Subject: Re: [PATCH v3 04/13] kmem accounting basic infrastructure
Posted by Tejun Heo on Sun, 30 Sep 2012 10:37:32 GMT
View Forum Message <> Reply to Message

Hello, James.

On Sun, Sep 30, 2012 at 09:56:28AM +0100, James Bottomley wrote:
> The beancounter approach originally used by OpenVZ does exactly this.
> There are two specific problems, though, firstly you can't count
> references in generic code, so now you have to extend the cgroup
> tentacles into every object, an invasiveness which people didn't really
> like.

Yeah, it will need some hooks. For dentry and inode, I think it would be pretty well isolated tho. Wasn't it?

> Secondly split accounting causes oddities too, like your total
> kernel memory usage can appear to go down even though you do nothing
> just because someone else added a share. Worse, if someone drops the
> reference, your usage can go up, even though you did nothing, and push
> you over your limit, at which point action gets taken against the
> container. This leads to nasty system unpredictability (The whole point
> of cgroup isolation is supposed to be preventing resource usage in one
> cgroup from affecting that in another).

In a sense, the fluctuating amount is the actual resource burden the cgroup is putting on the system, so maybe it just needs to be handled better or maybe we should charge fixed amount per refcnt? I don't know.

> We discussed this pretty heavily at the Containers Mini Summit in Santa
> Rosa. The emergent consensus was that no-one really likes first use
> accounting, but it does solve all the problems and it has the fewest
> unexpected side effects.

But that's like fitting the problem to the mechanism. Maybe that is the best which can be done, but the side effect there is way-off accounting under pretty common workload, which sounds pretty nasty to me.

Thanks.

--
tejun