 892402.00
08:36:39   1862684.00   1604.00     22.00      0.00 450718.00 452823.00      0.00 901408.00
08:36:40   1912636.00   1012.00     59.00      7.00 483042.00 485815.00      0.00 965788.00
08:36:41   1805068.00    252.00    781.00      0.00 459968.00 462066.00      0.00 918324.00
08:36:42   1753192.00    292.00    204.00     14.00 444417.00 462042.00      0.00 888710.00
08:36:43   1806640.00    868.00    102.00      3.00 437412.00 440025.00      0.00 874778.00
08:36:44   1915996.00   1188.00     14.00      3.00 485006.00 488247.00      0.00 969954.00
08:36:45   1888060.00    636.00      6.00      2.00 482974.00 485947.00      0.00 965790.00
08:36:46   1794816.00    812.00     50.00      0.00 441887.00 459891.00      0.00 883678.00
08:36:47   1822480.00   1396.00     34.00      9.00 452257.00 454652.00      0.00 904432.00
08:36:48   1898056.00    956.00    112.00     11.00 480725.00 483584.00      0.00 961344.00
08:36:49   1841264.00   1396.00     24.00      1.00 457446.00 459797.00      0.00 914844.00
08:36:50   1777428.00   2316.00    466.00    143.00 447691.00 464679.00      0.00 894362.00
08:36:51   1752588.00   3292.00      2.00      0.00 447774.00 449608.00      0.00 895476.00
08:36:52   1852696.00   1052.00      4.00      1.00 460700.00 463014.00      0.00 921216.00
08:36:53   1829572.00   1044.00    107.00      6.00 441776.00 444011.00      0.00 883488.00
08:36:54   1770812.00    880.00      7.00      2.00 449181.00 466480.00      0.00 898270.00
08:36:55   1714484.00   1844.00      7.00      5.00 425720.00 427673.00      0.00 851370.00
08:36:56   1840228.00   2364.00      5.00      0.00 469578.00 471321.00      0.00 939122.00
08:36:57   1186616.00    428.00     52.00     12.00 293856.00 294889.00      0.00 587688.00
08:36:58   1122412.00    368.00   2248.00   1427.00 277517.00 277487.00      0.00 552644.00
08:36:59   1111904.00    524.00     32.00      3.00 287897.00 289377.00      0.00 575702.00
08:37:00   1021036.00    484.00    134.00     22.00 260484.00 276913.00      0.00 520398.00
08:37:01    278200.00    256.00     36.00      4.00  68447.00  68659.00      0.00 136492.00
08:37:02       160.00      0.00    209.00     40.00     34.00      0.00      0.00      0.00
^C
Average:   1380304.37   1312.26    172.23     48.68 345607.34 350173.96      0.00 690909.15
```

buffer I/O 读文件速度不稳定，本地环境存在 150MB/s 的速度波动

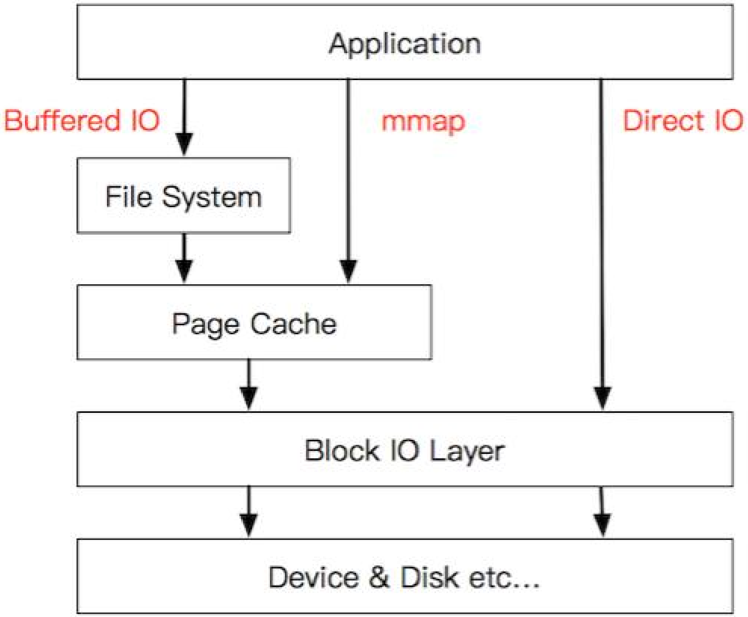


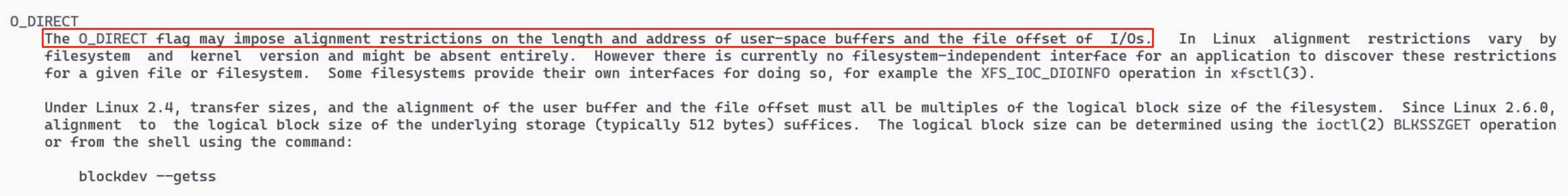
```
普通读
cmd: uncached_io_test 32768 0 1 25G.txt 25_1G.txt
Starting 2 threads
reading bs 32768, uncached 0
  1s: 1856MB/sec, MB=1856
  2s: 1907MB/sec, MB=3763
  3s: 1830MB/sec, MB=5594
  4s: 1745MB/sec, MB=7333
  5s: 1829MB/sec, MB=9162
  6s: 1903MB/sec, MB=11075
  7s: 1878MB/sec, MB=12942
  8s: 1763MB/sec, MB=14718
  9s: 1845MB/sec, MB=16549
 10s: 1915MB/sec, MB=18481
 11s: 1831MB/sec, MB=20295
 12s: 1750MB/sec, MB=22066
 13s: 1787MB/sec, MB=23832
 14s: 1913MB/sec, MB=25769
 15s: 1898MB/sec, MB=27668
 16s: 1795MB/sec, MB=29436
 17s: 1812MB/sec, MB=31248
 18s: 1890MB/sec, MB=33139
 19s: 1880MB/sec, MB=35020
 20s: 1754MB/sec, MB=36810
 21s: 1738MB/sec, MB=38511
 22s: 1854MB/sec, MB=40404
 23s: 1860MB/sec, MB=42224
 24s: 1753MB/sec, MB=43978
 25s: 1698MB/sec, MB=45676
 26s: 1861MB/sec, MB=47584
 27s: 1229MB/sec, MB=48766
 28s: 1108MB/sec, MB=49923
 29s: 1108MB/sec, MB=51031
 30s: 1035MB/sec, MB=52016
 30s: 891MB/sec, MB=129774
```



direct I/O 使用有严格内存对齐和大小限制

Direct I/O 虽然可以绕过 page cache，直接在用户空间和存储设备之间传输数据，进而避免了内存消耗，但这种方式要求严格的内存对齐和大小限制，使用起来较为复杂，且在 Android 上层无法直接被使用。

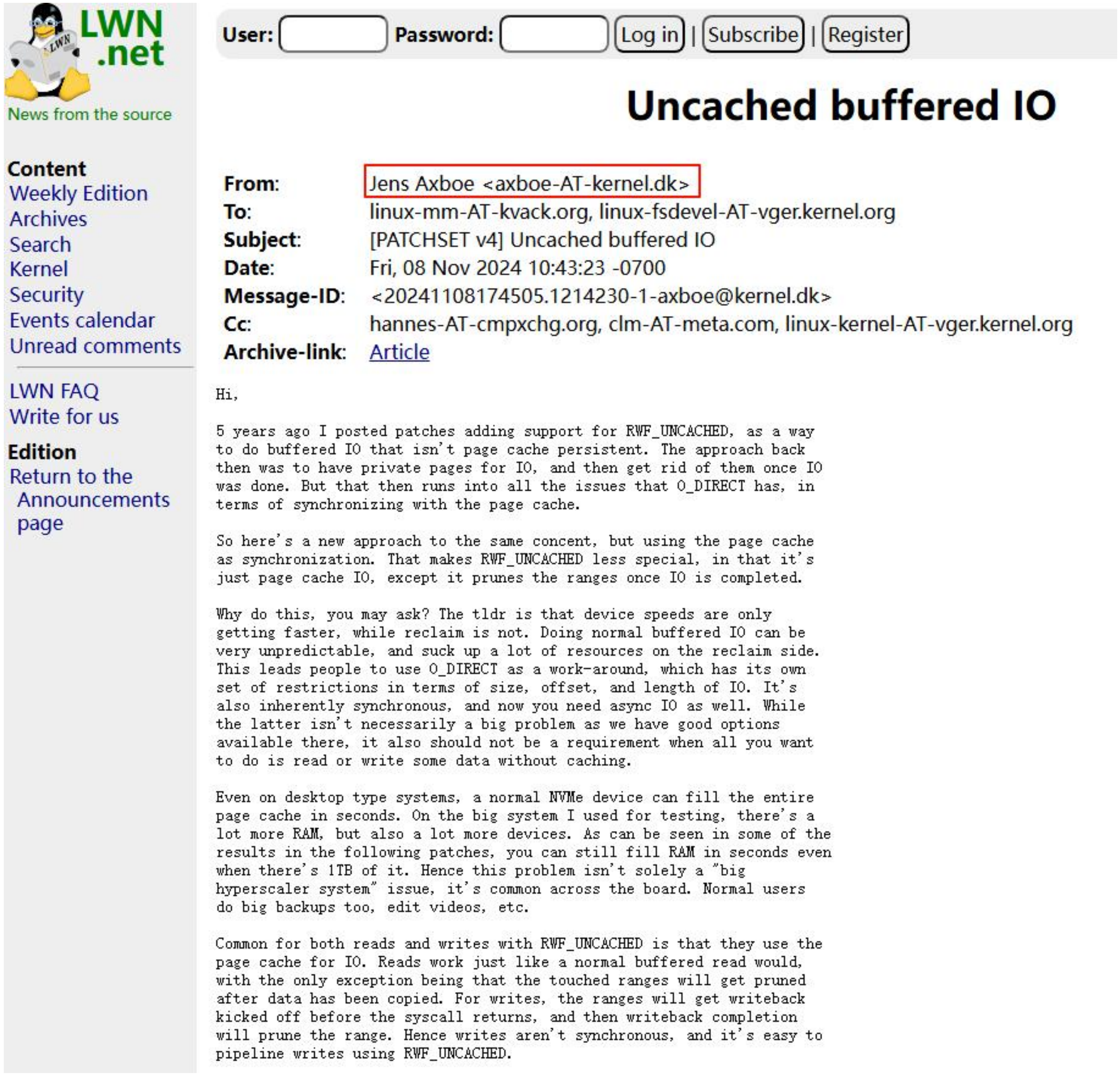




```
O_DIRECT
    The O_DIRECT flag may impose alignment restrictions on the length and address of user-space buffers and the file offset of  I/Os.   In  Linux  alignment  restrictions  vary  by
    filesystem  and  kernel  version and might be absent entirely.  However there is currently no filesystem-independent interface for an application to discover these restrictions
    for a given file or filesystem.  Some filesystems provide their own interfaces for doing so, for example the XFS_IOC_DIOINFO operation in xfsctl(3).

