Subject: Re: [PATCH] introduce task cgroup v2
Posted by KOSAKI Motohiro on Sat, 21 Jun 2008 09:10:39 GMT
View Forum Message <> Reply to Message

> > Bad performance on the charge/uncharge?
> >
> > The only difference I can see is that res_counter uses
> > spin_lock_irqsave()/spin_unlock_irqrestore(), and you're using plain
> > spin_lock()/spin_unlock().
> >
> > Is the overhead of a pushf/cli/popf really going to matter compared
> > with the overhead of forking/exiting a task?
> >
> > Or approaching this from the other side, does res_counter really need
> > irq-safe locking, or is it just being cautious?
>
> We really need irq-safe locking. We can end up uncharging from reclaim context
> (called under zone->lru_lock and mem->zone->lru_lock - held with interrupts
> disabled)
>
> I am going to convert the spin lock to a reader writers lock, so that reads from
> user space do not cause contention. I'll experiment and look at the overhead.

Sorry, late responce.
I'm working on fix current -mm tree regression recently ;)

Note:

I am going to convert spinlock in task limit cgroup to atomic_t.
task limit cgroup has following caractatics.
 - many write (fork, exit)
 - few read
 - fork() is performance sensitive systemcall.
   if increase fork overhead, system total performance cause degression.

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