

---

Subject: Re: [PATCH v3 12/28] slab: pass memcg parameter to  
kmem\_cache\_create

Posted by [Christoph Lameter](#) on Tue, 29 May 2012 16:52:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Tue, 29 May 2012, Glauber Costa wrote:

> > How do you detect that someone is touching it?  
>  
> kmem\_alloc\_cache will create mem\_cgroup\_get\_kmem\_cache.  
> (protected by static\_branches, so won't happen if you don't have at least  
> non-root memcg using it)  
>  
> \* Then it detects which memcg the calling process belongs to,  
> \* if it is the root memcg, go back to the allocation as quickly as we  
> can  
> \* otherwise, in the creation process, you will notice that each cache  
> has an index. memcg will store pointers to the copies and find them by  
> the index.  
>  
> From this point on, all the code of the caches is reused (except for  
> accounting the page)

Well kmem\_cache\_alloc cache is the performance critical hotpath.

If you are already there and doing all of that then would it not be better to simply count the objects allocated and freed per cgroup? Directly increment and decrement counters in a cgroup? You do not really need to duplicate the kmem\_cache structure and do not need to modify allocators if you are willing to take that kind of a performance hit. Put a wrapper around kmem\_cache\_alloc/free and count things.

---