
Subject: [PATCH 5/6 mm] memcgroup: fix zone isolation OOM
Posted by [Hugh Dickins](#) on Fri, 09 Nov 2007 07:13:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

mem_cgroup_charge_common shows a tendency to OOM without good reason, when a memhog goes well beyond its rss limit but with plenty of swap available. Seen on x86 but not on PowerPC; seen when the next patch omits swapcache from memcgroup, but we presume it can happen without.

mem_cgroup_isolate_pages is not quite satisfying reclaim's criteria for OOM avoidance. Already it has to scan beyond the nr_to_scan limit when it finds a !LRU page or an active page when handling inactive or an inactive page when handling active. It needs to do exactly the same when it finds a page from the wrong zone (the x86 tests had two zones, the PowerPC tests had only one).

Don't increment scan and then decrement it in these cases, just move the incrementation down. Fix recent off-by-one when checking against nr_to_scan. Cut out "Check if the meta page went away from under us", presumably left over from early debugging: no amount of such checks could save us if this list really were being updated without locking.

This change does make the unlimited scan while holding two spinlocks even worse - bad for latency and bad for containment; but that's a separate issue which is better left to be fixed a little later.

Signed-off-by: Hugh Dickins <hugh@veritas.com>

Insert just after

bugfix-for-memory-cgroup-controller-avoid-pagelru-page-in-mem_cgroup_isolate_pages-fix.patch
or just before memory-cgroup-enhancements

mm/memcontrol.c | 17 +++++-----
1 file changed, 4 insertions(+), 13 deletions(-)

```
--- patch4/mm/memcontrol.c 2007-11-08 16:03:33.000000000 +0000
+++ patch5/mm/memcontrol.c 2007-11-08 16:51:39.000000000 +0000
@@ -260,24 +260,20 @@ unsigned long mem_cgroup_isolate_pages(u
    spin_lock(&mem_cont->lru_lock);
    scan = 0;
    list_for_each_entry_safe_reverse(pc, tmp, src, lru) {
-   if (scan++ > nr_to_scan)
+   if (scan >= nr_to_scan)
        break;
        page = pc->page;
        VM_BUG_ON(!pc);

-   if (unlikely(!PageLRU(page))) {
```

```

- scan--;
+ if (unlikely(!PageLRU(page)))
    continue;
- }

    if (PageActive(page) && !active) {
        __mem_cgroup_move_lists(pc, true);
- scan--;
    continue;
    }
    if (!PageActive(page) && active) {
        __mem_cgroup_move_lists(pc, false);
- scan--;
    continue;
    }

@@ -288,13 +284,8 @@ unsigned long mem_cgroup_isolate_pages(u
    if (page_zone(page) != z)
        continue;

- /*
-  * Check if the meta page went away from under us
-  */
- if (!list_empty(&pc->lru))
-     list_move(&pc->lru, &pc_list);
- else
-     continue;
+ scan++;
+ list_move(&pc->lru, &pc_list);

    if (__isolate_lru_page(page, mode) == 0) {
        list_move(&page->lru, dst);

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
