

Memory Compaction in Linux Kernel

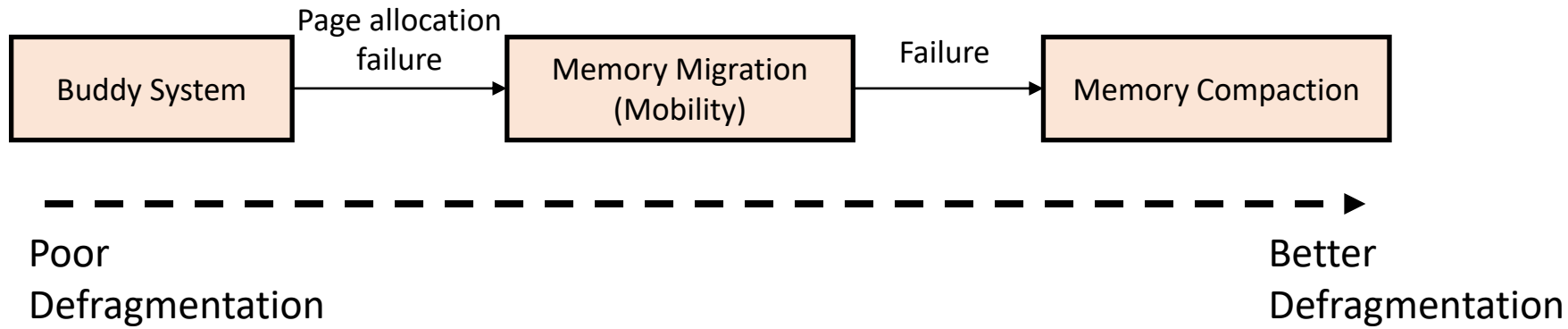
Adrian Huang | Aug, 2022

- * **Based on kernel 5.11 (x86_64) – QEMU**
- * **1-socket CPUs (8 cores/socket)**
- * **16GB memory**
- * **Kernel parameter: nokaslr norandmaps**
- * **Userspace: ASLR is disabled**
- * **Legacy BIOS**

Agenda

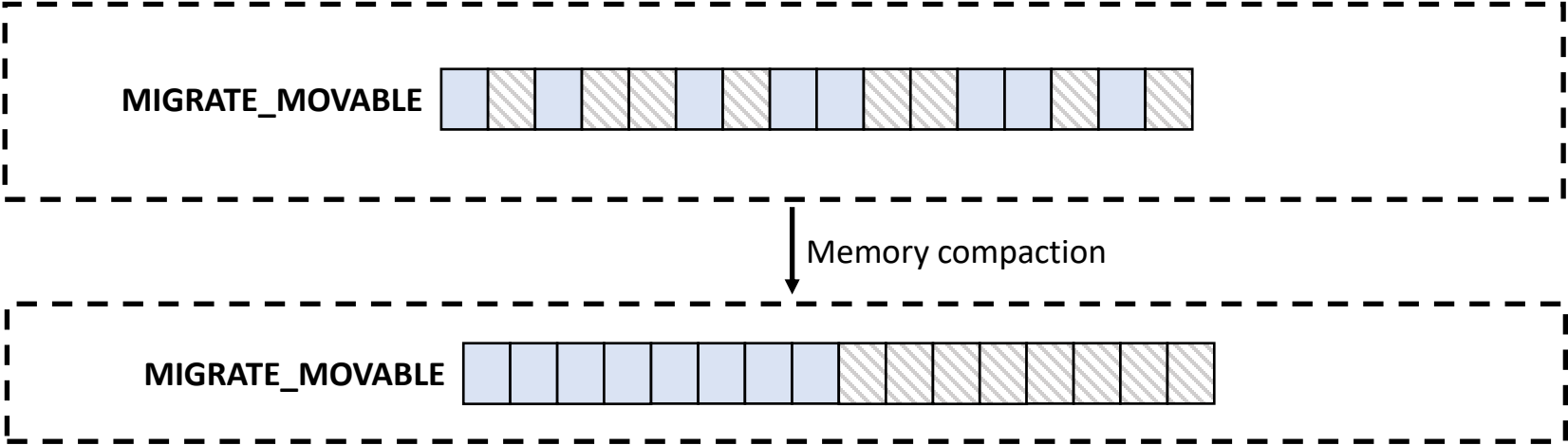
- Physical memory defragmentation (anti-fragmentation): Approaches
- Memory Compaction – Concept & Detail
- Implementation Detail
- Scenario Creation: gdb observation

Physical memory defragmentation (Anti-fragmentation): Approaches



Memory Compaction – Concept & Detail

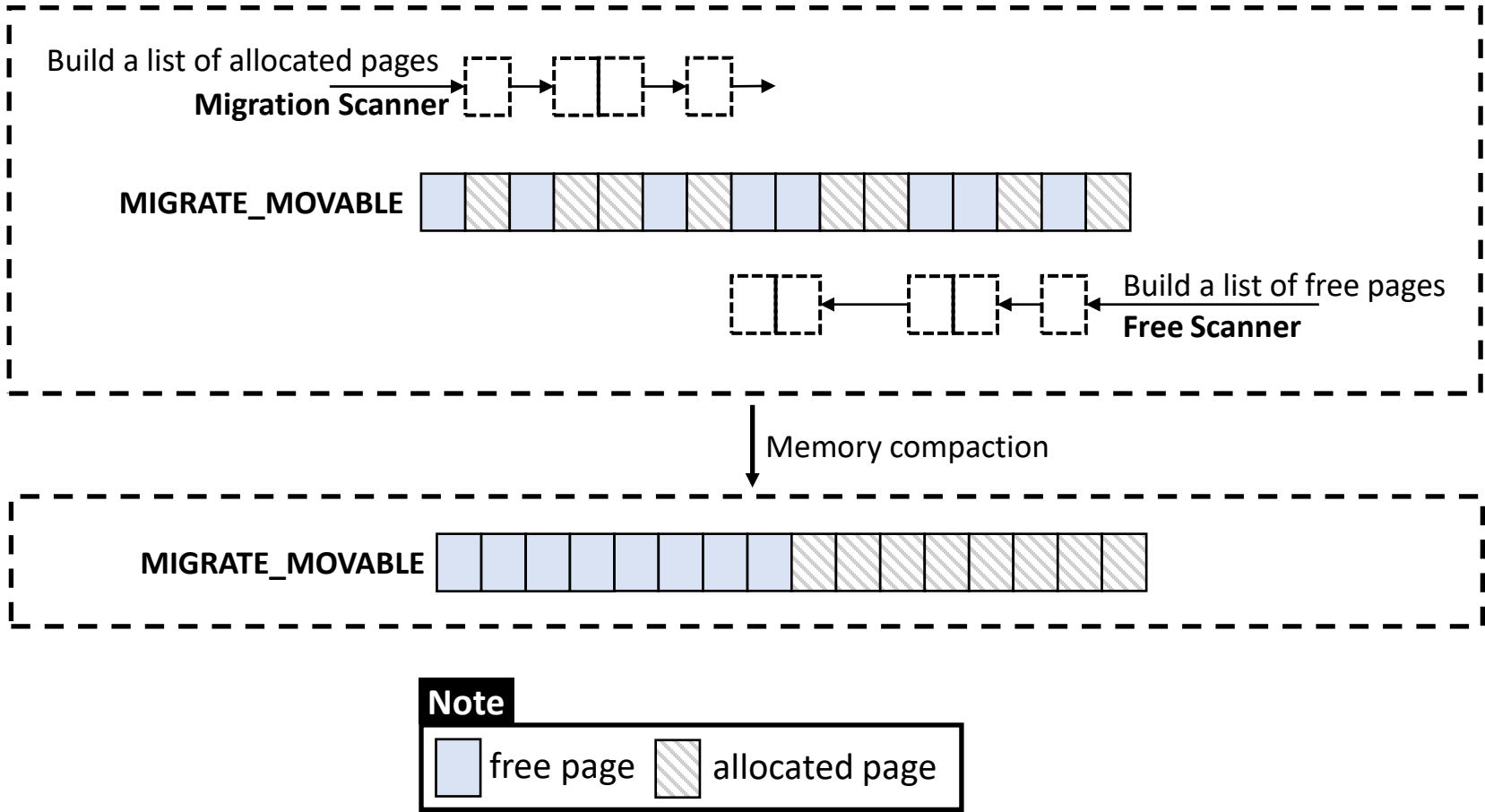
Memory Compaction – Concept (1/2)



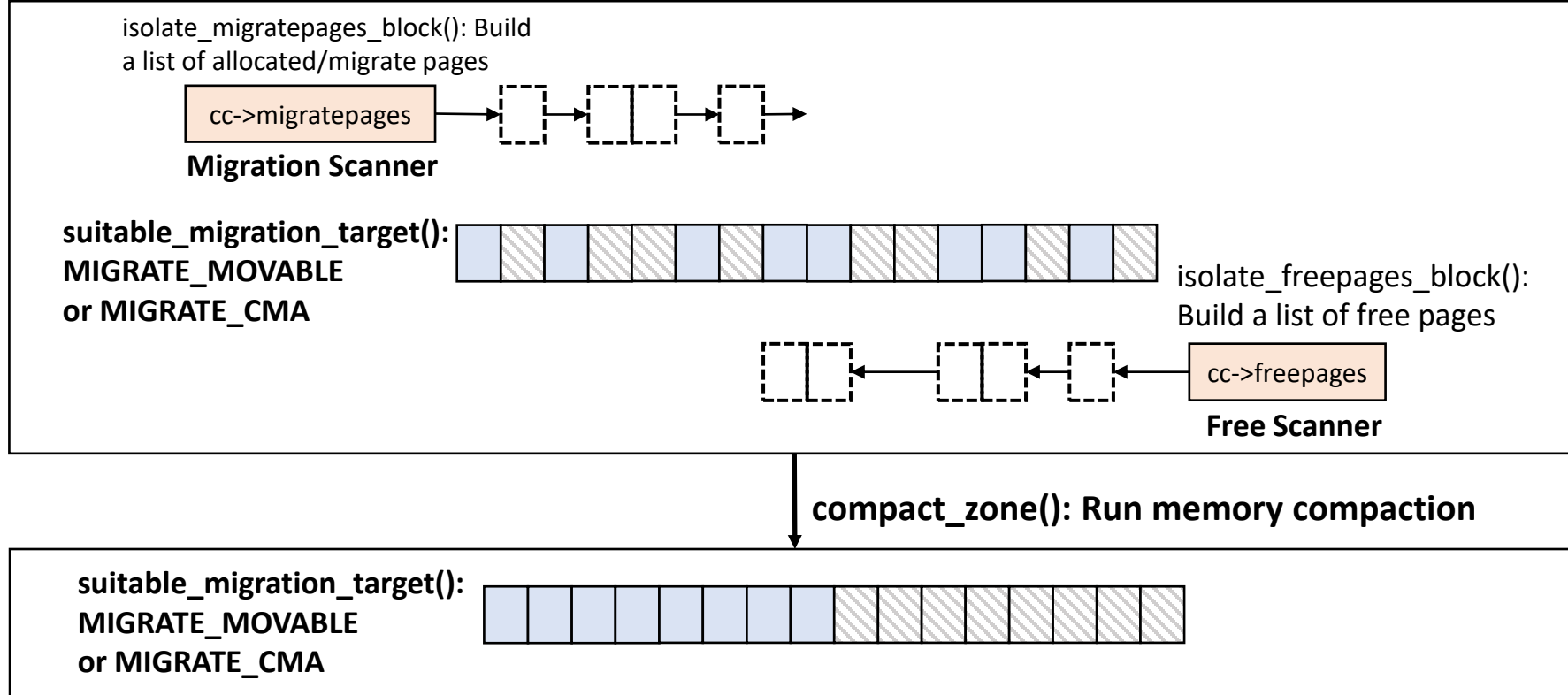
Note



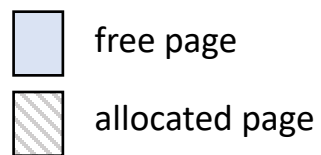
Memory Compaction – Concept (2/2)



