Subject: Re: [PATCH 16/23] slab: provide kmalloc_no_account
Posted by Glauber Costa on Wed, 25 Apr 2012 14:29:22 GMT
View Forum Message <> Reply to Message

On 04/24/2012 10:44 PM, KAMEZAWA Hiroyuki wrote:
> (2012/04/23 8:53), Glauber Costa wrote:
>
>> Some allocations need to be accounted to the root memcg regardless
>> of their context. One trivial example, is the allocations we do
>> during the memcg slab cache creation themselves. Strictly speaking,
>> they could go to the parent, but it is way easier to bill them to
>> the root cgroup.
>>
>> Only generic kmalloc allocations are allowed to be bypassed.
>>
>> The function is not exported, because drivers code should always
>> be accounted.
>>
>> This code is mosly written by Suleiman Souhlal.
>>
>> Signed-off-by: Glauber Costa<glommer@parallels.com>
>> CC: Christoph Lameter<cl@linux.com>
>> CC: Pekka Enberg<penberg@cs.helsinki.fi>
>> CC: Michal Hocko<mhocko@suse.cz>
>> CC: Kamezawa Hiroyuki<kamezawa.hiroyu@jp.fujitsu.com>
>> CC: Johannes Weiner<hannes@cmpxchg.org>
>> CC: Suleiman Souhlal<suleiman@google.com>
>
>
> Seems reasonable.
> Reviewed-by: KAMEZAWA Hiroyuki<kamezawa.hiroyu@jp.fujitsu.com>
>
> Hmm...but can't we find the 'context' in automatic way ?
>

Not that I can think of. Well, actually, not without adding some tests
to the allocation path I'd rather not (like testing for the return
address and then doing a table lookup, etc)

An option would be to store it in the task_struct. So we would allocate
as following:

```
memcg_skip_account_start(p);
do_a_bunch_of_allocations();
memcg_skip_account_stop(p);
```

The problem with that, is that it is quite easy to abuse.
but if we don't export that to modules, it would be acceptable.

Question is, given the fact that the number of kmalloc_no_account() is expected to be really small, is it worth it?