    Under Linux 2.4, transfer sizes, and the alignment of the user buffer and the file offset must all be multiples of the logical block size of the filesystem.  Since Linux 2.6.0,
    alignment  to  the logical block size of the underlying storage (typically 512 bytes) suffices.  The logical block size can be determined using the ioctl(2) BLKSSZGET operation
    or from the shell using the command:

        blockdev --getss
```

2、Uncached buffer I/O 诞生

为了能够使用 Buffer I/O 的便利性，同时又让 Buffer I/O 不占用 page cache，Jens Axboe 提出了 Uncached buffered I/O。



Uncached buffered IO

From:	Jens Axboe <axboe-AT-kernel.dk>
To:	linux-mm-AT-kvack.org, linux-fsdevel-AT-vger.kernel.org
Subject:	[PATCHSET v4] Uncached buffered IO
Date:	Fri, 08 Nov 2024 10:43:23 -0700
Message-ID:	<20241108174505.1214230-1-axboe@kernel.dk>
Cc:	hannes-AT-cmpxchg.org, clm-AT-meta.com, linux-kernel-AT-vger.kernel.org
Archive-link:	Article

Hi,

5 years ago I posted patches adding support for RWF_UNCACHED, as a way to do buffered IO that isn't page cache persistent. The approach back then was to have private pages for IO, and then get rid of them once IO was done. But that then runs into all the issues that O_DIRECT has, in terms of synchronizing with the page cache.

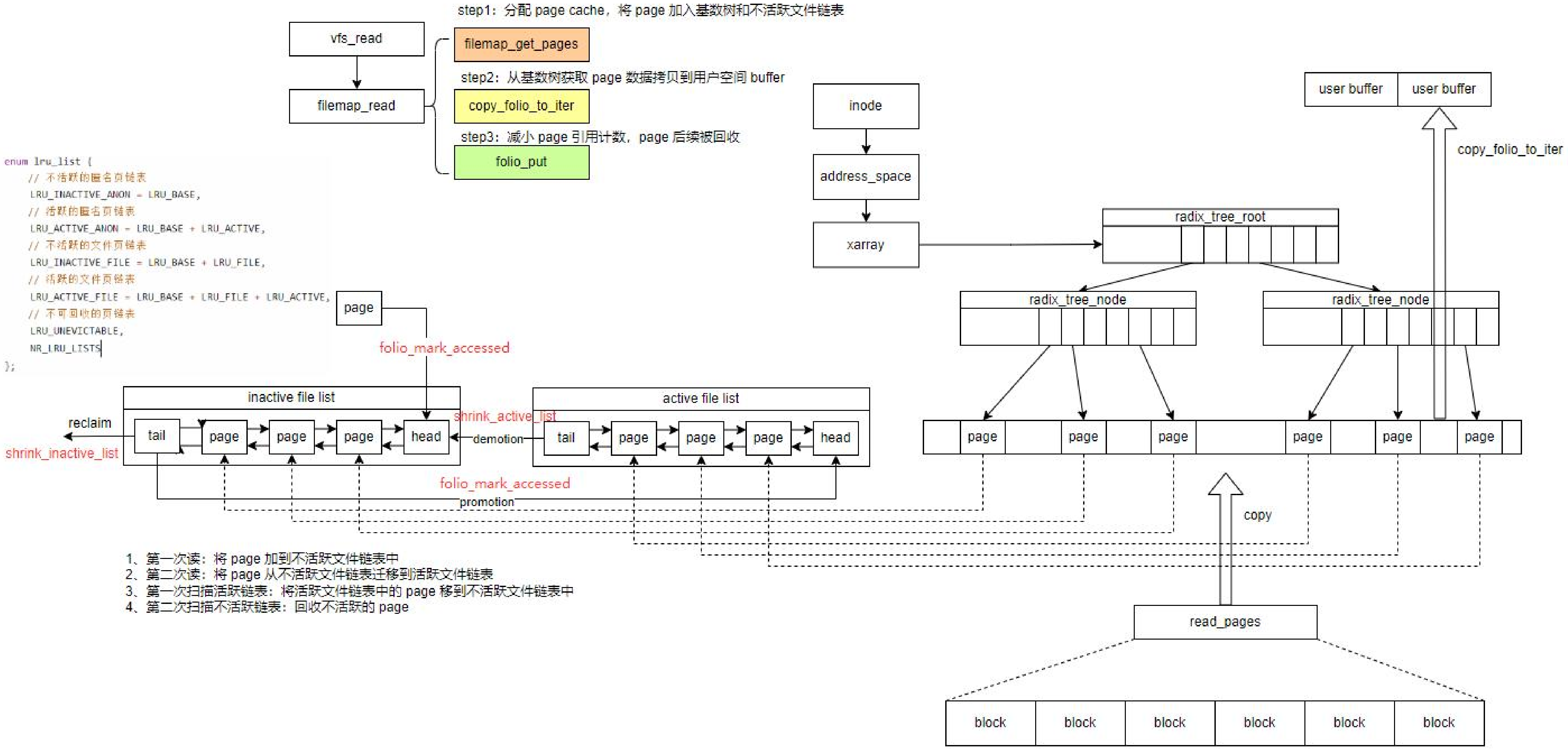
So here's a new approach to the same concent, but using the page cache as synchronization. That makes RWF_UNCACHED less special, in that it's just page cache IO, except it prunes the ranges once IO is completed.

Why do this, you may ask? The tldr is that device speeds are only getting faster, while reclaim is not. Doing normal buffered IO can be very unpredictable, and suck up a lot of resources on the reclaim side. This leads people to use O_DIRECT as a work-around, which has its own set of restrictions in terms of size, offset, and length of IO. It's also inherently synchronous, and now you need async IO as well. While the latter isn't necessarily a big problem as we have good options available there, it also should not be a requirement when all you want to do is read or write some data without caching.

Even on desktop type systems, a normal NVMe device can fill the entire page cache in seconds. On the big system I used for testing, there's a lot more RAM, but also a lot more devices. As can be seen in some of the results in the following patches, you can still fill RAM in seconds even when there's 1TB of it. Hence this problem isn't solely a "big hyperscaler system" issue, it's common across the board. Normal users do big backups too, edit videos, etc.

Common for both reads and writes with RWF_UNCACHED is that they use the page cache for IO. Reads work just like a normal buffered read would, with the only exception being that the touched ranges will get pruned after data has been copied. For writes, the ranges will get writeback kicked off before the syscall returns, and then writeback completion will prune the range. Hence writes aren't synchronous, and it's easy to pipeline writes using RWF_UNCACHED.

