Subject: Re: [RFC][PATCH 2/7] VPIDs: pid/vpid conversions
Posted by dev on Mon, 20 Feb 2006 14:55:42 GMT
View Forum Message <> Reply to Message

> Do you know how incomplete this patch is?
> You missed drivers/char/drm, and in your shipping OpenVZ patch.
> You missed get_xpid() on alpha.
> You missed nfs.
DRM/NFS code is correct.

The only correct thing you noticed is get_xpid on alpha. But this is in fact a simple bug and half a year before we didn't care much for archs others than i386/x86-64/ia64. That's it.

> I suspect the tagging of the VPIDS and the WARN_ON's help so you have
> a chance of catching things if someone uses a code path you haven't
> caught. But I don't see how you can possibly get full kernel
> coverage.
simple, the same way as you did, i.e. by renaming pid to tid or something like this.

> Is there a plan to catch all of the in-kernel use of pids that I am
> being to dense to see?
if Linus will be ready to take it into mainstream, it will be caught all. Actually only asm files should be investigated due to optimizations similar to those on IA64/Alpha. Everything else I suppose is correct and can be rechecked only.

And now a bit of contstructive ideas/things:

I propose to stop VPIDs discussion and switch to virtualization of networking, IPC and so on, which is essentially the same in yours and our solutions (openvz).

I took a look to your patch, it does actually the same things as openvz, almost thing by thing. But it is BUGGY! You have broken IPC/networking, many things to these subsytems are not virtualized etc. We need to get Linus comment about which approach is the best for him, with namespace pointers on task_struct involved by you or with effective container pointer. It is only a matter of his taste, but the result is effectively the same. Agree?

Actually we don't care whether virtualization introduces one container pointer on the task struct or as you proposed many pointers to namespaces. But you are WRONG IMHO thinking that this namespaces are independent and this allows you more fine grained virtualization. All these namespaces are tightly intergrated with each other(sic!).
For example, networking is coupled with sysctl, which in turn are

coupled with proc filesystem. And sysfs! You even added a piece of code
in net/core/net-sysfs.c in your patch, which is a dirty hack.
Another example, mqueues and other subsystems which use netlinks and
also depend on network context.
shmem/IPC is dependand on file system context and so on.
So it won't work when one have networking from one container and proc
from another. So I really see no much reasons to have separate
namespaces, but it is ok for me if someone really wants it this way.

We also don't care whether yours or our network virtualization will go
upstream. They do _exactly_ the same. You also virtualized IPv6 which is
good, since we have only IPv4, but you totally missed netfilters, which
is bad :) So again the only difference is that we have effective
container on the task, while you prefer to take it from sk/netdev or
bypass as an additional function argument.

So I propose the following:
1. ask Linus about the preffered approach. I prepared an email for him
with a description of approaches.
2. start from networking/netfilters/IPC which are essentially the same
in both projects and help each other.

Kirill