

---

Subject: Re: [PATCH v2 06/11] memcg: kmem controller infrastructure  
Posted by [Michal Hocko](#) on Wed, 15 Aug 2012 13:09:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Wed 15-08-12 13:42:24, Glauber Costa wrote:

[...]

> >> +

> >> + ret = 0;

> >> +

> >> + if (!memcg)

> >> + return ret;

> >> +

> >> + \_memcg = memcg;

> >> + ret = \_\_mem\_cgroup\_try\_charge(NULL, gfp, delta / PAGE\_SIZE,

> >> + &\_memcg, may\_oom);

> >

> > This is really dangerous because atomic allocation which seem to be

> > possible could result in deadlocks because of the reclaim.

>

> Can you elaborate on how this would happen?

Say you have an atomic allocation and we hit the limit so we get either to reclaim which can sleep or to oom which can sleep as well (depending on the oom\_control).

> > Also, as I have mentioned in the other email in this thread. Why

> > should we reclaim just because of kernel allocation when we are not

> > reclaiming any of it because shrink\_slab is ignored in the memcg

> > reclaim.

>

> Don't get too distracted by the fact that shrink\_slab is ignored. It is

> temporary, and while this being ignored now leads to suboptimal

> behavior, it will 1st, only affect its users, and 2nd, not be disastrous.

It's not just about shrink\_slab it is also about triggering memcg-oom which doesn't consider kmem accounted memory so the wrong tasks could be killed. It is true that the impact is packed inside the group (hierarchy) so you are right it won't be disastrous.

--

Michal Hocko  
SUSE Labs

---