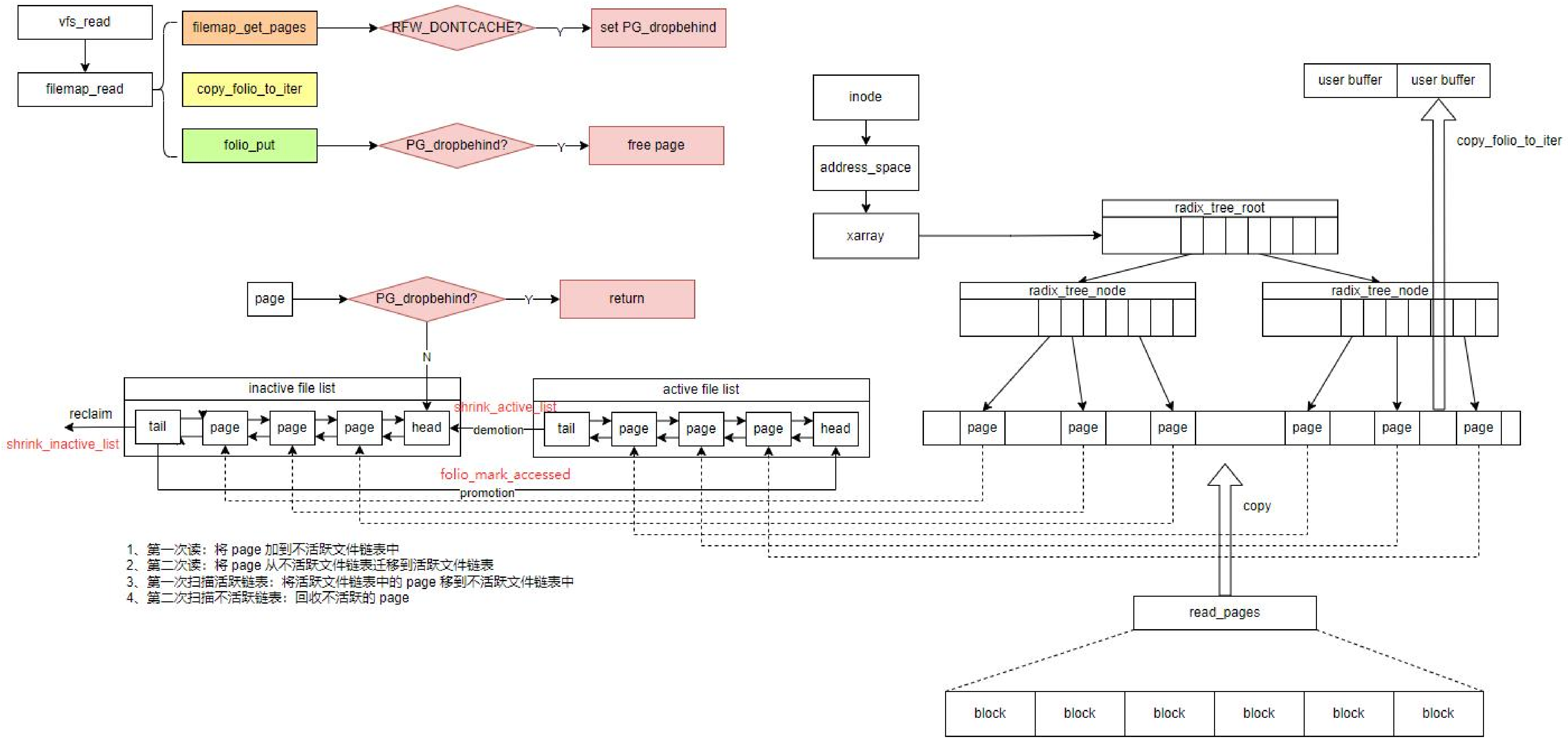
内核通用读文件与 page cache 回收流程



```
enum lru_list {
    // 不活跃的匿名页链表
    LRU_INACTIVE_ANON = LRU_BASE,
    // 活跃的匿名页链表
    LRU_ACTIVE_ANON = LRU_BASE + LRU_ACTIVE,
    // 不活跃的文件页链表
    LRU_INACTIVE_FILE = LRU_BASE + LRU_FILE,
    // 活跃的文件页链表
    LRU_ACTIVE_FILE = LRU_BASE + LRU_FILE + LRU_ACTIVE,
    // 不可回收的页链表
    LRU_UNEVICTABLE,
    NR_LRU_LISTS
};
```

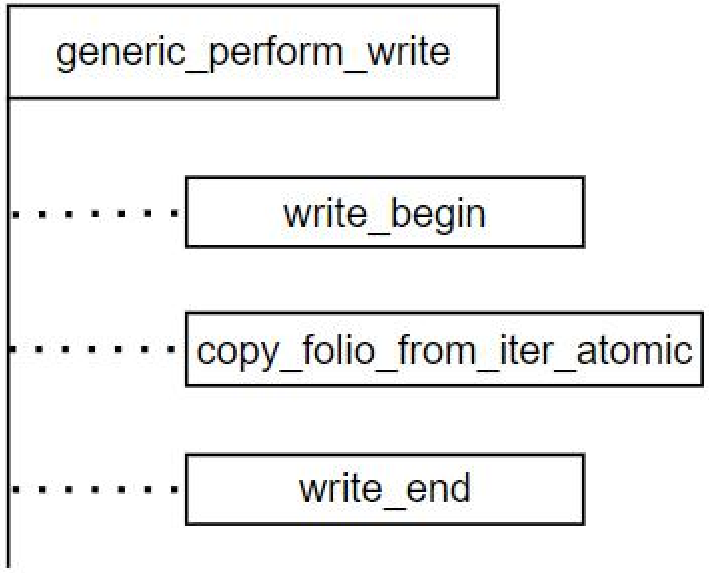
1、第一次读：将 page 加到不活跃文件链表中
2、第二次读：将 page 从不活跃文件链表迁移到活跃文件链表
3、第一次扫描活跃链表：将活跃文件链表中的 page 移到不活跃文件链表中
4、第二次扫描不活跃链表：回收不活跃的 page

uncached buffer io 读文件实现原理

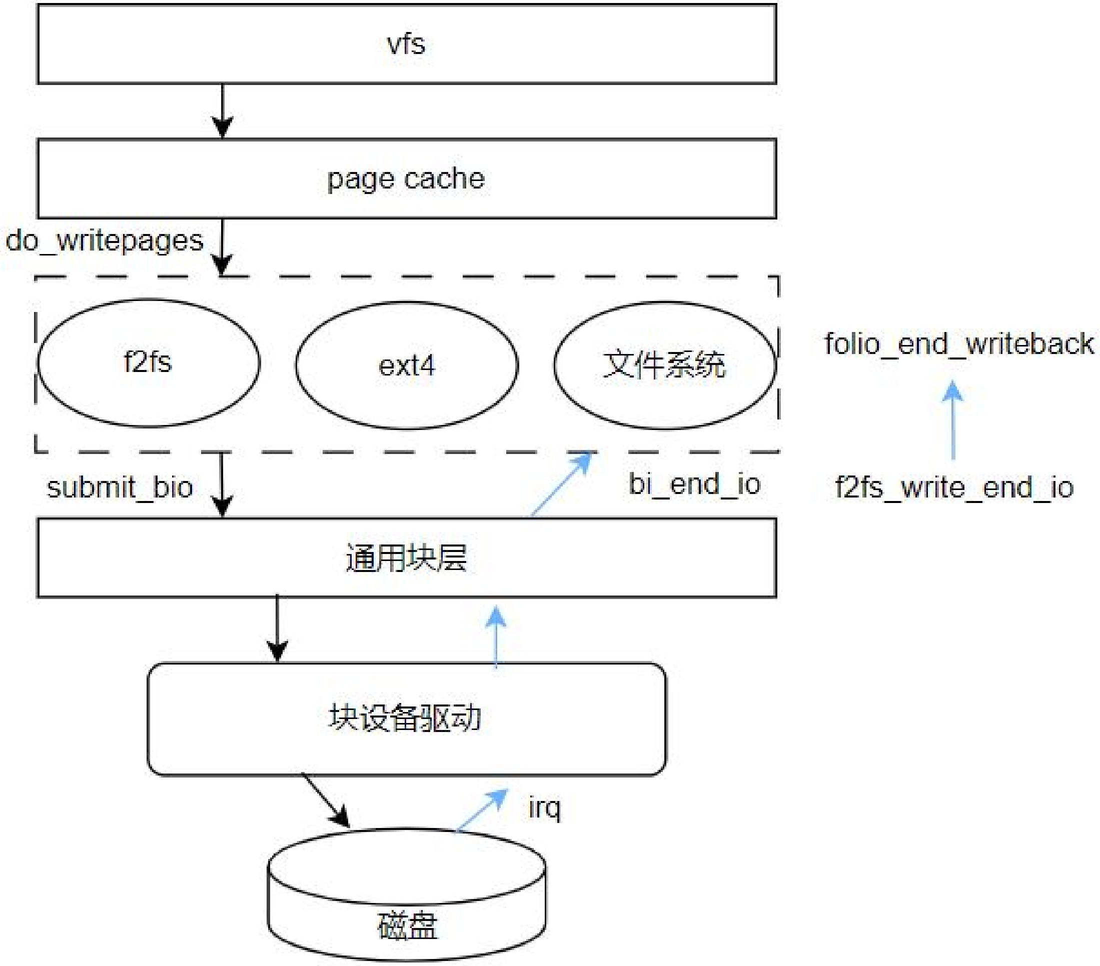


1、第一次读：将 page 加到不活跃文件链表中
2、第二次读：将 page 从不活跃文件链表迁移到活跃文件链表
3、第一次扫描活跃链表：将活跃文件链表中的 page 移到不活跃文件链表中
4、第二次扫描不活跃链表：回收不活跃的 page

内核通用写文件与 page cache 回收流程

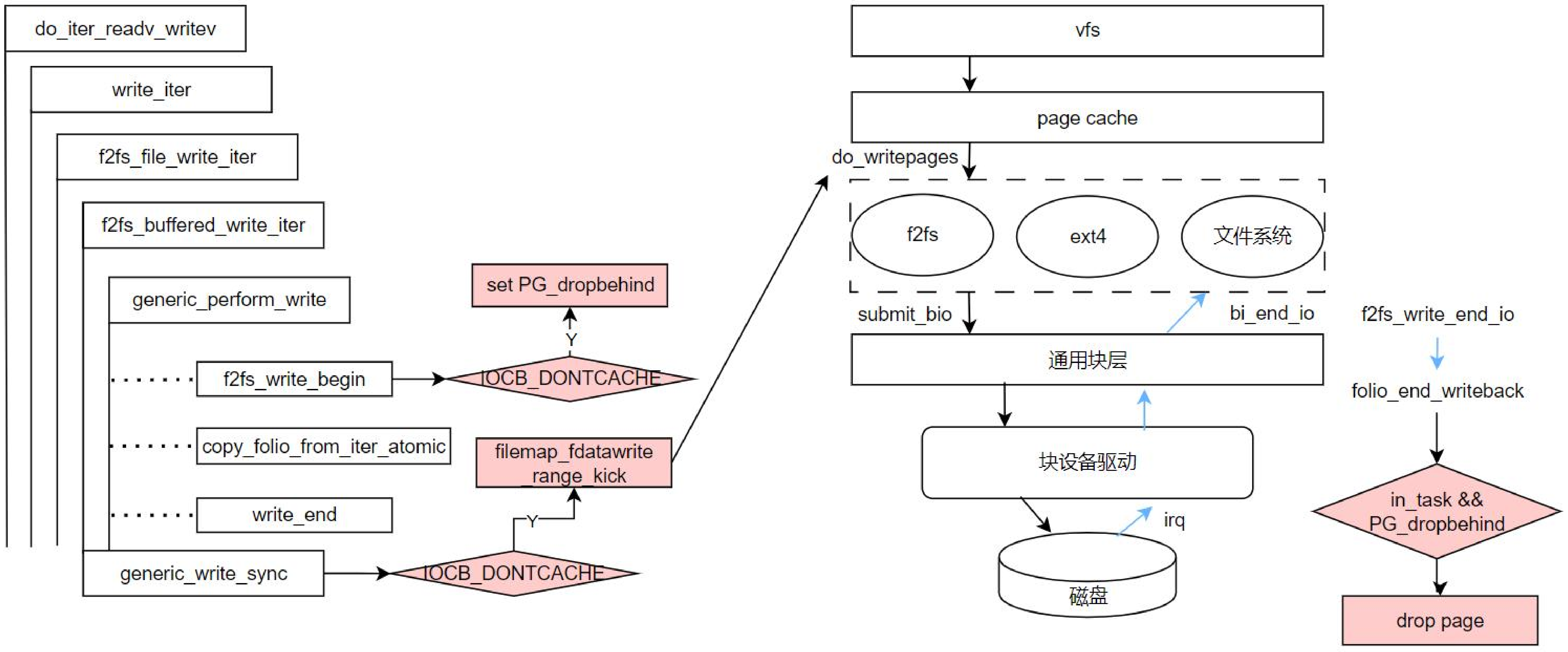


文件数据写内存

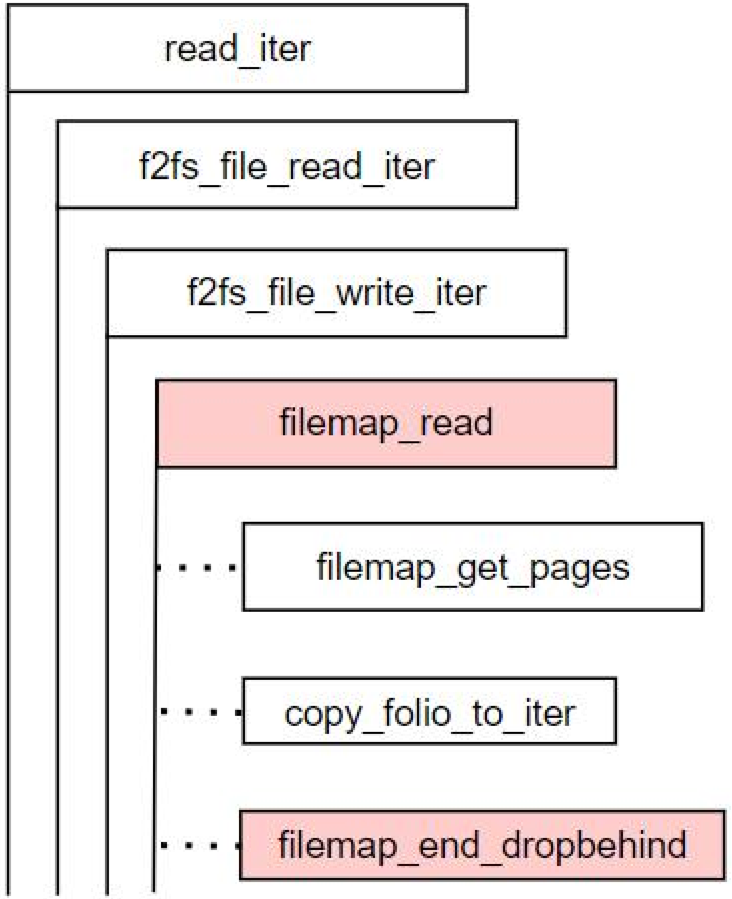


文件数据写磁盘

uncached buffer io 写文件实现原理

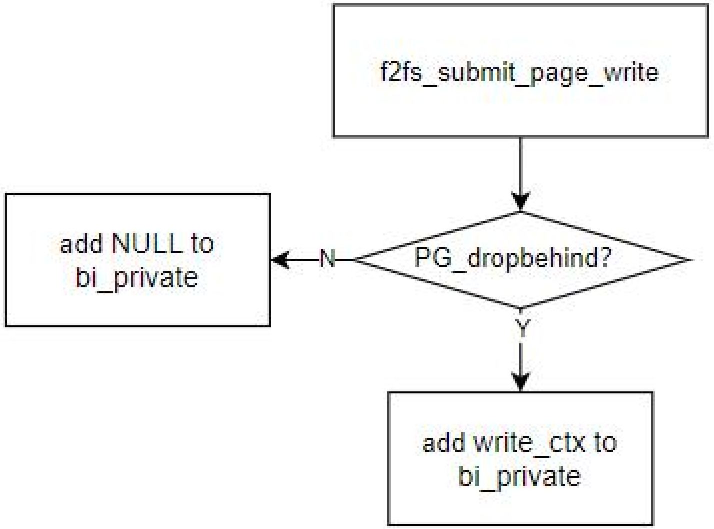


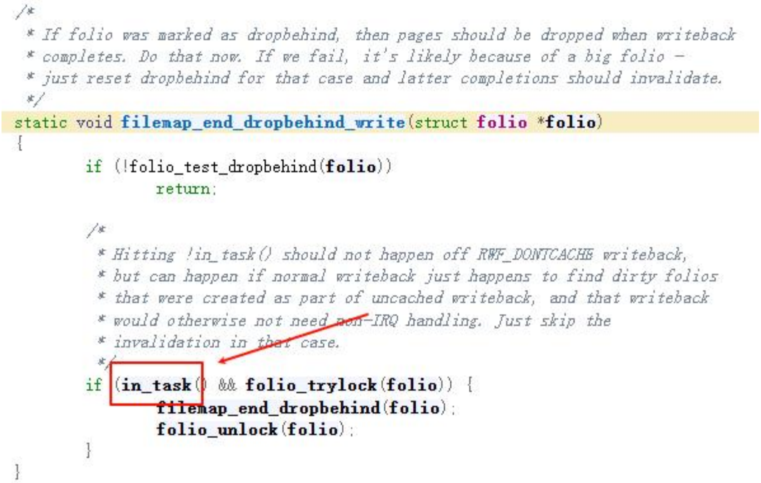
3、f2fs 适配 Uncached buffer I/O 读



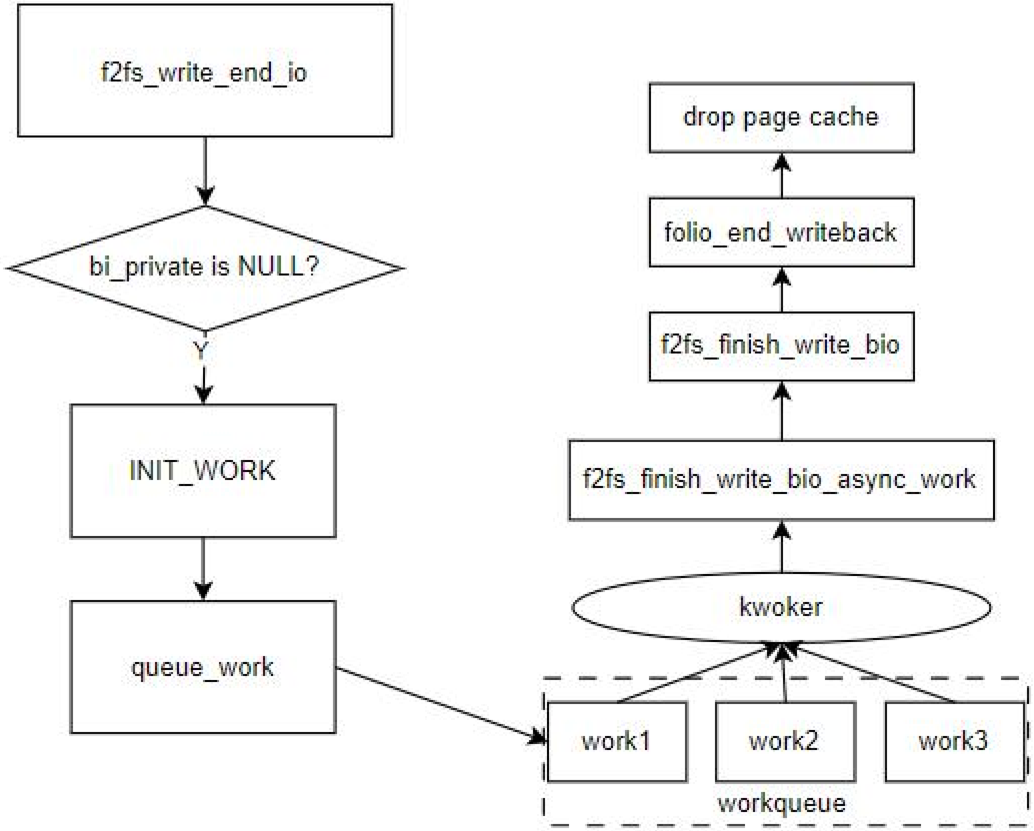
f2fs 适配 Uncached buffer IO 读提交链接：
https://lore.kernel.org/all/20250725075310.1614614-1-hanqi@vivo.com/

f2fs 适配 Uncached buffer I/O 写



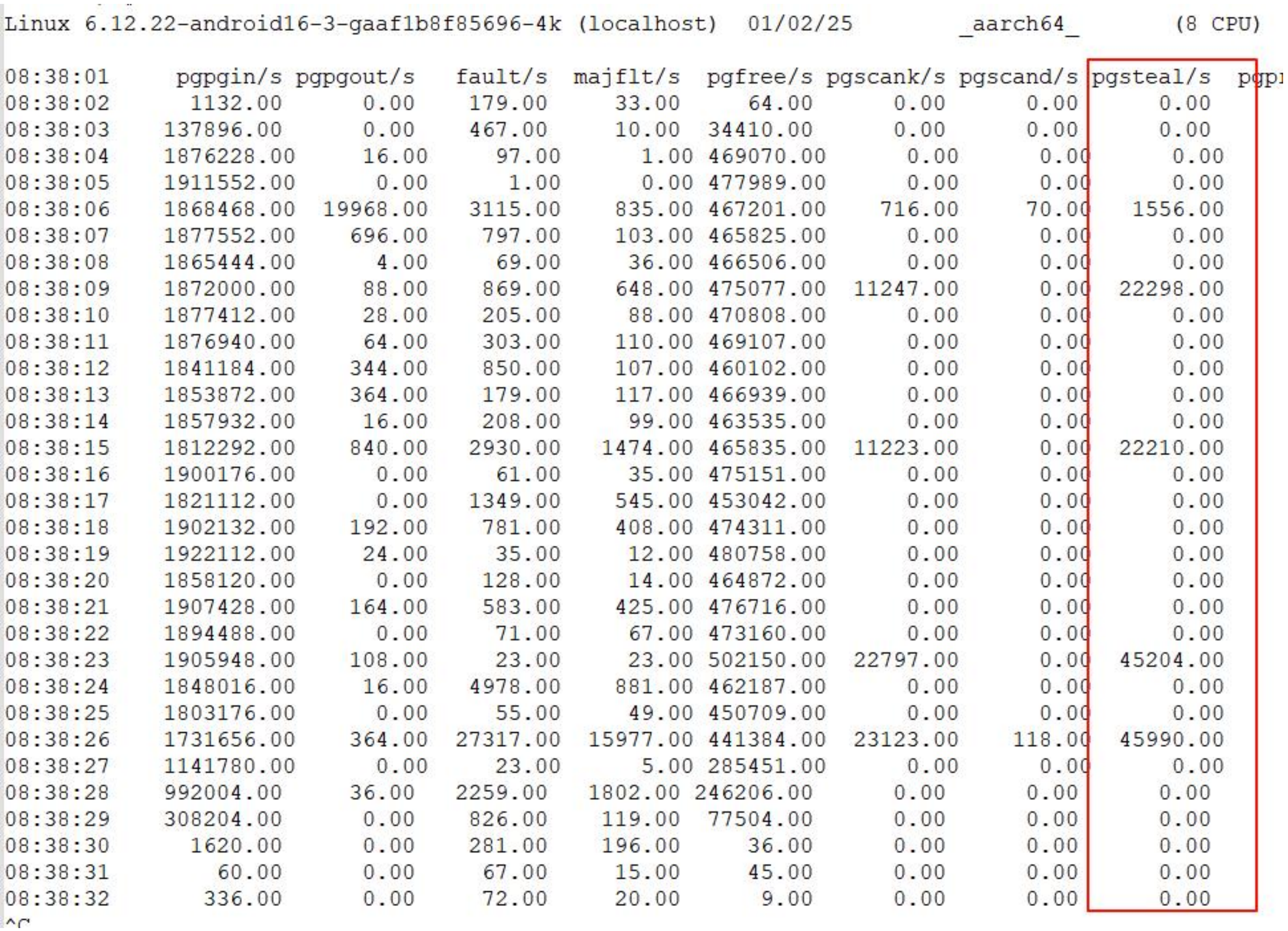


```
/*
 * If folio was marked as dropbehind, then pages should be dropped when writeback
 * completes. Do that now. If we fail, it's likely because of a big folio -
 * just reset dropbehind for that case and latter completions should invalidate.
 */
static void filemap_end_dropbehind_write(struct folio *folio)
{
	if (!folio_test_dropbehind(folio))
		return;

	/*
	 * Hitting !in_task() should not happen off RWF_DONTCACHE writeback,
	 * but can happen if normal writeback just happens to find dirty folios
	 * that were created as part of uncached writeback, and that writeback
	 * would otherwise not need non-IRQ handling. Just skip the
	 * invalidation in that case.
	 */
	if (in_task() && folio_trylock(folio)) {
		filemap_end_dropbehind(folio);
		folio_unlock(folio);
	}
}
```

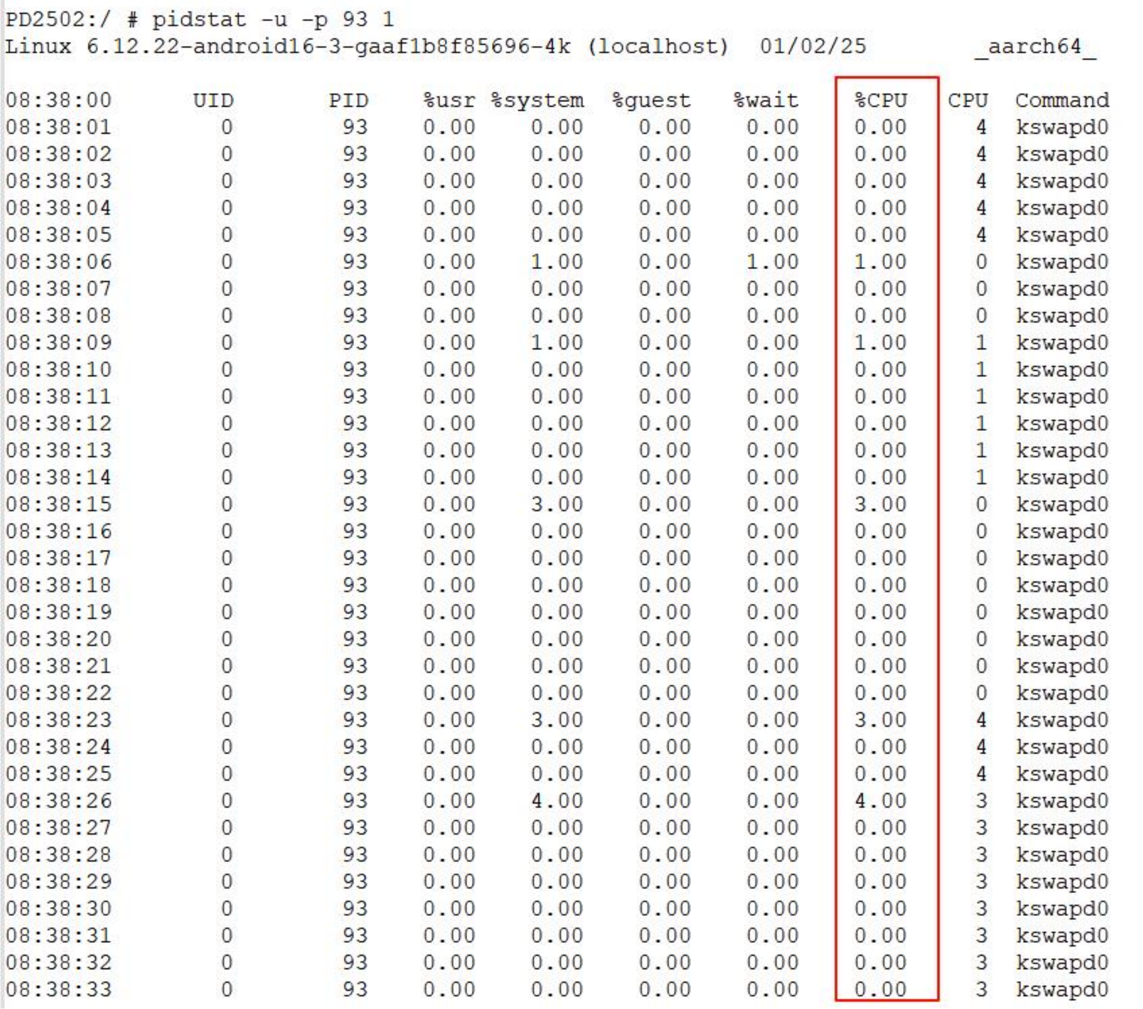


f2fs 适配 Uncached buffer IO 读提交链接:
https://lore.kernel.org/all/20250828121131.36941
54-1-hanqi@vivo.com/

4、f2fs Uncached buffer I/O 读场景后台无内存交换， kswapd 负载降为 0

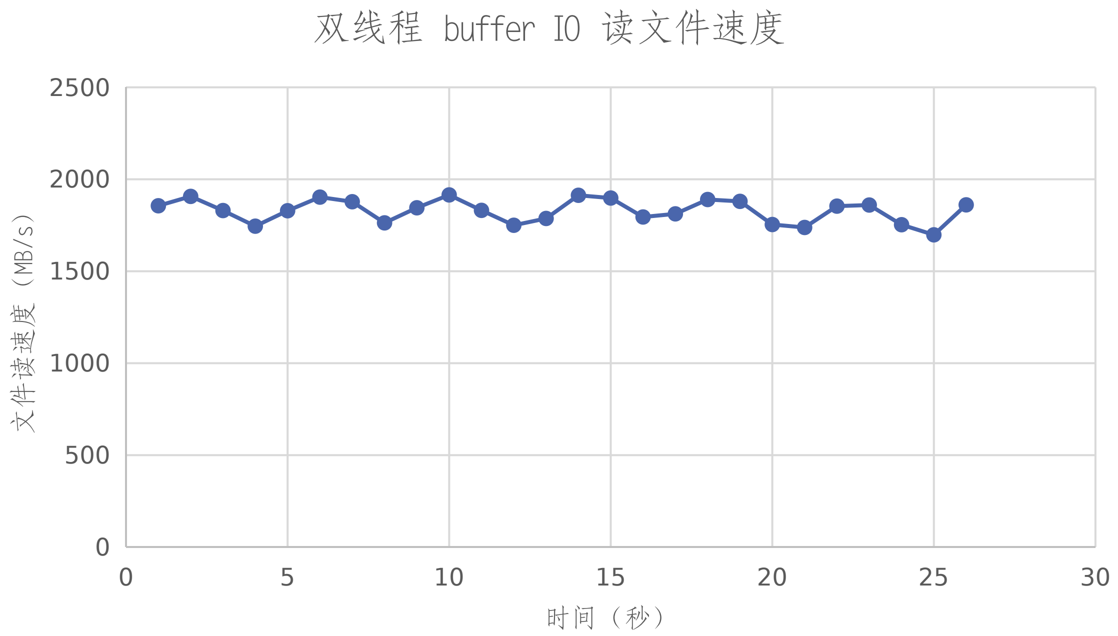


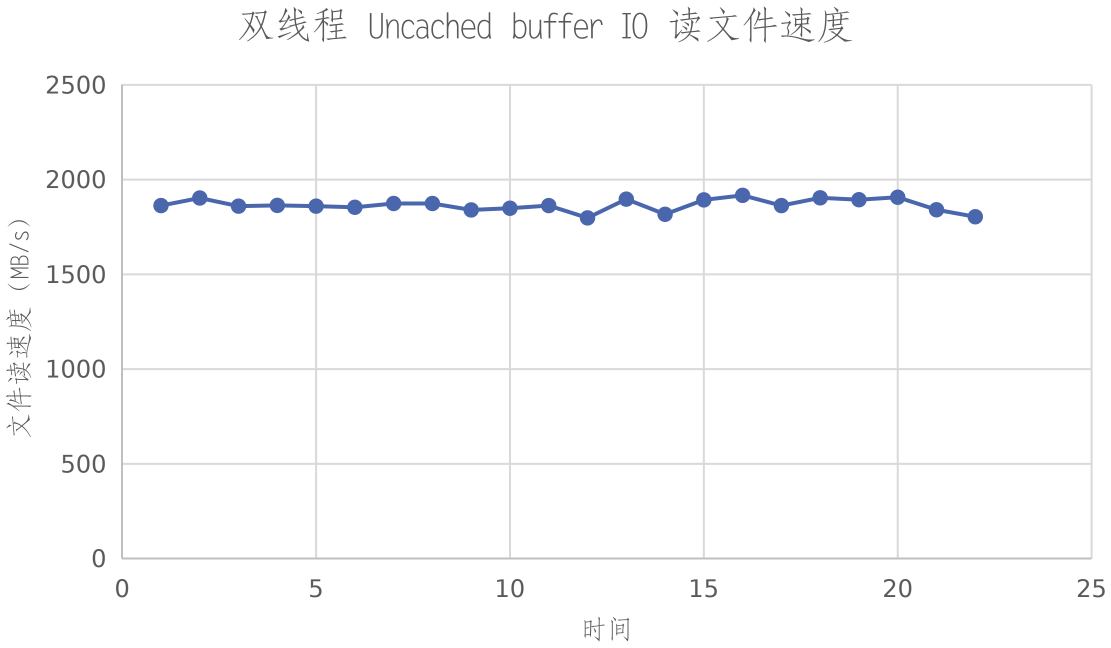
```
Linux 6.12.22-android16-3-gaaf1b8f85696-4k (localhost)  01/02/25        _aarch64_       (8 CPU)

08:38:01      pgpgin/s pgpgout/s   fault/s  majflt/s  pgfree/s pgscank/s pgscand/s pgsteal/s  pgp
08:38:02       1132.00      0.00    179.00     33.00     64.00      0.00      0.00      0.00
08:38:03     137896.00      0.00    467.00     10.00  34410.00      0.00      0.00      0.00
