

---

Subject: Re: [PATCH v5 06/14] memcg: kmem controller infrastructure  
Posted by [Glauber Costa](#) on Fri, 19 Oct 2012 10:08:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On 10/19/2012 01:59 AM, David Rientjes wrote:

> On Thu, 18 Oct 2012, Glauber Costa wrote:

>  
>>>> @@ -2630,6 +2634,171 @@ static void \_\_mem\_cgroup\_commit\_charge(struct  
mem\_cgroup \*memcg,  
>>>> memcg\_check\_events(memcg, page);  
>>>> }  
>>>>  
>>>> +#ifdef CONFIG\_MEMCG\_KMEM  
>>>> +static inline bool memcg\_can\_account\_kmem(struct mem\_cgroup \*memcg)  
>>>> +{  
>>>> + return !mem\_cgroup\_disabled() && !mem\_cgroup\_is\_root(memcg) &&  
>>>> + (memcg->kmem\_accounted & KMEM\_ACCOUNTED\_MASK);  
>>>> +}  
>>>> +  
>>>> +static int memcg\_charge\_kmem(struct mem\_cgroup \*memcg, gfp\_t gfp, u64 size)  
>>>> +{  
>>>> + struct res\_counter \*fail\_res;  
>>>> + struct mem\_cgroup \*\_memcg;  
>>>> + int ret = 0;  
>>>> + bool may\_oom;  
>>>> +  
>>>> + ret = res\_counter\_charge(&memcg->kmem, size, &fail\_res);  
>>>> + if (ret)  
>>>> + return ret;  
>>>> +  
>>>> + /\*  
>>>> + \* Conditions under which we can wait for the oom\_killer.  
>>>> + \* We have to be able to wait, but also, if we can't retry,  
>>>> + \* we obviously shouldn't go mess with oom.  
>>>> + \*/  
>>>> + may\_oom = (gfp & \_\_GFP\_WAIT) && !(gfp & \_\_GFP\_NORETRY);  
>>>>  
>>> What about gfp & \_\_GFP\_FS?  
>>>  
>>  
>> Do you intend to prevent or allow OOM under that flag? I personally  
>> think that anything that accepts to be OOM-killed should have GFP\_WAIT  
>> set, so that ought to be enough.  
>>  
>  
> The oom killer in the page allocator cannot trigger without \_\_GFP\_FS  
> because direct reclaim has little chance of being very successful and  
> thus we end up needlessly killing processes, and that tends to happen

> quite a bit if we dont check for it. Seems like this would also happen  
> with memcg if mem\_cgroup\_reclaim() has a large probability of failing?  
>

I can indeed see tests for GFP\_FS in some key locations in mm/ before calling the OOM Killer.

Should I test for GFP\_IO as well? If the idea is preventing OOM to trigger for allocations that can write their pages back, how would you feel about the following test:

may\_oom = (gfp & GFP\_KERNEL) && !(gfp & \_\_GFP\_NORETRY) ?

Michal, what is your take in here?

---