Subject: Re: [PATCH v5 00/14] kmem controller for memcg.
Posted by akpm on Wed, 17 Oct 2012 22:11:42 GMT
View Forum Message <> Reply to Message

On Tue, 16 Oct 2012 14:16:37 +0400
Glauber Costa <glommer@parallels.com> wrote:

> ...
>
> A general explanation of what this is all about follows:
>
> The kernel memory limitation mechanism for memcg concerns itself with
> disallowing potentially non-reclaimable allocations to happen in exaggerate
> quantities by a particular set of processes (cgroup). Those allocations could
> create pressure that affects the behavior of a different and unrelated set of
> processes.
>
> Its basic working mechanism is to annotate some allocations with the
> _GFP_KMEMCG flag. When this flag is set, the current process allocating will
> have its memcg identified and charged against. When reaching a specific limit,
> further allocations will be denied.

The need to set _GFP_KMEMCG is rather unpleasing, and makes one wonder
"why didn't it just track all allocations".

Does this mean that over time we can expect more sites to get the
_GFP_KMEMCG tagging? If so, are there any special implications, or do
we just go in, do the one-line patch and expect everything to work? If
so, why don't we go in and do that tagging right now?

And how *accurate* is the proposed code? What percentage of kernel
memory allocations are unaccounted, typical case and worst case?

All sorts of questions come to mind over this decision, but it was
unexplained. It should be, please. A lot!

>
> ...
>
> Limits lower than
> the user limit effectively means there is a separate kernel memory limit that
> may be reached independently than the user limit. Values equal or greater than
> the user limit implies only that kernel memory is tracked. This provides a
> unified vision of "maximum memory", be it kernel or user memory.
>

I'm struggling to understand that text much at all. Reading the
Documentation/cgroups/memory.txt patch helped.