08:38:04    1876228.00     16.00     97.00      1.00 469070.00      0.00      0.00      0.00
08:38:05    1911552.00      0.00      1.00      0.00 477989.00      0.00      0.00      0.00
08:38:06    1868468.00  19968.00   3115.00    835.00 467201.00    716.00     70.00   1556.00
08:38:07    1877552.00    696.00    797.00    103.00 465825.00      0.00      0.00      0.00
08:38:08    1865444.00      4.00     69.00     36.00 466506.00      0.00      0.00      0.00
08:38:09    1872000.00     88.00    869.00    648.00 475077.00  11247.00      0.00  22298.00
08:38:10    1877412.00     28.00    205.00     88.00 470808.00      0.00      0.00      0.00
08:38:11    1876940.00     64.00    303.00    110.00 469107.00      0.00      0.00      0.00
08:38:12    1841184.00    344.00    850.00    107.00 460102.00      0.00      0.00      0.00
08:38:13    1853872.00    364.00    179.00    117.00 466939.00      0.00      0.00      0.00
08:38:14    1857932.00     16.00    208.00     99.00 463535.00      0.00      0.00      0.00
08:38:15    1812292.00    840.00   2930.00   1474.00 465835.00  11223.00      0.00  22210.00
08:38:16    1900176.00      0.00     61.00     35.00 475151.00      0.00      0.00      0.00
08:38:17    1821112.00      0.00   1349.00    545.00 453042.00      0.00      0.00      0.00
08:38:18    1902132.00    192.00    781.00    408.00 474311.00      0.00      0.00      0.00
08:38:19    1922112.00     24.00     35.00     12.00 480758.00      0.00      0.00      0.00
08:38:20    1858120.00      0.00    128.00     14.00 464872.00      0.00      0.00      0.00
08:38:21    1907428.00    164.00    583.00    425.00 476716.00      0.00      0.00      0.00
08:38:22    1894488.00      0.00     71.00     67.00 473160.00      0.00      0.00      0.00
08:38:23    1905948.00    108.00     23.00     23.00 502150.00  22797.00      0.00  45204.00
08:38:24    1848016.00     16.00   4978.00    881.00 462187.00      0.00      0.00      0.00
08:38:25    1803176.00      0.00     55.00     49.00 450709.00      0.00      0.00      0.00
08:38:26    1731656.00    364.00  27317.00  15977.00 441384.00  23123.00    118.00  45990.00
08:38:27    1141780.00      0.00     23.00      5.00 285451.00      0.00      0.00      0.00
08:38:28     992004.00     36.00   2259.00   1802.00 246206.00      0.00      0.00      0.00
08:38:29     308204.00      0.00    826.00    119.00  77504.00      0.00      0.00      0.00
08:38:30       1620.00      0.00    281.00    196.00     36.00      0.00      0.00      0.00
08:38:31         60.00      0.00     67.00     15.00     45.00      0.00      0.00      0.00
08:38:32        336.00      0.00     72.00     20.00      9.00      0.00      0.00      0.00
^C
```



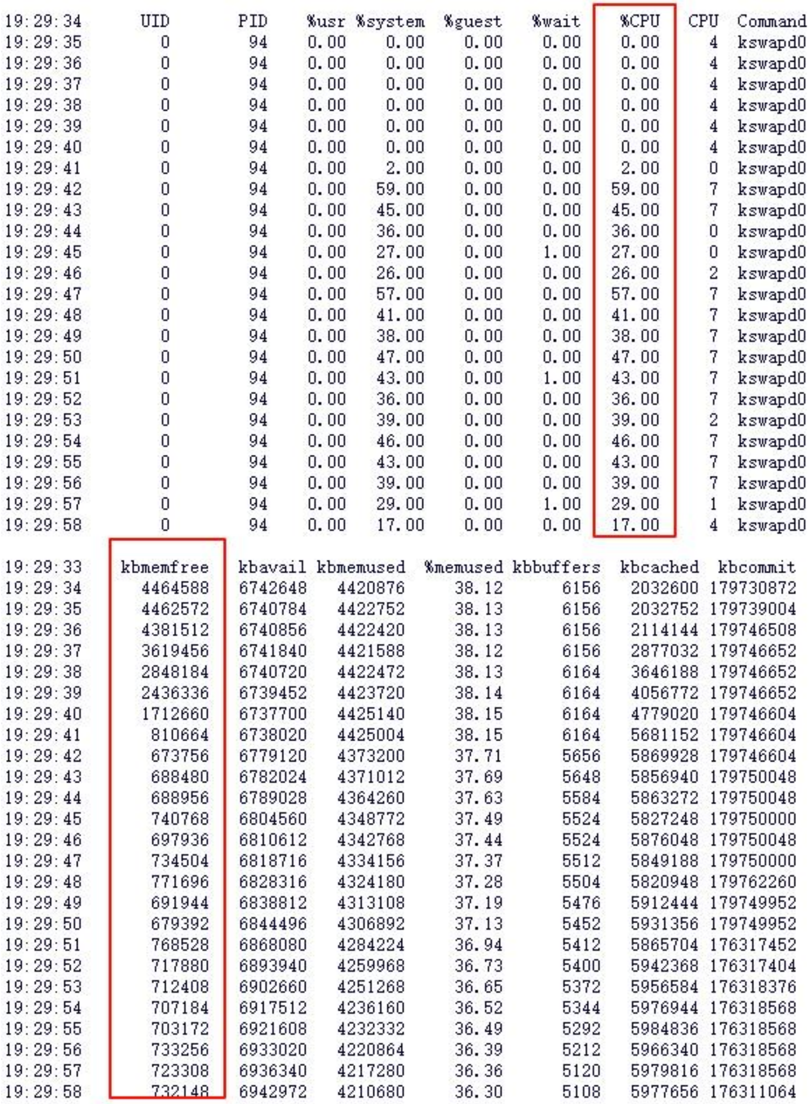
```