Memory Compaction – Detail



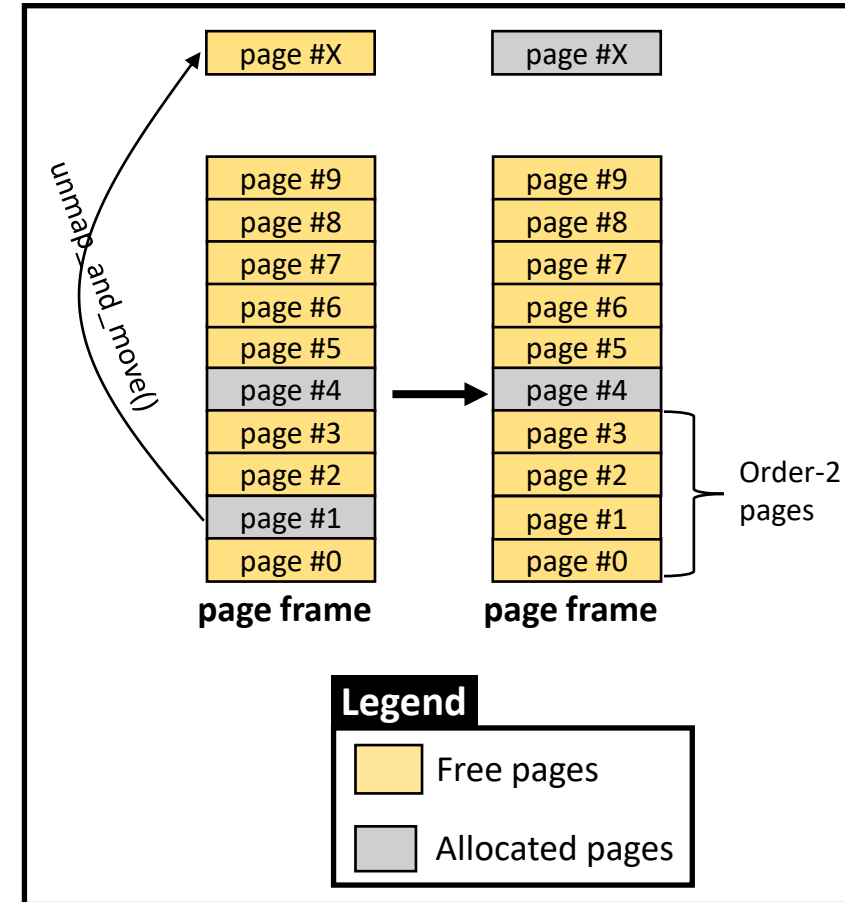
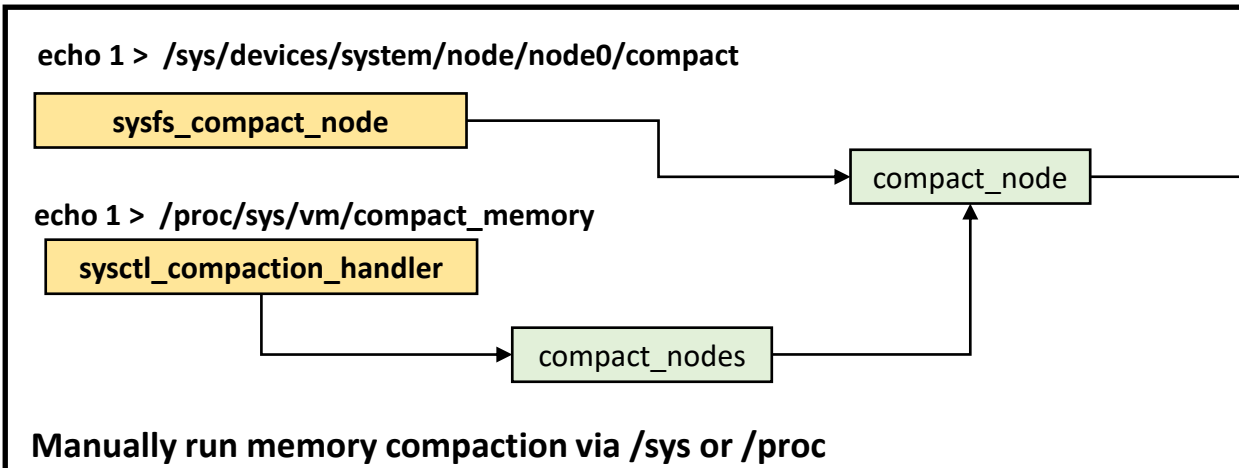
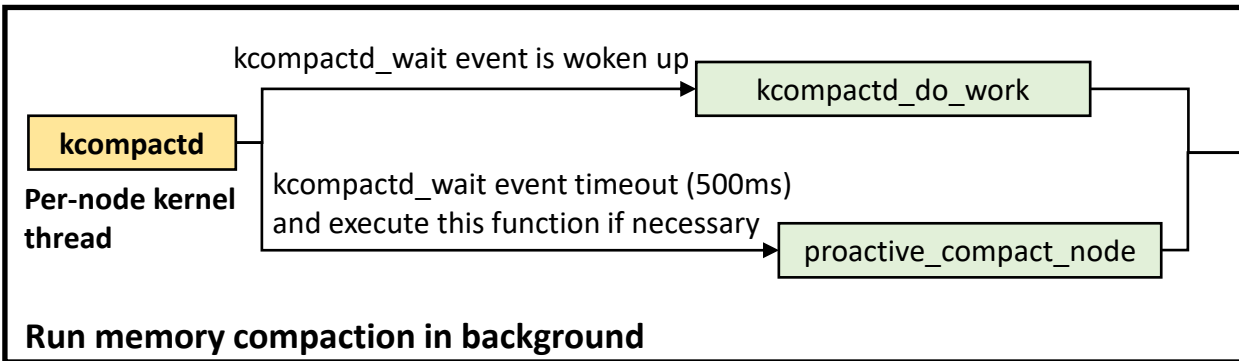
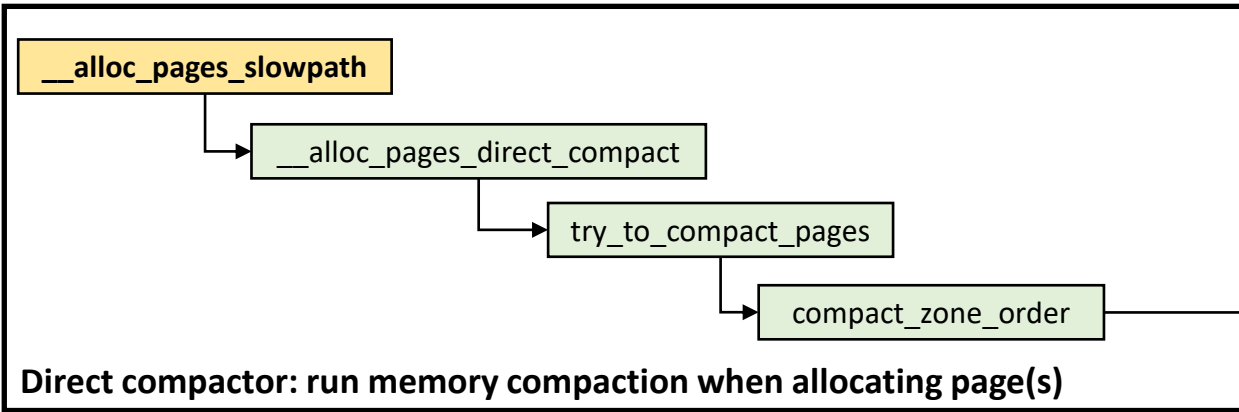
Legend



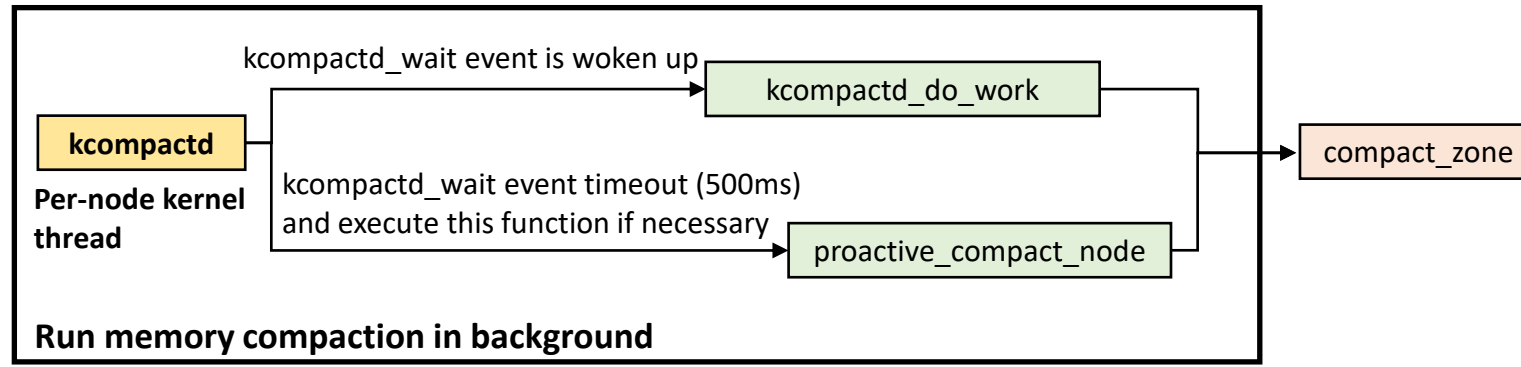
Implementation Detail

1. Call path
2. Direct compaction, proactive compaction (kernel thread), and manual compaction

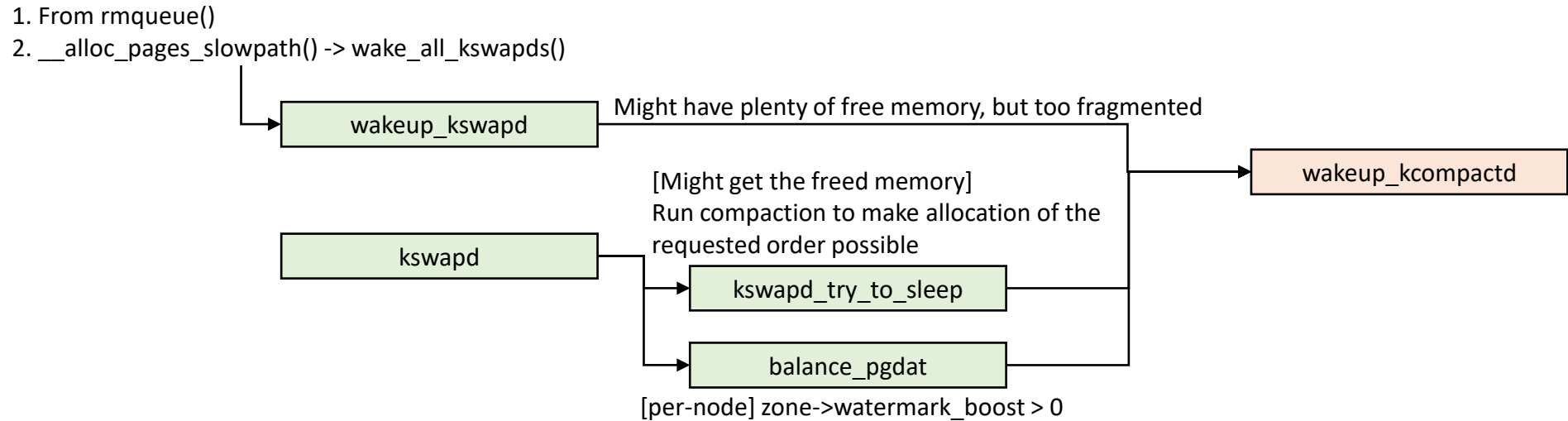
compact_zone() - callers



Who wakes up kcompactd?

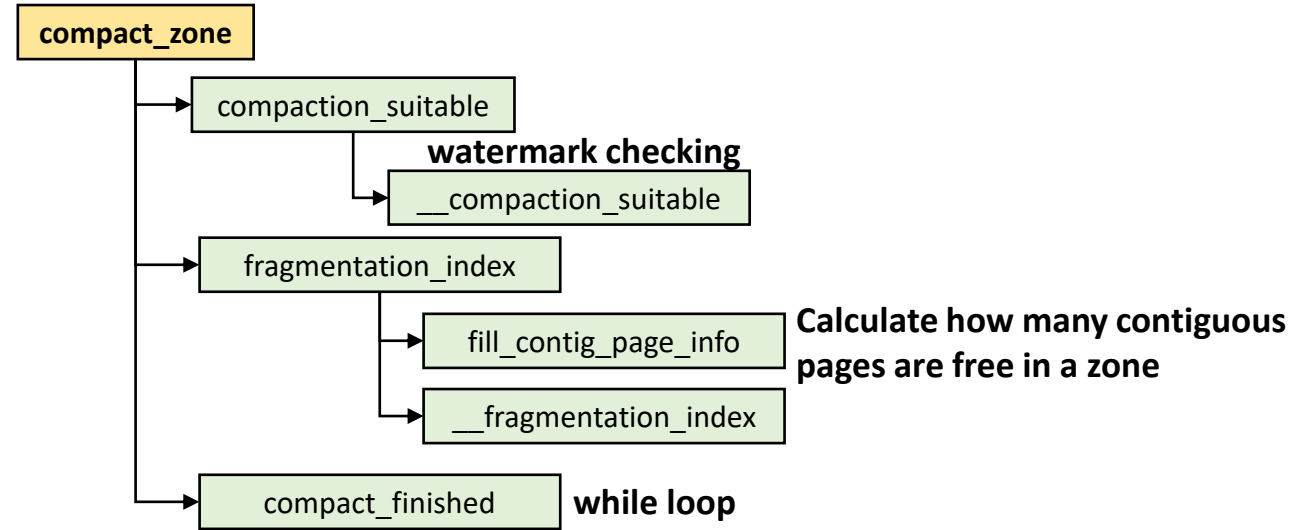


[Call path] Who wakes up kcompactd?

