
Subject: Re: [PATCH 0/2] resource control file system - aka containers on top of nsproxy!

Posted by [Srivatsa Vaddagiri](#) on Fri, 09 Mar 2007 16:34:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Wed, Mar 07, 2007 at 01:20:18PM -0800, Paul Menage wrote:

> On 3/7/07, Serge E. Hallyn <serue@us.ibm.com> wrote:

> >

> >All that being said, if it were going to save space without overly

> >complicating things I'm actually not opposed to using nsproxy, but it

>

> If space-saving is the main issue, then the latest version of my

> containers patches uses just a single pointer in the task_struct, and

> all tasks in the same set of containers (across all hierarchies) will

> share a single container_group object, which holds the actual pointers

> to container state.

Paul,

Some more thoughts, mostly coming from the point of view of vservers/containers/"whatever is the set of tasks sharing a nsproxy is called".

1. What is the fundamental unit over which resource-management is applied? Individual tasks or individual containers?

/me thinks latter. In which case, it makes sense to stick resource control information in the container somewhere. Just like when controlling a user's resource consumption, 'struct user_struct' may be a natural place to put these resource limits.

2. Regarding space savings, if 100 tasks are in a container (I dont know what is a typical number) -and- lets say that all tasks are to share the same resource allocation (which seems to be natural), then having a 'struct container_group *' pointer in each task_struct seems to be not very efficient (simply because we dont need that task-level granularity of managing resource allocation).

3. This next leads me to think that 'tasks' file in each directory doesnt make sense for containers. In fact it can lend itself to error situations (by administrator/script mistake) when some tasks of a container are in one resource class while others are in a different class.

Instead, from a containers pov, it may be usefull to write a 'container id' (if such a thing exists) into the tasks file which will move all the tasks of the container into the new resource class. This is the same requirement we discussed long back of moving all threads of a process into new

resource class.

--

Regards,
vatsa

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