Subject: Re: The issues for agreeing on a virtualization/namespaces implementation.
Posted by ebiederm on Thu, 09 Feb 2006 22:25:35 GMT
View Forum Message <> Reply to Message

ebiederm@xmission.com (Eric W. Biederman) writes:

> I think I can boil the discussion down into some of the fundamental
> questions that we are facing.
>
> Currently everyone seems to agree that we need something like
> my namespace concept that isolates multiple resources.
>
> We need these for
> PIDS
> UIDS
> SYSVIPC
> NETWORK
> UTSNAME
> FILESYSTEM
> etc.
>
> The questions seem to break down into:
> 1) Where do we put the references to the different namespaces?
> - Do we put the references in a struct container that we reference from
> struct task_struct?
> - Do we put the references directly in struct task_struct?

Answer in the task_struct. It is the simplest and most flexible
route and the other implementations are still possible.

> 2) What is the syscall interface to create these namespaces?
> - Do we add clone flags?
> (Plan 9 style)
> - Do we add a syscall (similar to setsid) per namespace?
> (Traditional unix style)?
> - Do we in addition add syscalls to manipulate containers generically?

The answer seems to be we decide on a per namespace basis with
additional syscalls being mandatory if we have any additional data to
pass.

> 3) How do we refer to namespaces and containers when we are not members?

I have seen no arguments against referring to namespaces or containers
by global ids. So it seems we do not need a container id.

> 4) How do we implement each of these namespaces?

> Besides being maintainable are there other constraints?

Largely quite. But I have not heard additional constraints.

> 5) How do we control the resource inside a namespace starting
> from a process that is outside of that namespace?
> - The filesystem mount namespace gave an interesting answer.
> So it is quite possible other namespaces will give
> equally interesting and surprising answers.

Not yet resolved, but a bit of speculation.

> 6) How do we do all of this efficiently without a noticeable impact on
> performance?
> - I have already heard concerns that I might be introducing cache
> line bounces and thus increasing tasklist_lock hold time.
> Which on big way systems can be a problem.

A little discussion. At the level of the last few cache line I
think this needs to be addressed when we merge. Simply not
messing up existing optimizations sounds like a good initial
target. Basically at this stage trying hard would be a
premature optimization.

> 7) How do we allow a process inside a container to create containers
> for it's children?
> - In general this is trivial but there are a few ugly issues
> here.

This look mostly like something to be discussed when we merge
namespaces. But as long as we keep it in mind it is easy.

Eric