Subject: Re: [PATCH v3 04/13] kmem accounting basic infrastructure
Posted by Glauber Costa on Wed, 26 Sep 2012 17:36:36 GMT
View Forum Message <> Reply to Message

On 09/26/2012 08:36 PM, Tejun Heo wrote:
> Hello, Michal, Glauber.
>
> On Wed, Sep 26, 2012 at 04:03:47PM +0200, Michal Hocko wrote:
>> Haven't we already discussed that a new memcg should inherit kmem_accounted
>> from its parent for use_hierarchy?
>> Say we have
>> root
>> |
>> A (kmem_accounted = 1, use_hierachy = 1)
>> \
>>   B (kmem_accounted = 0)
>>   \
>>   C (kmem_accounted = 1)
>>
>> B find's itself in an awkward situation becuase it doesn't want to
>> account u+k but it ends up doing so becuase C.
>
> Do we really want this level of flexibility? What's wrong with a
> global switch at the root? I'm not even sure we want this to be
> optional at all. The only reason I can think of is that it might
> screw up some configurations in use which are carefully crafted to
> suit userland-only usage but for that isn't what we need a transition
> plan rather than another ultra flexible config option that not many
> really understand the implication of?
>
> In the same vein, do we really need both .kmem_accounted and config
> option? If someone is turning on MEMCG, just include kmem accounting.
>

Yes, we do.

This was discussed multiple times. Our interest is to preserve existing
deployed setup, that were tuned in a world where kmem didn't exist.
Because we also feed kmem to the user counter, this may very well
disrupt their setup.

User memory, unlike kernel memory, may very well be totally in control
of the userspace application, so it is not unreasonable to believe that
extra pages appearing in a new kernel version may break them.

It is actually a much worse compatibility problem than flipping
hierarchy, in comparison