
Subject: Re: [PATCH v2 06/11] memcg: kmem controller infrastructure
Posted by [Glauber Costa](#) on Thu, 23 Aug 2012 07:51:31 GMT
[View Forum Message](#) < > [Reply to Message](#)

>>> Perhaps we're just trying to take a conservative initial implementation
>>> which is consistent with user visible pages.

>>>

>>

>> The way I see it, is not about being conservative, but rather about my
>> physical safety. It is quite easy and natural to assume that "all
>> modifications to page cgroup are done under lock". So someone modifying
>> this later will likely find out about this exception in a rather
>> unpleasant way. They know where I live, and guns for hire are everywhere.

>>

>> Note that it is not unreasonable to believe that we can modify this
>> later. This can be a way out, for example, for the memcg lifecycle problem.

>>

>> I agree with your analysis and we can ultimately remove it, but if we
>> cannot pinpoint any performance problems to here, maybe consistency
>> wins. Also, the locking operation itself is a bit expensive, but the
>> biggest price is the actual contention. If we'll have nobody contending
>> for the same page_cgroup, the problem - if exists - shouldn't be that
>> bad. And if we ever have, the lock is needed.

>

> Sounds reasonable. Another reason we might have to eventually revisit
> this lock is the fact that lock_page_cgroup() is not generally irq_safe.
> I assume that slab pages may be freed in softirq and would thus (in an
> upcoming patch series) call __memcg_kmem_free_page. There are a few
> factors that might make it safe to grab this lock here (and below in
> __memcg_kmem_free_page) from hard/softirq context:
> * the pc lock is a per page bit spinlock. So we only need to worry
> about interrupting a task which holds the same page's lock to avoid
> deadlock.

> * for accounted kernel pages, I am not aware of other code beyond
> __memcg_kmem_charge_page and __memcg_kmem_free_page which grab pc
> lock. So we shouldn't find __memcg_kmem_free_page() called from a
> context which interrupted a holder of the page's pc lock.

>

All very right.