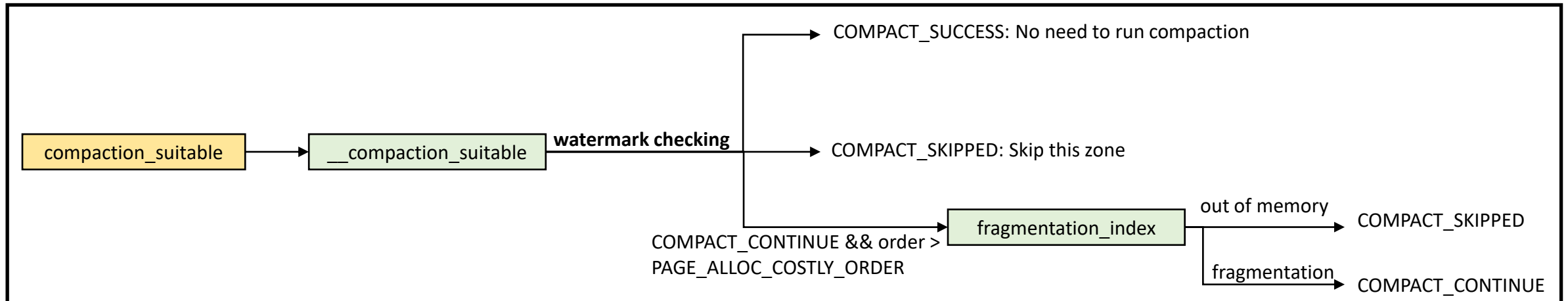
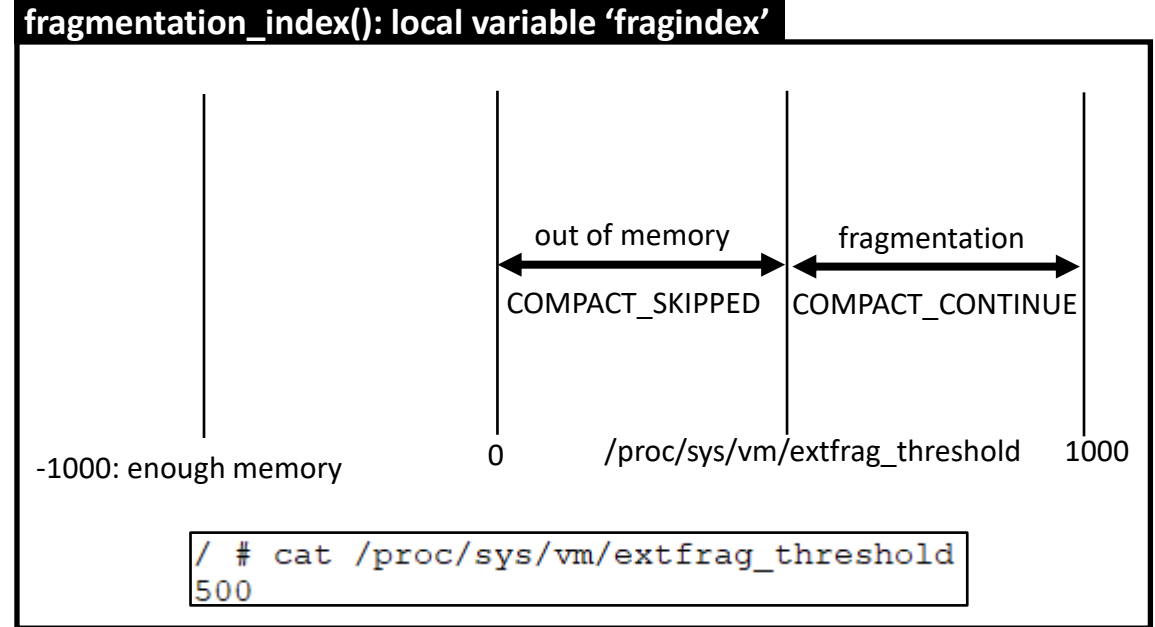
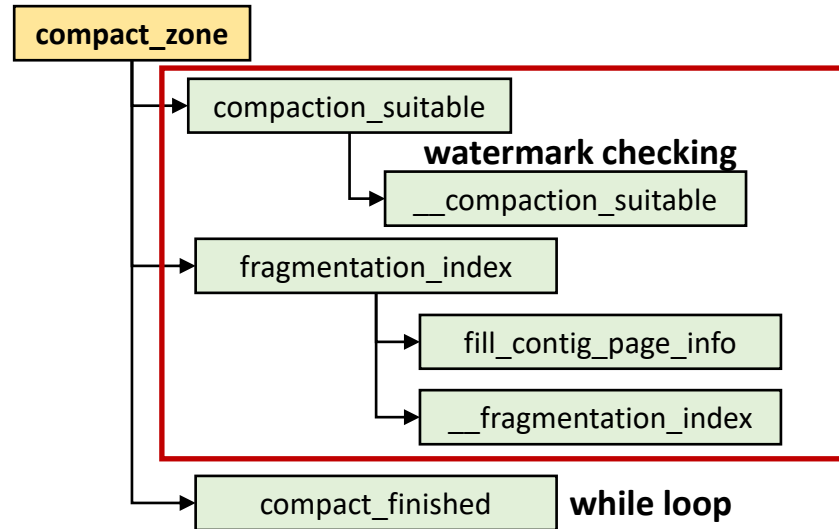


Mainly from kswapd

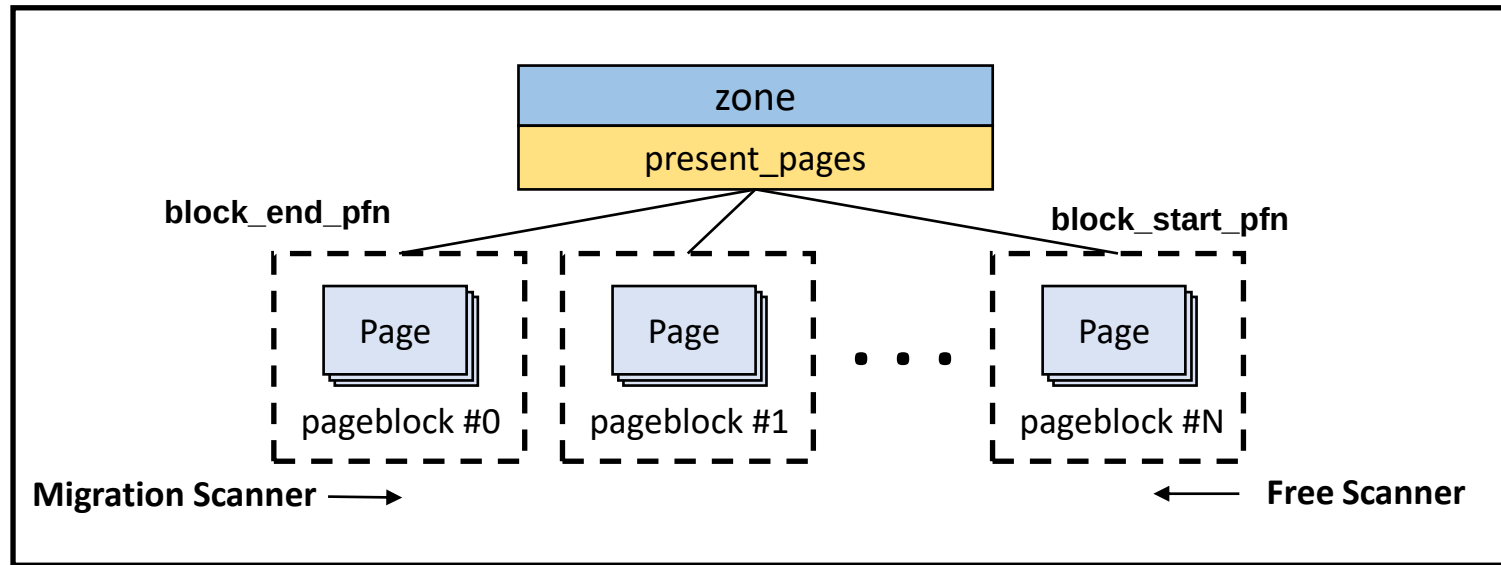
compact_zone() – call path



compact_zone() – call path

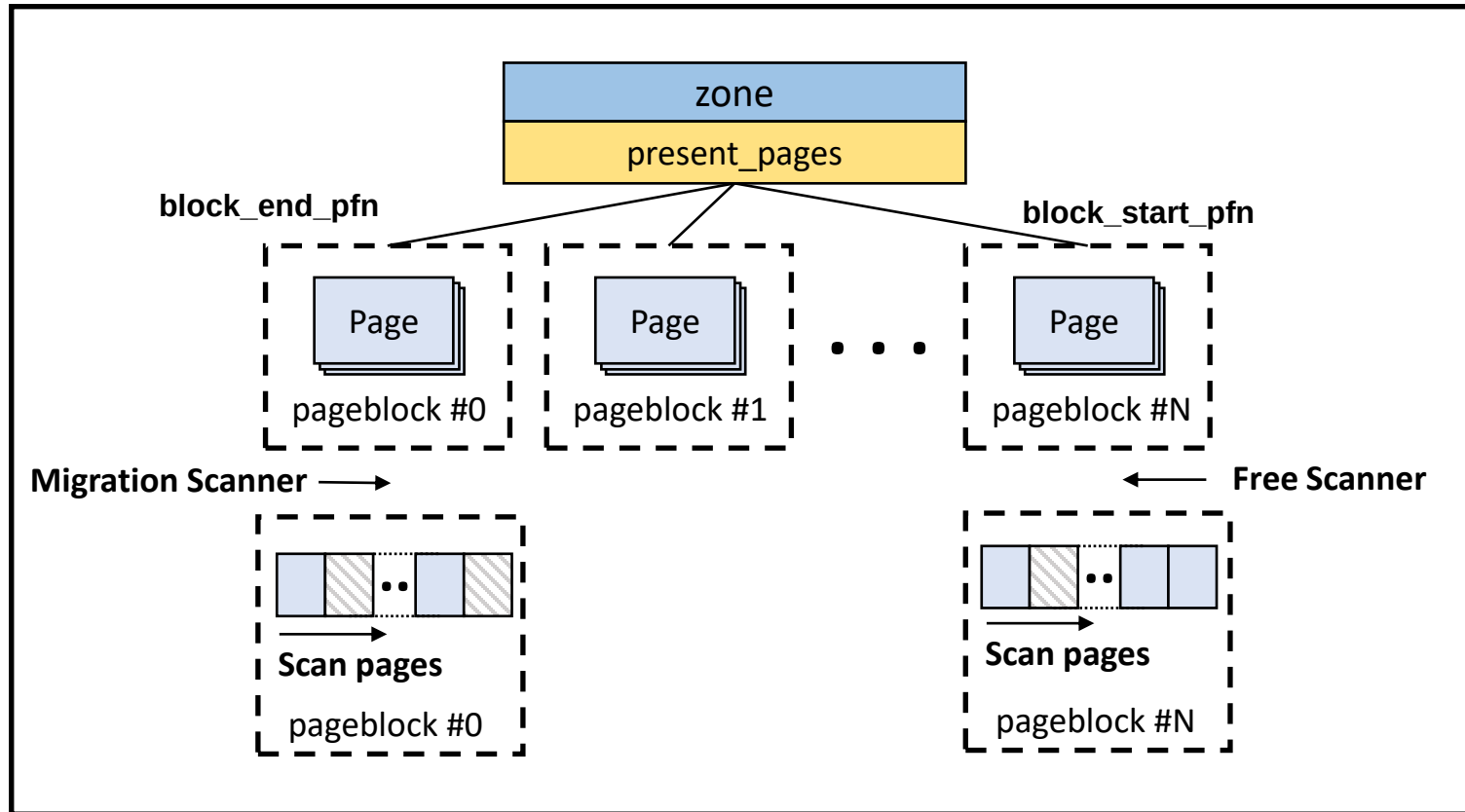


Migration Scanner & Free Scanner: concept (1/2)

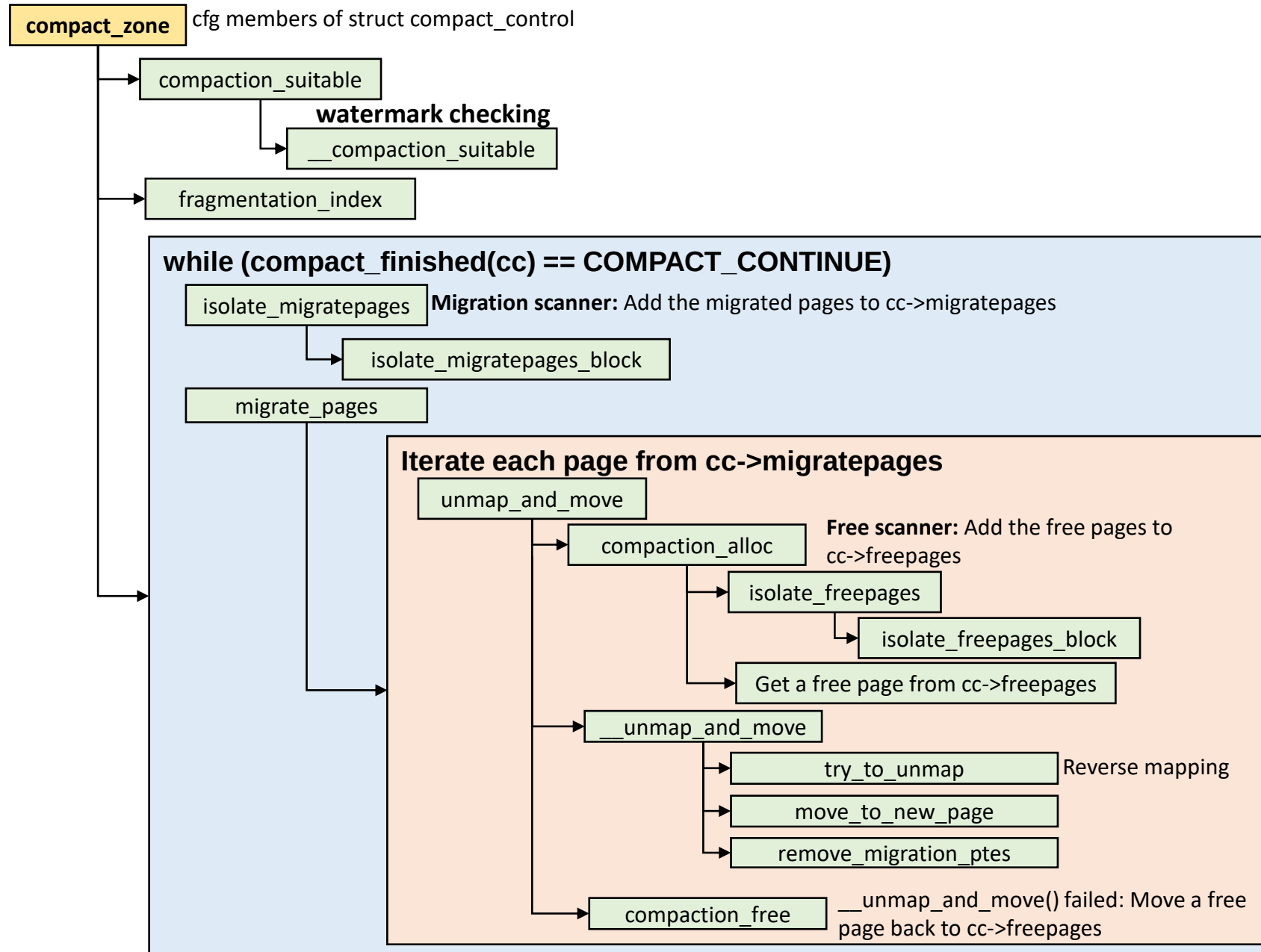


Migration Scanner: first pageblock -> last pageblock
Free Scanner: last pageblock -> first pageblock

