Subject: Re: [PATCH v7 00/10] Request for Inclusion: per-cgroup tcp memory pressure
Posted by KAMEZAWA Hiroyuki on Mon, 05 Dec 2011 02:06:19 GMT
View Forum Message <> Reply to Message

On Fri, 2 Dec 2011 16:04:08 -0200
Glauber Costa <glommer@parallels.com> wrote:

```
> On 11/30/2011 12:11 AM, KAMEZAWA Hiroyuki wrote:
> > On Tue, 29 Nov 2011 21:56:51 -0200
> > Glauber Costa<glommer@parallels.com>  wrote:
> >
> >> Hi,
> >>
> >> This patchset implements per-cgroup tcp memory pressure controls. It did not change
> >> significantly since last submission: rather, it just merges the comments Kame had.
> >> Most of them are style-related and/or Documentation, but there are two real bugs he
> >> managed to spot (thanks)
> >>
> >> Please let me know if there is anything else I should address.
> >>
> >
> > After reading all codes again, I feel some strange. Could you clarify ?
> >
> > Here.
> > ==
> > +void sock_update_memcg(struct sock *sk)
> > +{
> > + /* right now a socket spends its whole life in the same cgroup */
> > + if (sk->sk_cgrp) {
> > +  WARN_ON(1);
> > +  return;
> > + }
> > + if (static_branch(&memcg_socket_limit_enabled)) {
> > +  struct mem_cgroup *memcg;
> > +
> > +  BUG_ON(!sk->sk_prot->proto_cgroup);
> > +
> > +  rcu_read_lock();
> > +  memcg = mem_cgroup_from_task(current);
> > +  if (!mem_cgroup_is_root(memcg))
> > +   sk->sk_cgrp = sk->sk_prot->proto_cgroup(memcg);
> > +  rcu_read_unlock();
> > ==
> >
> > sk->sk_cgrp is set to a memcg without any reference count.
> >
> > Then, no check for preventing rmdir() and freeing memcgroup.
```

> >
> > Is there some css_get() or mem_cgroup_get() somewhere ?
> >
>
> There were a css_get in the first version of this patchset. It was
> removed, however, because it was deemed anti-intuitive to prevent rmdir,
> since we can't know which sockets are blocking it, or do anything about
> it. Or did I misunderstand something ?
>

Maybe I misuderstood. Thank you. Ok, there is no css_get/put and
rmdir() is allowed. But, hmm....what's guarding threads from stale
pointer access ?

Does a memory cgroup which is pointed by sk->sk_cgrp always exist ?

-Kame