Subject: Re: [ckrm-tech] [PATCH 7/7] containers (V7): Container interface to nsproxy subsystem
Posted by Paul Menage on Fri, 06 Apr 2007 21:55:54 GMT
View Forum Message <> Reply to Message

On 4/6/07, Pavel Emelianov <xemul@sw.ru> wrote:
>
> Well, it's less ugly than resetting pointer but it's worse from
> performance POV. See, we have an if (xxx) that is true only once
> during system lifetime. This is not that good...

Adding an extra comparison in the container creation path really isn't
a big deal - there are far far more heavyweight things going on in the
same path.

>
> Anyway. When are you going to send your patches with containers
> and what kernel will it be built upon? I'll try to prepare the RSS
> patches soon after this.

This afternoon. Still based on 2.6.20.

Paul

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: Re: [ckrm-tech] [PATCH 7/7] containers (V7): Container interface to nsproxy subsystem
Posted by Srivatsa Vaddagiri on Sat, 07 Apr 2007 01:35:02 GMT
View Forum Message <> Reply to Message

On Fri, Apr 06, 2007 at 02:55:54PM -0700, Paul Menage wrote:
> >Anyway. When are you going to send your patches with containers
> >and what kernel will it be built upon? I'll try to prepare the RSS
> >patches soon after this.
>
> This afternoon. Still based on 2.6.20.

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

For one, readers will be able to appreciate how the complexity of the patch stack is affected because of the various features progressively introduced. For other, it will make the patch stack more readable and (possibly) attact more review, since they will be dealing with smaller/focused changes at a time ..

--
Regards,
vatsa

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers