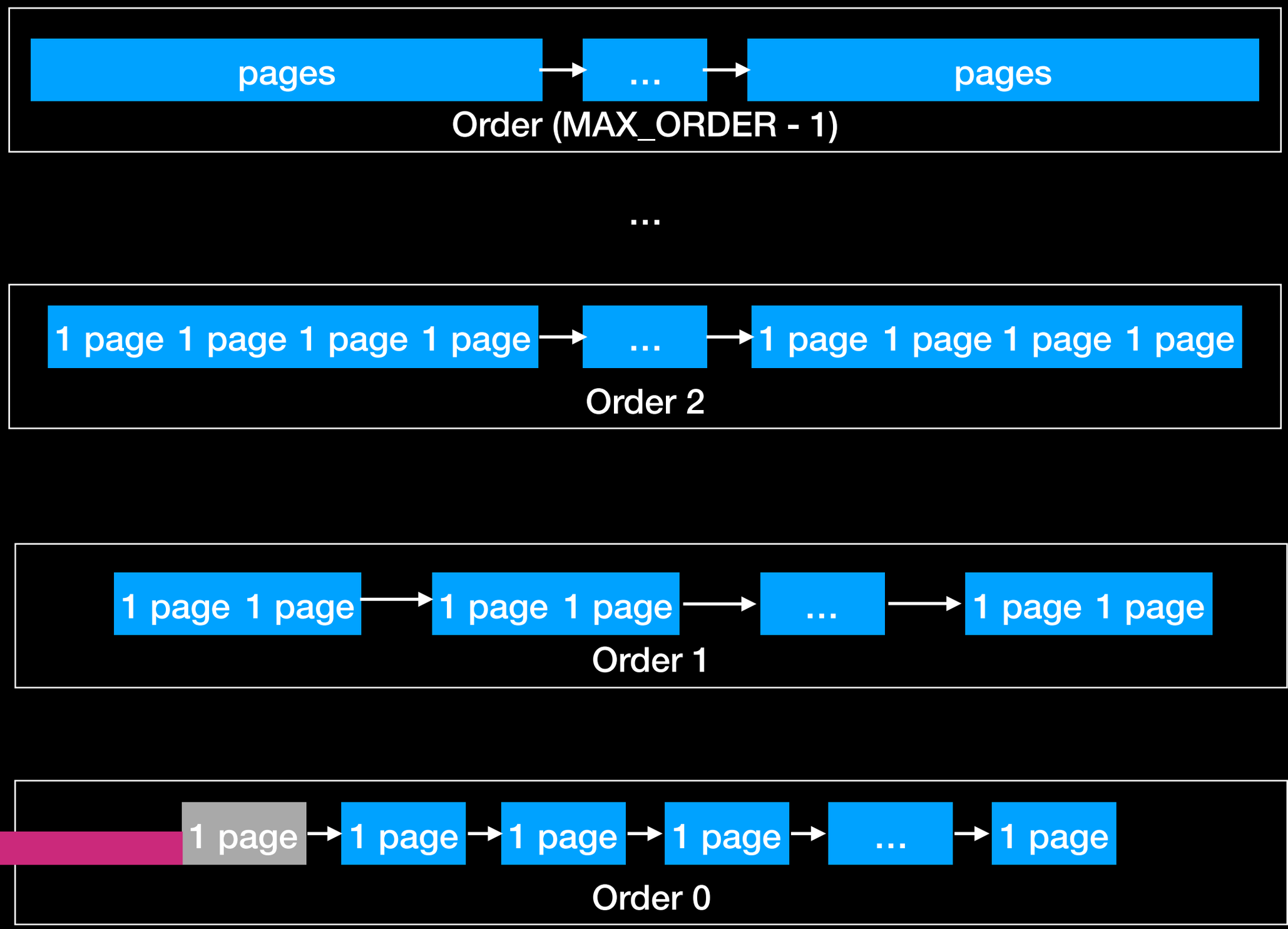
The implementations of anti pages fragmentation in Linux kernel

