Subject: Re: semantics for namespace naming
Posted by serue on Thu, 14 Dec 2006 15:47:22 GMT
View Forum Message <> Reply to Message

Quoting Sukadev Bhattiprolu (sukadev@us.ibm.com):
> Serge E. Hallyn [serue@us.ibm.com] wrote:
> | Let's say we have a vserver, from which we start some jobs
> | which we want to checkpoint/restart/migrate. These are two
> | of the usages we currently foresee for the namespaces, though
> | I'd say it's safe to assume there will be more.
> |
> | I'll want to be able to address the c/r jobs by some ID in
> | order to checkpoint and kill them. I'll also want to be
> | able to address the entire vserver by some ID, in order to
> | kill it. In that case the c/r jobs should also be killed.
> | So those jobs are known by at least two id's.
>
> For your calculate_pi example below, are the two ids "calculate_pi"
> and "/sergevserver/calculate_pi" (ie. are the two plus ids basically
> like relative and absolute pathnames or are they independent ?

Right, so if you're inside /servervserver, you will just know the
namespace as 'calculate_pi', but if you're in the initial namespace,
'/', then you know it as /servervserver/calculate_pi.

> And unrelated to the namespace naming - by "job" do you mean a single
> process or can a job include multiple processes ? If it can include

Multiple processes.

> multiple, can we checkpoint/restart/migrate just the job ? I am
> thinking that we would need to migrate the entire vserver to
> preserve process relationships - no ?

So presumably when you did your

/bin/checkpointable_exec -nsid calculate_pi /bin/calculate_pi

then /bin/checkpointable_exec would have unshared the pid namespace.
So /servervserver and /servervserver/calculate_pi would have
different pidspaces. If they don't, then presumably you knew what
you were doing.

Whether you do have to checkpoint/restart an entire pidspace I'm
not sure, though I guess it mainly depends on what the applications
do. So long as they only talk to each other, I don't see why
the restore couldn't just recognize that pid 245, which was the
one to start the job, but which itself is not in the container

and therefore not checkpointed. So it can tweak the children of
pid 245 so their ->parent pointers point to the container init task.
But maybe not. I haven't really thought through that all.

thanks,
-serge

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