Subject: Re: [PATCH v3 16/28] memcg: kmem controller charge/uncharge infrastructure
Posted by Glauber Costa on Wed, 30 May 2012 16:16:59 GMT

View Forum Message <> Reply to Message

On 05/30/2012 07:33 PM, Frederic Weisbecker wrote:
> On Wed, May 30, 2012 at 05:55:38PM +0400, Glauber Costa wrote:
>> On 05/30/2012 05:53 PM, Frederic Weisbecker wrote:
>>> On Wed, May 30, 2012 at 05:37:57PM +0400, Glauber Costa wrote:
>>>> On 05/30/2012 05:37 PM, Frederic Weisbecker wrote:
>>>>> Right. __mem_cgroup_get_kmem_cache() fetches the memcg of the owner
>>>>> and calls memcg_create_cache_enqueue() which does css_tryget(&memcg->css).
>>>>> After this tryget I think you're fine. And in-between you're safe against
>>>>> css_set removal due to rcu_read_lock().
>>>>>
>>>>> I'm less clear with __mem_cgroup_new_kmem_page() though...
>>>>
>>>> That one does not get memcg->css but it does call mem_cgroup_get(),
>>>> that does prevent against the memcg structure being freed, which I
>>>> believe to be good enough.
>>>
>>> What if the owner calls cgroup_exit() between mem_cgroup_from_task()
>>> and mem_cgroup_get()? The css_set which contains the memcg gets freed.
>>> Also the reference on the memcg doesn't even prevent the css_set to
>>> be removed, does it?
>> It doesn't, but we don't really care. The css can go away, if the
>> memcg structure stays.
>
> Ah right, the memcg itself is only freed at destroy time.
>
>> The caches will outlive the memcg anyway,
>> since it is possible that you delete it, with some caches still
>> holding objects that
>> are not freed (they will be marked as dead).
>
> I guess I need to look at how the destroy path is handled in your patchset
> then. Or how you ensure that __mem_cgroup_new_kmem_page() can't race against
> destroy.
Appreciate that, thanks.