PD2502:/ # pidstat -u -p 93 1
Linux 6.12.22-android16-3-gaaf1b8f85696-4k (localhost)  01/02/25        _aarch64_

08:38:00      UID       PID    %usr %system  %guest   %wait    %CPU   CPU  Command
08:38:01        0        93    0.00    0.00    0.00    0.00    0.00     4  kswapd0
08:38:02        0        93    0.00    0.00    0.00    0.00    0.00     4  kswapd0
08:38:03        0        93    0.00    0.00    0.00    0.00    0.00     4  kswapd0
08:38:04        0        93    0.00    0.00    0.00    0.00    0.00     4  kswapd0
08:38:05        0        93    0.00    0.00    0.00    0.00    0.00     4  kswapd0
08:38:06        0        93    0.00    1.00    0.00    1.00    1.00     0  kswapd0
08:38:07        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:08        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:09        0        93    0.00    1.00    0.00    0.00    1.00     1  kswapd0
08:38:10        0        93    0.00    0.00    0.00    0.00    0.00     1  kswapd0
08:38:11        0        93    0.00    0.00    0.00    0.00    0.00     1  kswapd0
08:38:12        0        93    0.00    0.00    0.00    0.00    0.00     1  kswapd0
08:38:13        0        93    0.00    0.00    0.00    0.00    0.00     1  kswapd0
08:38:14        0        93    0.00    0.00    0.00    0.00    0.00     1  kswapd0
08:38:15        0        93    0.00    3.00    0.00    0.00    3.00     0  kswapd0
08:38:16        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:17        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:18        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:19        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:20        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:21        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:22        0        93    0.00    0.00    0.00    0.00    0.00     0  kswapd0
08:38:23        0        93    0.00    3.00    0.00    0.00    3.00     4  kswapd0
08:38:24        0        93    0.00    0.00    0.00    0.00    0.00     4  kswapd0
08:38:25        0        93    0.00    0.00    0.00    0.00    0.00     4  kswapd0
08:38:26        0        93    0.00    4.00    0.00    0.00    4.00     3  kswapd0
08:38:27        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
08:38:28        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
08:38:29        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
08:38:30        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
08:38:31        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
08:38:32        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
08:38:33        0        93    0.00    0.00    0.00    0.00    0.00     3  kswapd0
```

f2fs Uncached buffer I/O 读速度波动由 150MB/s 降到 50MB/s