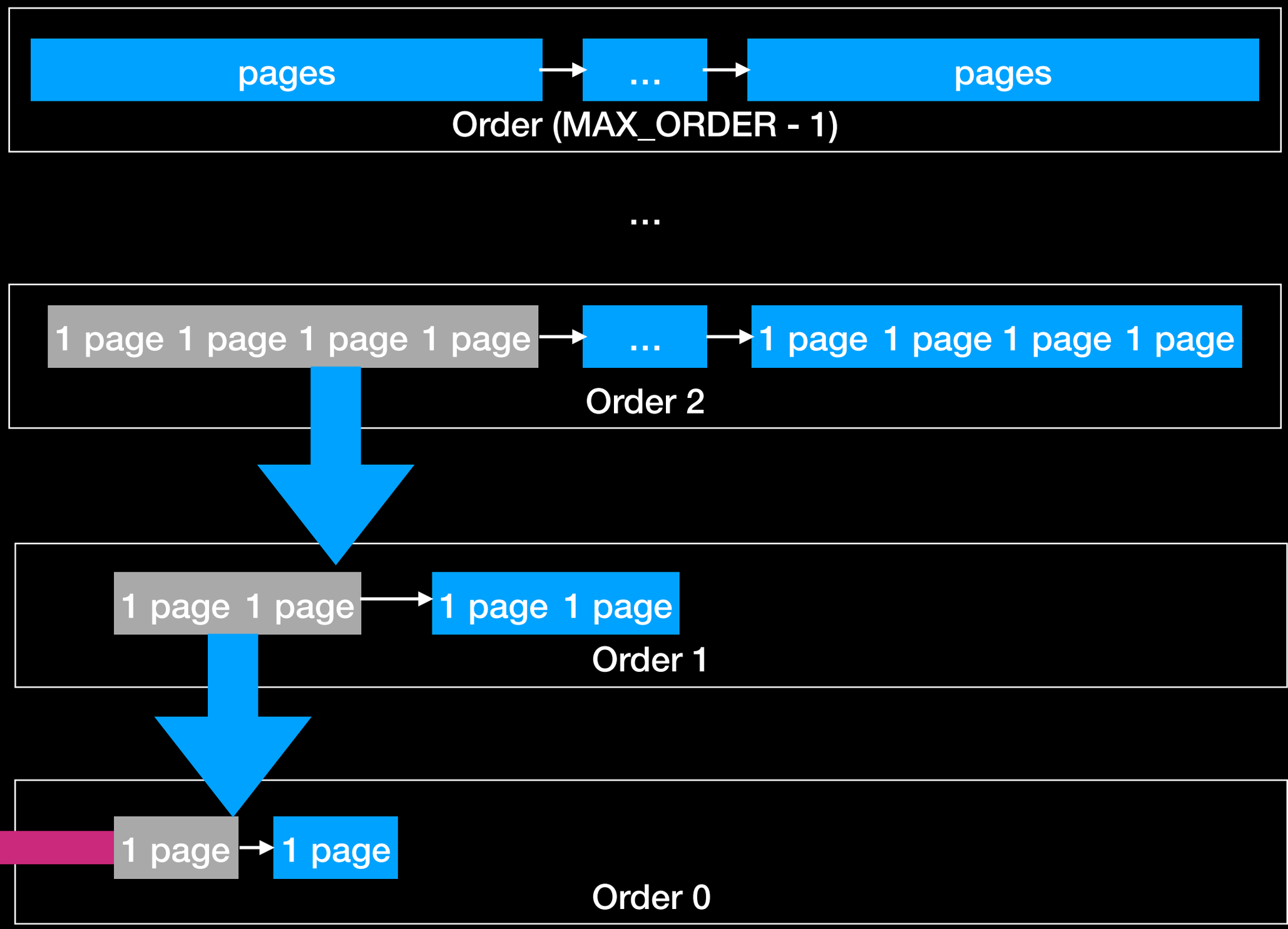
朱辉 teawater@hyper.sh

Buddy系统一个ZONE内空闲页基本结构

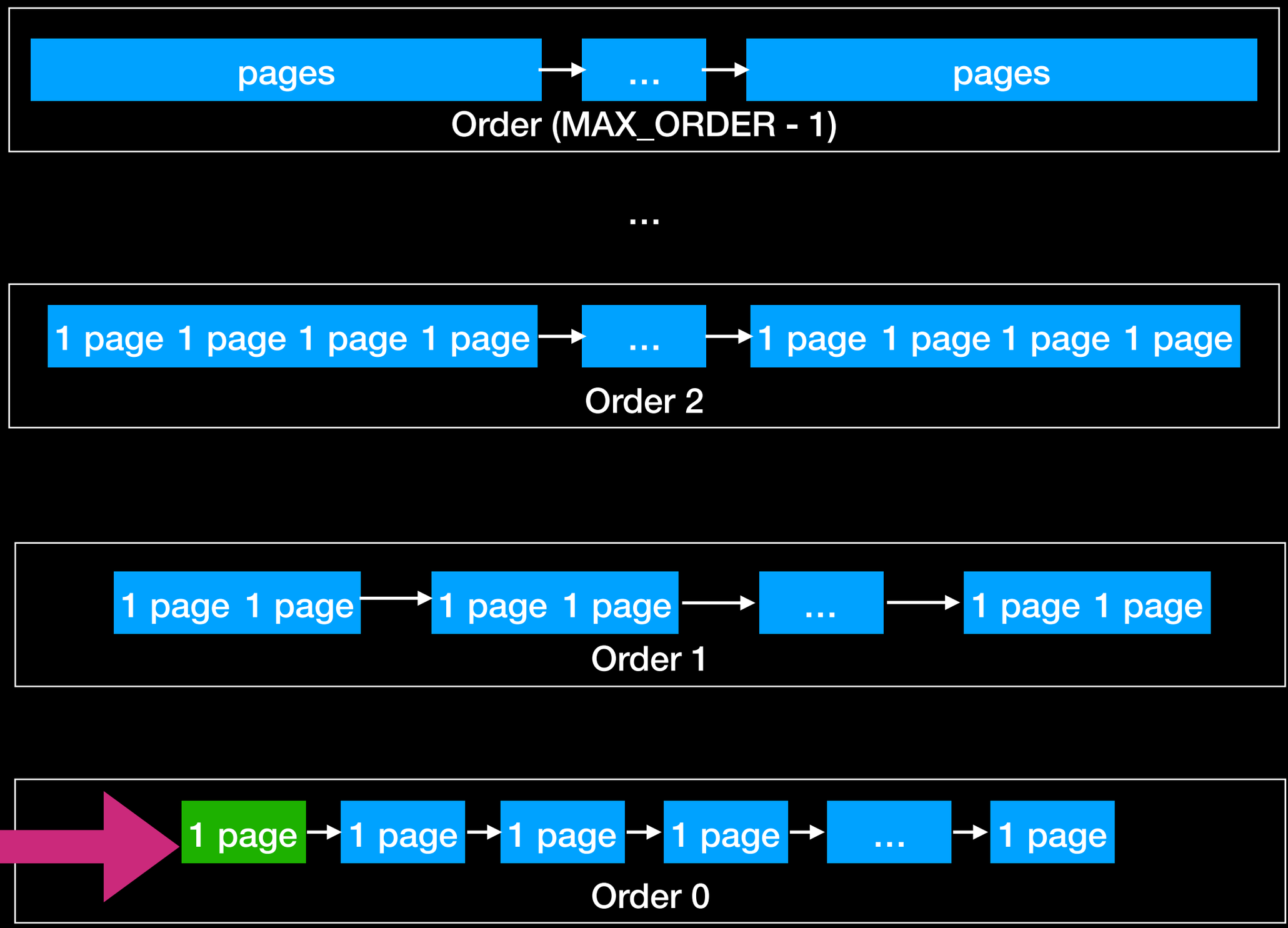
Buddy系统页面分配__rmqueue



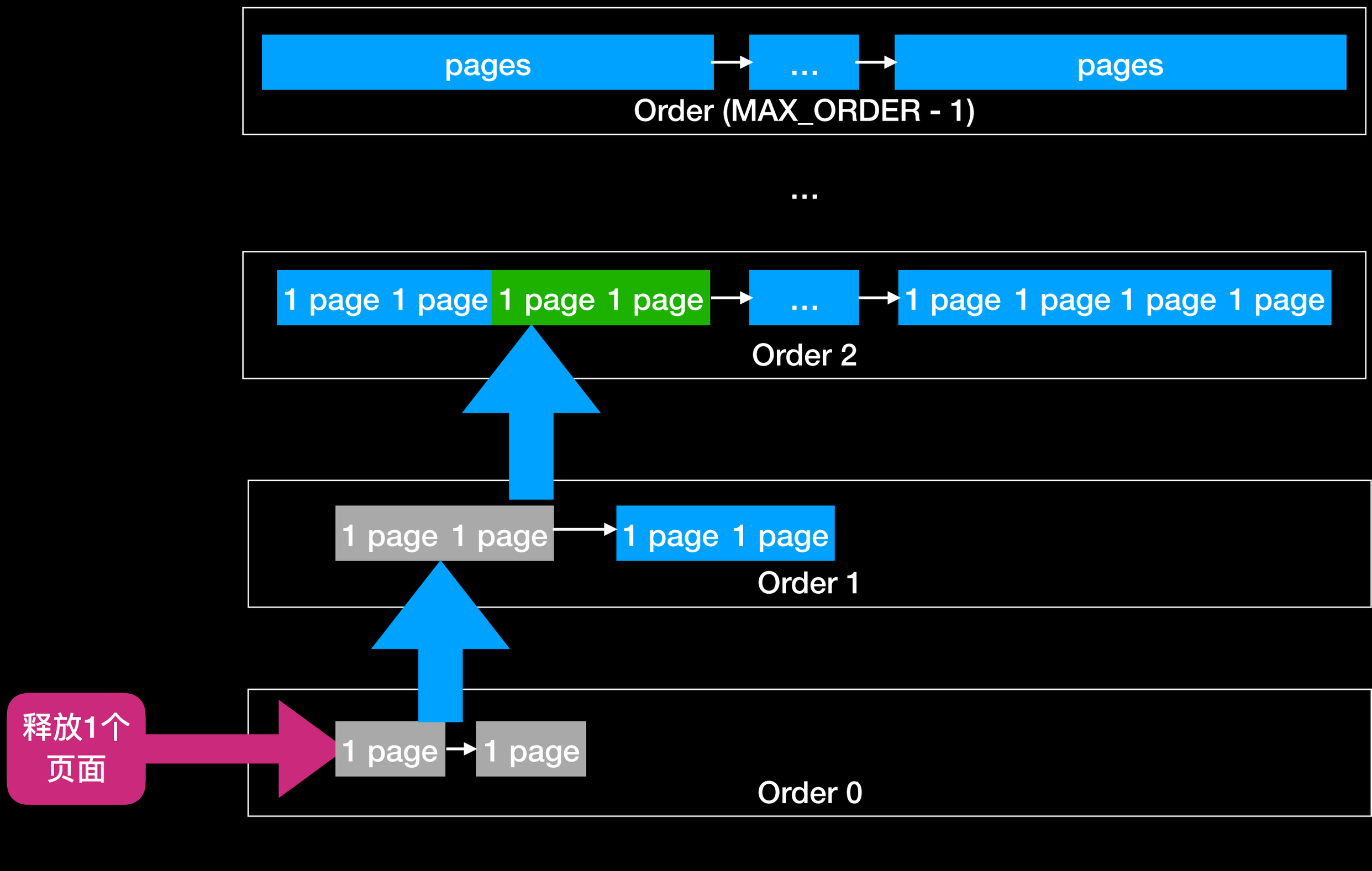