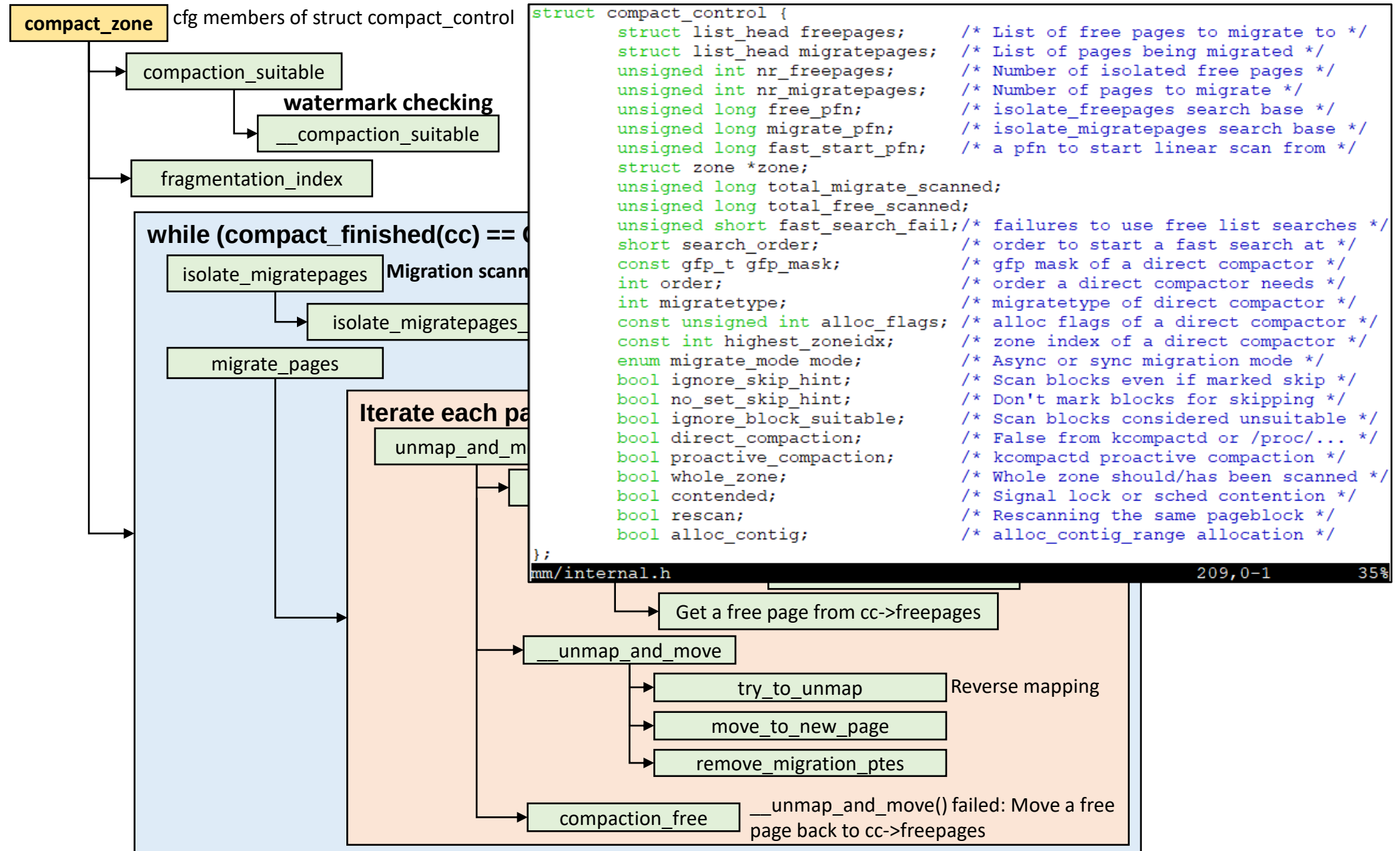
Migration Scanner & Free Scanner: concept (2/2)



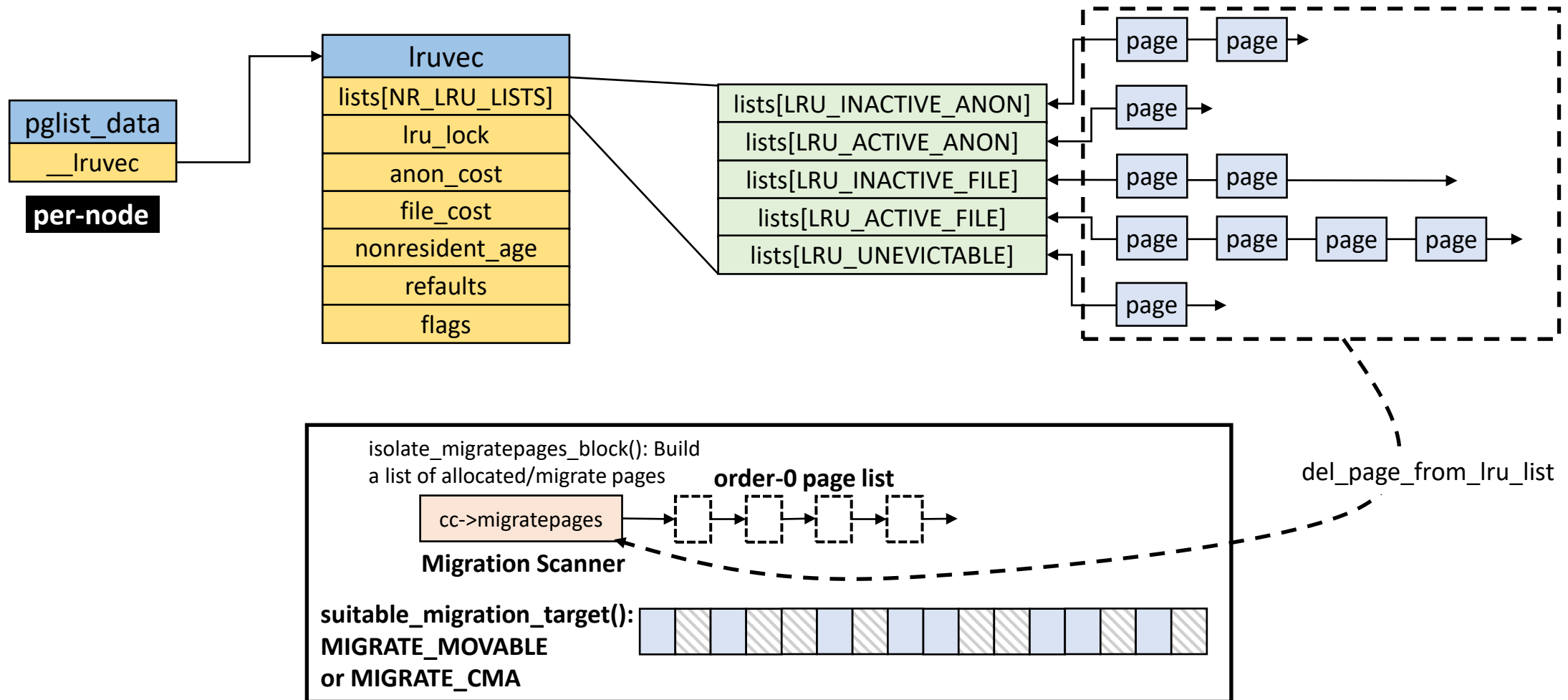
compact_zone() – call path



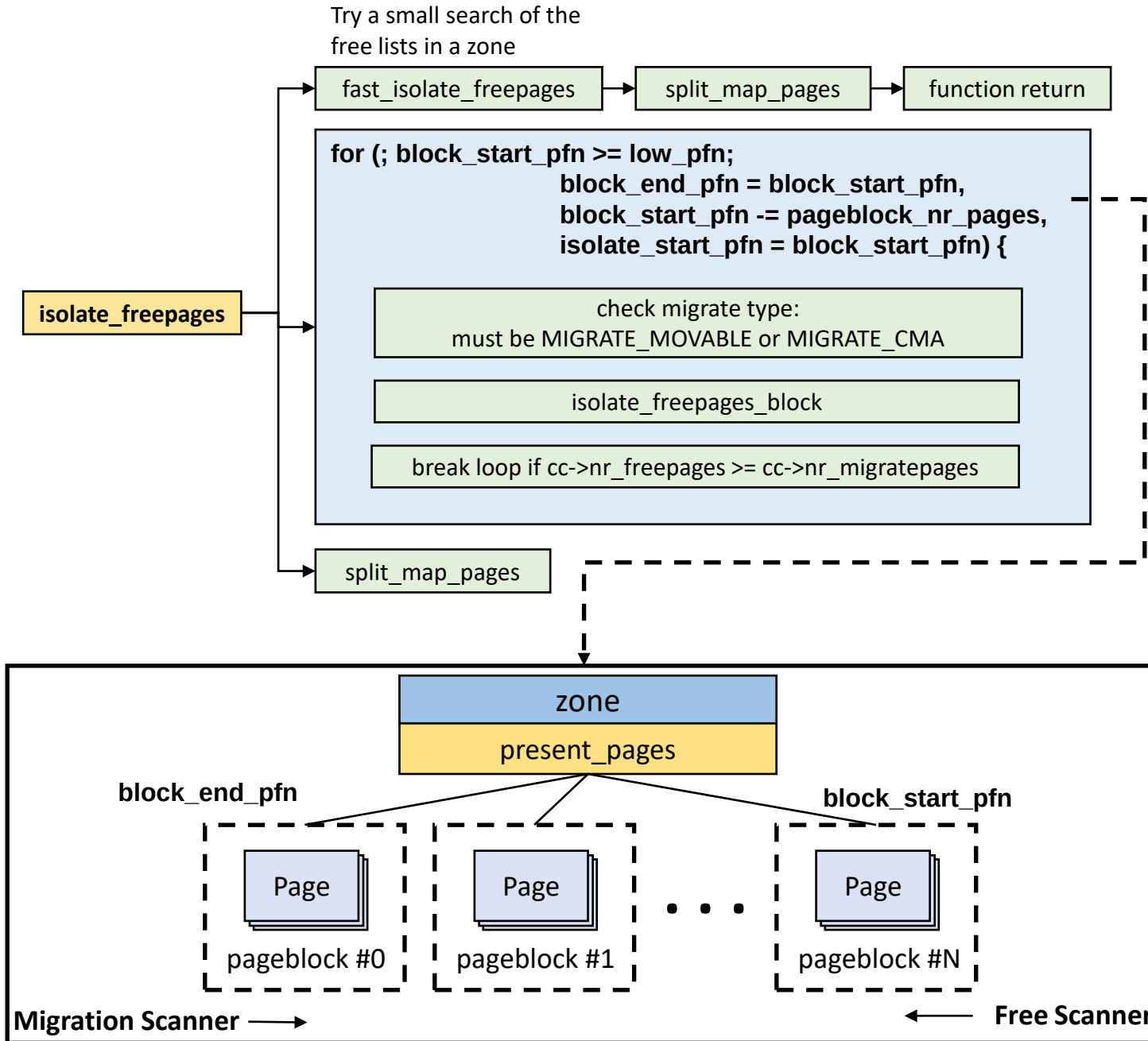
compact_zone() – call path



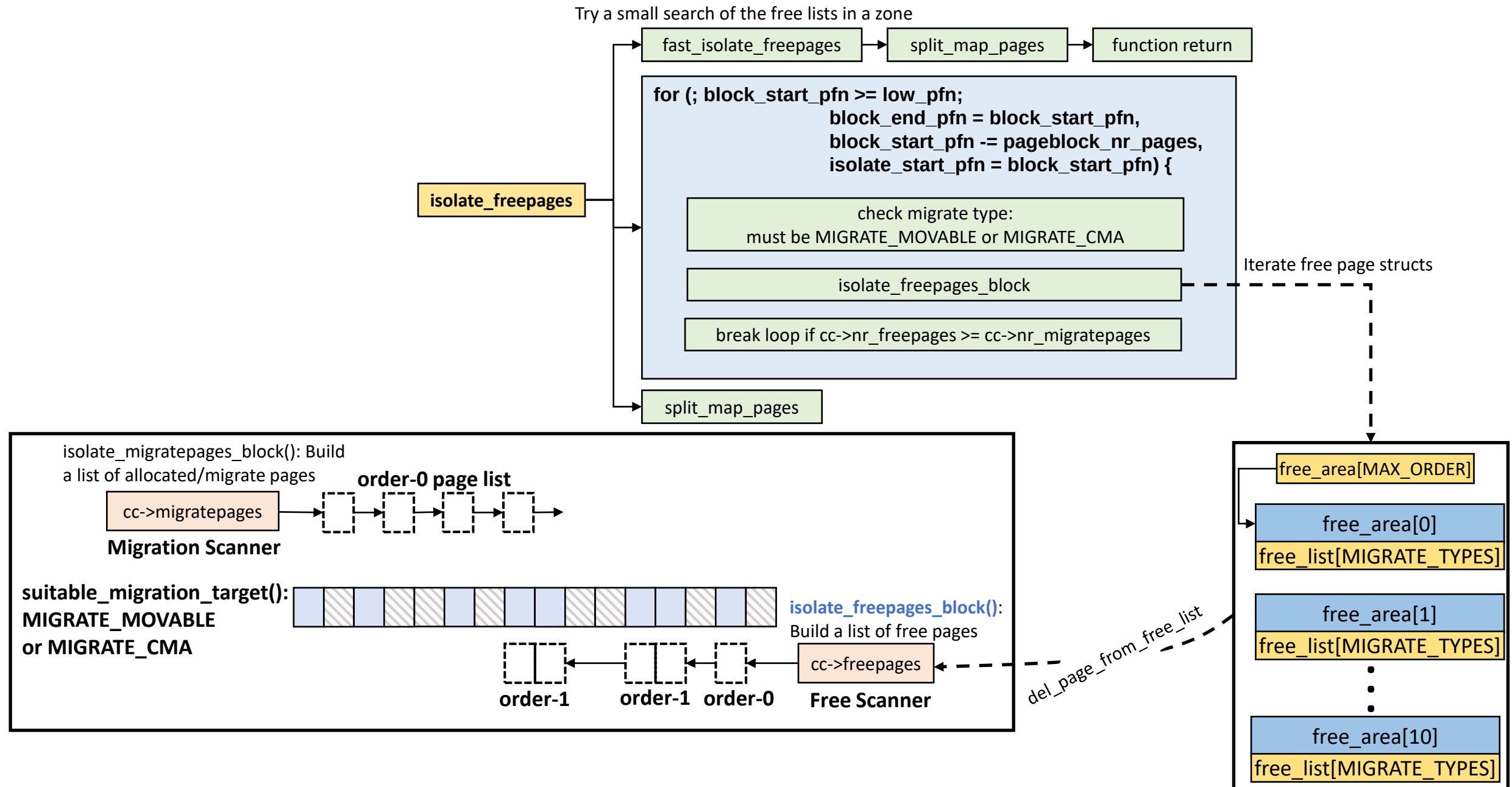
Migration Scanner: isolate_migratepages_block ()



