Subject: Re: [RFC][PATCH 1/5] Virtualization/containers: startup
Posted by Dave Hansen on Mon, 06 Feb 2006 16:35:33 GMT
View Forum Message <> Reply to Message

On Sun, 2006-02-05 at 18:05 +0300, Kirill Korotaev wrote:
> > "tsk->owner_container" That makes it sound like a pointer to the "task
> > owner's container". How about "owning_container"? The "container
> > owning this task". Or, maybe just "container"?
> This is why I don't like "container" name.

I worry that using something like "vps" obfuscates the real meaning a bit. The reason that "owner_vps" doesn't sound weird is that people, by default, usually won't understand what a "vps" is.

(if you like acronyms a lot, I'm sure I can find a job for you at IBM or in the US military :)

> Please, also note, in OpenVZ we have 2 pointers on task_struct:
> One is owner of a task (owner_env), 2nd is a current context (exec_env).
> exec_env pointer is used to avoid adding of additional argument to all
> the functions where current context is required.

That makes sense. However, are there many cases in the kernel where a task ends up doing something temporary like this:

```
tsk->exec_vnc = bar;
do_something_here(task);
tsk->exec_vnc = foo;
```

If that's the case very often, we probably want to change the APIs, just to make the common action explicit. If it never happens, or is a rarity, I think it should be just fine.

> > Any particular reason for the "u32 id" in the vps_info struct as opposed
> > to one of the more generic types? Do we want to abstract this one in
> > the same way we do pid_t?
> VPS ID is passed to/from user space APIs and when you have a cluster
> with different archs and VPSs it is better to have something in common
> for managing this.

I guess it does keep you from running into issues with mixing 32 and 64-bit processes. But, haven't we solved those problems already? Is it just a pain?

> > Lastly, is this a place for krefs? I don't see a real need for a
> > destructor yet, but the idea is fresh in my mind.
> I don't see much need for krefs, do you?
> In OpenVZ we have 2-level refcounting (mentioned recently by Linus as in

> mm). Process counter is used to decide when container should
> collapse/cleanuped and real refcounter is used to free the structures
> which can be referenced from somewhere else.

It sounds to me like anything that needs to have an action taken when a refcount reaches zero is a good candidate for a kref. Both of those uses sound like they need that. Probably not too big of a deal, though.

-- Dave