Subject: Re: [PATCH v3 04/13] kmem accounting basic infrastructure
Posted by Tejun Heo on Mon, 01 Oct 2012 00:57:17 GMT
View Forum Message <> Reply to Message

Hello, James.

On Sun, Sep 30, 2012 at 12:25:52PM +0100, James Bottomley wrote:
> But you've got to ask yourself who cares about accurate accounting per
> container of dentry and inode objects? They're not objects that any
> administrator is used to limiting. What we at parallels care about
> isn't accurately accounting them, it's that one container can't DoS
> another by exhausting system resources. That's achieved equally well by
> first charge slab accounting, so we don't really have an interest in
> pushing object accounting code for which there's no use case.

Isn't it more because the use cases you have on mind don't share
dentries/inodes too much? Wildly incorrect accounting definitely
degrades container isolation and can lead to unexpected behaviors.

> All we need kernel memory accounting and limiting for is DoS prevention.
> There aren't really any system administrators who care about Kernel
> Memory accounting (at least until the system goes oom) because there are
> no absolute knobs for it (all there is are a set of weird and wonderful
> heuristics, like dirty limit ratio and drop caches). Kernel memory

I think that's because the mechanism currently doesn't exist. If one
wants to control how memory is distributed across different cgroups,
it's logical to control kernel memory too. The resource in question
is the actual memory after all. I think at least google would be
interested in it, so, no, I don't agree that nobody wants it. If that
is the case, we're working towards the wrong direction.

> usage has a whole set of regulatory infrastructure for trying to make it
> transparent to the user.
>
> Don't get me wrong: if there were some easy way to get proper memory
> accounting for free, we'd be happy but, because it has no practical
> application for any of our customers, there's a limited price we're
> willing to pay to get it.

Even on purely technical ground, it could be that first-use is the
right trade off if other more accurate approaches are too difficult
and most workloads are happy with such approach. I'm still a bit
weary to base userland interface decisions on that tho.

Thanks.

--

tejun