Free Scanner - isolate_freepages(): Iterate pageblock (1/3)

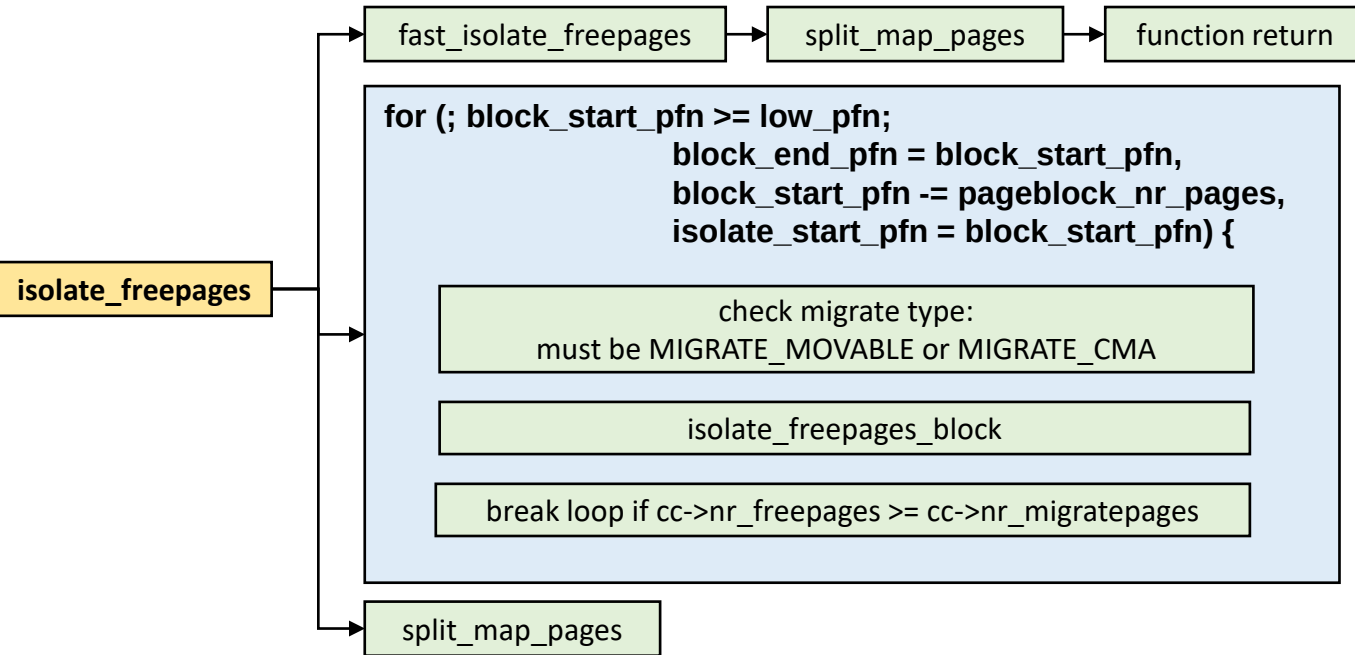


Free Scanner - isolate_freepages_block(): Iterate free page structs (2/3)

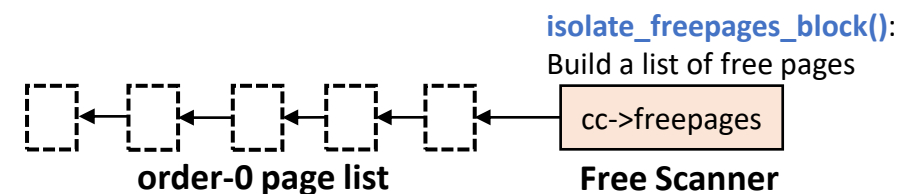
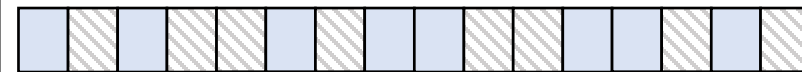
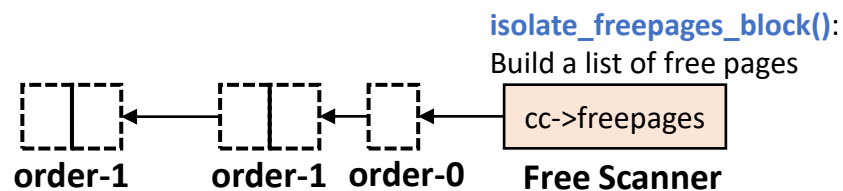


Free Scanner - split_map_pages(): Build order-0 page list (3/3)

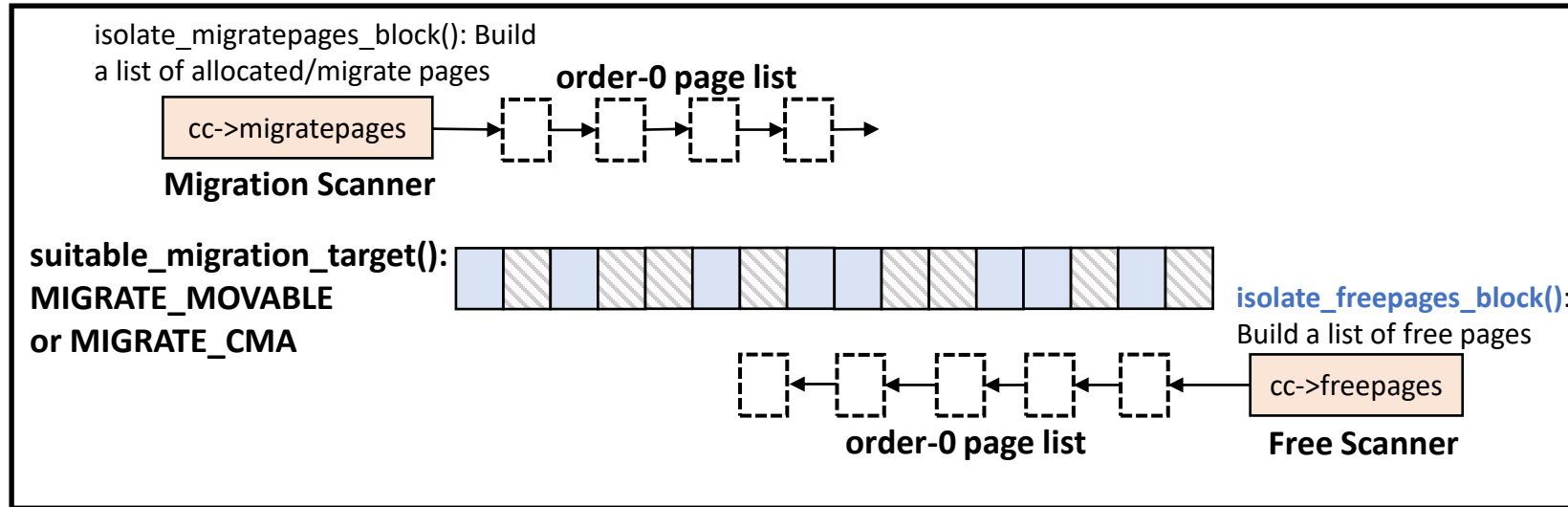
Try a small search of the free lists in a zone



