Subject: Re: [PATCH 3/4] swapcgroup: implement charge/uncharge
Posted by Daisuke Nishimura on Fri, 23 May 2008 11:52:29 GMT
View Forum Message <> Reply to Message

On 2008/05/22 16:37 +0900, KAMEZAWA Hiroyuki wrote:
> On Thu, 22 May 2008 15:20:05 +0900
> Daisuke Nishimura <nishimura@mxp.nes.nec.co.jp> wrote:
>

```
>> +#ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> +int swap_cgroup_charge(struct page *page,
>> +   struct swap_info_struct *si,
>> +   unsigned long offset)
>> +{
>> + int ret;
>> + struct page_cgroup *pc;
>> + struct mem_cgroup *mem;
>> +
>> + lock_page_cgroup(page);
>> + pc = page_get_page_cgroup(page);
>> + if (unlikely(!pc))
>> +  mem = &init_mem_cgroup;
>> + else
>> +  mem = pc->mem_cgroup;
>> + unlock_page_cgroup(page);
>
```

> If !pc, the page is used before memory controller is available. But is it
> good to be charged to init_mem_cgroup() ?
I'm sorry, but I can't understand this situation.
memory controller is initialized at kernel initialization,
so aren't processes created after it is initialized?

> How about returning 'failure' in this case ? I think returning 'failure' here
> is not so bad.
>
>
Which of below do you mean by 'failure'?

A. make it fail to get swap entry, so the page cannot be swapped out.
B. don't charge this swap entry to any cgroup, but the page
   would be swapped out.

I don't want to do B, because I don't want to make such
not-charged-to-anywhere entries.
And I've seen several times this condition(!pc) becomes true,
so I charged to init_mem_cgroup.

BTW, I noticed that almost all of functions I added by this patch set

should check "mem_cgroup_subsys.disabled" first because it depend on
memory cgroup.

>> +
>> + css_get(&mem->css);
>
> move this css_get() before unlock_page_cgroup() is safer.
>
OK, thanks.

Daisuke Nishimura.

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