第 0019 讲 5 内存管理 2 个调优参数

内存管理--2 个调优参数分析

根据 Linux 内核内存管理模块所提供的常用调优参数 ,所支持内存管理调优参数在/proc/sys/vm 目录，具体如下：

```
File  Edit  View  Search  Terminal  Help
vico@ubuntu:/proc/sys/vm$ ls
admin_reserve_kbytes          mmap_rnd_bits
block_dump                    mmap_rnd_compat_bits
compact_memory                nr_hugepages
compact_unevictable_allowed   nr_hugepages_mempolicy
dirty_background_bytes        nr_overcommit_hugepages
dirty_background_ratio        numa_stat
dirty_bytes                   numa_zonelist_order
dirty_expire_centisecs        oom_dump_tasks
dirty_ratio                   oom_kill_allocating_task
dirtytime_expire_seconds      overcommit_kbytes
dirty_writeback_centisecs     overcommit_memory
drop_caches                   overcommit_ratio
extfrag_threshold             page-cluster
hugetlb_shm_group             panic_on_oom
laptop_mode                   percpu_pagelist_fraction
legacy_va_layout              stat_interval
lowmem_reserve_ratio          stat_refresh
max_map_count                 swappiness
memory_failure_early_kill     unprivileged_userfaultfd
memory_failure_recovery       user_reserve_kbytes
min_free_kbytes               vfs_cache_pressure
min_slab_ratio                watermark_boost_factor
min_unmapped_ratio            watermark_scale_factor
mmap_min_addr                 zone_reclaim_mode
vico@ubuntu:/proc/sys/vm$
```

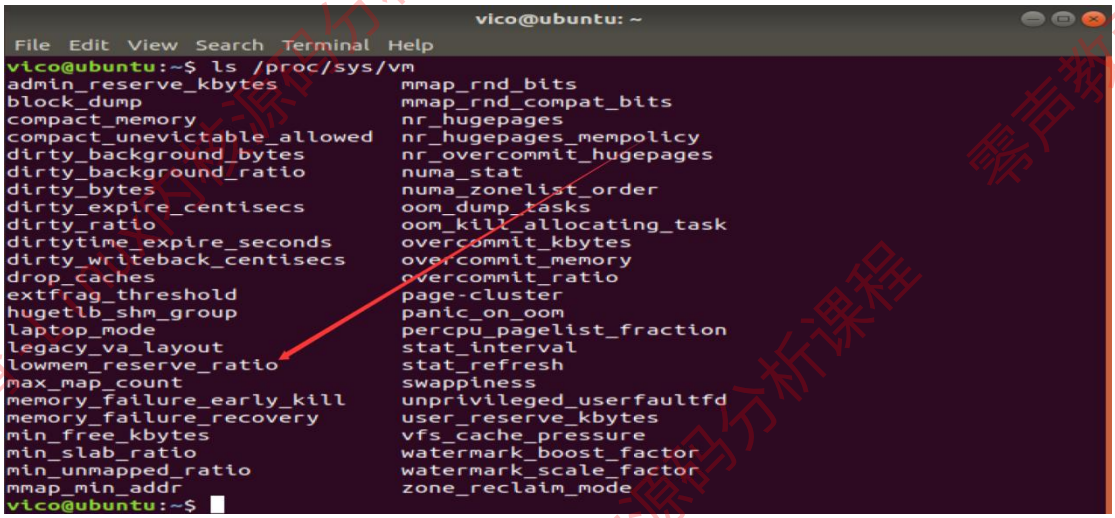
一、内存管理区水位调优参数 min_free_kbytes

每个 min_free_kbytes 的数值会影响内存管理区的水位操作，具体操作对应内核源码函数设计如下：

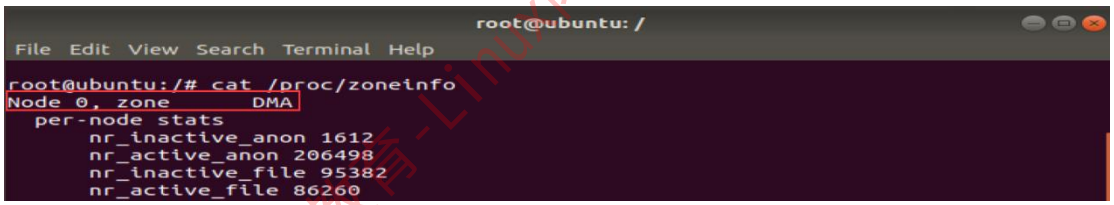
```
mm > C page_alloc.c > ⓜ _setup_per_zone_wmarks(void)
7473  static void __setup_per_zone_wmarks(void)
7474  {
7475      unsigned long pages_min = min_free_kbytes >> (PAGE_SHIFT - 10);
7476      unsigned long lowmem_pages = 0;
7477      struct zone *zone;
7478      unsigned long flags;
7479
7480      /* Calculate total number of !ZONE_HIGHMEM pages */
7481      for_each_zone(zone) {
7482          if (!is_highmem(zone))
7483              lowmem_pages += zone_managed_pages(zone);
7484      }
7485
```