split_map_pages

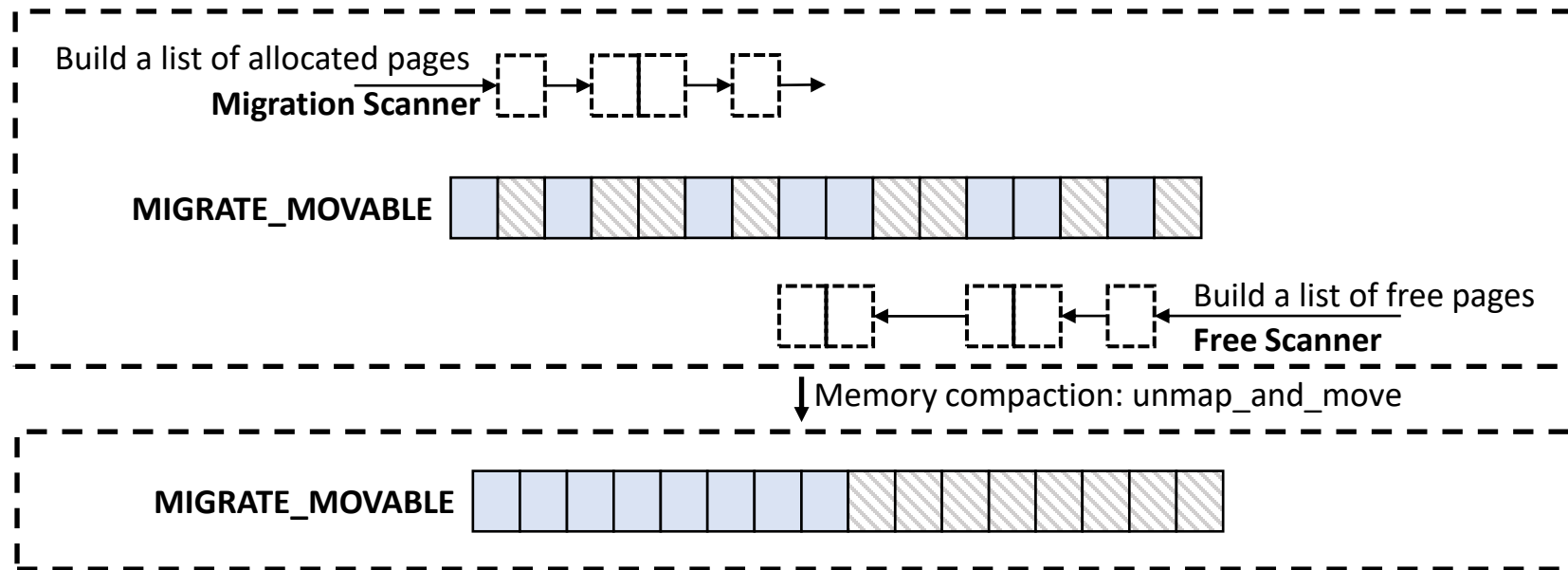


Migration/Free Scanner: Mission Completed

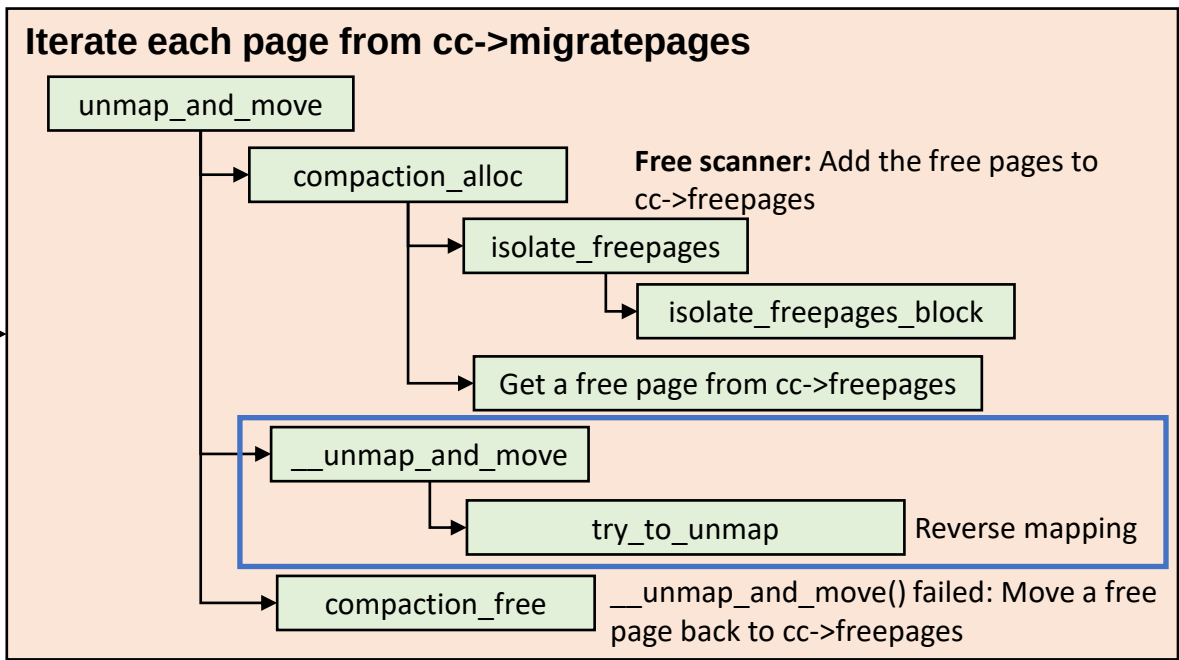


1. Buddy system (`free_list`): order-N
2. `lru/cc->migratepages/cc->freepages`: order-0 basis

__unmap_and_move()



migrate_pages

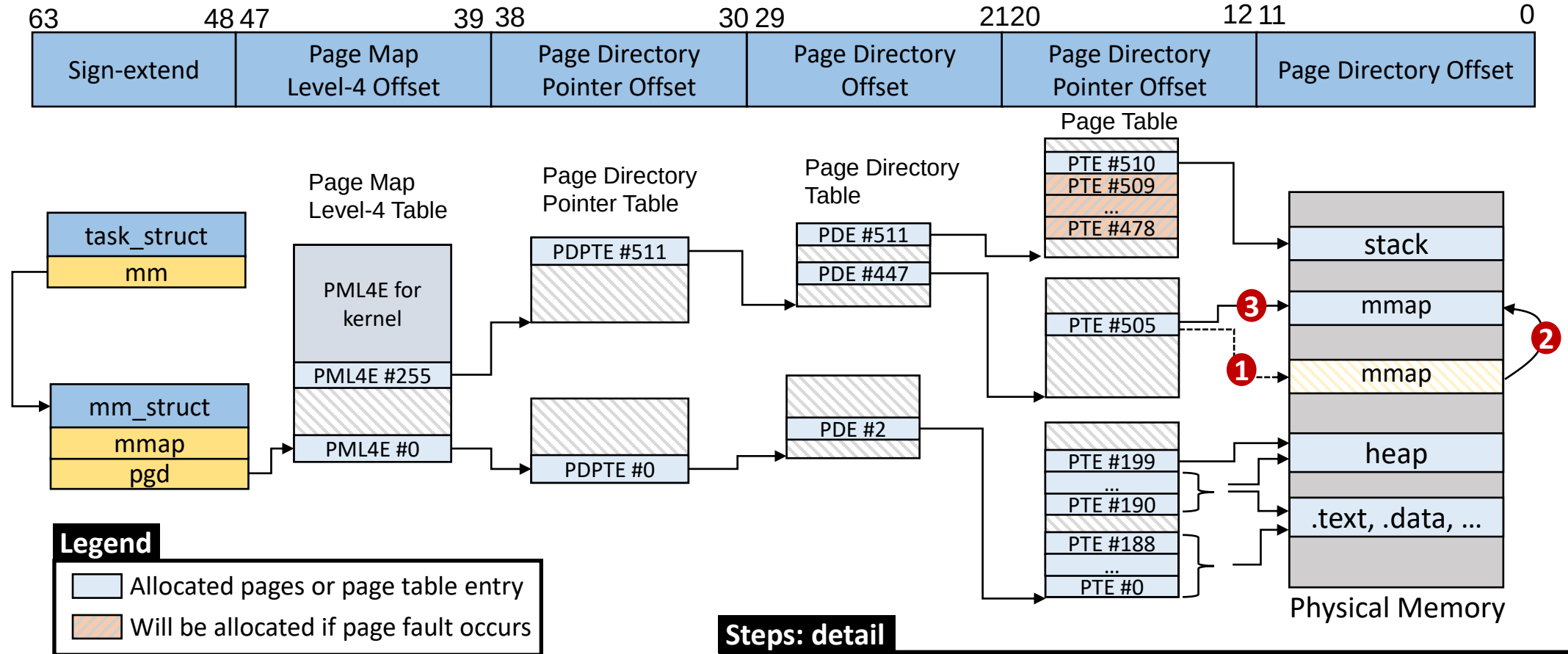


Note

free page allocated page

__unmap_and_move(): page table changes

Linear address of the moved page: 0x7fff7ff9000



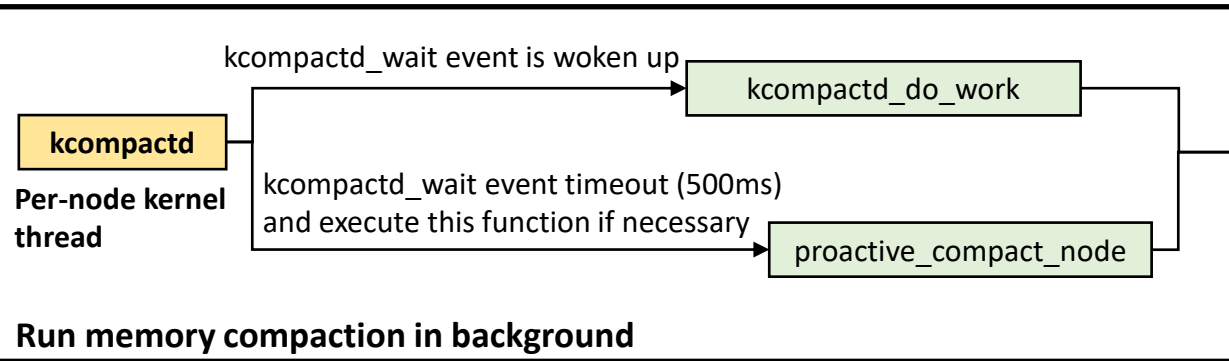
Steps: detail

- 1 `__unmap_and_move` -> `try_to_unmap`
 - Reverse mapping: unmap all PTEs associated with this physical page
- 2 `__unmap_and_move` -> `move_to_new_page`: data copy
- 3 `__unmap_and_move` -> `remove_migration_ptes`
 - Reverse Mapping: restore a migration pte to a new physical page

Scenario Creation: gdb observation

1. Migration scanner & free scanner
2. unmap -> copy page -> pte change

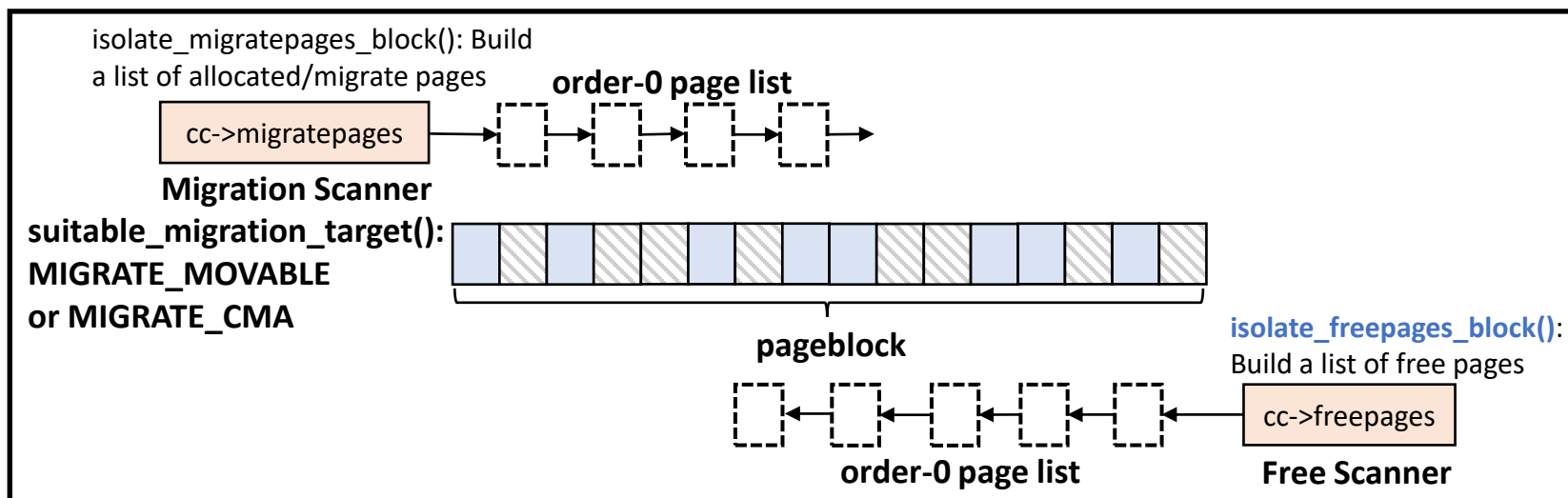
Scenario Creation



```
/ # ./stress-ng --vm 8 --vm-bytes 90% -t 10m
```

```
(gdb) bt
#0  try_to_unmap (page=page@entry=0xfffffea00040b07c0,
    flags=flags@entry=(TTU_MIGRATION | TTU_IGNORE_MLOCK))
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/rmap.c:1742
#1  0xffffffff81117249 in __unmap_and_move (mode=<optimized out>, force=0,
    newpage=0xfffffea0010ffcdc0, page=0xfffffea00040b07c0)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:112
#2  unmap_and_move (ret=0xffffc900001ffd08, reason=MR_COMPACTON,
    mode=<optimized out>, force=0, page=0xfffffea00040b07c0,
    private=18446683600572120576,
    put_new_page=0xffffffff810d01a0 <compaction_free>,
    get_new_page=0xffffffff810d0ff0 <compaction_alloc>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:119
#3  migrate_pages (from=from@entry=0xffffc900001ffe10,
    get_new_page=get_new_page@entry=0xffffffff810d0ff0 <compaction_alloc>,
    put_new_page=put_new_page@entry=0xffffffff810d01a0 <compaction_free>,
    private=private@entry=18446683600572120576, mode=<optimized out>,
    reason=reason@entry=0)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:146
#4  0xffffffff810d297e in compact_zone (cc=cc@entry=0xffffc900001ffe00,
    capc=capc@entry=0x0 <fixed_percpu_data>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2358
#5  0xffffffff810d30ba in proactive_compact_node (
    pgdat=pgdat@entry=0xffff88843ffdf000)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2604
#6  0xffffffff810d345d in kcompactd (p=0xffff88843ffdf000)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2877
#7  0xffffffff8105892c in kthread (_create=0xffff88810025c240)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/kernel/kthread.c:
292
#8  0xffffffff810011bf in ret_from_fork ()
```

Scenario Creation: Confirm order-0 page list (1/3)

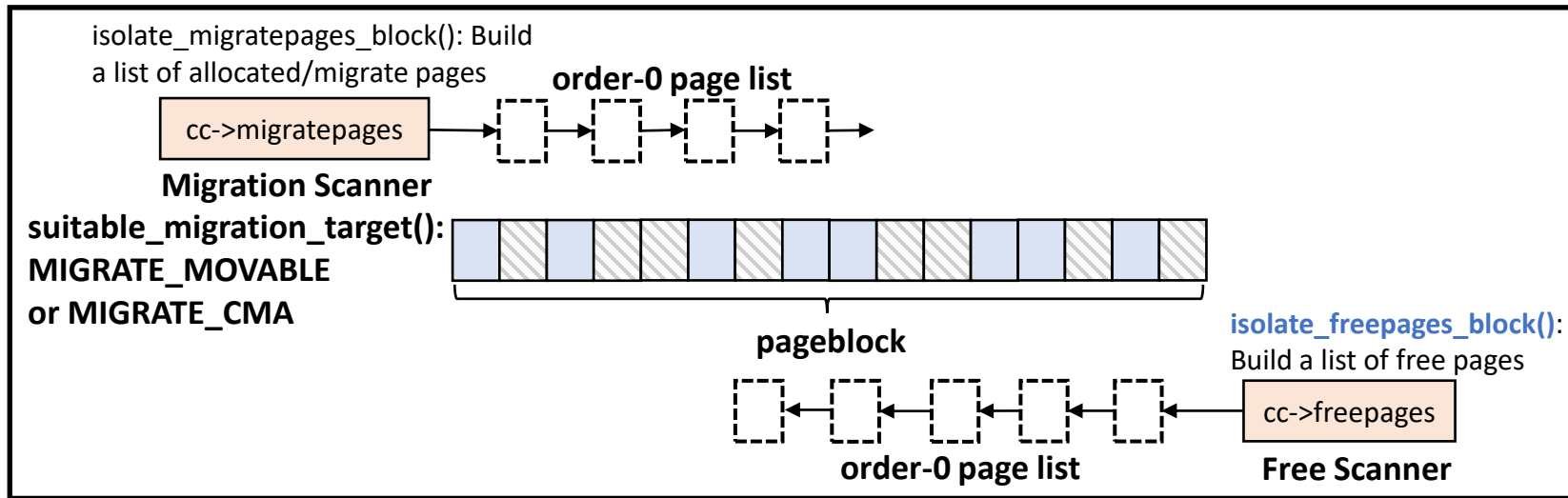


```
(gdb) bt 5
#0  try_to_unmap (page=page@entry=0xfffffea00040b07c0,
    flags=flags@entry=(TTU_MIGRATION | TTU_IGNORE_MLOCK))
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/rmap.c:1742
#1  0xffffffff81117249 in __unmap_and_move (mode=<optimized out>, force=0,
    newpage=0xfffffea0010ffcdc0, page=0xfffffea00040b07c0)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:112
#2  unmap_and_move (ret=0xffffc900001ffd08, reason=MR_COMPACTON,
    mode=<optimized out>, force=0, page=0xfffffea00040b07c0,
    private=18446683600572120576,
    put_new_page=0xffffffff810d01a0 <compaction_free>,
    get_new_page=0xffffffff810d0ff0 <compaction_alloc>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:119
#3  migrate_pages (from=from@entry=0xffffc900001ffe10,
    get_new_page=get_new_page@entry=0xffffffff810d0ff0 <compaction_alloc>,
    put_new_page=put_new_page@entry=0xffffffff810d01a0 <compaction_free>,
    private=private@entry=18446683600572120576, mode=<optimized out>,
    reason=reason@entry=0)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:146
#4  0xffffffff810d297e in compact_zone (cc=cc@entry=0xffffc900001ffe00,
    capc=capc@entry=0x0 <fixed_percpu_data>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2358
(More stack frames follow...)
```

```
(gdb) f 4
#4  0xffffffff810d297e in compact_zone (cc=cc@entry=0xffffc900001ffe00,
    capc=capc@entry=0x0 <fixed_percpu_data>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2358
2358         err = migrate_pages(&cc->migratepages, compaction_alloc,
(gdb) p *cc
$63 = {
  freepages = {
    next = 0xfffffea0010ffcd88,
    prev = 0xfffffea0010ff8408
  },
  migratepages = {
    next = 0xfffffea00040b07c8,
    prev = 0xfffffea00040b0008
  },
  nr_freepages = 34,
  nr_migratepages = 32,
  free_pfn = 4456248,
  migrate_pfn = 1059872,
  fast_start_pfn = 0,
  zone = 0xffff88843ffdf580,
  total_migrate_scanned = 9878,
  total_free_scanned = 290,
  fast_search_fail = 0,
  search_order = 0,
  gfp_mask = 3264,
  order = -1,
  migratetype = 0,
}
```

Free scanner stops
if `cc->nr_freepages >= cc->nr_migratepages`

Scenario Creation: Confirm order-0 page list (2/3)



```
(gdb) f 4
#4 0xffffffff810d297e in compact_zone (cc=cc@entry=0xffffc900001ffe00,
    capc=capc@entry=0x0 <fixed_percpu_data>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2358
2358     err = migrate_pages(&cc->migratepages, compaction_alloc,
(gdb) p *cc
$63 = {
  freepages = {
    next = 0xfffffea0010ffcd88,
    prev = 0xfffffea0010ff8408
  },
  migratepages = {
    next = 0xfffffea00040b07c8,
    prev = 0xfffffea00040b0008
  },
  nr_freepages = 34,
  nr_migratepages = 32,
  free_pfn = 4456248,
  migrate_pfn = 1059872,
  fast_start_pfn = 0,
  zone = 0xffff88843ffdf580,
  total_migrate_scanned = 9878,
  total_free_scanned = 290,
  fast_search_fail = 0,
  search_order = 0,
  gfp_mask = 3264,
  order = -1,
  migratetype = 0,
}
```

