Subject: Re: [PATCH v3 04/13] kmem accounting basic infrastructure
Posted by Glauber Costa on Thu, 04 Oct 2012 11:55:14 GMT
View Forum Message <> Reply to Message

On 10/04/2012 02:54 AM, Tejun Heo wrote:
> Hello, Glauber.
>
> On Mon, Oct 01, 2012 at 12:45:11PM +0400, Glauber Costa wrote:
>>> where kmemcg_slab_idx is updated from sched notifier (or maybe add and
>>> use current->kmemcg_slab_idx?). You would still need __GFP_* and
>>> in_interrupt() tests but current->mm and PF_KTHREAD tests can be
>>> rolled into index selection.
>>
>> How big would this array be? there can be a lot more kmem_caches than
>> there are memcgs. That is why it is done from memcg side.
>
> The total number of memcgs are pretty limited due to the ID thing,
> right? And kmemcg is only applied to subset of caches. I don't think
> the array size would be a problem in terms of memory overhead, would
> it? If so, RCU synchronize and dynamically grow them?
>
> Thanks.
>

I don't want to assume the number of memcgs will always be that limited. Sure, the ID limitation sounds pretty much a big one, but people doing VMs usually want to stack as many VMs as they possibly can in an environment, and the less things preventing that from happening, the better.

That said, now that I've experimented with this a bit, indexing from the cache may have some advantages: it can get too complicated to propagate new caches appearing to all memcgs that already in-flight. We don't have this problem from the cache side, because instances of it are guaranteed not to exist at this point by definition.

I don't want to bloat unrelated kmem_cache structures, so I can't embed a memcg array in there: I would have to have a pointer to a memcg array that gets assigned at first use. But if we don't want to have a static number, as you and christoph already frowned upon heavily, we may have to do that memcg side as well.

The array gets bigger, though, because it pretty much has to be enough to accomodate all css_ids. Even now, they are more than the 400 I used in this patchset. Not allocating all of them at once will lead to more complication and pointer chasing in here.

I'll take a look at the alternatives today and tomorrow.