Buddy系统页面分配__rmqueue



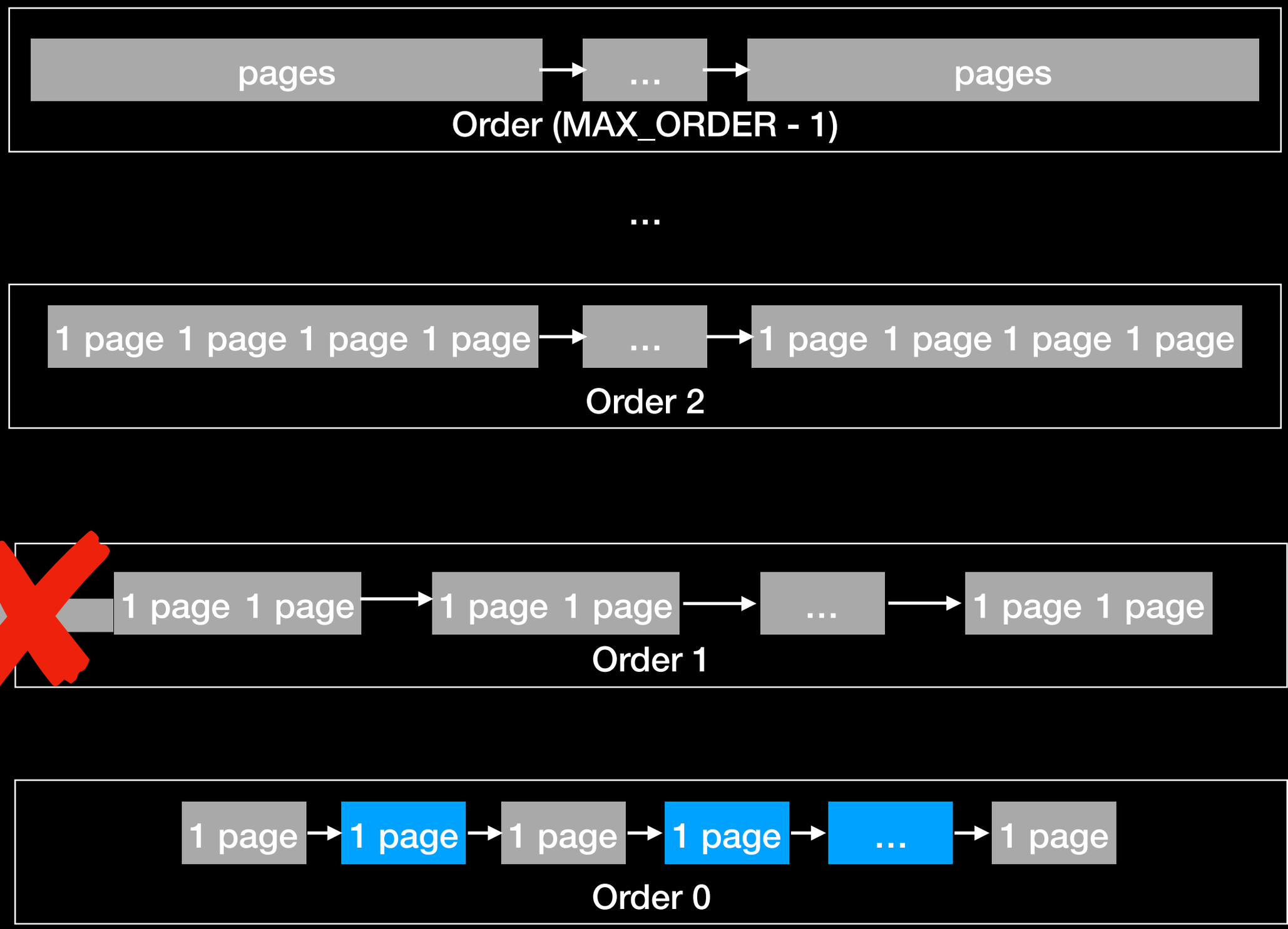
Buddy系统页面释放__free_one_page



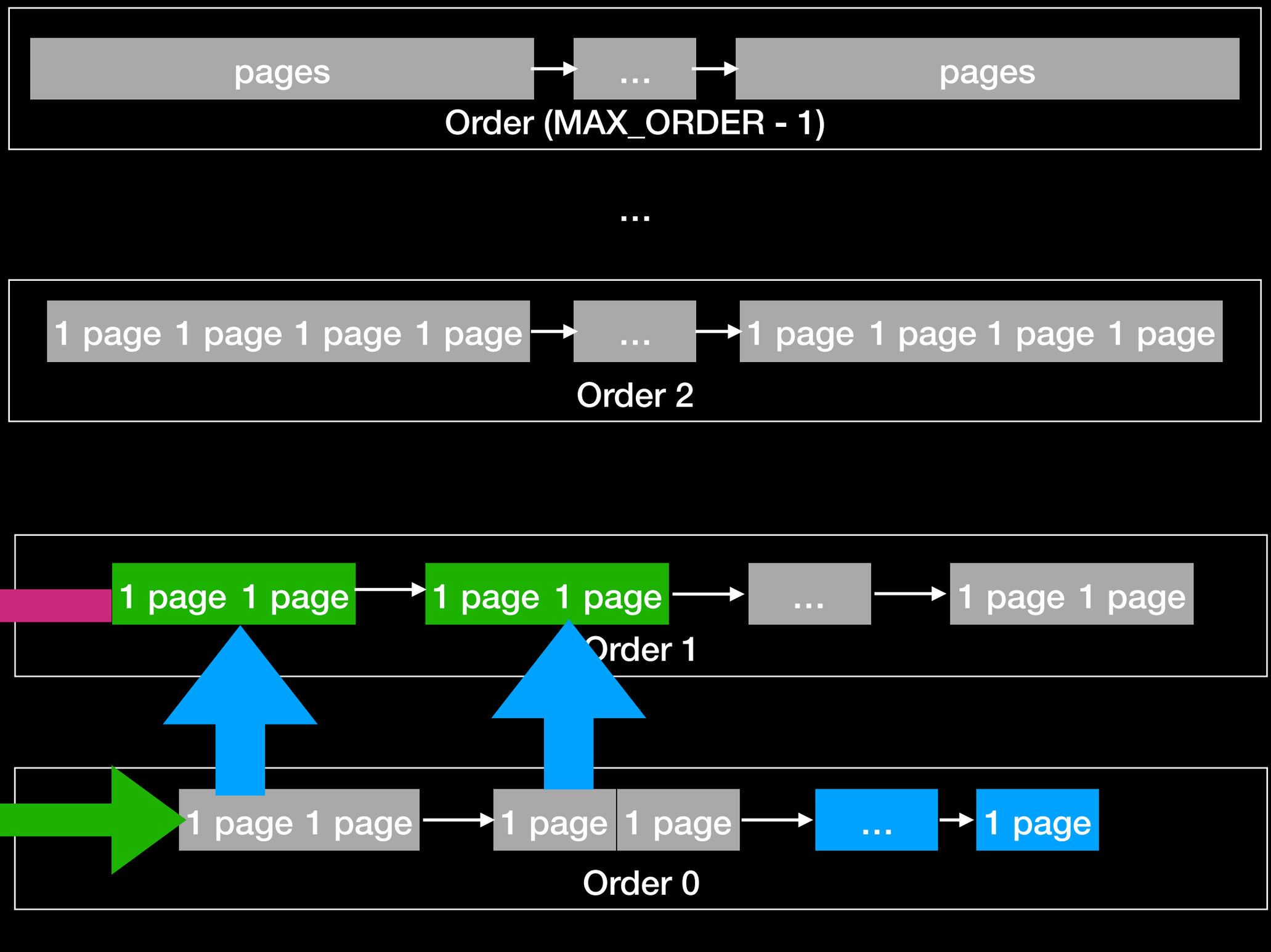
Buddy系统页面释放__free_one_page



碎页问题


pages
...
pages
Order (MAX_ORDER - 1)
...
1 page 1 page 1 page 1 page
...
1 page 1 page 1 page 1 page
Order 2
分配2个
页面
1 page 1 page
1 page 1 page
...
1 page 1 page
Order 1
1 page
1 page
1 page
1 page
...
1 page
Order 0

