

---

Subject: [RFC] [-mm PATCH] Memory controller fix swap charging context in  
unuse\_pte()

Posted by [Balbir Singh](#) on Fri, 05 Oct 2007 04:14:06 GMT

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

---

Found-by: Hugh Dickins <[hugh@veritas.com](mailto:hugh@veritas.com)>

mem\_cgroup\_charge() in unuse\_pte() is called under a lock, the pte\_lock. That's clearly incorrect, since we pass GFP\_KERNEL to mem\_cgroup\_charge() for allocation of page\_cgroup.

This patch release the lock and reacquires the lock after the call to mem\_cgroup\_charge().

Tested on a powerpc box by calling swapon in the middle of a cgroup running a workload that pushes pages to swap.

Signed-off-by: Balbir Singh <[balbir@linux.vnet.ibm.com](mailto:balbir@linux.vnet.ibm.com)>

---

mm/swapfile.c | 16 ++++++++-----  
1 file changed, 12 insertions(+), 4 deletions(-)

diff -puN mm/swapfile.c~memory-controller-fix-unuse-pte-charging mm/swapfile.c  
--- linux-2.6.23-rc8/mm/swapfile.c~memory-controller-fix-unuse-pte-charging 2007-10-03  
13:45:56.000000000 +0530

+++ linux-2.6.23-rc8-balbir/mm/swapfile.c 2007-10-05 08:49:54.000000000 +0530

@@ -507,11 +507,18 @@ unsigned int count\_swap\_pages(int type,  
\* just let do\_wp\_page work it out if a write is requested later - to  
\* force COW, vm\_page\_prot omits write permission from any private vma.  
\*/

```
-static int unuse_pte(struct vm_area_struct *vma, pte_t *pte,  
- unsigned long addr, swp_entry_t entry, struct page *page)  
+static int unuse_pte(struct vm_area_struct *vma, pte_t *pte, pmd_t *pmd,  
+ unsigned long addr, swp_entry_t entry, struct page *page,  
+ spinlock_t **ptl)  
{  
- if (mem_cgroup_charge(page, vma->vm_mm, GFP_KERNEL))  
+ pte_unmap_unlock(pte - 1, *ptl);  
+  
+ if (mem_cgroup_charge(page, vma->vm_mm, GFP_KERNEL)) {  
+ pte_offset_map_lock(vma->vm_mm, pmd, addr, ptl);  
+ return -ENOMEM;  
+ }  
+  
+ pte_offset_map_lock(vma->vm_mm, pmd, addr, ptl);
```

```
inc_mm_counter(vma->vm_mm, anon_rss);
```

```

get_page(page);
@@ -543,7 +550,8 @@ static int unuse_pte_range(struct vm_area
 * Test inline before going to call unuse_pte.
 */
if (unlikely(pte_same(*pte, swp_pte))) {
- ret = unuse_pte(vma, pte++, addr, entry, page);
+ ret = unuse_pte(vma, pte++, pmd, addr, entry, page,
+   &ptl);
  break;
}
} while (pte++, addr += PAGE_SIZE, addr != end);

```

—

--

Warm Regards,  
Balbir Singh  
Linux Technology Center  
IBM, ISTL

---

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

---