Subject: Re: [PATCH v2 07/11] mm: Allocate kernel pages to the right memcg
Posted by Michal Hocko on Wed, 15 Aug 2012 09:24:05 GMT
View Forum Message <> Reply to Message

On Thu 09-08-12 17:01:15, Glauber Costa wrote:
[...]
> diff --git a/mm/page_alloc.c b/mm/page_alloc.c
> index b956cec..da341dc 100644
> --- a/mm/page_alloc.c
> +++ b/mm/page_alloc.c
> @@ -2532,6 +2532,7 @@ __alloc_pages_nodemask(gfp_t gfp_mask, unsigned int order,
> struct page *page = NULL;
> int migratetype = allocflags_to_migratetype(gfp_mask);
> unsigned int cpuset_mems_cookie;
> + void *handle = NULL;
>
> gfp_mask &= gfp_allowed_mask;
>
> @@ -2543,6 +2544,13 @@ __alloc_pages_nodemask(gfp_t gfp_mask, unsigned int order,
> return NULL;
>
> /*
> + * Will only have any effect when __GFP_KMEMCG is set.
> + * This is verified in the (always inline) callee
> + */
> + if (!memcg_kmem_new_page(gfp_mask, &handle, order))
> + return NULL;

When the previous patch introduced this function I thought the handle
obfuscantion is to prevent from spreading struct mem_cgroup inside the
page allocator but memcg_kmem_commit_page uses the type directly. So why
that obfuscation? Even handle as a name sounds unnecessarily confusing.
I would go with struct mem_cgroup **memcgp or even return the pointer on
success or NULL otherwise.

[...]
> +EXPORT_SYMBOL(__free_accounted_pages);

Why exported?

Btw. this is called from call_rcu context but it itself calls call_rcu
down the chain in mem_cgroup_put. Is it safe?

[...]
> +EXPORT_SYMBOL(free_accounted_pages);

here again
--

Michal Hocko
SUSE Labs