通过内存回收解决分配问题



一些措施

- 因为总体需求不强，碎页问题并不严重。
- 2007年左右开始比较关注碎页问题，因为hugepage用的多?
- 页面预留
- slub_max_order
- 用vmalloc换get_pages

内存回收lumpy reclaim

- 在页面回收isolate_lru_pages也就是选择要回收页面的时候，根据要分配的order，isolate一个页面成功后，将同buddy页面也isolate，最终回收，这样就更容易给系统增加某order的空闲页面。

- 只以位置为回收标准，而不是active和inactive这种更不容易影响性能的标准，对系统影响比较大。
且因为compaction功能的出现，2012年被去掉。

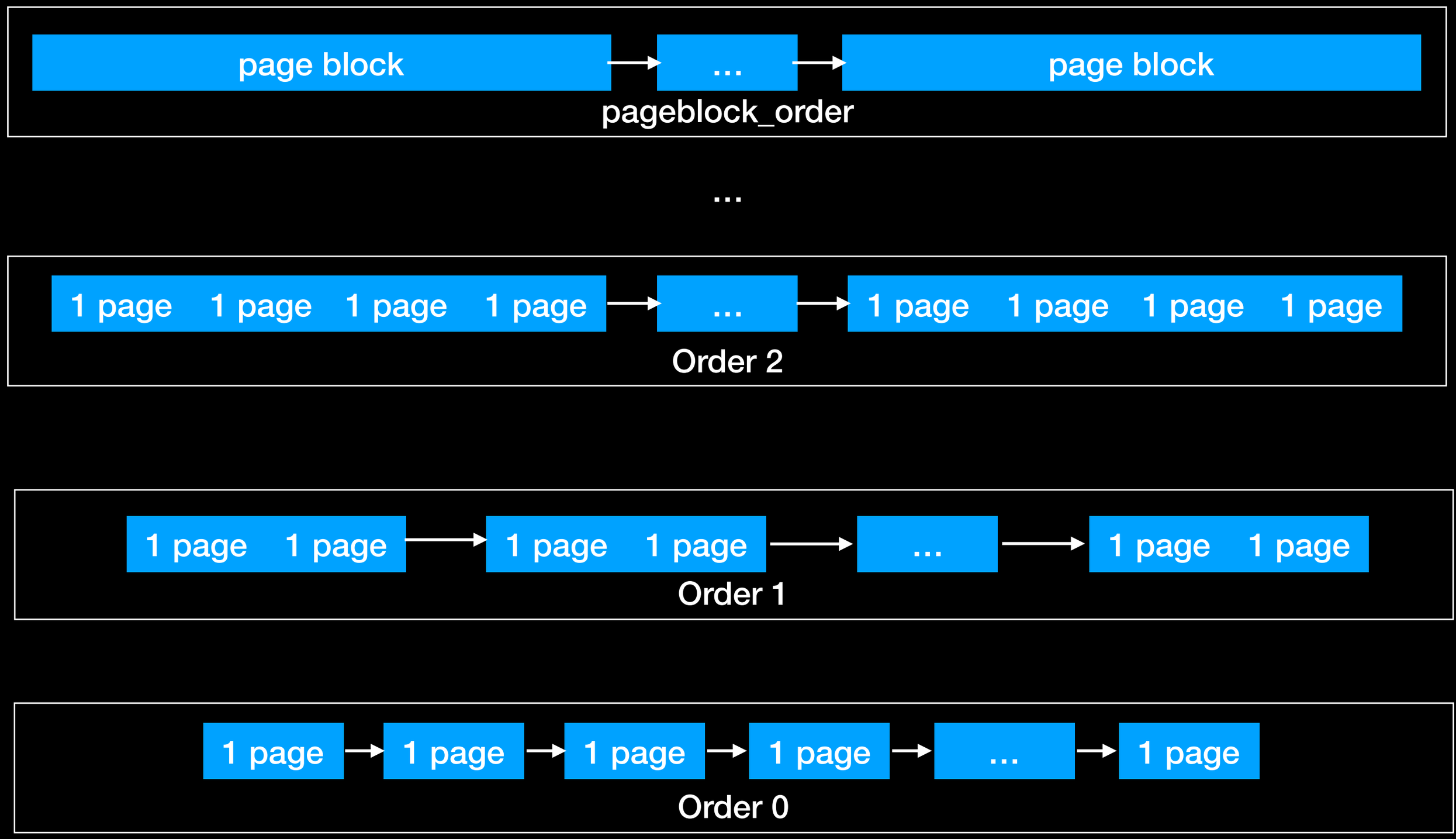
内存回收
PAGE_ALLOC_COSTLY_ORDER

- 一个宏，默认为3。
- 分配超过这个值的order的页面，分配中引起的内存回收就会做更重的操作。
- 给内存申请也提供了标准。
- 和lumpy reclaim同一个commit，commit里都没提，好像买东西随手给的小礼品，但是存活到了现在。

