
Subject: Re: [PATCH 17/23] kmem controller charge/uncharge infrastructure
Posted by [Frederic Weisbecker](#) on Fri, 27 Apr 2012 11:38:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Tue, Apr 24, 2012 at 01:21:43PM -0700, David Rientjes wrote:

> On Tue, 24 Apr 2012, Frederic Weisbecker wrote:

>

> > > This seems horribly inconsistent with memcg charging of user memory since
> > > it charges to p->mm->owner and you're charging to p. So a thread attached
> > > to a memcg can charge user memory to one memcg while charging slab to
> > > another memcg?

> >

> > Charging to the thread rather than the process seem to me the right behaviour:
> > you can have two threads of a same process attached to different cgroups.

> >

> > Perhaps it is the user memory memcg that needs to be fixed?

> >

>

> No, because memory is represented by mm_struct, not task_struct, so you
> must charge to p->mm->owner to allow for moving threads amongst memcgs
> later for memory.move_charge_at_immigrate. You shouldn't be able to
> charge two different memcgs for memory represented by a single mm.

The idea I had was more that only the memcg of the thread that does the allocation is charged. But the problem is that this allocation can be later deallocated from another thread. So probably charging the owner is indeed the only sane way to go with user memory.
