

---

Subject: Re: [PATCH v5] slab: Ignore internal flags in cache creation

Posted by [Glauber Costa](#) on Fri, 19 Oct 2012 09:32:43 GMT

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

---

On 10/19/2012 02:42 AM, Andrew Morton wrote:

> On Wed, 17 Oct 2012 15:36:51 +0400

> Glauber Costa <glommer@parallels.com> wrote:

>

>> Some flags are used internally by the allocators for management  
>> purposes. One example of that is the CFLGS\_OFF\_SLAB flag that slab uses  
>> to mark that the metadata for that cache is stored outside of the slab.

>>

>> No cache should ever pass those as a creation flags. We can just ignore  
>> this bit if it happens to be passed (such as when duplicating a cache in  
>> the kmem memcg patches).

>

> I may be misunderstanding this, but...

>

> If some caller to kmem\_cache\_create() is passing in bogus flags then  
> that's a bug, and it is undesirable to hide such a bug in this fashion?

>

Not necessarily.

This part is part of the kmemcg-slab series. In that use case, I copy the flags from the original kmem cache, and create a duplicate. That duplicate need to have the same flags, but only the creation flags.

We had many attempts to mask it out in different places, and after some discussion, it seemed best to independently do it from common code in slab\_common.c at creation time. It gets quite independent from the kmemcg-slab this way, and so I posted independently to reduce my churn

---