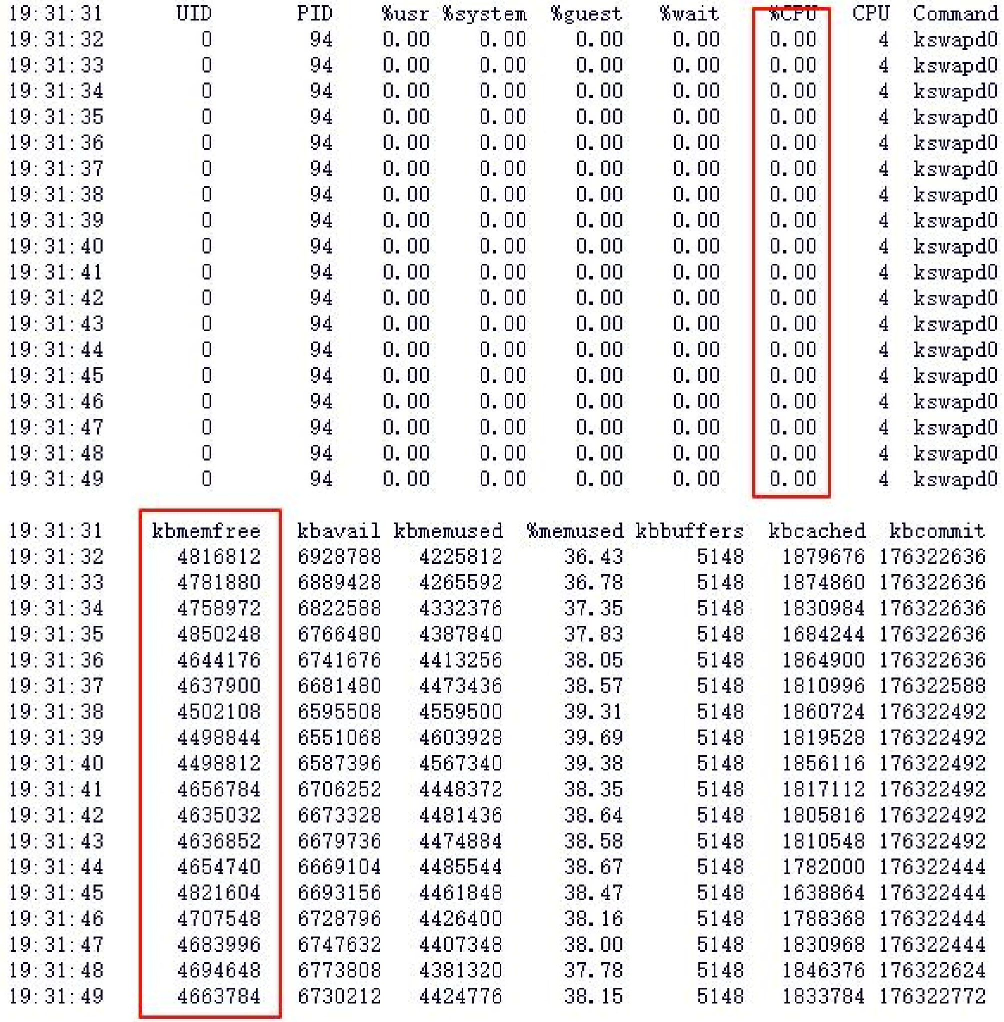


f2fs Uncached buffer I/O 写场景后台无 page cacahe 占用， kswapd 负载降为 0



```
19:29:34      UID      PID    %usr %system  %guest   %wait    %CPU   CPU  Command
19:29:35        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:29:36        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:29:37        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:29:38        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:29:39        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:29:40        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:29:41        0       94    0.00    2.00    0.00    0.00    2.00     0  kswapd0
19:29:42        0       94    0.00   59.00    0.00    0.00   59.00     7  kswapd0
19:29:43        0       94    0.00   45.00    0.00    0.00   45.00     7  kswapd0
19:29:44        0       94    0.00   36.00    0.00    0.00   36.00     0  kswapd0
19:29:45        0       94    0.00   27.00    0.00    1.00   27.00     0  kswapd0
19:29:46        0       94    0.00   26.00    0.00    0.00   26.00     2  kswapd0
19:29:47        0       94    0.00   57.00    0.00    0.00   57.00     7  kswapd0
19:29:48        0       94    0.00   41.00    0.00    0.00   41.00     7  kswapd0
19:29:49        0       94    0.00   38.00    0.00    0.00   38.00     7  kswapd0
19:29:50        0       94    0.00   47.00    0.00    0.00   47.00     7  kswapd0
19:29:51        0       94    0.00   43.00    0.00    1.00   43.00     7  kswapd0
19:29:52        0       94    0.00   36.00    0.00    0.00   36.00     7  kswapd0
19:29:53        0       94    0.00   39.00    0.00    0.00   39.00     2  kswapd0
19:29:54        0       94    0.00   46.00    0.00    0.00   46.00     7  kswapd0
19:29:55        0       94    0.00   43.00    0.00    0.00   43.00     7  kswapd0
19:29:56        0       94    0.00   39.00    0.00    0.00   39.00     7  kswapd0
19:29:57        0       94    0.00   29.00    0.00    1.00   29.00     1  kswapd0
19:29:58        0       94    0.00   17.00    0.00    0.00   17.00     4  kswapd0

19:29:33  kbmemfree  kbavail kbmemused  %memused kbbuffers  kbcached  kbcommit
19:29:34    4464588  6742648   4420876     38.12      6156   2032600 179730872
19:29:35    4462572  6740784   4422752     38.13      6156   2032752 179739004
19:29:36    4381512  6740856   4422420     38.13      6156   2114144 179746508
19:29:37    3619456  6741840   4421588     38.12      6156   2877032 179746652
19:29:38    2848184  6740720   4422472     38.13      6164   3646188 179746652
19:29:39    2436336  6739452   4423720     38.14      6164   4056772 179746652
19:29:40    1712660  6737700   4425140     38.15      6164   4779020 179746604
19:29:41     810664  6738020   4425004     38.15      6164   5681152 179746604
19:29:42     673756  6779120   4373200     37.71      5656   5869928 179746604
19:29:43     688480  6782024   4371012     37.69      5648   5856940 179750048
19:29:44     688956  6789028   4364260     37.63      5584   5863272 179750048
19:29:45     740768  6804560   4348772     37.49      5524   5827248 179750000
19:29:46     697936  6810612   4342768     37.44      5524   5876048 179750048
19:29:47     734504  6818716   4334156     37.37      5512   5849188 179750000
19:29:48     771696  6828316   4324180     37.28      5504   5820948 179762260
19:29:49     691944  6838812   4313108     37.19      5476   5912444 179749952
19:29:50     679392  6844496   4306892     37.13      5452   5931356 179749952
19:29:51     768528  6868080   4284224     36.94      5412   5865704 176317452
19:29:52     717880  6893940   4259968     36.73      5400   5942368 176317404
19:29:53     712408  6902660   4251268     36.65      5372   5956584 176318376
19:29:54     707184  6917512   4236160     36.52      5344   5976944 176318568
19:29:55     703172  6921608   4232332     36.49      5292   5984836 176318568
19:29:56     733256  6933020   4220864     36.39      5212   5966340 176318568
19:29:57     723308  6936340   4217280     36.36      5120   5979816 176318568
19:29:58     732148  6942972   4210680     36.30      5108   5977656 176311064
```

Buffer IO 写

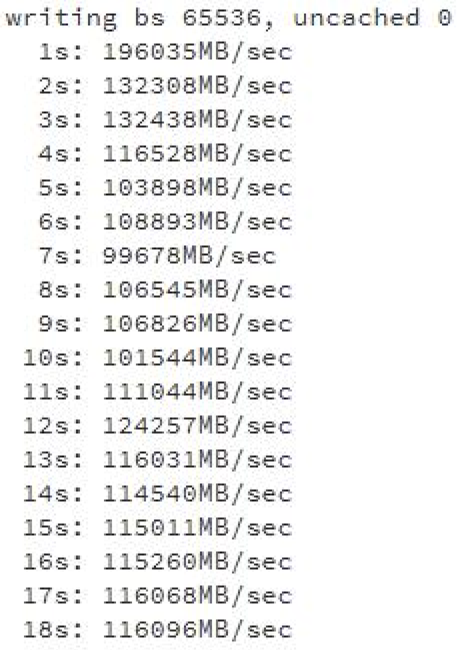


```
19:31:31      UID      PID    %usr %system  %guest   %wait    %CPU   CPU  Command
19:31:32        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:33        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:34        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:35        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:36        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:37        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:38        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:39        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:40        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:41        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:42        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:43        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:44        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:45        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:46        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:47        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:48        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0
19:31:49        0       94    0.00    0.00    0.00    0.00    0.00     4  kswapd0

