Subject: Re: [PATCH v5 14/14] Add documentation about the kmem controller
Posted by Glauber Costa on Tue, 16 Oct 2012 19:02:46 GMT
View Forum Message <> Reply to Message

On 10/16/2012 10:25 PM, Christoph Lameter wrote:
> On Tue, 16 Oct 2012, Glauber Costa wrote:
>
>>
>> + memory.kmem.limit_in_bytes   # set/show hard limit for kernel memory
>> + memory.kmem.usage_in_bytes   # show current kernel memory allocation
>> + memory.kmem.failcnt   # show the number of kernel memory usage hits limits
>> + memory.kmem.max_usage_in_bytes # show max kernel memory usage recorded
>
> Does it actually make sense to limit kernel memory?

Yes.

> The user generally has
> no idea how much kernel memory a process is using and kernel changes can
> change the memory footprint. Given the fuzzy accounting in the kernel a
> large cache refill (if someone configures the slab batch count to be
> really big f.e.) can account a lot of memory to the wrong cgroup. The
> allocation could fail.
>

It heavily depends on the type of the user. The user may not know how
much kernel memory precisely will be used, but he/she usually knows
quite well that it shouldn't be all cgroups together shouldn't use more
than available in the system.

IOW: It is usually safe to overcommit user memory, but not kernel
memory. This is absolutely crucial in any high-density container host,
and we've been doing this in OpenVZ for ages (in an uglier form than this)

> Limiting the total memory use of a process (U+K) would make more sense I
> guess. Only U is probably sufficient? In what way would a limitation on
> kernel memory in use be good?
>

The kmem counter is also fed into the u counter. If the limit value of
"u" is equal or greater than "k", this is actually what you are doing.

For a lot of application yes, only U is sufficient. This is the default,
btw, since "k" is only even accounted if you set the limit.

All those use cases are detailed a bit below in this file.

A limitation of kernel memory use would be good, for example, to prevent

abuse from non-trusted containers in a high density, shared, container environment.