Subject: Re: [PATCH v9 1/9] Basic kernel memory functionality for the Memory Controller
Posted by Glauber Costa on Fri, 16 Dec 2011 13:02:51 GMT
View Forum Message <> Reply to Message

On 12/16/2011 04:32 PM, Michal Hocko wrote:
> On Thu 15-12-11 16:29:18, Glauber Costa wrote:
>> On 12/14/2011 09:04 PM, Michal Hocko wrote:
>>> [Now with the current patch version, I hope]
>>> On Mon 12-12-11 11:47:01, Glauber Costa wrote:
> [...]
>>>> @@ -3848,10 +3862,17 @@ static inline u64 mem_cgroup_usage(struct mem_cgroup *memcg, bool swap)
>>>> u64 val;
>>>>
>>>> if (!mem_cgroup_is_root(memcg)) {
>>>> + val = 0;
>>>> +#ifdef CONFIG_CGROUP_MEM_RES_CTLR_KMEM
>>>> + if (!memcg->kmem_independent_accounting)
>>>> + val = res_counter_read_u64(&memcg->kmem, RES_USAGE);
>>>> +#endif
>>>> if (!swap)
>>>> - return res_counter_read_u64(&memcg->res, RES_USAGE);
>>>> + val += res_counter_read_u64(&memcg->res, RES_USAGE);
>>>> else
>>>> - return res_counter_read_u64(&memcg->memsw, RES_USAGE);
>>>> + val += res_counter_read_u64(&memcg->memsw, RES_USAGE);
>>>> +
>>>> + return val;
>>>> }
>>>
>>> So you report kmem+user but we do not consider kmem during charge so one
>>> can easily end up with usage_in_bytes over limit but no reclaim is going
>>> on. Not good, I would say.
>
> I find this a problem and one of the reason I do not like !independent
> accounting.
>
>>>
>>> OK, so to sum it up. The biggest problem I see is the (non)independent
>>> accounting. We simply cannot mix user+kernel limits otherwise we would
>>> see issues (like kernel resource hog would force memcg-oom and innocent
>>> members would die because their rss is much bigger).
>>> It is also not clear to me what should happen when we hit the kmem
>>> limit. I guess it will be kmem cache dependent.
>>
>> So right now, tcp is completely independent, since it is not
>> accounted to kmem.

>
> So why do we need kmem accounting when tcp (the only user at the moment)
> doesn't use it?

Well, a bit historical. I needed a basic placeholder for it, since it
tcp is officially kmem. As the time passed, I took most of the stuff out
of this patch to leave just the basics I would need for tcp.
Turns out I ended up focusing on the rest, and some of the stuff was
left here.

At one point I merged tcp data into kmem, but then reverted this
behavior. the kmem counter stayed.

I agree deferring the whole behavior would be better.

>> In summary, we still never do non-independent accounting. When we
>> start doing it for the other caches, We will have to add a test at
>> charge time as well.
>
> So we shouldn't do it as a part of this patchset because the further
> usage is not clear and I think there will be some real issues with
> user+kmem accounting (e.g. a proper memcg-oom implementation).
> Can you just drop this patch?

Yes, but the whole set is in the net tree already. (All other patches
are tcp-related but this) Would you mind if I'd send a follow up patch
removing the kmem files, and leaving just the registration functions and
basic documentation? (And sorry for that as well in advance)

>> We still need to keep it separate though, in case the independent
>> flag is turned on/off
>
> I don't mind to have kmem.tcp.* knobs.
>