Subject: Re: [RFC][PATCH 5/7] VPIDs: vpid/pid conversion in VPID enabled case
Posted by Cedric Le Goater on Tue, 07 Feb 2006 09:15:12 GMT
View Forum Message <> Reply to Message

Bonjour!

Alexey Kuznetsov wrote:

> [...]
>
> We could force each process visiting container to daemonize and to setsid().
> But do not forget that pid space is just a little part of the whole engine,
> to force full isolation we have to close all the files opened
> in root container, to get rid of its memory space before entering container
> etc. But it makes not so much of sense, because in any case we have to keep
> at least some way to communicate to host. F.e. even when we allow to pass
> only an open pipe, we immediately encounter the situation when a file
> owned by one container could refer to processes of another container.
>
> So that, the only way to enforce full isolation is to prohibit
> "vzctl exec/enter" as whole.

containers are useful, even without migration. No doubt on that.

But, at the end, long long term probably, if we want to have a mobile
container under linux, we need to address all the issues from the start and
take them into account in the design. So, if we need to add some
constraints on the container init process (child reaper) or the resource
isolation, pid for example, to make sure a container is migratable, I think
we should start to think about it now.

By the time we reach that state, openvz would be have been rewritten a few
times already like any good software. nope ? :)

>>We've been living with the vpid approach also for years and we found issues
>>that we haven't solve at restart. So we think we might do a better job with
>>another. But, this still needs to be confirmed :)
>
> What are the issues?

The one above.

Having containers which are not migratable because their execution
environment was not contrained enough is a pity I think.

Containers are useful for isolation but being able to swsuspend them and
migrate them is even more interesting ! and fun.

> The only inconvenience which I encountered until now
> is a little problem with stray pids. F.e. this happens with flock().
> Corresponding kernel structure contains some useless (actually, illegal
> and contradicting to the nature of flock()) reference to pid.
> If the process took the lock and exited, stray pid remains forever and points
> to nowhere. In this case it is silly to prohibit checkpointing,
> but we have to restore the flock to a lock with pointing to the same point
> in the sky, i.e. to nowhere. With (container, pid) approach we would
> restore it pointing to exactly the same empty place in the sky, with
> vpids we have to choose a new place. Ugly, but not a real issue.

thanks for your insights ! I hope we will have plenty of these issues to
talk about.

c.
