Subject: Re: [RFC/PATCH] cgroup swap subsystem
Posted by Daisuke Nishimura on Thu, 06 Mar 2008 12:20:58 GMT
View Forum Message <> Reply to Message

Hi.

Paul Menage wrote:
```
>> +       pc = page_get_page_cgroup(page);
>> +       if (WARN_ON(!pc))
>> +               mm = &init_mm;
>> +       else
>> +               mm = pc->pc_mm;
>> +       BUG_ON(!mm);
>
> Is this safe against races with the mem.force_empty operation?
>
```
I've not considered yet about force_empty operation
of memory subsystem.
Thank you for pointing it out.

```
>> +
>> +       rcu_read_lock();
>> +       swap = rcu_dereference(mm->swap_cgroup);
>> +       rcu_read_unlock();
>> +       BUG_ON(!swap);
>
> Is it safe to do rcu_read_unlock() while you are still planning to
> operate on the value of "swap"?
>
```
You are right.
I think I should css_get() before rcu_read_unlock() as
memory subsystem does.

```
>> +
>> +#ifdef CONFIG_CGROUP_SWAP_LIMIT
>> +               p->swap_cgroup = vmalloc(maxpages * sizeof(*swap_cgroup));
>> +               if (!(p->swap_cgroup)) {
>> +                       error = -ENOMEM;
>> +                       goto bad_swap;
>> +               }
>> +               memset(p->swap_cgroup, 0, maxpages * sizeof(*swap_cgroup));
>> +#endif
>
> It would be nice to only allocate these the first time the swap cgroup
> subsystem becomes active, to avoid the overhead for people not using
> it; even better if you can free it again if the swap subsystem becomes
> inactive again.
>
```

Hmm.. good idea.
I think this is possible by adding a flag file, like "swap.enable_limit",
to the top of cgroup directory, and charging all the swap entries
which are used when the flag is enabled to the top cgroup.

Thanks,
Daisuke Nishimura.

_____

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