ZONE_MOVABLE

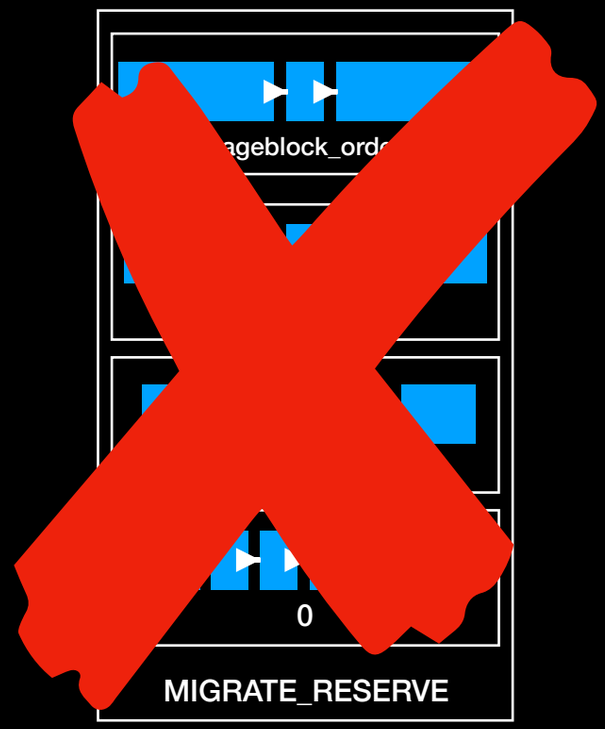
- 虚拟ZONE，在初始化的时候通过kernelcore或者movablecore指定大小。
- 只能被(__GFP_HIGHMEM | __GFP_MOVABLE)申请使用。
- commit上介绍的作用是分开可移动页面和不可移动页面，从而抗碎页。
- 不过感觉更主要的作用是因内存使用者都是100%可迁移的所以更倾向方便memory hotplug。
- 打开hugepages_treat_as_movable后hugepage可从zone_movable分配内存。
  此选项于2018年1月被删除，因为hugepage的不可移动性会影响ZONE_MOVABLE memory hotplug。

Migratetype
或称
CONFIG_PAGE_GROUP_BY_MOBILITY



...

MIGRATE_UNMOVABLE	MIGRATE_MOVABLE	MIGRATE_RECLAIMABLE	MIGRATE_HIGHATOMIC	MIGRATE_CMA	MIGRATE_ISOLATE
pageblock_order	pageblock_order	pageblock_order	pageblock_order	pageblock_order	pageblock_order
...	...	...	...	...	...
2	2	2	2	2	2
1	1	1	1	1	1
0	0	0	0	0	0



__rmqueue当某个migratetype列表中没有足够页面时__rmqueue_fallback

- static int fallbacks[MIGRATE_TYPES][4] = {
- [MIGRATE_UNMOVABLE] = { MIGRATE_RECLAIMABLE, MIGRATE_MOVABLE, MIGRATE_TYPES },
- [MIGRATE_RECLAIMABLE] = { MIGRATE_UNMOVABLE, MIGRATE_MOVABLE, MIGRATE_TYPES },
- [MIGRATE_MOVABLE] = { MIGRATE_RECLAIMABLE, MIGRATE_UNMOVABLE, MIGRATE_TYPES },
- #ifdef CONFIG_CMA
- [MIGRATE_CMA] = { MIGRATE_TYPES }, /* Never used */
- #endif
- #ifdef CONFIG_MEMORY_ISOLATION
- [MIGRATE_ISOLATE] = { MIGRATE_TYPES }, /* Never used */
- #endif
- };

Migratetype简介

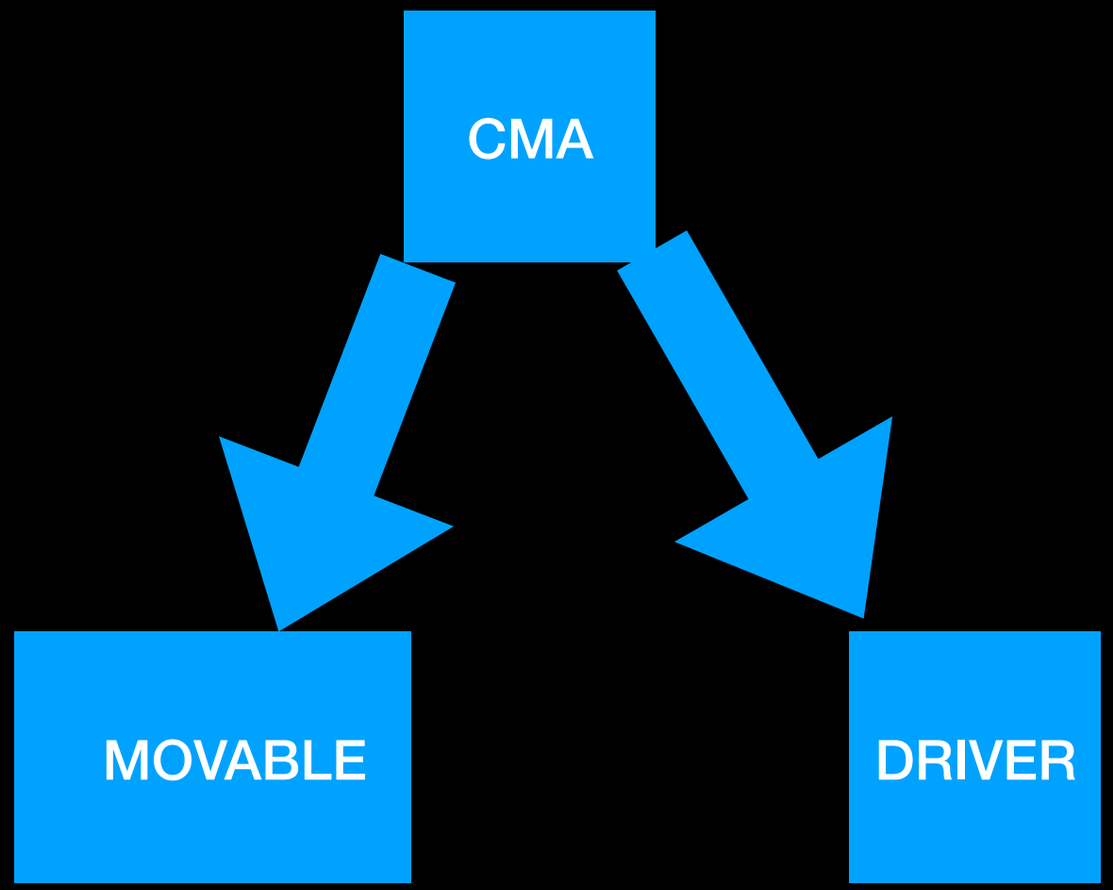
- MIGRATE_UNMOVABLE，不可migration的页面，一般是内核分配来自己用的页面。
- MIGRATE_MOVABLE，可migration的页面，一般是应用层使用，分配时候会指定__GFP_MOVABLE。
- MIGRATE_RECLAIMABLE，可回收的页面，主要是slab分配的时候指定了SLAB_RECLAIM_ACCOUNT的kmem，指定这个内存都自带shrink_slab接口，内存回收的时候可以被释放掉，比如inode。
- MIGRATE_ISOLATE，其中页面不会被分配，用来帮助isolate页面。isolate页面的时候会将页块先设置为isolate防止其被释放。
- ~~MIGRATE_RESERVE，把min_free_kbytes大小的内存存于特殊的group，作为全部fallback的备份。最终被MIGRATE_HIGHATOMIC反戈一击，被替换掉。~~
- 还有两个类型比较复杂，需要单独来讲。

