Subject: [RFC][PATCH] memory cgroup enhancements updated [3/10] remember pagecache
Posted by KAMEZAWA Hiroyuki on Fri, 19 Oct 2007 09:31:06 GMT
View Forum Message <> Reply to Message

Add PCGF_PAGECACHE flag to page_cgroup to remember "this page is
charged as page-cache."
This is very useful for implementing precise accounting in memory cgroup.

Changelog v1 -> v2
 - moved #define to out-side of struct definition

Signed-off-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>
Signed-off-by: YAMAMOTO Takashi <yamamoto@valinux.co.jp>

 mm/memcontrol.c | 18 ++++++++++++++---
 1 file changed, 15 insertions(+), 3 deletions(-)

Index: devel-2.6.23-mm1/mm/memcontrol.c
===
--- devel-2.6.23-mm1.orig/mm/memcontrol.c
+++ devel-2.6.23-mm1/mm/memcontrol.c
@@ -83,7 +83,9 @@ struct page_cgroup {
 struct mem_cgroup *mem_cgroup;
 atomic_t ref_cnt; /* Helpful when pages move b/w */
     /* mapped and cached states */
+ int flags;
 };
+#define PCGF_PAGECACHE (0x1) /* charged as page-cache */

 enum {
 MEM_CGROUP_TYPE_UNSPEC = 0,
@@ -315,8 +317,8 @@ unsigned long mem_cgroup_isolate_pages(u
 * 0 if the charge was successful
 * < 0 if the cgroup is over its limit
 */
-int mem_cgroup_charge(struct page *page, struct mm_struct *mm,
-   gfp_t gfp_mask)
+static int mem_cgroup_charge_common(struct page *page, struct mm_struct *mm,
+   gfp_t gfp_mask, int is_cache)
 {
 struct mem_cgroup *mem;
 struct page_cgroup *pc;
@@ -418,6 +420,10 @@ noreclaim:
 atomic_set(&pc->ref_cnt, 1);
 pc->mem_cgroup = mem;
 pc->page = page;
+ if (is_cache)

```
+  pc->flags = PCGF_PAGECACHE;
+ else
+  pc->flags = 0;
  if (page_cgroup_assign_new_page_cgroup(page, pc)) {
   /*
    * an another charge is added to this page already.
@@ -442,6 +448,12 @@ err:
  return -ENOMEM;
 }

+int mem_cgroup_charge(struct page *page, struct mm_struct *mm,
+   gfp_t gfp_mask)
+{
+ return mem_cgroup_charge_common(page, mm, gfp_mask, 0);
+}
+
 /*
  * See if the cached pages should be charged at all?
  */
@@ -454,7 +466,7 @@ int mem_cgroup_cache_charge(struct page

  mem = rcu_dereference(mm->mem_cgroup);
  if (mem->control_type == MEM_CGROUP_TYPE_ALL)
-  return mem_cgroup_charge(page, mm, gfp_mask);
+  return mem_cgroup_charge_common(page, mm, gfp_mask, 1);
  else
   return 0;
 }
```

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