

---

Subject: [PATCH][BUGFIX][for -mm] Misc fix for memory cgroup [1/5] -- fix refcnt check in mem\_cgroup\_charge

Posted by [KAMEZAWA Hiroyuki](#) on Thu, 11 Oct 2007 04:55:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

The logic of uncharge is

- decrement refcnt -> lock page cgroup -> remove page cgroup.

But the logic of charging is

- lock page cgroup -> increment refcnt -> return.

Then, one charge will be added to a page\_cgroup under being removed.

This makes no big trouble (like panic) but one charge is lost.

This patch add a test at charging to verify page\_cgroup's refcnt is greater than 0. If not, unlock and retry.

Changelog v1->v2:

- \* added cpu\_relax() before retry.

Signed-off-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>

mm/memcontrol.c | 10 ++++++--

1 file changed, 8 insertions(+), 2 deletions(-)

Index: devel-2.6.23-rc8-mm2/mm/memcontrol.c

=====

--- devel-2.6.23-rc8-mm2.orig/mm/memcontrol.c

+++ devel-2.6.23-rc8-mm2/mm/memcontrol.c

@ @ -271,14 +271,20 @ @ int mem\_cgroup\_charge(struct page \*page,

    \* to see if the cgroup page already has a page\_cgroup associated

    \* with it

    \*/

+retry:

    lock\_page\_cgroup(page);

    pc = page\_get\_page\_cgroup(page);

    /\*

    \* The page\_cgroup exists and the page has already been accounted

    \*/

    if (pc) {

- atomic\_inc(&pc->ref\_cnt);

- goto done;

+ if (unlikely(!atomic\_inc\_not\_zero(&pc->ref\_cnt))) {

+ /\* this page is under being uncharged ? \*/

+ unlock\_page\_cgroup(page);

+ cpu\_relax();

+ goto retry;

+ } else

```
+ goto done;  
}  
  
unlock_page_cgroup(page);
```

---

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

---