Subject: Re: [RFC][PATCH] VPIDs: Virtualization of PIDs (OpenVZ approach)
Posted by Kirill Korotaev on Fri, 03 Feb 2006 10:30:05 GMT
View Forum Message <> Reply to Message

> well, IMHO this approach lacks a few things which would
> be very useful in a mainline pid virtualization, which
> pretty much explains why it is relatively small
>
> - hierarchical structure (it's flat, only one level)

PID virtualization has nothing to do with containers and its structure.
This VPID patch can be used both in environments with hierarchical and
flat structure.
I mean that this VPID patch introduces some kind of abstraction, which
means that global unique pid is not always what user sees. Thats it.
And kernel should not be modified in all places where access checks
current->pid == allowed_pid are done. Global unique PIDs are preserved.

> - a proper administration scheme

Can't catch what you mean. What kind of administration do you want for
VPIDs? Do you have one for usual PIDs? It is assigned by kernel and
that's it.

> - a 'view' into the child pid spaces

it has. see proc patch.

> - handling of inter context signalling

How is it related to inter context signalling???
virtualization of signaling is a separate task and has nothing to do
with pids.

> and, more important, it does not deal with the existing
> issues and error cases, where references to pids, tasks,
> task groups and sessions aren't handled properly ...

1. if kernel has some errors, these errors should be fixed.
Virtualization doesn't deal with it, doesn't solve such issues, doesn't
make it worse. So what do you mean?
2. if kernel has some issues and error cases can you point them to me? I
will be glad to fix it. Without pointing to the facts your words sound
like a pure speculation. What issues? What error cases? Where task
groups and sessiona aren't handled properly?

> I think that in real world virtualization scenarios
> with hundreds of namespaces those 'imprecisions' will
> occasionally lead to very strange and random behaviour
> which in many cases will go completely unnoticed.

OpenVZ successfully works with >1000 VPSs on a single server.
What scenarios do you mean?
I see no arguments from your side except for some guesses.

> so I really prefer to cleanup the existing pid handling
> first, to avoid big surprises later ...

I'm not against pid handling cleanups, am I?
This can be done in parallel/before/after.

Kirill