Subject: Re: [PATCH v5 01/14] memcg: Make it possible to use the stock for more than one page.
Posted by Glauber Costa on Thu, 18 Oct 2012 08:55:32 GMT
View Forum Message <> Reply to Message

On 10/18/2012 02:11 AM, Andrew Morton wrote:
> On Tue, 16 Oct 2012 14:16:38 +0400
> Glauber Costa <glommer@parallels.com> wrote:
>
>> From: Suleiman Souhlal <ssouhlal@FreeBSD.org>
>>
>> We currently have a percpu stock cache scheme that charges one page at a
>> time from memcg->res, the user counter. When the kernel memory
>> controller comes into play, we'll need to charge more than that.
>>
>> This is because kernel memory allocations will also draw from the user
>> counter, and can be bigger than a single page, as it is the case with
>> the stack (usually 2 pages) or some higher order slabs.
>>
>> ...
>>
>> -/*
>> - * Try to consume stocked charge on this cpu. If success, one page is consumed
>> - * from local stock and true is returned. If the stock is 0 or charges from a
>> - * cgroup which is not current target, returns false. This stock will be
>> - * refilled.
>> +/**
>> + * consume_stock: Try to consume stocked charge on this cpu.
>> + * @memcg: memcg to consume from.
>> + * @nr_pages: how many pages to charge.
>> + *
>> + * The charges will only happen if @memcg matches the current cpu's memcg
>> + * stock, and at least @nr_pages are available in that stock. Failure to
>> + * service an allocation will refill the stock.
>> + *
>> + * returns true if succesfull, false otherwise.
>
> spello.
>
Thanks. I can never successfuly write successfull =(

>>   */
>> -static bool consume_stock(struct mem_cgroup *memcg)
>> +static bool consume_stock(struct mem_cgroup *memcg, int nr_pages)
>
> I don't believe there is a case for nr_pages < 0 here? If not then I
> suggest that it would be clearer to use an unsigned type, like
> memcg_stock_pcp.stock.

>

Done, thanks.