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

Posted by [Srivatsa Vaddagiri](#) on Thu, 08 Mar 2007 11:30:54 GMT

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

On Thu, Mar 08, 2007 at 01:50:01PM +1300, Sam Vilain wrote:

> 7. resource namespaces

It should be. Imagine giving 20% bandwidth to a user X. X wants to divide this bandwidth further between multi-media (10%), kernel compilation (5%) and rest (5%). So,

> Is the subservient namespace's resource usage counting against ours too?

Yes, the resource usage of children should be accounted when capping parent resource usage.

> Can we dynamically alter the subservient namespace's resource allocations?

Should be possible yes. That lets user X completely manage his allocation among whatever sub-groups he creates.

> So let's bring this back to your patches. If they are providing
> visibility of ns_proxy, then it should be called namesfs or some such.

The patches should give visibility to both nsproxy objects (by showing what tasks share the same nsproxy objects and letting tasks move across nsproxy objects if allowed) and the resource control objects pointed to by nsproxy (struct cpuset, struct cpu_limit, struct rss_limit etc).

> It doesn't really matter if processes disappear from namespace
> aggregates, because that's what's really happening anyway. The only
> problem is that if you try to freeze a namespace that has visibility of
> things at this level, you might not be able to reconstruct the
> filesystem in the same way. This may or may not be considered a problem,
> but open filehandles and directory handles etc surviving a freeze/thaw
> is part of what we're trying to achieve. Then again, perhaps some
> visibility is better than none for the time being.

>

> If they are restricted entirely to resource control, then don't use the
> nsproxy directly - use the structure or structures which hang off the
> nsproxy (or even task_struct) related to resource control.

--

Regards,
vatsa

Containers mailing list

