Subject: Re: [PATCH v3 18/28] slub: charge allocation to a memcg
Posted by Glauber Costa on Tue, 29 May 2012 16:06:45 GMT
View Forum Message <> Reply to Message

On 05/29/2012 06:51 PM, Christoph Lameter wrote:
> On Fri, 25 May 2012, Glauber Costa wrote:
>
>> This patch charges allocation of a slab object to a particular
>> memcg.
>
> I am wondering why you need all the other patches. The simplest approach
> would just to hook into page allocation and freeing from the slab
> allocators as done here and charge to the currently active cgroup. This
> avoids all the duplication of slab caches and per node as well as per cpu
> structures. A certain degree of fuzziness cannot be avoided given that
> objects are cached and may be served to multiple cgroups. If that can be
> tolerated then the rest would be just like this patch which could be made
> more simple and non intrusive.
>
Just hooking into the page allocation only works for caches with very
big objects. For all the others, we need to relay the process to the
correct cache.

Some objects may be shared, yes, but in reality most won't.

Let me give you an example:

We track task_struct here. So as a nice side effect of this, a fork bomb
will be killed because it will not be able to allocate any further.

But if we're accounting only at page allocation time, it is quite
possible to come up with a pattern while I always let other cgroups
pay the price for the page, but I will be the one filling it.

Having an eventual dentry, for instance, shared among caches, is okay.
But the main use case is for process in different cgroups dealing with
totally different parts of the filesystem.

So we can't really afford to charge to the process touching the nth
object where n is the number of objects per page. We need to relay it to
the right one.