
Subject: Re: [PATCH 06/14] sysfs: Rewrite sysfs_get_dentry
Posted by [ebiederm](#) on Tue, 31 Jul 2007 11:23:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Tejun Heo <teheo@suse.de> writes:

```
> Eric W. Biederman wrote:
>> Currently sysfs_get_dentry is very hairy and requires all kinds
>> of locking magic. This patch rewrites sysfs_get_dentry to
>> not use the cached sd->s_dentry, and instead simply lookup
>> and create dcache entries.
>
> I wanted to kill sd->s_dentry from the beginning. The obstacle was
> actually the shadow directory support, ironic.
>
>> + struct qstr name;
>> + struct inode *inode;
>>
>> + parent_dentry = NULL;
>> + dentry = dget(sysfs_sb->s_root);
>>
>> + 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.

>> + 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)

>> + dentry = d_alloc(parent_dentry, &name);
>> + if (!dentry) {
```

```

>> + dentry = ERR_PTR(-ENOMEM);
>> + goto out;
>> }
>> + inode = sysfs_get_inode(cur);
>> + if (!inode) {
>>     dput(dentry);
>> + dentry = ERR_PTR(-ENOMEM);
>> + goto out;
>> }
>> + d_instantiate(dentry, inode);
>> + sysfs_attach_dentry(cur, dentry);
>> + } while (cur != sd);
>>
>> +out:
>> + dput(parent_dentry);
>> + return dentry;
>> +}
>>
>> @@ -750,6 +725,12 @@ static struct dentry * sysfs_lookup(struct inode *dir,
> struct dentry *dentry,
>> struct inode *inode;
>>
>> mutex_lock(&sysfs_mutex);
>> +
>> + /* Guard against races with sysfs_get_dentry */
>> + result = d_hash_and_lookup(dentry->d_parent, &dentry->d_name);
>> + if (result)
>> + goto out;
>> +
>
> Hmmm... This is tricky but probably better than the previous hairy
> sysfs_get_dentry(). I think it would be worthwhile to comment about
> creating dentry/inode behind vfs's back in __sysfs_get_dentry().

```

Yes. Good point. That is sufficiently non-intuitive and non-obvious to deserve a comment.

```

> 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.

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?

Anyway back to bed with me. I've been dreaming up to many silly ways to abuse the dcache...

Eric

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