Subject: Re: Getting the new RxRPC patches upstream
Posted by David Howells on Tue, 24 Apr 2007 13:37:04 GMT
View Forum Message <> Reply to Message

Oleg Nesterov <oleg@tv-sign.ru> wrote:

> > > We only care when del_timer() returns true. In that case, if the timer
> > > function still runs (possible for single-threaded wqs), it has already
> > > passed __queue_work().
> >
> > Why do you assume that?

Sorry, I should have been more clear. I meant the assumption that we only care about a true return from del_timer().

> If del_timer() returns true, the timer was pending. This means it was
> started by work->func() (note that __run_timers() clears timer_pending()
> before calling timer->function). This in turn means that
> delayed_work_timer_fn() has already called __queue_work(dwork), otherwise
> work->func() has no chance to run.

But if del_timer() returns 0, then there may be a problem. We can't tell the difference between the following two cases:

(1) The timer hadn't been started.

(2) The timer had been started, has expired and is no longer pending, but another CPU is running its handler routine.

try_to_del_timer_sync() _does_, however, distinguish between these cases: the first is the 0 return, the second is the -1 return, and the case where it dequeued the timer is the 1 return.

BTW, can a timer handler be preempted? I assume not... But it can be delayed by interrupt processing.

David

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