Subject: Re: [PATCH 17/23] kmem controller charge/uncharge infrastructure
Posted by Glauber Costa on Tue, 24 Apr 2012 21:36:02 GMT
View Forum Message <> Reply to Message

On 04/24/2012 05:25 PM, David Rientjes wrote:
> On Tue, 24 Apr 2012, Glauber Costa wrote:
>
>> I think memcg is not necessarily wrong. That is because threads in a process
>> share an address space, and you will eventually need to map a page to deliver
>> it to userspace. The mm struct points you to the owner of that.
>>
>> But that is not necessarily true for things that live in the kernel address
>> space.
>>
>> Do you view this differently ?
>>
>
> Yes, for user memory, I see charging to p->mm->owner as allowing that
> process to eventually move and be charged to a different memcg and there's
> no way to do proper accounting if the charge is split amongst different
> memcgs because of thread membership to a set of memcgs. This is
> consistent with charges for shared memory being moved when a thread
> mapping it moves to a new memcg, as well.

But that's the problem.

When we are dealing with kernel memory, we are allocating a whole slab
page. It is essentially impossible to track, given a page, which task
allocated which object.