Subject: Re: [PATCH 06/14] sysfs: Rewrite sysfs_get_dentry
Posted by ebiederm on Wed, 01 Aug 2007 09:22:40 GMT
View Forum Message <> Reply to Message

Tejun Heo <htejun@gmail.com> writes:

> On Tue, Jul 31, 2007 at 08:34:47PM +0900, Tejun Heo wrote:
>> > If sysfs_mutex nested the other way things would be easier,
>> > and we could grab all of the i_mutexes we wanted. I wonder if we can
>> > be annoying in sysfs_lookup and treat that as the lock inversion
>> > case using mutex_trylock etc. And have sysfs_mutex be on the
>> > outside for the rest of the cases?
>>
>> The problem with treating sysfs_lookup as inversion case is that vfs
>> layer grabs i_mutex outside of sysfs_lookup. Releasing i_mutex from
>> inside sysfs_lookup would be a hacky layering violation.
>>
>> Then again, the clean up which can come from the new sysfs_looukp_dentry
>> is very significant. I'll think about it a bit more.
>
> How about something like this? __sysfs_get_dentry() never creates any
> dentry, it just looks up existing ones. sysfs_get_dentry() calls
> __sysfs_get_dentry() and if it fails, it builds a path string and look
> up using regular vfs_path_lookup(). Once in the creation path,
> sysfs_get_dentry() is allowed to fail, so allocating path buf is fine.
>
> It still needs to retry when vfs_path_lookup() returns -ENOENT or the
> wrong dentry but things are much simpler now. It doesn't violate any
> VFS locking rule while maintaining all the benefits of
> sysfs_get_dentry() cleanup.
>
> Something like LOOKUP_KERNEL is needed to ignore security checks;
> otherwise, we'll need to resurrect lookup_one_len_kern() and open code
> look up.
>
> The patch is on top of all your patches and is in barely working form.

I will look a little more and see. But right now it looks like the
real problem with locking is that we use sysfs_mutex to lock the
sysfs_dirent s_children list.

Instead it really looks like we should use i_mutex from the appropriate
inode. Or is there a real performance problem with forcing the directory
inodes in core when we modify the directories?

Using i_mutex to lock the s_children list. Allows us to make sysfs_mutex
come before i_mutex, and it removes the need for an additional lock in
sysfs_lookup. So generally it looks like the right thing to do and

it should noticeably simplify the sysfs locking.

Was this an oversight or is there some good reason we aren't using
i_mutex to lock the s_children list?

Eric

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers