Subject: Re: [PATCH v3 15/16] memcg/sl[au]b: shrink dead caches
Posted by JoonSoo Kim on Fri, 21 Sep 2012 04:48:30 GMT
View Forum Message <> Reply to Message

Hi Glauber.

2012/9/18 Glauber Costa <glommer@parallels.com>:
> diff --git a/mm/slub.c b/mm/slub.c
> index 0b68d15..9d79216 100644
> --- a/mm/slub.c
> +++ b/mm/slub.c
> @@ -2602,6 +2602,7 @@ redo:
> } else
> __slab_free(s, page, x, addr);
>
> + kmem_cache_verify_dead(s);
> }

As far as u know, I am not a expert and don't know anything about memcg.
IMHO, this implementation may hurt system performance in some case.

In case of memcg is destoried, remained kmem_cache is marked "dead".
After it is marked,
every free operation to this "dead" kmem_cache call
kmem_cache_verify_dead() and finally call kmem_cache_shrink().
kmem_cache_shrink() do invoking kmalloc and flush_all() and taking a
lock for online node and invoking kfree.
Especially, flush_all() may hurt performance largely, because it call
has_cpu_slab() against all the cpus.

And I know some other case it can hurt system performance.
But, I don't mention it, because above case is sufficient to worry.

And, I found one case that destroying memcg's kmem_cache don't works properly.
If we destroy memcg after all object is freed, current implementation
doesn't destroy kmem_cache.
kmem_cache_destroy_work_func() check "cachep->memcg_params.nr_pages == 0",
but in this case, it return false, because kmem_cache may have
cpu_slab, and cpu_partials_slabs.
As we already free all objects, kmem_cache_verify_dead() is not invoked forever.
I think that we need another kmem_cache_shrink() in
kmem_cache_destroy_work_func().

I don't convince that I am right, so think carefully my humble opinion.

Thanks.