Subject: Re: [PATCH 06/14] sysfs: Rewrite sysfs_get_dentry
Posted by Tejun Heo on Tue, 31 Jul 2007 11:34:47 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
>>> + do {
>>> + /* Find the first ancestor I have not looked up */
>>> + cur = sd;
>>> + while (cur->s_parent != dentry->d_fsdata)
>>>   cur = cur->s_parent;
>>>
>>>   /* look it up */
>>>   dput(parent_dentry);
>> Shouldn't this be done after looking up the child?
> Yes and that is what this is. Delaying it a little longer
> made the conditionals easier to write and verify for correctness.

Right, missed the next line.

>>> + parent_dentry = dentry;
>>> + name.name = cur->s_name;
>>> + name.len = strlen(cur->s_name);
>>> + dentry = d_hash_and_lookup(parent_dentry, &name);
>>> + if (dentry)
>>> +   continue;
>>> + if (!create)
>>> +   goto out;
>> Probably missing dentry = NULL?
> d_hash_and_lookup sets dentry, and we can't get here if (dentry != NULL)

Yes.

>> One thing I'm worried about is that dentry can now be looked up without
>> holding i_mutex. In sysfs proper, there's no synchronization problem
>> but I'm not sure whether we're willing to cross that line set by vfs.
>> It might come back and bite our asses hard later.
>
> It's a reasonable concern. I'm wondering if there are any precedents
> set by distributed filesystems. Because in a sense that is what
> we are.

Yeah, that's the weird thing about sysfs. sysfs interface acts as a
different access point to the filesystem making it virtually distributed.

> As for crossing that line I don't know what to say it makes the
> code a lot cleaner, and if we are merged into the kernel at
> least it will be visible if someone rewrites the vfs.
>

> 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.

Thanks.

--
tejun