Subject: Re: [ckrm-tech] [PATCH 7/7] containers (V7): Container interface to nsproxy subsystem
Posted by Paul Menage on Sat, 07 Apr 2007 01:36:19 GMT
View Forum Message <> Reply to Message

On 4/6/07, Srivatsa Vaddagiri <vatsa@in.ibm.com> wrote:
>
> I would suggest to split the patches well to the extent possible. For ex: in my
> stack, this is what I tried doing:
>
> - A basic file system which registers itself and lets itself be
> mounted
> - Add tasks file
> - Add container_clone/exit support (for namespace to use)
> - Basic namespace changes to use this filesystem
> - Add a hostname file in every container dir (demonstrates what
> it takes to add a file)
> - Add support for user to mkidr/rmdir (so far the patch stack
> will just show what directrories kernel (i.e namespace) has
> auto-created/destroyed - like /proc)
> - Add attach task support - This shows why attach task is nasty
> - Add another subsystem like cpu accounting and show how it
> leads to unnecessary repetitve lines of code, to avoid which
> - Generalize using a register mechanism, so that we can do
> for_each_subsys()
> - Introduce multi hierarchy support ..so far everything was
> capable of being mounted under a diff hierarchy.
> - Introduce notify_on_release

The progression of patches is already a bit like that - first the simple container filesystem, then cpusets on containers, and then generic containers. I agree that the third patch does add more changes in one patch than I'd ideally want to.

The problem with separating out patches to that extent is that bouncing up and down through a quilt stack of patches becomes a total nightmare when underlying patches have changed sufficiently that they conflict.

I think I'd rather move to that model once there seems to be general consensus from people actively involved in container-like work in the kernel that the structure is about right, and then split into smaller patches that people can examine for the details.

Paul

_____
Containers mailing list
Containers@lists.linux-foundation.org