

Hello!

> - hierarchical structure (it's flat, only one level)

Er... if I understand the question correctly, it is not flat,
it is rather universal.

The main idea of the VPID approach is that it preserves existing
logic completely, all the references, interprocess linkage and process
grouping are made using usual unique pids.

Virtual pids are just the numbers, which user level applications see.
Of course, they depend on the point where you view from.

> - a proper administration scheme

There is nothing to administrate. pids are generated and assigned
automatically.

Virtual pids come to the game only when you move a process from
one host to another, new process is created with `do_fork_pid()`,
which assigns the same virtual pid and generates brand new global pid,
unique for destination host.

> - a 'view' into the child pid spaces

In order to access a virtual pid space the process must enter
the corresponding container. You are not going to add the whole set
of syscalls, which use not only pid but also a container identifier, right?

For purpose of debugging, vpids are shown `/proc/*/status`, it is enough.

> - handling of inter context signalling

Not an issue. All the inter-container communication is made using
global pids.

It is necessary to emphasize again, VPID patch preserves currently
existing capabilities: all the processes in all the containers
can be accessed by global pids. F.e. plain "ps axf" executed in "initial
container" shows the whole process tree.

It is one of main points: "virtual" pids are not used to `_limit_` access, is not a "no go" thing, it allows to go on when we have to reinstate VPS at another host. Access checks are not based on some pids, they are defined by policy rules, which are checked after corresponding objects (tasks in our case) are found.

- > 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 ...

I am sorry, I am not aware about such bugs (expect for a few well-known cases, when someone holds a pid, which can remain stray and be recycled. But it is apparently orthogonal to the problem under discussion). If they do exist they apparently must be fixed asap not depending on anything else.

Alexey
