Subject: Re: Re: [patch -mm 08/17] nsproxy: add hashtable
Posted by Cedric Le Goater on Wed, 13 Dec 2006 15:09:27 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Kirill Korotaev <dev@sw.ru> writes:
>
>>> I think what those projects need is _some_ way to group tasks. I'm not
>>> sure they actually need nsproxies.
>>>
>>> Two tasks in the same container could very well have different
>>> nsproxies.
>> what is container then from your POV?
>
> A nested instance of user space. User space may unshare things
> such as the mount namespace so it can give users the ability to
> control their own mounts and the like.
>
>>> The nsproxy defines how the pid namespace, and pid<->task
>>> mappings happen for a given task. The init process for a container is
>>> special and might actually appear in more than one pid namespace, while
>>> its children might only appear in one. That means that this init
>>> process's nsproxy can and should actually be different from its
>>> children's. This is despite the fact that they are in the same
>>> container.
>> nsproxy has references to all namespaces, not just pid namespace.
>> Thus it is a container "view" effectively.
>> If container is something different, then please define it.
>
> nsproxy has exactly one instance of all namespaces. A container
> in the general case can hold other containers, and near containers
> (like processes with separate mount namespaces). As well as
> processes.
>
> So nsproxy currently captures the common case for containers but not
> the general case.

i don't see the difference. honestly. what is the general case ? the full
system holding all the instances of all the namespaces ? but that's not
usable by user space. you need only one instance of all namespaces, not all.

I don't get it. we don't want the general case to be exposed to user space
we only want one instance == nsproxy.

C.