Subject: Re: [RFC][PATCH] Make access to taks's nsproxy liter
Posted by paulmck on Wed, 08 Aug 2007 18:48:00 GMT
View Forum Message <> Reply to Message

On Wed, Aug 08, 2007 at 09:36:47PM +0400, Oleg Nesterov wrote:
> On 08/08, Paul E. McKenney wrote:
> >
> > On Wed, Aug 08, 2007 at 08:41:07PM +0400, Oleg Nesterov wrote:
> > > > +void switch_task_namespaces(struct task_struct *p, struct nsproxy *new)
> > > > +{
> > > > + struct nsproxy *ns;
> > > > +
> > > > + might_sleep();
> > > > +
> > > > + ns = p->nsproxy;
> > > > + if (ns == new)
> > > > + return;
> > > > +
> > > > + if (new)
> > > > + get_nsproxy(new);
> > > > + rcu_assign_pointer(p->nsproxy, new);
> > > > +
> > > > + if (ns && atomic_dec_and_test(&ns->count)) {
> > > > + /*
> > > > +   * wait for others to get what they want from this
> > > > +   * nsproxy. cannot release this nsproxy via the
> > > > +   * call_rcu() since put_mnt_ns will want to sleep
> > > > +   */
> > > > + synchronize_rcu();
> > > > + free_nsproxy(ns);
> > > > + }
> > > > +}
> > >
> > > (I may be wrong, Paul cc'ed)
> > >
> > > This is correct with the current implementation of RCU, but strictly speaking,
> > > we can't use synchronize_rcu() here, because write_lock_irq() doesn't imply
> > > rcu_read_lock() in theory.
> >
> > Can you use synchronize_sched() instead? The synchronize_sched()
> > primitive will wait until all preempt/irq-disable code sequences complete.
> > Therefore, it would wait for all write_lock_irq() code sequences to
> > complete.
>
> Thanks Paul!
>
> But we also need to cover the case when ->nsproxy is used under rcu_read_lock(),
> so if we go this way, we'd better add rcu_read_lock() to do_notify_parent.*() as

> Eric suggested.

Makes sense. Just for completeness, the other thing you could do would be to do both a synchronize_sched() and a synchronize_rcu() in the switch_task_namespaces() function, but I believe that Eric's approach would be better. (The only counter-example I can come up with off-hand would be if there were tons of read paths, and you needed a quick fix. But even in that case, hopefully the quick fix would be followed by taking Eric's approach.)

Thanx, Paul

_____

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