Subject: Re: [PATCH v3 00/28] kmem limitation for memcg
Posted by Frederic Weisbecker on Thu, 07 Jun 2012 14:00:45 GMT
View Forum Message <> Reply to Message

On Thu, Jun 07, 2012 at 02:53:07PM +0400, Glauber Costa wrote:
> On 06/07/2012 02:26 PM, Frederic Weisbecker wrote:
> >On Fri, May 25, 2012 at 05:03:20PM +0400, Glauber Costa wrote:
> >>Hello All,
> >>
> >>This is my new take for the memcg kmem accounting. This should merge
> >>all of the previous comments from you, plus fix a bunch of bugs.
> >>
> >>At this point, I consider the series pretty mature. Since last submission
> >>2 weeks ago, I focused on broadening the testing coverage. Some bugs were
> >>fixed, but that of course doesn't mean no bugs exist.
> >>
> >>I believe some of the early patches here are already in some trees around.
> >>I don't know who should pick this, so if everyone agrees with what's in here,
> >>please just ack them and tell me which tree I should aim for (-mm? Hocko's?)
> >>and I'll rebase it.
> >>
> >>I should point out again that most, if not all, of the code in the caches
> >>are wrapped in static_key areas, meaning they will be completely patched out
> >>until the first limit is set. Enabling and disabling of static_keys incorporate
> >>the last fixes for sock memcg, and should be pretty robust.
> >>
> >>I also put a lot of effort, as you will all see, in the proper separation
> >>of the patches, so the review process is made as easy as the complexity of
> >>the work allows to.
> >
> >So I believe that if I want to implement a per kernel stack accounting/limitation,
> >I need to work on top of your patchset.
> >
> >What do you think about having some sub kmem accounting based on the caches?
> >For example there could be a specific accounting per kmem cache.
> >
> >Like if we use a specific kmem cache to allocate the kernel stack
> >(as is done by some archs but I can generalize that for those who want
> >kernel stack accounting), allocations are accounted globally in the memcg as
> >done in your patchset but also on a seperate counter only for this kmem cache
> >on the memcg, resulting in a kmem.stack.usage somewhere.
> >
> >The concept of per kmem cache accounting can be expanded more for any
> >kind of finegrained kmem accounting.
> >
> >Thoughts?
>
> I believe a general separation is too much, and will lead to knob

> explosion. So I don't think it is a good idea.

Right. This could be an option in kmem_cache_create() or something.

>
> Now, for the stack itself, it can be justified. The question that
> remains to be answered is:
>
> Why do you need to set the stack value separately? Isn't accounting
> the stack value, and limiting against the global kmem limit enough?

Well, I may want to let my container have a full access to some kmem
resources (net, file, etc...) but defend against fork bombs or NR_PROC
rlimit exhaustion of other containers.

So I need to be able to set my limit precisely on kstack.