

[RFC][PATCH -mm 6/7] memcg: make-mapping-null-before-calling-uncharge.patch

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-08/msg08815.html>

- *From:* KAMEZAWA Hiroyuki <kamezawa.hiroyu@xxxxxxxxxxxxxxxx>
 - *Date:* Wed, 20 Aug 2008 19:07:02 +0900
-

This patch tries to make page->mapping to be NULL before mem_cgroup_uncharge_cache_page() is called.

"page->mapping == NULL" is a good check for "whether the page is still radix-tree or not".

This patch also adds VM_BUG_ON() to mem_cgroup_uncharge_cache_page();

Signed-off-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@xxxxxxxxxxxxxxxx>

```

---
mm/filemap.c | 2 +-
mm/memcontrol.c | 1 +
mm/migrate.c | 11 ++++++++--
3 files changed, 11 insertions(+), 3 deletions(-)

```

Index: mmtom-2.6.27-rc3+/mm/filemap.c

```

=====
--- mmtom-2.6.27-rc3+.orig/mm/filemap.c
+++ mmtom-2.6.27-rc3+/mm/filemap.c
@@ -116,12 +116,12 @@ void __remove_from_page_cache(struct pag
{
struct address_space *mapping = page->mapping;

- mem_cgroup_uncharge_cache_page(page);
radix_tree_delete(&mapping->page_tree, page->index);
page->mapping = NULL;
mapping->npages--;
__dec_zone_page_state(page, NR_FILE_PAGES);
BUG_ON(page_mapped(page));
+ mem_cgroup_uncharge_cache_page(page);

```

```

/*
* Some filesystems seem to re-dirty the page even after

```

Index: mmtom-2.6.27-rc3+/mm/memcontrol.c

```

=====
--- mmtom-2.6.27-rc3+.orig/mm/memcontrol.c

```

[RFC][PATCH -mm 6/7] memcg: make-mapping-null-before-calling-uncharge.patch

```
+++ mmtom-2.6.27-rc3+/mm/memcontrol.c
@@ -854,6 +854,7 @@ void mem_cgroup_uncharge_page(struct pag
void mem_cgroup_uncharge_cache_page(struct page *page)
{
VM_BUG_ON(page_mapped(page));
+ VM_BUG_ON(page->mapping);
__mem_cgroup_uncharge_common(page, MEM_CGROUP_CHARGE_TYPE_CACHE);
}
```

Index: mmtom-2.6.27-rc3+/mm/migrate.c

```
----- mmtom-2.6.27-rc3+.orig/mm/migrate.c
+++ mmtom-2.6.27-rc3+/mm/migrate.c
@@ -330,8 +330,6 @@ static int migrate_page_move_mapping(str
__inc_zone_page_state(newpage, NR_FILE_PAGES);

spin_unlock_irq(&mapping->tree_lock);
- if (!PageSwapCache(newpage))
- mem_cgroup_uncharge_cache_page(page);

return 0;
}
@@ -379,6 +377,15 @@ static void migrate_page_copy(struct pag
ClearPagePrivate(page);
set_page_private(page, 0);
page->mapping = NULL;
+ /* page->mapping contains a flag for PageAnon() */
+ if (PageAnon(page)) {
+ /* This page is uncharged at try_to_unmap(). */
+ page->mapping = NULL;
+ } else {
+ /* Obsolete file cache should be uncharged */
+ page->mapping = NULL;
+ mem_cgroup_uncharge_cache_page(page);
+ }
```

```
/*
* If any waiters have accumulated on the new page then
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>