MIGRATE_CMA

- 用来帮助嵌入式设备驱动分配某个特定位置和大块连续内存。基本原理可以见我2014年的演讲《Buddy和CMA简介》。

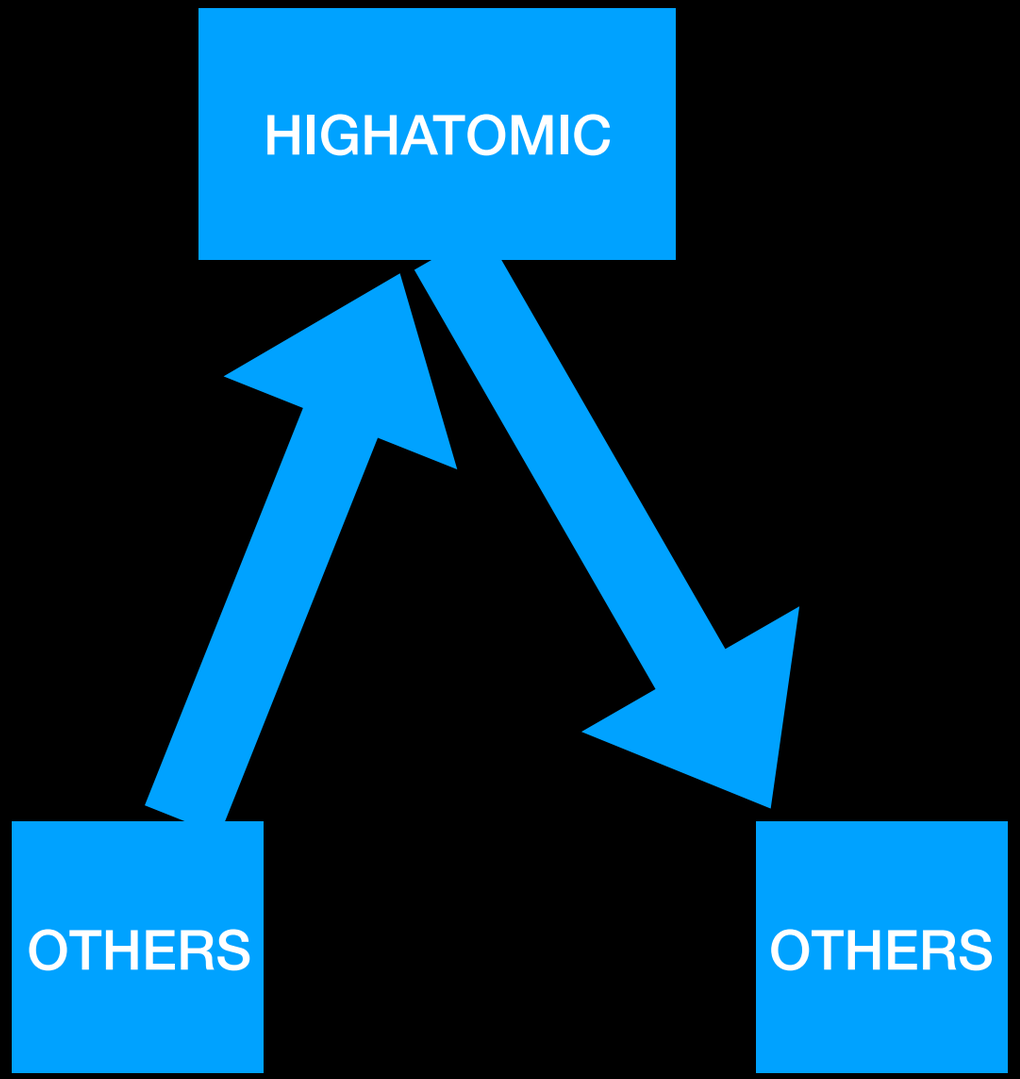
- 存在一堆坑人问题，用了很可能不如不用。

- 从2015年就开始喊要弄成类似ZONE_MOVABLE的虚拟ZONE（不能解决其大部分问题），现在还没提交。主要意义是有效阻止了修正问题PATCH的提交。

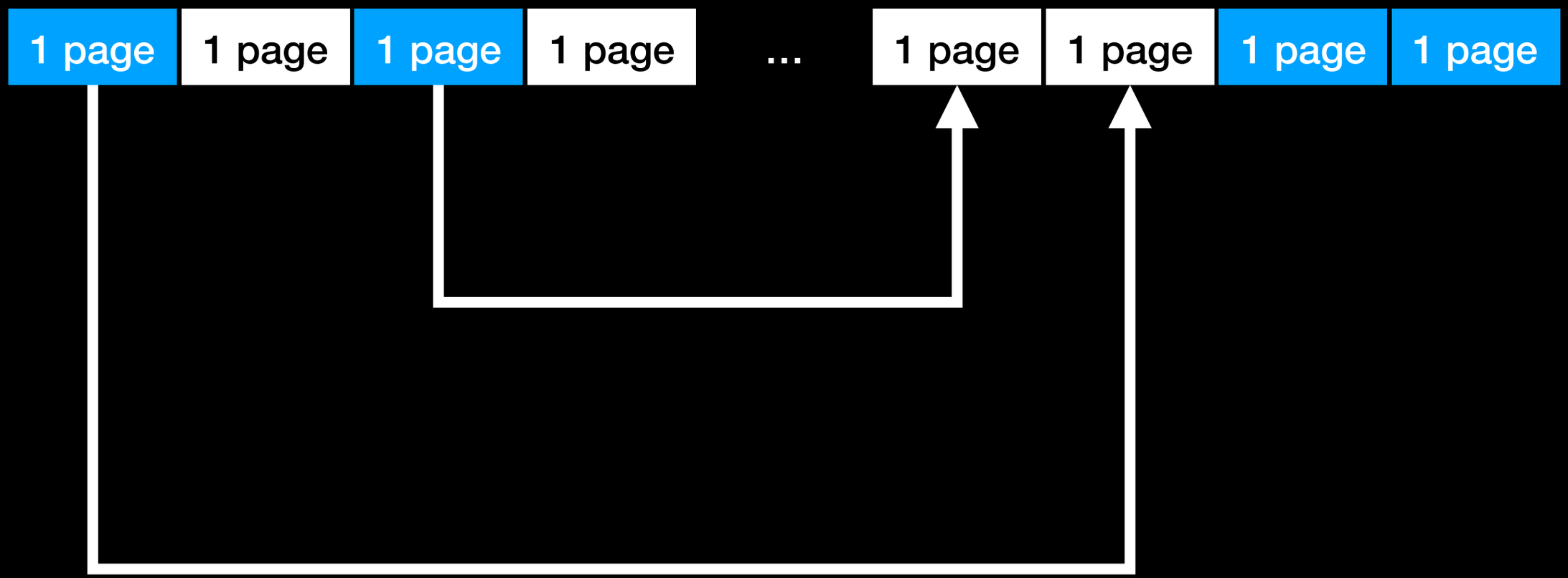


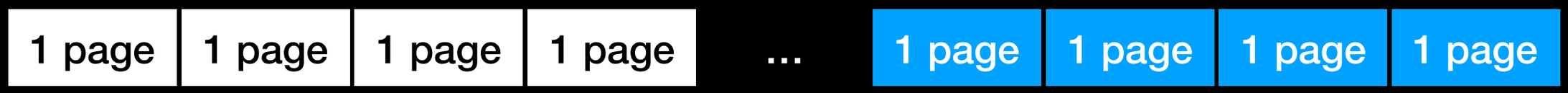
MIGRATE_HIGHATOMIC

- MIGRATE_HIGHATOMIC，最初版本存在过，存在于普通fallback列表，很快因为被MIGRATE_RESERVE取代。于2015年以更灵活形式回归。
- 当有原子连续页面分配时，自动优先从MIGRATE_HIGHATOMIC分配。
- 以前原子连续页面分配依赖MIGRATE_RESERVE和High-order watermark一起来保证，因为会引起普通连续页面watermark检查失败，所以被替换掉。
- MIGRATE_HIGHATOMIC按需reserve，同时拿掉High-order watermark保证了连续页面分配率。
- reserve_highatomic_pageblock
- unreserve_highatomic_pageblock
- https://lkml.org/lkml/2017/9/26/232



