Subject: Re: [PATCH 01/11] SYSCTL: export root and set handling routines
Posted by ebiederm on Wed, 11 Jan 2012 17:20:17 GMT
View Forum Message <> Reply to Message

Stanislav Kinsbursky <skinsbursky@parallels.com> writes:

>> Stanislav Kinsbursky<skinsbursky@parallels.com> writes:
>>

>>>> Stanislav Kinsbursky<skinsbursky@parallels.com> writes:
>>>>

>>>>>> Stanislav Kinsbursky<skinsbursky@parallels.com> writes:
>>>>>>
>>>>>> Doing that independently of the rest of the sysctls is pretty horrible
>>>>>> and confusing to users. What I am planning might suit your needs and
>>>>>> if not we need to talk some more about how to get the vfs to do
>>>>>> something reasonable.
>>>>>>
>>>>>
>>>>> Ok, Eric. Would be glad to discuss your sysctls plans.
>>>>> But actually you already know my needs: I would like to make sysctls work in the
>>>>> way like sysfs does: i.e. content of files depends on mount maker -
>>>>> not viewer.
>>>>
>>>> What drives the desire to have sysctls depend on the mount maker?
>>>
>>> Because we can (will, actually) have nested fs root's for containers. IOW,
>>> container's root will be accessible from it's creator context. And I want to
>>> tune container's fs from creators context.
>>
>> Tuning the child context from the parent context is an entirely
>> reasonable thing to do. To affect a namespace that is not yours
>> the requirement is simply that we don't use current to lookup the
>> sysctl. So what I am proposing should work for your case.
>>
>
> Could you explain, what are you proposing?
> I still don't know any details about it.

I am proposing treating /proc/sys like /proc/net is currently treated.
See below.

>>>> Especially what drives that desire not to have it have a /proc/<pid>/sys
>>>> directory that reflects the sysctls for a given process.
>>>>
>>>

>>> This is not so important for me, where to access sysctl's. But I'm worrying
>>> about backward compatibility. IOW, I'm afraid of changing path
>>> "/proc/sys/sunprc/*" to "/proc/<pid>/sys/sunrpc". This would break a lot of
>>> user-space programs.
>>
>> The part that keeps it all working is by adding a symlink from /proc/sys
>> to /proc/self/sys. That technique has worked well for /proc/net, and I
>> don't expect there will be any problems with /proc/sys either. It is
>> possible but is very rare for the introduction of a symlink in a path
>> to cause problems.
>>
>
> Probably I don't understand you, but as I see it now, symlink to "/proc/self/"
> is unacceptable because of the following:
> 1) will be used current context (any) instead of desired one
(Using the current context is the desirable outcome for existing tools).
> 1) if CT has other pid namespace - then we just have broken link.

Assuming the process in question is not in the pid namespace available
to proc then yes you will indeed have a broken link. But a broken
link is only a problem for new applications that are doing something strange.

I am proposing treating /proc/sys like /proc/net has already been
treated. Aka move have the version of /proc/sys that relative to a
process be visible at: /proc/<pid>/sys, and with a compat symlink
from /proc/sys -> /proc/self/sys.

Just like has already been done with /proc/net.

Semantically this should be easy to understand, and about as backwards
compatible as it gets.

Eric