Free scanner stops if $cc \rightarrow nr_freepages \geq cc \rightarrow nr_migratepages$

sizeof(struct page) = 64 = 0x40

```
(gdb) p cc->migratepages->next
$68 = (struct list_head *) 0xfffffea00040b07c8
(gdb) p cc->migratepages->next->next
$69 = (struct list_head *) 0xfffffea00040b0788
(gdb) p cc->migratepages->next->next->next
$70 = (struct list_head *) 0xfffffea00040b0748
```

addr offset=0x40

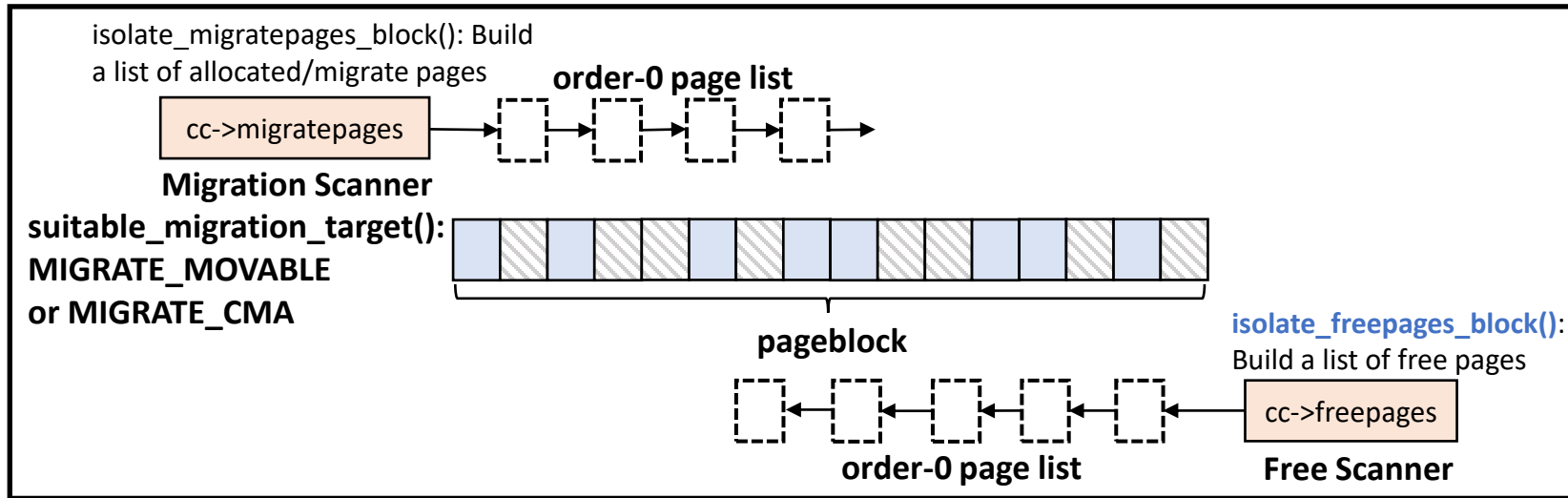
addr offset=0x40

```
(gdb) p cc->freepages->next
$65 = (struct list_head *) 0xfffffea0010ffcd88
(gdb) p cc->freepages->next->next
$66 = (struct list_head *) 0xfffffea0010ffcd48
(gdb) p cc->freepages->next->next->next
$67 = (struct list_head *) 0xfffffea0010ffcd08
```

addr offset=0x40

addr offset=0x40

Scenario Creation: Confirm order-0 page list (3/3)



`sizeof(struct page) = 64 = 0x40`

```
(gdb) p cc->migratepages->next
$68 = (struct list_head *) 0xffffea00040b07c8
(gdb) p cc->migratepages->next->next
$69 = (struct list_head *) 0xffffea00040b0788
(gdb) p cc->migratepages->next->next->next
$70 = (struct list_head *) 0xffffea00040b0748
```

addr offset=0x40

addr offset=0x40

```
(gdb) p cc->freepages->next
$65 = (struct list_head *) 0xffffea0010ffcd88
(gdb) p cc->freepages->next->next
$66 = (struct list_head *) 0xffffea0010ffcd48
(gdb) p cc->freepages->next->next->next
$67 = (struct list_head *) 0xffffea0010ffcd08
```

addr offset=0x40

addr offset=0x40

- Pages managed by buddy system (free pages or free_area)
 - ✓ Page order-N
 - `page->private` denotes order 'N'
- Once pages are removed from buddy system, `page->private` will be set as '0'.
 - ✓ `__rmqueue -> __rmqueue_smallest -> del_page_from_free_list -> set_page_private(page, 0)`
 - ✓ Examples:
 - lru pages (page cache & anonymous page)
 - Kernel memory allocation (GFP_KERNEL)
 - Kernel data structure, kernel page table...

1. Unmap: try_to_unmap()

```
(gdb) bt
#0  map_pte (pvmw=0xffffc900001ffb78)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/page_vma_mappe
ed.c:19
#1  page_vma_mapped_walk (pvmw=pvmw@entry=0xffffc900001ffb78)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/page_vma_mappe
ed.c:217
#2  0xffffffff810ec9be in try_to_unmap_one (page=0xffffea00083d5180,
    vma=<optimized out>, address=<optimized out>, arg=<optimized out>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/rmap.c:1428
#3  0xffffffff810ebbe9 in rmap_walk_anon (page=0xffffea00083d5180,
    rwc=0xffffc900001ffc50, locked=locked@entry=false)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/rmap.c:1873
#4  0xffffffff810ed840 in rmap_walk (page=<optimized out>, rwc=<optimized out>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/rmap.c:1943
#5  0xffffffff810edb09 in try_to_unmap (page=page@entry=0xffffea00083d5180,
    flags=flags@entry=(TTU_MIGRATION | TTU_IGNORE_MLOCK))
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/rmap.c:1764
#6  0xffffffff81117249 in __unmap_and_move (mode=<optimized out>, force=0,
    newpage=0xffffea000fba3fc0, page=0xffffea00083d5180)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:112
#7  unmap_and_move (ret=0xffffc900001ffcf8, reason=MR_COMPACTON,
    mode=<optimized out>, force=0, page=0xffffea00083d5180,
    private=18446683600572120560,
    put_new_page=0xffffffff810d01a0 <compaction_free>,
    get_new_page=0xffffffff810d0ff0 <compaction_alloc>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:119
#8  migrate_pages (from=from@entry=0xffffc900001ffe00,
    get_new_page=get_new_page@entry=0xffffffff810d0ff0 <compaction_alloc>,
    put_new_page=put_new_page@entry=0xffffffff810d01a0 <compaction_free>,
    private=private@entry=18446683600572120560, mode=<optimized out>,
    reason=reason@entry=0)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:146
#9  0xffffffff810d297e in compact_zone (cc=cc@entry=0xffffc900001ffdf0,
    capc=capc@entry=0x0 <fixed_percpu_data>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2358
#10 0xffffffff810d321a in kcompactd_do_work (
    pgdat=pgdat@entry=0xffff88843ffdf000)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2768
#11 0xffffffff810d334a in kcompactd (p=0xffff88843ffdf000)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/compaction.c:
2863
```

```
(gdb) p *pvmw
$11 = {
    page = 0xffffea00083d5180,
    vma = 0xffff88810089c0c0,
    address = 139817161609216,
    pmd = 0xffff88810036ce10,
    pte = 0xffff8881809c0028,
    ptl = 0x0 <fixed_percpu_data>,
    flags = 0
}
(gdb) p ((struct page *) 0xffffea00083d5180)->_mapcount
$12 = {
    counter = 0
}
(gdb) p /x *(unsigned long *) 0xffff8881809c0028
$13 = 0x800000020f546067
```

Only one pte mapped

pte: physical address

```
(gdb) info b
Num  Type           Disp Enb Address                               What
1    breakpoint     keep y   0xffffffff81001760 in emulate_vsyscall
                                at /home/adrian/git-repo/gdb-
linux-real-mode/src/linux-5.11/arch/x86/entry/vsyscall/vsyscall_64.c:131
2    breakpoint     keep y   0xffffffff81116959 in __unmap_and_move
                                at /home/adrian/git-repo/gdb-
linux-real-mode/src/linux-5.11/arch/x86/include/asm/current.h:15
3    breakpoint     keep y   0xffffffff810e9f10 in page_vma_mapped_walk
                                at /home/adrian/git-repo/gdb-
linux-real-mode/src/linux-5.11/mm/page_vma_mapped.c:145
5    hw watchpoint  keep y   *(0xffff8881809c0028)
```

1. Unmap: try_to_unmap()

sizeof(struct page) = 64 = 0x40

```
(gdb) p *pvmw
$11 = {
  page = 0xffffea00083d5180,
  vma = 0xffff88810089c0c0,
  address = 139817161609216,
  pmd = 0xffff88810036ce10,
  pte = 0xffff8881809c0028,
  ptl = 0x0 <fixed_percpu_data>,
  flags = 0
}
(gdb) p ((struct page *) 0xffffea00083d5180)->_mapcount
$12 = {
  counter = 0
}
(gdb) p /x *(unsigned long *) 0xffff8881809c0028
$13 = 0x800000020f546067
```

BIOS-provided physical RAM map:

BIOS-e820:	[mem 0x0000000000000000-0x0000000000009fbff]	usable
BIOS-e820:	[mem 0x0000000000009fc00-0x0000000000009ffff]	reserved
BIOS-e820:	[mem 0x000000000000f0000-0x000000000000ffffff]	reserved
BIOS-e820:	[mem 0x00000000000100000-0x000000000bffdffff]	usable
BIOS-e820:	[mem 0x00000000bffe0000-0x00000000bfffffff]	reserved
BIOS-e820:	[mem 0x00000000fffc0000-0x00000000ffffffff]	reserved
BIOS-e820:	[mem 0x0000000100000000-0x000000043fffffff]	usable

page= 0xffffea00083d5180
pfn = (0xffffea00083d5180 - 0xffffea0000000000) / sizeof(struct page) = 2159942 = 0x20F546
physical address = 2159942 * 4096 = 0x2_0F54_6000

- Kernel direct mapping address (64TB – 4-level paging):
‘page_offset_base’
 - Reference: Documentation/x86/x86_64/mm.rst

```
(gdb) x /24g 0xffff88820f546000
0xffff88820f546000: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff88820f546010: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff88820f546020: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff88820f546030: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff88820f546040: 0x3131313131313131 0x3131313131313131
0xffff88820f546050: 0x3131313131313131 0x3131313131313131
0xffff88820f546060: 0x3131313131313131 0x3131313131313131
0xffff88820f546070: 0x3131313131313131 0x3131313131313131
0xffff88820f546080: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff88820f546090: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff88820f5460a0: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff88820f5460b0: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
```

1. Unmap: try_to_unmap() -> try_to_unmap_one()

```
(gdb) c
Continuing.
-----[ STACK ]-----
0x1fffb18:      Error while running hook_stop:
Cannot access memory at address 0x1fffb18

Thread 4 hit Hardware watchpoint 5: *(0xffff8881809c0028)

Old value = <unreadable>
New value = 0
ptep_clear_flush (vma=vma@entry=0xffff88810089c0c0, address=139817161609216,
    ptep=0xffff8881809c0028)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/pgtable-gener
ic.c:96
96          if (pte_accessible(mm, pte))
```

```
static bool try_to_unmap_one(struct page *page, struct vm_area_struct *vma,
    unsigned long address, void *arg)
{
+---162 lines: struct mm_struct *mm = vma->vm_mm;-----
    if (should_defer_flush(mm, flags)) {
        /*
         * We clear the PTE but do not flush so potentially
         * a remote CPU could still be writing to the page.
         * If the entry was previously clean then the
         * architecture must guarantee that a clear->dirty
         * transition on a cached TLB entry is written through
         * and traps if the PTE is unmapped.
         */
        pteval = ptep_get_and_clear(mm, address, pvmw.pte);

        set_tlb_ufc_flush_pending(mm, pte_dirty(pteval));
    } else {
        pteval = ptep_clear_flush(vma, address, pvmw.pte);
    }
}
```

1. Unmap: try_to_unmap() -> try_to_unmap_one()

```
(gdb) c
Continuing.
-----[ STACK ]-----
0x1ffb30:      Error while running hook_stop:
Cannot access memory at address 0x1ffb30

Thread 4 hit Hardware watchpoint 5: *(0xffff8881809c0028)

Old value = <unreadable>
New value = -1105890816
try_to_unmap_one (page=0xffffea00083d5180, vma=<optimized out>,
address=<optimized out>, arg=<optimized out>)
at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/arch/x86/include
/asm/pgtable_64.h:67
67      WRITE_ONCE(*ptep, pte);
```

```
(gdb) bt 3
#0  try_to_unmap_one (page=0xffffea00083d5180, vma=<optimized out>,
address=<optimized out>, arg=<optimized out>)
at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/arch/x86/include
/asm/pgtable_64.h:67
#1  0xffffffff810ebb9 in rmap_walk_anon (page=0xffffea00083d5180,
rcw=0xfffffc900001ffc50, locked=locked@entry=false)
at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/rmap.c:1873
#2  0xffffffff810ed840 in rmap_walk (page=<optimized out>, rcw=<optimized out>)
at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/rmap.c:1943
(More stack frames follow...)
(gdb) p /x *((unsigned long *) 0xffff8881809c0028)
$19 = 0xfffffffffb157200
```

```
static bool try_to_unmap_one(struct page *page, struct vm_area_struct *vma,
unsigned long address, void *arg)
{
+--212 lines: struct mm_struct *mm = vma->vm_mm;-----
    } else if (IS_ENABLED(CONFIG_MIGRATION) &&
               (flags & (TTU_MIGRATION|TTU_SPLIT_FREEZE))) {
        swp_entry_t entry;
        pte_t swp_pte;

        if (arch_unmap_one(mm, vma, address, pteval) < 0) {
            set_pte_at(mm, address, pvmw.pte, pteval);
            ret = false;
            page_vma_mapped_walk_done(&pvmw);
            break;
        }

        /*
         * Store the pfn of the page in a special migration
         * pte. do_swap_page() will wait until the migration
         * pte is removed and then restart fault handling.
         */
        entry = make_migration_entry(subpage,
                                     pte_write(pteval));
        swp_pte = swp_entry_to_pte(entry);
        if (pte_soft_dirty(pteval))
            swp_pte = pte_swp_mksoft_dirty(swp_pte);
        if (pte_uffd_wp(pteval))
            swp_pte = pte_swp_mkuffd_wp(swp_pte);
        set_pte_at(mm, address, pvmw.pte, swp_pte);
        /*
         * No need to invalidate here it will synchronize on
         * against the special swap migration pte.
         */
    } else if (PageAnon(page)) {
        swp_entry_t entry = { .val = page_private(subpage) };
mm/rmap.c 1618,4-25
```


