Subject: Re: [PATCH 02/11] memcg: Reclaim when more than one page needed.
Posted by Glauber Costa on Tue, 26 Jun 2012 09:08:48 GMT
View Forum Message <> Reply to Message

On 06/26/2012 12:54 PM, David Rientjes wrote:
> On Tue, 26 Jun 2012, Glauber Costa wrote:
>
>>>> + * retries
>>>> + */
>>>> +#define NR_PAGES_TO_RETRY 2
>>>> +
>>>
>>> Should be 1 << PAGE_ALLOC_COSTLY_ORDER? Where does this number come from?
>>> The changelog doesn't specify.
>>
>> Hocko complained about that, and I changed. Where the number comes from, is
>> stated in the comments: it is a number small enough to have high changes of
>> had been freed by the previous reclaim, and yet around the number of pages of
>> a kernel allocation.
>>
>
> PAGE_ALLOC_COSTLY_ORDER _is_ the threshold used to determine where reclaim
> and compaction is deemed to be too costly to continuously retry, I'm not
> sure why this is any different?
>
> And this is certainly not "around the number of pages of a kernel
> allocation", that depends very heavily on the slab allocator being used;
> slub very often uses order-2 and order-3 page allocations as the default
> settings (it is capped at, you guessed it, PAGE_ALLOC_COSTLY_ORDER
> internally by default) and can be significantly increased on the command
> line.

I am obviously okay with either.

Maybe Michal can comment on this?

>> Of course there are allocations for nr_pages > 2. But 2 will already service
>> the stack most of the time, and most of the slab caches.
>>
>
> Nope, have you checked the output of /sys/kernel/slab/.../order when
> running slub? On my workstation 127 out of 316 caches have order-2 or
> higher by default.
>

Well, this is still on the side of my argument, since this is still a
majority of them being low ordered. The code here does not necessarily
have to retry - if I understand it correctly - we just retry for very

small allocations because that is where our likelihood of succeeding is.