
Subject: Re: [PATCH 17/23] kmem controller charge/uncharge infrastructure
Posted by [David Rientjes](#) on Fri, 27 Apr 2012 18:13:31 GMT

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

On Fri, 27 Apr 2012, Frederic Weisbecker wrote:

> > 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.
>

It's all really the same concept: if we want to move memory of a process, willingly free memory in the process itself, or free memory of a process by way of the oom killer, we need a way to do that for the entire process so the accounting makes sense afterwards. And since we have that requirement for user memory, it makes sense that its consistent with slab as well. I don't think a thread of a process should be able to charge slab to one memcg while its user memory is charged to another memcg.
