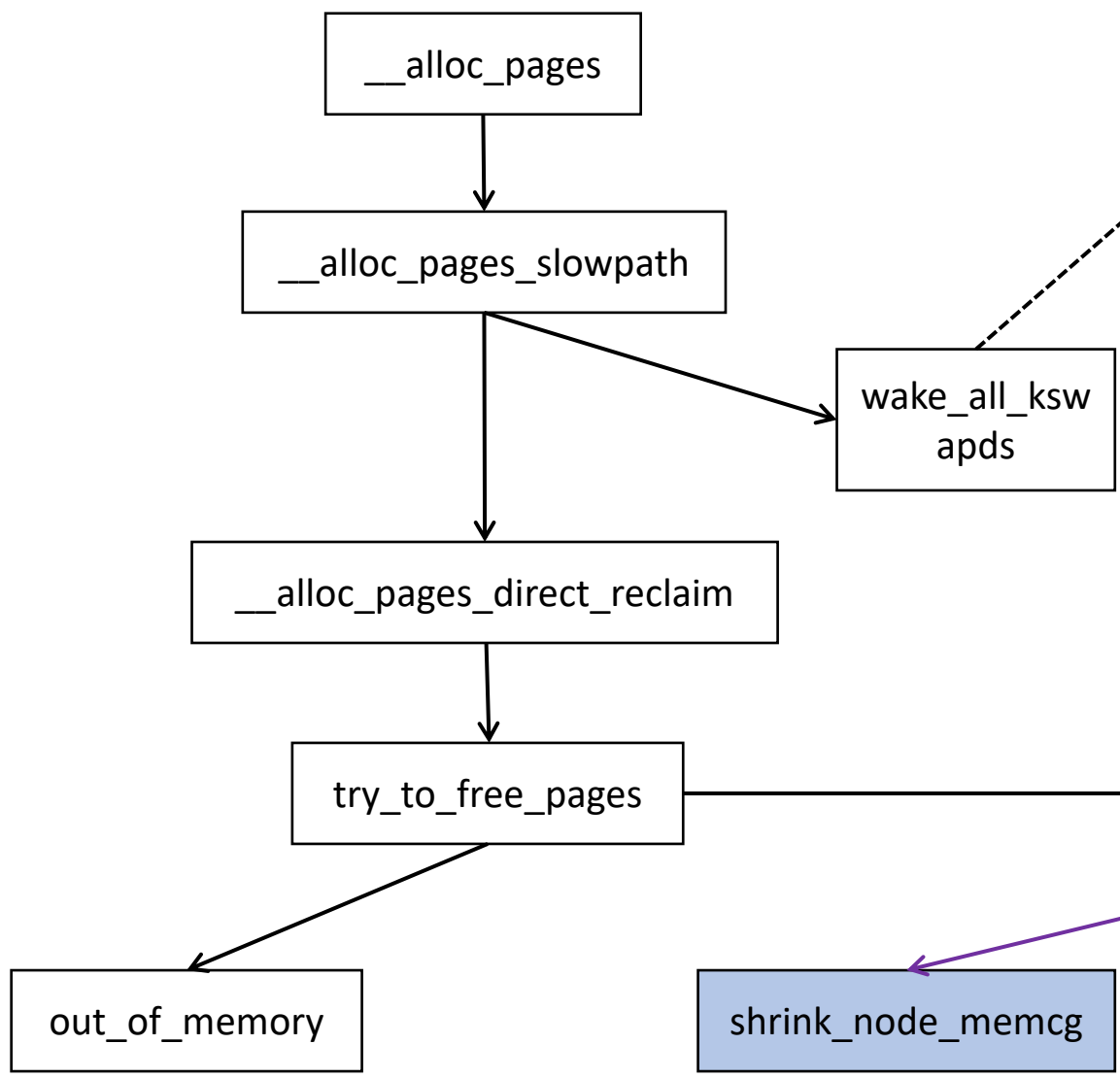
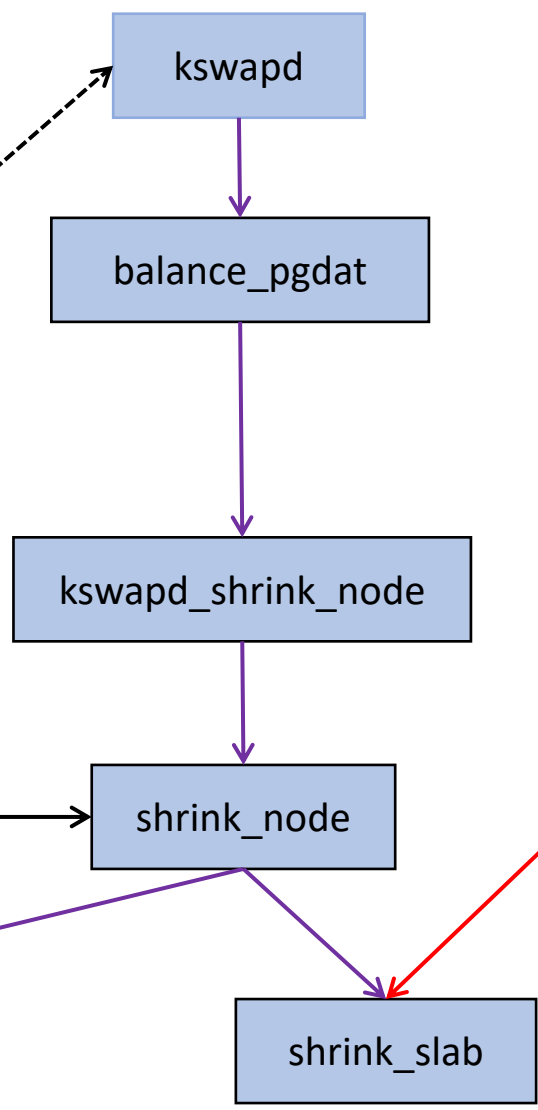


