Subject: Re: [ckrm-tech] [PATCH 0/2] resource control file system - aka containers on top of nsproxy!
Posted by Srivatsa Vaddagiri on Wed, 07 Mar 2007 17:52:57 GMT
View Forum Message <> Reply to Message

On Wed, Mar 07, 2007 at 09:29:12AM -0800, Paul Menage wrote:
> That seems bad. With the current way you're doing it, if I mount
> hierarchies A and B on /mnt/A and /mnt/B, then initially all tasks are
> in /mnt/A/tasks and /mnt/B/tasks. If I then create /mnt/A/foo and move
> a process into it, that process disappears from /mnt/B/tasks, since
> its nsproxy no longer matches the nsproxy of B's root container. Or am
> I missing something?

I realized that bug as I was doing cpuset conversion.

Basically, we can't use just tsk->nsproxy to find what tasks are in
a directory (/mnt/B for ex). Here's what I was think we should be doing instead:

```
struct nsproxy *ns;
void *data;

ns = dentry_of(/mnt/B/tasks)->d_parent->d_fsdata;
data = ns->ctlr_data[some subsystem id which is bound in /mnt/B hierarchy]
```

we now scan tasklist and find a match if:

```
 tsk->nsproxy->ctlr_data[the above id] == data
```

(maybe we need to match on all data from all subsystems bound to B)

There is a similar bug in rcfs_rmdir also. We can't just use the nsproxy
pointed to by dentry to know whether the resource objects are free or
not. I am thinking (if at all resource control has to be provided on top
of nsproxy) that we should have a get_res_ns, similar to get_mnt_ns or
get_uts_ns, which will track number of nsproxies pointing to the same
resource object. If we do that, then rmdir() needs to go and check
those resource object's refcounts to see if a dir is in use or not.

--
Regards,
vatsa

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers