
Subject: Re: [PATCH 2/4] swapcgroup: add member to swap_info_struct for cgroup
Posted by [Daisuke Nishimura](#) on Thu, 22 May 2008 08:46:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi.

On 2008/05/22 16:23 +0900, KAMEZAWA Hiroyuki wrote:

> On Thu, 22 May 2008 15:18:51 +0900

> Daisuke Nishimura <nishimura@mxp.nes.nec.co.jp> wrote:

>

>> This patch add a member to swap_info_struct for cgroup.

>>

>> This member, array of pointers to mem_cgroup, is used to
>> remember to which cgroup each swap entries are charged.

>>

>> The memory for this array of pointers is allocated on swapon,
>> and freed on swapoff.

>>

> Hi, in general, #ifdefs in the middle of functions are not good style.

> I'd like to comment some hints.

>

I completely agree that it's not good style.

>> Signed-off-by: Daisuke Nishimura <nishimura@mxp.nes.nec.co.jp>

>>

>> ---

>> include/linux/swap.h | 3 +++

>> mm/swapfile.c | 32 ++++++

>> 2 files changed, 35 insertions(+), 0 deletions(-)

>>

>> diff --git a/include/linux/swap.h b/include/linux/swap.h

>> index de40f16..67de27b 100644

>> --- a/include/linux/swap.h

>> +++ b/include/linux/swap.h

>> @@ -141,6 +141,9 @@ struct swap_info_struct {

>> struct swap_extent *curr_swap_extent;

>> unsigned old_block_size;

>> unsigned short * swap_map;

>> + #ifdef CONFIG_CGROUP_SWAP_RES_CTLR

>> + struct mem_cgroup **memcg;

>> + #endif

>> unsigned int lowest_bit;

>> unsigned int highest_bit;

>> unsigned int cluster_next;

>> diff --git a/mm/swapfile.c b/mm/swapfile.c

>> index d3caf3a..232bf20 100644

>> --- a/mm/swapfile.c

>> +++ b/mm/swapfile.c

```
>> @@ -1207,6 +1207,9 @@ asmlinkage long sys_swapoff(const char __user * specialfile)
>> {
>> struct swap_info_struct * p = NULL;
>> unsigned short *swap_map;
>> + #ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> + struct mem_cgroup **memcg;
>> + #endif
> Remove #ifdef.
>
> struct mem_cgroup **memcg = NULL;
>
good idea.
I'll do it.
```

```
>> struct file *swap_file, *victim;
>> struct address_space *mapping;
>> struct inode *inode;
>> @@ -1309,10 +1312,17 @@ asmlinkage long sys_swapoff(const char __user * specialfile)
>> p->max = 0;
>> swap_map = p->swap_map;
>> p->swap_map = NULL;
>> + #ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> + memcg = p->memcg;
>> + p->memcg = NULL;
>> + #endif
>
>
> ==
> #ifdef CONFIG_CGROUP_SWAP_RES_CTR
> void swap_cgroup_init_memcg(p, memcg)
> {
>     do something.
> }
> #else
> void swap_cgroup_init_memcg(p, memcg)
> {
> }
> #endif
> ==
>
```

I think swap_cgroup_init_memcg should return old value of p->memcg, and I would like to name it swap_cgroup_clear_memcg, because it is called by sys_swapoff, so "clear" rather than "init" would be better.

How about something like this?

```
struct mem_cgroup **swap_cgroup_clear_memcg(p, memcg)
```

```

{
    struct mem_cgroup **mem;

    mem = p->memcg;
    p->memcg = NULL;

    return mem;
}

```

and at sys_swapoff():

```

struct mem_cgroup **memcg;
:
memcg = swap_cgroup_clear_memcg(p, memcg);
:
if (memcg)
    vfree(memcg);

>> p->flags = 0;
>> spin_unlock(&swap_lock);
>> mutex_unlock(&swapon_mutex);
>> vfree(swap_map);
>> +ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> + vfree(memcg);
>> +endif
> if (memcg)
>     vfree(memcg);
>
>

```

will do.

```

>> inode = mapping->host;
>> if (S_ISBLK(inode->i_mode)) {
>>     struct block_device *bdev = I_BDEV(inode);
>> @@ -1456,6 +1466,9 @@ asmlinkage long sys_swapon(const char __user * specialfile, int
swap_flags)
>>     unsigned long maxpages = 1;
>>     int swapfilesize;
>>     unsigned short *swap_map;
>> +ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> + struct mem_cgroup **memcg;
>> +endif
> Remove #ifdefs
>

```

will do.

```

>> struct page *page = NULL;
>> struct inode *inode = NULL;

```

```

>> int did_down = 0;
>> @@ -1479,6 +1492,9 @@ asmlinkage long sys_swapon(const char __user * specialfile, int
swap_flags)
>> p->swap_file = NULL;
>> p->old_block_size = 0;
>> p->swap_map = NULL;
>> + #ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> + p->memcg = NULL;
>> + #endif
>
> void init_swap_ctlr_memcg(p);
>
I would like to call this one swap_cgroup_init_memcg.

```

```

>> p->lowest_bit = 0;
>> p->highest_bit = 0;
>> p->cluster_nr = 0;
>> @@ -1651,6 +1667,15 @@ asmlinkage long sys_swapon(const char __user * specialfile, int
swap_flags)
>>     1 /* header page */;
>>     if (error)
>>         goto bad_swap;
>> +
>> + #ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> + p->memcg = vmalloc(maxpages * sizeof(struct mem_cgroup *));
>> + if (!p->memcg) {
>> +     error = -ENOMEM;
>> +     goto bad_swap;
>> + }
>> + memset(p->memcg, 0, maxpages * sizeof(struct mem_cgroup *));
>> + #endif
> void alloc_swap_ctlr_memcg(p)
>
OK.
I'll implement swap_cgroup_alloc_memcg.

```

> But this implies swapon will fail at memory shortage. Is it good ?

>

Hum.

Would it be better to just disabling this feature?

```

>> }
>>
>> if (nr_good_pages) {
>> @@ -1710,11 +1735,18 @@ bad_swap_2:
>>     swap_map = p->swap_map;
>>     p->swap_file = NULL;
>>     p->swap_map = NULL;

```

```
>> +#ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> + memcg = p->memcg;
>> + p->memcg = NULL;
>> +#endif
>> p->flags = 0;
>> if (!(swap_flags & SWAP_FLAG_PREFER))
>> ++least_priority;
>> spin_unlock(&swap_lock);
>> vfree(swap_map);
>> +#ifdef CONFIG_CGROUP_SWAP_RES_CTLR
>> + vfree(memcg);
>> +#endif
>> if (swap_file)
>> filp_close(swap_file, NULL);
>> out:
>>
I'll handle these 2 #ifdefs as sys_swapoff.
```

```
>>
>
> Thanks,
> -Kame
>
```

Thank you for your advice!

Daisuke Nishimura.

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