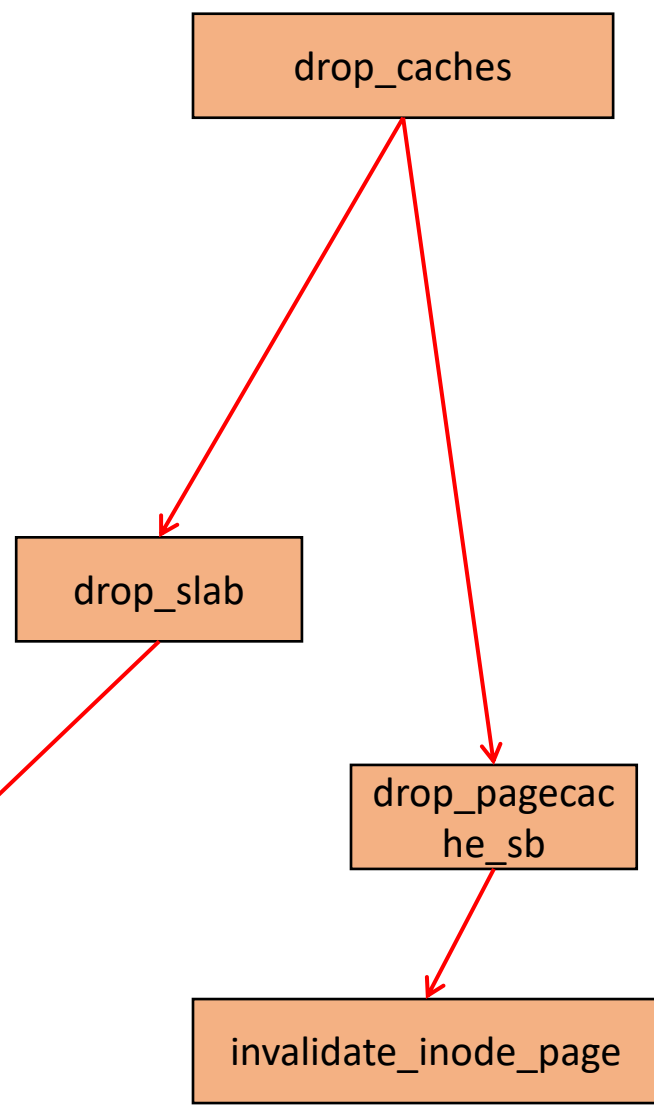
内存**紧缺**回收



内存**定期**回收



内存**手动**回收



# 内存回收原则

页面回收，并**不是**回收得越多越好，而是力求达到一种**balanced**。  
因为页面回收总是以cache丢弃、内存swap等为代价的，  
对系统性能会有一定程度的影响。  
而balanced，就是既要保证**性能**，又要应付好新来的页面**分配请求**

# 内存回收目标

## 内核空间

内核是所有进程公用的，内核中使用的页通常是伴随整个系统运行周期的，频繁的页换入和换出是非常影响性能的，所以内核中的这部分内存基本上**不能回收**，包括：

- 内核代码段
- 数据段
- 内核kmalloc()/vmalloc()出来的内存
- 内核线程占用的内存等。

内核中除上述之外的内存是**可以回收**的，包括：

- slab中的dcache、icache
- page cache
- mmap()文件时的有名映射所使用的物理内存等。

# 内存回收目标

## 用户空间

应用程序主动申请锁定的页（使用mlock api将页主动锁定），不允许它进行回收。  
用户空间除此之外的内存是可以回收的，包括：  
进程使用各种api（malloc,mmap,brk/sbrk）申请到的物理内存  
(这些api通常只是申请虚拟地址，真实的页分配发生在page fault中)，  
包括堆、栈，进程间通信中的共享内存，pipe，bss段，数据段，tmpfs的页等。

# 内存回收

成员	描述
ret_nr_dirty	统计脏页数量，即设置了PGDAT_DIRTY标志位的页面
ret_nr_unqueued_dirty	统计还没有开始回写的脏页以及还没有在块设备I/O上排队等待回写的页面数量，即设置了PGDAT_DIRTY标志位，但是还没有设置PGDAT_WRITEBACK标志位的页面，即PageDirty(page)&&! PageWriteback(page)
ret_nr_congested	表示这个页面正在块设备I/O上进行数据回写，我们统计这个页面是因为是一个阻塞源 1.脏页或正在回写的页面，这个页面有回写的存储设备，如匿名页面分配了交换空间 2.设置了PG_reclaim说明这个页面正在往交换分区或者文件写入
ret_nr_writeback	统计正在回写的页面数量
ret_nr_immediate	处理正在回写的页面时，发现已经有大量的页面在等待回写，因此表示需要立即特殊处理，该页面需要等待一段时间