Subject: Re: [PATCH 1/4] Virtualization/containers: introduction
Posted by Hubertus Franke on Tue, 07 Feb 2006 23:18:31 GMT
View Forum Message <> Reply to Message

Sam Vilain wrote:
> Hubertus Franke wrote:
>
>> The container is just an umbrella object that ties every "virtualized"
>> subsystem together.
>
>
> I like this description; it matches roughly with the concepts as
> presented by vserver; there is the process virtualisation (vx_info), and
> the network virtualisation (nx_info) of Eric's that has been integrated
> to the vserver 2.1.x development branch. However the vx_info has become
> the de facto umbrella object space as well. These could almost
> certainly be split out without too much pain or incurring major
> rethinks.
>
> Sam.
>

Agreed.. here are some issued we learned from other projects that had
similar interception points.

Having a central umbrella object (let's stick to the name container)
is useful, but being the only object through which every access has to
pass may have drawbacks..

task->container->pspace->pidmap[offset].page implies potential
cachemisses etc.

If overhead becomes too large, then we can stick (cache) the pointer
additionally in the task struct. But ofcourse that should be carefully
examined on a per subsystem base...

==
Another thing to point out is that container's can have overlaps.

C/R should be a policy thing. So if each "subsystem"

> Quote Eric>>>
> PIDS
> UIDS
> SYSVIPC
> NETWORK
> UTSNAME

> FILESYSTEM

is represented as a NAMESPACE, then one can pick and choose as a
policy how these constitute at a conceptual level as a container.
You want something migratable you better make sure that
container implies unique subsystems.
Maybe you want to nest containers, but only want to create a
separate pidspaces for performance isolation (see planetlab work
with vserver).
So, there are many possibilities, that might make perfect sense
for different desired solutions and it seems with the
clone (CLONE_FLAGS_NSPACE_[PIDS/UIDS/SYS.../FS]) one gets a solution
that is flexible, yet embodies may requirements.....

-- Hubertus