2. copy page: migrate_page_move_mapping()

sizeof(struct page) = 64 = 0x40

```
(gdb) bt 4
#0 migrate_page_move_mapping (mapping=0x0 <fixed_percpu_data>,
    newpage=newpage@entry=0xffffea000fba3fc0,
    page=page@entry=0xffffea00083d5180, extra_count=extra_count@entry=0)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:401
#1 0xffffffff81115bf4 in migrate_page (mapping=<optimized out>,
    newpage=0xffffea000fba3fc0, page=0xffffea00083d5180,
    mode=MIGRATE_SYNC_LIGHT)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:702
#2 0xffffffff81116375 in move_to_new_page (
    newpage=newpage@entry=0xffffea000fba3fc0,
    page=page@entry=0xffffea00083d5180, mode=mode@entry=MIGRATE_SYNC_LIGHT)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:953
#3 0xffffffff81117267 in __unmap_and_move (mode=<optimized out>, force=0,
    newpage=0xffffea000fba3fc0, page=0xffffea00083d5180)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:112
7
(More stack frames follow...)
```

page= 0xffffea00083d5180
pfn = (0xffffea00083d5180 - 0xffffea0000000000) / sizeof(struct page) = 2159942 = 0x20F546
physical address = 2159942 * 4096 = 0x2_0F54_6000

newpage= 0xffffea000fba3fc0
newpfn = (0xffffea000fba3fc0 - 0xffffea0000000000) / sizeof(struct page) = 4122879 = 0x3EE8FF
New physical address = 4122879 * 4096 = 0x3_EE8F_F000

2. copy page: migrate_page_move_mapping()

```
page= 0xffffea00083d5180
pfn = (0xffffea00083d5180 - 0xffffea0000000000) / sizeof(struct page) = 2159942 = 0x20F546
physical address = 2159942 * 4096 = 0x2_0F54_6000

newpage= 0xffffea000fba3fc0
newpfn = (0xffffea000fba3fc0 - 0xffffea0000000000) / sizeof(struct page) = 4122879 = 0x3EE8FF
New physical address = 4122879 * 4096 = 0x3_EE8F_F000
```

page: access from kernel direct mapping address

```
(gdb) x /24g 0xfffff88820f546000
0xfffff88820f546000: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xfffff88820f546010: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xfffff88820f546020: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xfffff88820f546030: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xfffff88820f546040: 0x3131313131313131 0x3131313131313131
0xfffff88820f546050: 0x3131313131313131 0x3131313131313131
0xfffff88820f546060: 0x3131313131313131 0x3131313131313131
0xfffff88820f546070: 0x3131313131313131 0x3131313131313131
0xfffff88820f546080: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xfffff88820f546090: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xfffff88820f5460a0: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xfffff88820f5460b0: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
```

newpage: access from kernel direct mapping address

```
(gdb) x /24g 0xfffff8883ee8ff000
0xfffff8883ee8ff000: 0x3030303030303030 0x7676767676767676
0xfffff8883ee8ff010: 0xedededededededed 0x6262626262626262
0xfffff8883ee8ff020: 0x8f8f8f8f8f8f8f8f 0x0f0f0f0f0f0f0f0f
0xfffff8883ee8ff030: 0xd6d6d6d6d6d6d6d6 0x7575757575757575
0xfffff8883ee8ff040: 0x4e4e4e4e4e4e4e4e 0x1212121212121212
0xfffff8883ee8ff050: 0xb4b4b4b4b4b4b4b4 0xd3d3d3d3d3d3d3d3
0xfffff8883ee8ff060: 0x0303030303030303 0x9090909090909090
0xfffff8883ee8ff070: 0x9595959595959595 0xafafafafafafafafaf
0xfffff8883ee8ff080: 0x7777777777777777 0xfffffffffffffffffff
0xfffff8883ee8ff090: 0xddddddddddddddddd 0xcdcdcdcdcdcdcdcdcd
0xfffff8883ee8ff0a0: 0xb8b8b8b8b8b8b8b8 0x8b8b8b8b8b8b8b8b
0xfffff8883ee8ff0b0: 0x0404040404040404 0x1f1f1f1f1f1f1f1f
```

migrate_page_copy

* more info for “kernel direct mapping”: page #5 of [Decompressed vmlinux: linux kernel initialization from page table configuration perspective](#)

Call path: migrate_page_copy->copy_highpage->{kmap_atomic, copy_page}

2. migrate_page_move_mapping() -> migrate_page_copy()

```
(gdb) c
Continuing.
-----[ STACK ]-----
0x1ffbf8:      Error while running hook_stop:
Cannot access memory at address 0x1ffbf8

Thread 4 hit Breakpoint 7, migrate_page_copy (
  newpage=newpage@entry=0xffffea000fba3fc0,
  page=page@entry=0xffffea00083d5180)
  at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:675
675      if (PageHuge(page) || PageTransHuge(page))
(gdb) bt 4
#0  migrate_page_copy (newpage=newpage@entry=0xffffea000fba3fc0,
  page=page@entry=0xffffea00083d5180)
  at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:675
#1  0xffffffff81115c0b in migrate_page (mapping=<optimized out>,
  newpage=0xffffea000fba3fc0, page=0xffffea00083d5180,
  mode=MIGRATE_SYNC_LIGHT)
  at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:708
#2  0xffffffff81116375 in move_to_new_page (
  newpage=newpage@entry=0xffffea000fba3fc0,
  page=page@entry=0xffffea00083d5180, mode=mode@entry=MIGRATE_SYNC_LIGHT)
  at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:953
#3  0xffffffff81117267 in __unmap_and_move (mode=<optimized out>, force=0,
  newpage=0xffffea000fba3fc0, page=0xffffea00083d5180)
  at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:112
7
(More stack frames follow...)
```

page: access from kernel direct mapping address

```
(gdb) x /24g 0xffff88820f546000
0xffff88820f546000:    0xd2d2d2d2d2d2d2d2    0xd2d2d2d2d2d2d2d2
0xffff88820f546010:    0xd2d2d2d2d2d2d2d2    0xd2d2d2d2d2d2d2d2
0xffff88820f546020:    0xd2d2d2d2d2d2d2d2    0xd2d2d2d2d2d2d2d2
0xffff88820f546030:    0xd2d2d2d2d2d2d2d2    0xd2d2d2d2d2d2d2d2
0xffff88820f546040:    0x3131313131313131    0x3131313131313131
0xffff88820f546050:    0x3131313131313131    0x3131313131313131
0xffff88820f546060:    0x3131313131313131    0x3131313131313131
0xffff88820f546070:    0x3131313131313131    0x3131313131313131
0xffff88820f546080:    0xc0c0c0c0c0c0c0c0    0xc0c0c0c0c0c0c0c0
0xffff88820f546090:    0xc0c0c0c0c0c0c0c0    0xc0c0c0c0c0c0c0c0
0xffff88820f5460a0:    0xc0c0c0c0c0c0c0c0    0xc0c0c0c0c0c0c0c0
0xffff88820f5460b0:    0xc0c0c0c0c0c0c0c0    0xc0c0c0c0c0c0c0c0
```

newpage: access from kernel direct mapping address

```
(gdb) x /24g 0xffff8883ee8ff000
0xffff8883ee8ff000:    0x3030303030303030    0x7676767676767676
0xffff8883ee8ff010:    0xededededededededed    0x6262626262626262
0xffff8883ee8ff020:    0x8f8f8f8f8f8f8f8f    0x0f0f0f0f0f0f0f0f
0xffff8883ee8ff030:    0xd6d6d6d6d6d6d6d6    0x7575757575757575
0xffff8883ee8ff040:    0x4e4e4e4e4e4e4e4e    0x1212121212121212
0xffff8883ee8ff050:    0xb4b4b4b4b4b4b4b4    0xd3d3d3d3d3d3d3d3
0xffff8883ee8ff060:    0x0303030303030303    0x9090909090909090
0xffff8883ee8ff070:    0x9595959595959595    0xafafafafafafafaf
0xffff8883ee8ff080:    0x7777777777777777    0xffffffffffffffff
0xffff8883ee8ff090:    0xdddddddddddddddd    0xcdcdcdcdcdcdcdcd
0xffff8883ee8ff0a0:    0xb8b8b8b8b8b8b8b8    0x8b8b8b8b8b8b8b8b
0xffff8883ee8ff0b0:    0x0404040404040404    0x1f1f1f1f1f1f1f1f
```

migrate_page_copy

2. migrate_page_move_mapping() -> migrate_page_copy()

```
page= 0xffffea00083d5180
pfn = (0xffffea00083d5180 - 0xffffea0000000000) / sizeof(struct page) = 2159942 = 0x20F546
physical address = 2159942 * 4096 = 0x2_0F54_6000
```

```
newpage= 0xffffea000fba3fc0
newpfn = (0xffffea000fba3fc0 - 0xffffea0000000000) / sizeof(struct page) = 4122879 = 0x3EE8FF
New physical address = 4122879 * 4096 = 0x3_EE8F_F000
```

page: access from kernel direct mapping address

```
(gdb) x /24g 0xffff88820f546000
0xffff88820f546000: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff88820f546010: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff88820f546020: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff88820f546030: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff88820f546040: 0x3131313131313131 0x3131313131313131
0xffff88820f546050: 0x3131313131313131 0x3131313131313131
0xffff88820f546060: 0x3131313131313131 0x3131313131313131
0xffff88820f546070: 0x3131313131313131 0x3131313131313131
0xffff88820f546080: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff88820f546090: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff88820f5460a0: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff88820f5460b0: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
```

migrate_page_copy

```
(gdb) x /24g 0xffff8883EE8FF000
0xffff8883ee8ff000: 0x3030303030303030 0x7676767676767676
0xffff8883ee8ff010: 0xedededededededed 0x6262626262626262
0xffff8883ee8ff020: 0x8f8f8f8f8f8f8f8f 0x0f0f0f0f0f0f0f0f
0xffff8883ee8ff030: 0xd6d6d6d6d6d6d6d6 0x7575757575757575
0xffff8883ee8ff040: 0x4e4e4e4e4e4e4e4e 0x1212121212121212
0xffff8883ee8ff050: 0xb4b4b4b4b4b4b4b4 0xd3d3d3d3d3d3d3d3
0xffff8883ee8ff060: 0x0303030303030303 0x9090909090909090
0xffff8883ee8ff070: 0x9595959595959595 0xafafafafafafafaf
0xffff8883ee8ff080: 0x7777777777777777 0xffffffffffffffff
0xffff8883ee8ff090: 0xdddddddddddddddd 0xcdcdcdcdcdcdcdcd
0xffff8883ee8ff0a0: 0xb8b8b8b8b8b8b8b8 0x8b8b8b8b8b8b8b8b
0xffff8883ee8ff0b0: 0x0404040404040404 0x1f1f1f1f1f1f1f1f
(gdb) c
Continuing.
-----[ STACK ]-----
0x1ffc88: Error while running hook_stop:
Cannot access memory at address 0x1ffc88

Thread 4 hit Breakpoint 8, remove_migration_ptes (
  old=old@entry=0xffffea00083d5180, new=0xffffea000fba3fc0,
  locked=locked@entry=false)
  at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:292
292      struct rmap walk_control rwc = {
(gdb) x /24g 0xffff8883EE8FF000
0xffff8883ee8ff000: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff8883ee8ff010: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff8883ee8ff020: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff8883ee8ff030: 0xd2d2d2d2d2d2d2d2 0xd2d2d2d2d2d2d2d2
0xffff8883ee8ff040: 0x3131313131313131 0x3131313131313131
0xffff8883ee8ff050: 0x3131313131313131 0x3131313131313131
0xffff8883ee8ff060: 0x3131313131313131 0x3131313131313131
0xffff8883ee8ff070: 0x3131313131313131 0x3131313131313131
0xffff8883ee8ff080: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff8883ee8ff090: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff8883ee8ff0a0: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
0xffff8883ee8ff0b0: 0xc0c0c0c0c0c0c0c0 0xc0c0c0c0c0c0c0c0
```

3. pte change: remove_migration_ptes()

```
page= 0xffffea00083d5180
pfn = (0xffffea00083d5180 - 0xffffea0000000000) / sizeof(struct page) = 2159942 = 0x20F546
physical address = 2159942 * 4096 = 0x2_0F54_6000

newpage= 0xffffea000fba3fc0
newpfn = (0xffffea000fba3fc0 - 0xffffea0000000000) / sizeof(struct page) = 4122879 = 0x3EE8FF
New physical address = 4122879 * 4096 = 0x3_EE8F_F000
```

```
(gdb) p /x *(unsigned long *) 0xfffff8881809c0028
$21 = 0xffffffffbbe157200
(gdb) c
Continuing.
-----[ STACK ]-----
0x1ffb80:      Error while running hook_stop:
Cannot access memory at address 0x1ffb80

Thread 4 hit Hardware watchpoint 5: *(0xfffff8881809c0028)

Old value = <unreadable>
New value = -292556793
remove_migration_pte (page=0xffffea000fba3fc0, vma=0xfffff88810089c0c0,
    addr=<optimized out>, old=<optimized out>)
    at /home/adrian/git-repo/gdb-linux-real-mode/src/linux-5.11/mm/migrate.c:268
268         if (PageAnon(new))
(gdb) p /x *(unsigned long *) 0xfffff8881809c0028
$22 = 0x80000003ee8ff007
```

Reference

- [Linux Kernel vs. Memory Fragmentation \(Part 1\)](#)
- <https://www.cnblogs.com/LoyenWang/p/11746357.html>