19:31:31  kbmemfree  kbavail kbmemused  %memused kbbuffers  kbcached  kbcommit
19:31:32    4816812  6928788   4225812     36.43      5148   1879676 176322636
19:31:33    4781880  6889428   4265592     36.78      5148   1874860 176322636
19:31:34    4758972  6822588   4332376     37.35      5148   1830984 176322636
19:31:35    4850248  6766480   4387840     37.83      5148   1684244 176322636
19:31:36    4644176  6741676   4413256     38.05      5148   1864900 176322636
19:31:37    4637900  6681480   4473436     38.57      5148   1810996 176322588
19:31:38    4502108  6595508   4559500     39.31      5148   1860724 176322492
19:31:39    4498844  6551068   4603928     39.69      5148   1819528 176322492
19:31:40    4498812  6587396   4567340     39.38      5148   1856116 176322492
19:31:41    4656784  6706252   4448372     38.35      5148   1817112 176322492
19:31:42    4635032  6673328   4481436     38.64      5148   1805816 176322492
19:31:43    4636852  6679736   4474884     38.58      5148   1810548 176322492
19:31:44    4654740  6669104   4485544     38.67      5148   1782000 176322444
19:31:45    4821604  6693156   4461848     38.47      5148   1638864 176322444
19:31:46    4707548  6728796   4426400     38.16      5148   1788368 176322444
19:31:47    4683996  6747632   4407348     38.00      5148   1830968 176322444
19:31:48    4694648  6773808   4381320     37.78      5148   1846376 176322624
19:31:49    4663784  6730212   4424776     38.15      5148   1833784 176322772
```

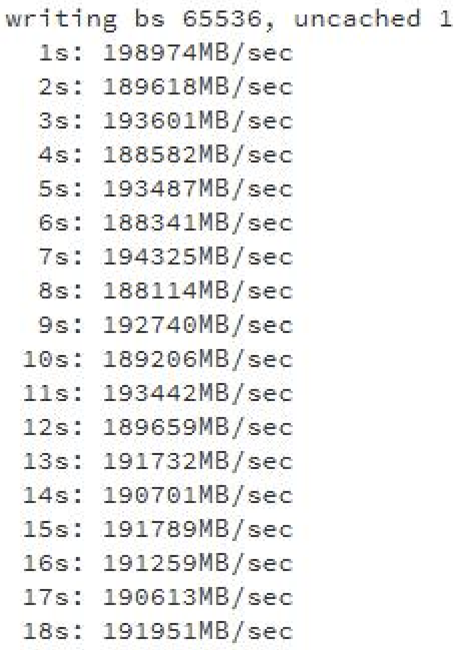
Uncached buffer IO 写

是否有更好方案解决 f2fs Uncached buffer I/O 写场景下性能下降问题?

Uncached buffer I/O 写为了能尽快释放 page cache 占用内存，其会在数据写入 page cache 后主动触发一次回写，这种方法类似于 sync 操作，进而会导致写性能降低。但是在 Jens Axboe 的测试环境，即磁盘性能非常好的环境下 UBIO 写性能会有提升，但是在绝大部分场景下磁盘的性能是低于内存的，故 UBIO 写会导致写性能降低。是否有更优方案来解决 UBIO 写性能降低的问题?



```
writing bs 65536, uncached 0
  1s: 196035MB/sec
  2s: 132308MB/sec
  3s: 132438MB/sec
  4s: 116528MB/sec
  5s: 103898MB/sec
  6s: 108893MB/sec
  7s: 99678MB/sec
  8s: 106545MB/sec
  9s: 106826MB/sec
 10s: 101544MB/sec
 11s: 111044MB/sec
 12s: 124257MB/sec
 13s: 116031MB/sec
 14s: 114540MB/sec
 15s: 115011MB/sec
 16s: 115260MB/sec
 17s: 116068MB/sec
 18s: 116096MB/sec
```



```
writing bs 65536, uncached 1
  1s: 198974MB/sec
  2s: 189618MB/sec
  3s: 193601MB/sec
  4s: 188582MB/sec
  5s: 193487MB/sec
  6s: 188341MB/sec
  7s: 194325MB/sec
  8s: 188114MB/sec
  9s: 192740MB/sec
 10s: 189206MB/sec
 11s: 193442MB/sec
 12s: 189659MB/sec
 13s: 191732MB/sec
 14s: 190701MB/sec
 15s: 191789MB/sec
 16s: 191259MB/sec
 17s: 190613MB/sec
 18s: 191951MB/sec
```

Jens Axboe 环境写性能:
https://git.kernel.org/pub/scm/linux/kernel/git/akpm/mm.git/commit/?h=mm-new&id=d47c670061b5f9481ce494cd6c45078be301620e

THANKS