Subject: Re: [PATCH v2 04/29] slub: always get the cache from its page in kfree
Posted by Glauber Costa on Fri, 11 May 2012 19:24:11 GMT
View Forum Message <> Reply to Message

On 05/11/2012 04:20 PM, Christoph Lameter wrote:
> On Fri, 11 May 2012, Glauber Costa wrote:
>
>>> I see that. But there are other subsystems from slab allocators that do
>>> the same. There are also objects that may be used by multiple processes.
>>
>> This is also true for normal user pages. And then, we do what memcg does:
>> first one to touch, gets accounted. I don't think deviating from the memcg
>> behavior for user pages makes much sense here.
>>
>> A cache won't go away while it still have objects, even after the memcg is
>> removed (it is marked as dead)
>
> Ok so we will have some dead pages around that are then repatriated to
> the / set?

No, they are not repatriated. I actually wrote code for that once in my
first series, but it was the general feeling at the time that it was too
complicated. (and I only tried for the slub, not slab)

So instead, we just keep the cache around, until the objects go away.
It will show in slabinfo as dentry(css_id:memcgname)dead

For the record, I wrote that code because I found a nice feature, but I
totally agree with the complicated part.

Also, in normal scenarios, dead caches are not expected to be common.
Most of them should go away as memcg dies.

>>> Hmmm.. Would be better to have a hierachy there. /proc/slabinfo is more
>>> legacy.
>>
>> I can take a look at that then. Assuming you agree with all the rest, is
>> looking into that a pre-requisite for merging, or is something that can be
>> deferred for a phase2 ? (We still don't do shrinkers, for instance, so this is
>> sure to have a phase2)
>
> Not a prerequisite for merging but note that I intend to rework the
> allocators to extract common code so that they have the same sysfs
> interface, error reporting and failure scenarios. We can at that time
> also add support for /sys/kernel/slab to memcg. (/sys/memcg/<name>/slab/* ?)

Yes, that would be a good plan.