Subject: Re: [PATCH v3 04/13] kmem accounting basic infrastructure
Posted by Glauber Costa on Wed, 26 Sep 2012 22:29:06 GMT
View Forum Message <> Reply to Message

On 09/27/2012 02:10 AM, Tejun Heo wrote:
> Hello, Glauber.
>
> On Thu, Sep 27, 2012 at 01:24:40AM +0400, Glauber Costa wrote:
>> "kmem_accounted" is not a switch. It is an internal representation only.
>> The semantics, that we discussed exhaustively in San Diego, is that a
>> group that is not limited is not accounted. This is simple and consistent.
>>
>> Since the limits are still per-cgroup, you are actually proposing more
>> user-visible complexity than me, since you are adding yet another file,
>> with its own semantics.
>
> I was confused. I thought it was exposed as a switch to userland (it
> being right below .use_hierarchy tripped red alert).

Remember I was the one more vocally and radically so far trying to get rid of use_hierarchy. I should have been more clear - and I was, as soon as I better understood the nature of your opposition - but this is precisely what I meant by "inherently different".

>
> So, the proposed behavior is to allow enabling kmemcg anytime but
> ignore what happened inbetween? Where the knob is changes but the
> weirdity seems all the same. What prevents us from having a single
> switch at root which can only be flipped when there's no children?

So I view this very differently from you. We have no root-only switches in memcg. This would be a first, and this is the kind of thing that adds complexity, in my view.

You have someone like libvirt or a systemd service using memcg. It probably starts at boot. Once it is started, it will pretty much prevent switching of any global switch like this.

And then what? If you want a different behavior you need to go kill all your services that are using memcg so you can get the behavior you want? And if they happen to be making a specific flag choice by design, you just say "you really can't run A + B together" ?

I myself think global switches are an unnecessary complication. And let us not talk about use_hierarchy, please. If it becomes global, it is going to be as part of a phase out plan anyway. The problem with that is not that it is global, is that it shouldn't even exist.

>
> Backward compatibility is covered with single switch and I really
> don't think "you can enable limits for kernel memory anytime but we
> don't keep track of whatever happened before it was flipped the first
> time because the first time is always special" is a sane thing to
> expose to userland. Or am I misunderstanding the proposed behavior
> again?
>

You do keep track. Before you switch it for the first time, it all
belongs to the root memcg.