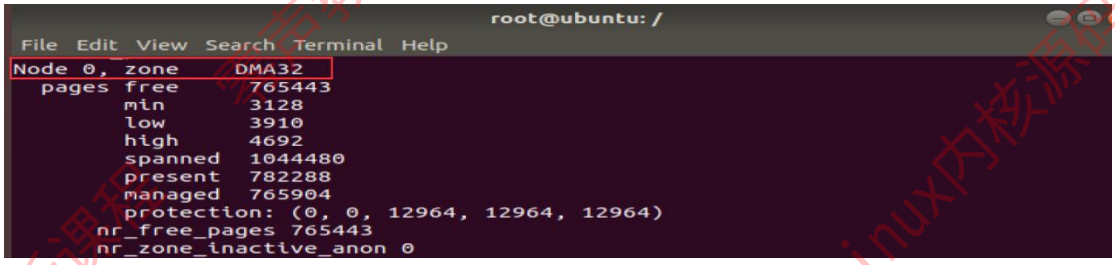
二、页面分配参数 lowmem_reserve_ratio

我们通过终端查看/proc/zoneinfo 节点来获得每个内存管理区中的核心参数（高水位、低水位等）。



具体操作对应内核源码函数设计如下：





```
Node 0, zone   Movable
  pages free     0
        min      0
        low      0
        high     0
        spanned  0
        present  0
        managed  0
        protection: (0, 0, 0, 0, 0)
Node 0, zone   Device
  pages free     0
        min      0
        low      0
        high     0
        spanned  0
        present  0
        managed  0
        protection: (0, 0, 0, 0, 0)
root@ubuntu:/#
```

通过对 lowmem_reserve[]数组的单位为页面 ,通过对它的设备来防止页面分配器过度从低端内存管理区域分配内存。

include > linux > C mmzone.h > zone

```
378     * recalculated at runtime if the sysctl_lowmem_reserve_ratio sysctl
379     * changes.
380     */
381    long lowmem_reserve[MAX_NR_ZONES];
382
```

影响页面回收参数：

- swappiness：主要用于控制 kswapd 内核线程把页面写入交换区的活跃程序，此参数值的设置为 0--100 ，此参数默认设置为 60。
- zone_reclaim_mode：当页面分配器在一个内存管理区分配失败，zone_reclaim_mode 为 0，表示可以从下一个内存管理区或者下一个内存节点分配内存；否则表示可以在这个内存管理区中进行回收内存。
- watermark_boost_factor：主要用于优化内存外碎片化。它临时提高内存管理区的水位，即 zone->watermark_boost。提高内存管理区的水平，kswapd 可以回收更多的内存。
- watermark_scale_factor：除刚才讲的 min_free_kbytes 以外，watermark_scale_factor 此参数也会影响每个内存管理区的低水位和高水位。通过__setup_per_zone_wmarks 函数将 watermark_scale_factor 参数值默认设置为 10，分母为 10000。watermark_scale_factor 参数最大值为 1000。

有时间大家可以查看一下官方文档（影响脏页回写的参数：dirty_background_ratio、dirty_bytes 等等）。

1、Linux内核在系统初始化是通过函数来计算min_free_kbytes值的大小，然后计算每个内存管理区的水位，具体计算公式如下：

$$min\_free\_kbytes=4\sqrt{lowmem\_kbytes}$$

lowmem_keytes此变量是系统中所有内存管理区的管理页面数量减少高水位页面数量的总和。最后计算出来的min_free_kbytes有范围孤帆，最小值为128KB，最大值为64MB。

mm > C page_alloc.c > _setup_per_zone_wmarks(void)

```
7473  static void __setup_per_zone_wmarks(void)
7474  {
7475      unsigned long pages_min = min_free_kbytes >> (PAGE_SHIFT - 10);
7476      unsigned long lowmem_pages = 0;
7477      struct zone *zone;
7478      unsigned long flags;
```

内存管理区的3个水位计算与min_free_kbytes有关。当系统只有一个内存管理区时，最低警戒水位等于min_free_kbytes，低水位、高水位与watermark_scale_factor参数、内存管理区管理的内存大小managed_pages有关。waterwark_boost表示临时提高的水位。