compaction碎页整理





compaction碎页整理的一些优化

- 在紧急情况下，来源和目标页块可更多的使用非MIGRATE_MOVABLE页块。

- 在紧急情况下，迁移（isolate_migratepages_block）失败后的处理变严格。（持怀疑态度，建议在用的可以拿掉测试一下）

- kcompacted

Kernel page movable

- 内存需求量大的内核内功能，会造成碎页。比如A64的安卓上的ZRAM的后端ZSMALLOC。

- 给这部分页面增加相应接口，令其可以migration。这些页面可以分配为MIGRATE_MOVABLE。

- ZSMALLOC，可以见我的演讲《ZRAM那点事》。
  77ff465799c60294e248000cd22ae8171da3304c
  另外ZSMALLOC因为migration成功率高，作为CMA后端也是不错的选择。

- F2FS

VMAP_STACK

- 打开此选项，alloc_thread_stack_node改alloc_pages_node为vmalloc，不再需要连续页面。
- 除了用来侦测内核栈溢出，其实也可以抗碎页。
- 在内存不足，且内核栈巨大的系统上，连续的内核栈分配既是碎页产生的原因，又是碎页问题的受害者。
- 而且VMAP_STACK backporting难度低，比之前介绍的难度都低几个等级。
- 缺点是因为内核栈变成vmalloc出来的，很多驱动会受到影响。

谢谢！ 问题?

顺便做个广告，我司招聘GO程序员。jobs@hyper.sh

我的微信公众号 茶水侃山