Subject: Re: [PATCH 0/13] Pid namespaces (OpenVZ view)
Posted by ebiederm on Tue, 29 May 2007 12:36:06 GMT
View Forum Message <> Reply to Message

Pavel Emelianov <xemul@openvz.org> writes:

> Eric W. Biederman wrote:
>> Pavel Emelianov <xemul@openvz.org> writes:
>> 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.

We walk through the pidmap. That should not change either way.

I'm actually not horribly fond of walking through the pidmap but
it was needed for correctness so we could have a stable token we could
return to user space for restarting readdir in /proc.

> 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.

We should be comparing struct pid pointers not user space pid_t values.
With that being the case we should convert at the edge of user space
and all should be good.

>> 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 */
> ...
> }
> ?

Initially yes. 3 levels should be enough. Ultimately we may want more
but that should be a small tweak at the implementation level. Nothing

outside of the pid functions should care.

Eric