Subject: Re: [PATCH 1/4] Virtualization/containers: introduction
Posted by dev on Wed, 08 Feb 2006 15:35:10 GMT
View Forum Message <> Reply to Message

> The pid-namespace (pspace) provides an approach of fully separate
> the allocation and maintenance of the pids and treating the <pspace,pid>
> tuple as an entity to uniquely identify a task and vice versa.
> As a result the logic of lookup can be pushed down the find_task_by_pid()
> lookup. There are specific cases where the init_task of a container or
> pspace needs to be checked to ensure that signals/waits and alike are
> properly
> handled across pspace boundaries. I think this is an intuitive and clean
> way.
> It also completely avoids the problem of having to think about all the
> locations
> at the user/kernel boundary where a vpid/pid conversion needs to take
> place.
> It also avoids the problems that logically vpids and pids are different
> types and
> therefore it would have been good to have type checking automatically
> identify
> problem spots.
> On the negative side, it does require to maintain a pidmap per pidspace.

Additional negative sides:
- full isolation can be inconvinient from containers management point of
view. You will need to introduce new modified tools such as top/ps/kill
and many many others. You won't be able to strace/gdb processes from the
host also.
- overhead when virtualization is off, result is not the same.
- additional args everywhere (stack usage, etc.)

> The vpid approach has the drawbacks of having to identify the conversion
> spots
> of all vpid vs. pid semantics. On the otherhand it does take advantage
> of the fact that no virtualization has to take place until a "container"
> has been migrated, thus rendering most of the vpid<->pid calls to be
> noops.

It has some other additional advantages:
- flexible: you can select full isolation or weak is required. I really
believe this is very important.

> The container is just an umbrella object that ties every "virtualized"
> subsystem
> together.

Yep. And containers were what I wanted to start with actually. Not VPIDs.

Kirill