Subject: Re: [PATCH v5 06/14] memcg: kmem controller infrastructure
Posted by Glauber Costa on Fri, 19 Oct 2012 10:00:23 GMT

View Forum Message <> Reply to Message

On 10/19/2012 01:31 PM, David Rientjes wrote:
> On Fri, 19 Oct 2012, Glauber Costa wrote:
>
>>>>> Do we actually need to test PF_KTHREAD when current->mm == NULL?
>>>>> Perhaps because of aio threads whcih temporarily adopt a userspace mm?
>>>>
>>>> I believe so. I remember I discussed this in the past with David
>>>> Rientjes and he advised me to test for both.
>>>>
>>>
>>> PF_KTHREAD can do use_mm() to assume an ->mm but hopefully they aren't
>>> allocating slab while doing so. Have you considered actually charging
>>> current->mm->owner for that memory, though, since the kthread will have
>>> freed the memory before unuse_mm() or otherwise have charged it on behalf
>>> of a user process, i.e. only exempting PF_KTHREAD?
>>>
>> I always charge current->mm->owner.
>>
>
> Yeah, I'm asking have you considered charging current->mm->owner for the
> memory when a kthread (current) assumes the mm of a user process via
> use_mm()? It may free the memory before calling unuse_mm(), but it's also
> allocating the memory on behalf of a user so this exemption might be
> dangerous if use_mm() becomes more popular. I don't think there's
> anything that prevents that charge, I'm just wondering if you considered
> doing it even for kthreads with an mm.
>
Well, I thought about it.

And I personally don't like it. I think all kthreads should be treated
the same. We have control over it, unlike any userspace application. We
never expect its memory consumption to explode.

Specially considering that those allocations are supposed to be
short-lived, we are only paying the res_counters count for no reason.