Subject: Re: [PATCH 0/7] memcg kernel memory tracking
Posted by Suleiman Souhlal on Tue, 21 Feb 2012 23:25:18 GMT
View Forum Message <> Reply to Message

Hi Glauber,

On Tue, Feb 21, 2012 at 3:34 AM, Glauber Costa <glommer@parallels.com> wrote:
> This is a first structured approach to tracking general kernel
> memory within the memory controller. Please tell me what you think.

Thanks for posting these.

> As previously proposed, one has the option of keeping kernel memory
> accounted separatedly, or together with the normal userspace memory.
> However, this time I made the option to, in this later case, bill
> the memory directly to memcg->res. It has the disadvantage that it becomes
> complicated to know which memory came from user or kernel, but OTOH,
> it does not create any overhead of drawing from multiple res_counters
> at read time. (and if you want them to be joined, you probably don't care)

It would be nice to still keep a kernel memory counter (that gets
updated at the same time as memcg->res) even when the limits are not
independent, because sometimes it's important to know how much kernel
memory is being used by a cgroup.

> Kernel memory is never tracked for the root memory cgroup. This means
> that a system where no memory cgroups exists other than the root, the
> time cost of this implementation is a couple of branches in the slub
> code - none of them in fast paths. At the moment, this works only
> with the slub.
>
> At cgroup destruction, memory is billed to the parent. With no hierarchy,
> this would mean the root memcg. But since we are not billing to that,
> it simply ceases to be tracked.
>
> The caches that we want to be tracked need to explicit register into
> the infrastructure.

Why not track every cache unless otherwise specified? If you don't,
you might end up polluting code all around the kernel to create
per-cgroup caches.
>From what I've seen, there are a fair amount of different caches that
can end up using a significant amount of memory, and having to go
around and explicitly mark each one doesn't seem like the right thing
to do.

-- Suleiman