

---

Subject: Re: [RFC][PATCH] memory cgroup enhancements updated [3/10]  
remember pagecache

Posted by [Balbir Singh](#) on Wed, 24 Oct 2007 13:56:34 GMT

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

---

KAMEZAWA Hiroyuki wrote:

> 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;
```

I prefer PAGE\_CGROUP\_CACHE since cache can be page cache/swap cache.

```
> + 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);
```

Could we define

```
enum {
MEM_CGROUP_CHARGE_TYPE_CACHE = 0,
MEM_CGROUP_CHARGE_TYPE_MAPPED = 1,
};
```

and use the enums here and below.

```
> +}
> +
> /*
>  * 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;
> }
>
```

--

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>

---