Subject: Re: [PATCH 0/9] namespaces: Introduction
Posted by ebiederm on Fri, 19 May 2006 11:41:45 GMT
View Forum Message <> Reply to Message

Andrew Morton <akpm@osdl.org> writes:

> All of which begs the question "now what?".

I think we are at the point where it is time to start merging patches
into -mm, and having the discussion on what the merge plans are
for the rest of this code.

> What we do _not_ want to do is to merge up a pile of infrastructural stuff
> which never gets used.  On the other hand, we don't want to be in a
> position where nothing is merged into mainline until the entirety of
> vserver &&/|| openvs is ready to be merged.

The namespaces I see needed for a useable result are:
- fs namespace (already merged)
- uts namespace
- sysvipc namespace
- time namespace
- uid/gid (keys?) namespace
- network namespace
- pid namespace

> I see two ways of justifying a mainline merge of things such as this
>
> a) We make an up-front decision that Linux _will_ have OS-virtualisation
>    capability in the future and just start putting in place the pieces for
>    that, even if some of them are not immediately useful.
>
>    I suspect that'd be acceptable, although I worry that we'd get
>    partway through and some issues would come up which are irreconcilable
>    amongst the various groups.

I think I see a third way of justifying a mainline merge.  We make an
up-front decision that we will improve the existing chroot jail
functionality in Linux and start making improvements.  Even if some of
the improvements are quite small.

Except for partial steps while the code is being refactored, we should
never have steps that are not immediately useful.

This reduces the danger of irreconcilable differences, because being
part way through is still useful.

The only namespace that I see as really contentious is the pid
namespace, and even there I don't think we have read an impasse.
There remains a bunch of patches left to write that replace raw pid_t
values with struct pid references, but once that happens the patches
to implement the pid namespace will be small, and I don't see any
previous problems that we can't resolve when the conversation happens.

> It would help set minds at ease if someone could produce a
> bullet-point list of what features the kernel will need to get it to the
> stage where "most or all vserver and openvz functionality can be
> implemented by controlling resource namespaces from userspace." Then we
> can discuss that list, make sure that everyone's pretty much in
> agreement.

So this is slightly the wrong question.  If you look at Sam's list you
will see that there are several independent dimensions to the complete
solution.  Most of them dealing with the increase in the number of users
and the amount of work that is happening on a single kernel in this
context.

Basically we need to expect a lot of kernel tuning after we get the
basics working.

The proper question is: What needs to happen before we can run separate
user space instances?

The namespaces I have previously listed.  There is also a lot of
cleanup work with sysctl, proc, sysfs, netlink and some other
fundamental interfaces that needs to happen as well.  Until each
namespace gets merged we are in a race with other people looking
at enhancing those namespaces.  So a complete of what needs to be
fixed is impossible.

> b) Only merge into mainline those feature which make sense in a
> standalone fashion.  eg, we don't merge this patchset unless the
> "per-process utsname namespace" feature is useful to and usable by a
> sufficiently broad group of existing Linux users.
>
> I suspect this will be a difficult approach.

I agree if the feature must be useful and usable by a sufficiently
broad group of existing Linux users.  Of course I suspect the current
fs namespace fails this test.

I would rather the criteria be, that the functionality that is well
defined and not detrimental to the rest of users.

> The third way would be to buffer it all up in -mm until everything is
> sufficiently in place and then slam it all in.  That might not be feasible
> for various reasons - please advise..

Fundamentally I don't think there are problems buffering things up in -mm,
but I worry that we would start having -mm too different from the
stable kernel at some point.

For some of the pieces like the networking stack we need to go through
the respective maintainers, and their development trees to avoid
conflicts.  For the sysvipc, utsname, and  we have avoided that
because they are absolutely trivial namespaces and they don't have
active maintainers.

> A fourth way would be for someone over there to run a git tree - you all
> happily work away, I redistribute it in -mm for testing and one day it's
> all ready to merge.  I don't really like this approach.  It ends up meaning
> that nobody else reviews the new code, nobody else understands what it's
> doing, etc.  It's generally subversive of the way we do things.

The only part of this picture that might make sense is if we have a
process by which we can decide if patches are good and acceptable to
the various projects independent of deciding if they are good for
the kernel proper, which might take some of the burden off of the
rest of the kernel maintainers.

If we were working in an area of the kernel where we didn't affect
anyone else it would be business as usual and not really subversive.
But since we can't implement things this way I agree that this
code needs to be reviewed as much as possible.

> Eric, Kirill, Herbert: let us know your thoughts, please.

Eric