Subject: Re: [RFD][PATCH] memcg: Move Usage at Task Move
Posted by KAMEZAWA Hiroyuki on Wed, 11 Jun 2008 03:24:56 GMT
View Forum Message <> Reply to Message

On Wed, 11 Jun 2008 12:03:45 +0900
Daisuke Nishimura <nishimura@mxp.nes.nec.co.jp> wrote:
> > > Or, instead of implementing rollback in kernel,
> > > how about making user(or middle ware?) re-echo pid to rollbak
> > > on failure?
> > >
> >
> > "If the users does well, the system works in better way" is O.K.
> > "If the users doesn't well, the system works in broken way" is very bad.
> >
> Hum...
>
> I think users must know what they are doing.
>
yes. but it's a different problem,
 - "a user must know what they does."
 - "a system works without BUG even if the user is crazy."

> They must know that moving a process to another group
> that doesn't have enough room for it may fail with half state,
> if it is the behavior of kernel.
> And they should handle the error by themselves, IMHO.
>

I'm now considering following logic. How do you think ?

Assume: move TASK from group:CURR to group:DEST.

== move_task(TASK, CURR, DEST)

if (DEST's limit is unlimited)
 moving TASK
 return success.

usage = check_usage_of_task(TASK).

/* try to reserve enough room in destionation */
if (try_to_reserve_enough_room(DEST, usage)) {
 move TASK to DEST and move pages AMAP.
 /* usage_of_task(TASK) can be changed while we do this.
    Then, we move AMAP. */
 return success;
}

return failure.
==

The difficult point will be reservation but can be implemented without complexity.

Thanks,
-Kame

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