Subject: Re: [PATCH v5 07/14] mm: Allocate kernel pages to the right memcg
Posted by Glauber Costa on Tue, 16 Oct 2012 18:55:30 GMT
View Forum Message <> Reply to Message

On 10/16/2012 07:31 PM, Christoph Lameter wrote:
> On Tue, 16 Oct 2012, Glauber Costa wrote:
>
>> To avoid adding markers to the page - and a kmem flag that would
>> necessarily follow, as much as doing page_cgroup lookups for no reason,
>> whoever is marking its allocations with __GFP_KMEMCG flag is responsible
>> for telling the page allocator that this is such an allocation at
>> free_pages() time. This is done by the invocation of
>> __free_accounted_pages() and free_accounted_pages().
>
> Hmmm... The code paths to free pages are often shared between multiple
> subsystems. Are you sure that this is actually working and accurately
> tracks the MEMCG pages?
>

As described above, only call sites that are switched to
free_accounted_pages are affected. There are very few of them. The stack
case is particularly easy to test: every time a process appears, usage
is increased in 8k. Every time a process dies, usage decreases by 8k.

In my other patchseries, I include the object allocators into this. So
again: there are very few call sites actually being patched.

>> +/*
>> + * __free_accounted_pages and free_accounted_pages will free pages allocated
>> + * with __GFP_KMEMCG.
>> + *
>> + * Those pages are accounted to a particular memcg, embedded in the
>> + * corresponding page_cgroup. To avoid adding a hit in the allocator to search
>> + * for that information only to find out that it is NULL for users who have no
>> + * interest in that whatsoever, we provide these functions.
>> + *
>> + * The caller knows better which flags it relies on.
>> + */
>> +void __free_accounted_pages(struct page *page, unsigned int order)
>> +{
>> + memcg_kmem_uncharge_page(page, order);
>> + __free_pages(page, order);
>> +}
>
> If we already are introducing such an API: Could it not be made more
> general so that it can also be used in the future to communicate other
> characteristics of a page on free?

>

I guess so. Which other use case do you have in mind?
In any case, I don't see this as a blocker to this patchset. There is no reason why it can't be done should the need arise.
