
Subject: Re: [patch -mm 08/17] nsproxy: add hashtable

Posted by [dev](#) on Tue, 12 Dec 2006 08:43:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

>>>Even letting the concept of nsproxy escape to user space sounds wrong.

>>>nsproxy is an internal space optimization. It's not struct container

>>>and I don't think we want it to become that.

>>

>>i don't agree here. we need that, so does openvz, vserver, people working

>>on resource management.

>

>

> 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?

> 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.

> If we really need this 'container' grouping, it can easily be something

> pointed to by the nsproxy, but it shouldn't be the nsproxy.

You can add another indirection if really want it so much...

But is it required?

We created nsproxy which adds another level of indirection, but from performance POV

it is questionable. I can say that we had a nice experience, when adding

a single dereference in TCP code resulted in ~0.5% performance degradation.

Thanks,

Kirill

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
