Subject: Re: [RFC][PATCH] Do not set /proc inode->pid for non-pid-related inodes
Posted by ebiederm on Mon, 26 Mar 2007 17:12:36 GMT
View Forum Message <> Reply to Message

"Serge E. Hallyn" <serue@us.ibm.com> writes:

> Quoting Eric W. Biederman (ebiederm@xmission.com):
>> Dave Hansen <hansendc@us.ibm.com> writes:
>>
>> > So, doesn't that problem go away (or at least move to be umount's duty)
>> > if we completely disconnect those inodes' lifetime from that of any
>> > process or pid namespace?
>>
>> If the last process has exited the pid namespace I would like the
>> code to continue to behave as it currently does.
>>
>> I would like readdir on /proc/ to not even try to show any pids when
>> there are no pids or pid related files in the pid namespace.
>
> In (at least one version of) Dave's patches, the /proc your pidns is
> automatically used when you use /proc. In that case a /proc should
> just go away when the last task goes away, since noone else can use
> that /proc.

Unless I am rather confused that does extremely nasty things to
the VFS dentry cache. Because a dentry can point at one process
one minute and another process the next. It is doable but only
at the cost of decreased performance.

> I like that behavior, because otherwise (a) we require every new
> pid_namespace to start by remounting /proc ere they get undefined
> behavior,

The behavior won't be undefined just unexpected. Given the way
the vfs caching works the requirement for mounting /proc after
an we create a new copy of the pid namespace is a hard requirement.

> and (b) to gain anything from it, we would need a way
> to refer to another pidspace for the sake of mounting it's proc,
> i.e.
>
> mount -t proc -o init_pid=7501 proc_vserver1 /vserver1/proc

First we gain by not thrashing the dcache, and destroying /proc
performance.

Second we can use it if we unshare the mount namespace after we
create a separate pid namespace.

Third an option that points at the pid of a child process to dig
out the mount namespace isn't that hard, and is a simple extension.

Fourth there is an additional issue. There is the process related
part of /proc that is in fs/proc/base.c and then there is the
non-process related part of /proc in fs/proc/generic.c that probably
should have different rules.

Eric