Subject: Re: [PATCH 0/2] resource control file system - aka containers on top of nsproxy!
Posted by Paul Menage on Fri, 09 Mar 2007 22:09:35 GMT
View Forum Message <> Reply to Message

On 3/9/07, Srivatsa Vaddagiri <vatsa@in.ibm.com> wrote:

```
>
> 1. What is the fundamental unit over which resource-management is
> applied? Individual tasks or individual containers?
>
>       /me thinks latter.
```

Yes

```
> In which case, it makes sense to stick
>       resource control information in the container somewhere.
```

Yes, that's what all my patches have been doing.

```
> 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).
```

I think you should re-read my patches.

Previously, each task had N pointers, one for its container in each
potential hierarchy. The container_group concept means that each task
has 1 pointer, to a set of container pointers (one per hierarchy)
shared by all tasks that have exactly the same set of containers (in
the various different hierarchies).

It doesn't give task-level granularity of resource management (unless
you create a separate container for each task), it just gives a space
saving.

```
>
> 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.

I think you need to give a more concrete example and use case of what you're trying to propose here. I don't really see what advantage you're getting.

Paul

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