Subject: Re: [PATCH 0/9] namespaces: Introduction
Posted by Sam Vilain on Thu, 18 May 2006 23:28:51 GMT
View Forum Message <> Reply to Message

Andrew Morton wrote:

>Generally, I think that the whole approach of virtualising the OS so it can
>run multiple independent instances of userspace is a good one.
>[...]
>All of which begs the question "now what?".
>[...]
> 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.
>
>

This is a heartening position to hear from someone such as yourself; we seem to be at a near consensus of the way forward.

Here's a list based on the one I came up with when I originally started my line of development, which got shot down so badly it lost a few priority points on my workqueue scheduler :-).

0. features that don't need namespaces per se

a. Bind Mount Options (mount --bind -o ro, etc)
b. FS - immutable linkage invert (immulink)

1. core vserver patch - no features (this stuff is succeeded by Serge's set)
a. struct and ps addition; internal API and refcounting
b. syscall, and switch (to be canned)
c. /proc visibility
d. debugging
e. history

2. isolation features

a. IPC, semaphore, and signal restrictions
b. proc/array filtering
c. IPv4 chbind
d. FS chroot() barrier
e. general /proc filtering
f. ptrace

g. process admin: alloc_uid, find_user, sys_setpriority

3. virtualisation features

a. uts information
b. initpid virtualisation
c. uptime
d. time
e. load average
f. ksyslog
g. vshelper (reboot support)
h. vroot (quota, fs IOCTL, etc)
i. general PID virtualisation (eric)
j. ngnet (network stack virtualisation)

4. resource tracking features

a. scheduler tracking hook
b. FS namespace counting
c. FS namespace tagging
d. ulimits
e. RSS usage
f. IO - async tracking

5. resource sharing features

a. scheduling v1 - TBF and vavavoom
b. disk scheduler integration
c. RSS limits
d. FS - mad cow

6. resource limit features

a. scheduler
b. rlimits
c. disklimits

7. super whizzy features

a. Namespace checkpointing
b. Namespace migration
c. HA Cluster Computing (think Tandem)

Can anyone see any that are missed?

As far as how it is tested etc, I have no particular preferences, whatever people are happy with. I'll continue to track submissions in the utsl.gen.nz repository:

http://utsl.gen.nz/gitweb/?p=vserver

I'll import Serge's new submission there now.

Sam.