Subject: Re: [PATCH 0/13] Pid namespaces (OpenVZ view)
Posted by Pavel Emelianov on Tue, 29 May 2007 07:47:35 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Pavel Emelianov <xemul@openvz.org> writes:
>
>> Hmm. I see. So you don't care that the pids in the namespace #2 are still
>> the same. I can understand that politics for namespace #1, but for #2...
>
> I'm confused, I think the statement above is wrong.
>
> If we just checkpoint/restart a leaf pid namespace we don't care about
> the other pids, in other namespace.
>
> If we checkpoint/restart a pid namespace with another pid namespace
> nested inside it we need to preserve the pids in the pid namespace we
> are checkpointing and in a nested pid namespaces.
>
> Pids in namespaces that none of the process we are migrating cannot
> see we do not care about. (i.e. the init pid namespace, and possibly
> some of it's children)
>
>> OK, if you need this let us go on with such model, but I'd like to see
>> the CONFIG_PID_NS_MULTILEVEL for this. Or at least CONFIG_PID_NS_FLAT for
>> my model as we do not need to sacrifice the performance to such generic
>> behavior.
>
> Where is the world would a performance sacrafice come in? If you

Easy! Consider the problem of getting a list of pids for proc. In case
of flat layout we just take a number from a known structure. In case of
nested pids we have to scan through the list of pid_elem-s or lookup
the hash or something similar.

The same stays true for wait() when we have to compare pids in the
eligible_child(), for setpgid(), terminal ioctls and so on and so forth.

Not to be unfounded I will measure booth cases with unixbench's spawn,
execl and shell tests and with "ps -xaf" and report the results. All will
be run in init namespace and in "level one" namespace. If the flat layout
wins (with noticeable difference) I would insist having two of them. Agree?

> happen to be using a deeply nested pid namespace I can see a small
> performance hit, there is fundamentally more to do. However if you
> don't use a nested pid namespace there should not be more work todo
> and it should be impossible to measure the over head.
>

> Further 3 levels should be as simple to implement and as cheap as two
> levels. Because we can continue to use static allocation.

Wait a bit. Do you mean that there's enough to have only 3 levels of
namespaces? I.e. to have a struct pid look like
struct pid {
 int pid;
 int pid1; /* for first level */
 int pid2; /* for 2nd level */
 ...
}
?

> Eric
>