Subject: Re: [PATCH 2/4] sysfs: Implement sysfs manged shadow directory support.
Posted by ebiederm on Mon, 30 Jul 2007 15:51:28 GMT
View Forum Message <> Reply to Message

Tejun Heo <teheo@suse.de> writes:

> Kirill Korotaev wrote:
>> Tejun Heo wrote:
>>> I thought something like supermount plus some twists or fuse based sysfs
>>> proxy would fit better. Dunno whether or how uevent and polling stuff
>>> can work that way tho. Note that sysfs no longer keeps dentries and
>>> inodes pinned. It might make the shared dentry stuff harder.
>>
>> We simply don't share sysfs dentries/inodes between containers.
>> It's not that frequently used time critical fs to be super-optimized... :)
>
> OIC, dentries and inodes are not shared. Good then. Agreed that sysfs
> doesn't need to be super-optimized as long as big machines aren't
> penalized too much (both memory and cpu cycle wise).
>
>> I don't like the idea with fuse, since sysfs exports kernel-related stuff,
>> so doing it via user-space would be pain.
>
> Yeah, it would be cumbersome to setup but it's also fast and easy to toy
> with for prototypes at least.

How close are we to the point where we can get mount sysfs multiple
times and get multiple dentry trees with different super blocks?

That really does sound like the right way to go. Especially as it
simplifies the monitoring of containers. If you want to watch what
the view looks like in some container your bind mount his sysfs and
look at that.

If we can do that the dcache side at least will be beautiful. And
with a little care we may be able to reduce the work to a special case
in lookup, some extra handling to mark directories as belonging only
to a certain mount of sysfs.

If we can find something that is stupid and simple I'm all for that.

To reach the no-kobj utopia we may also need a special device_migrate
that is a super set of device_rename (because sometimes we need to
rename devices when we move them between namespaces).

So are we close to having a sysfs that we can have multiple super
blocks for?

I'm on a sysctl tangent, but I should be able to look at that in
just a little bit.

Eric

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

---

Subject: Re: [PATCH 2/4] sysfs: Implement sysfs manged shadow directory support.
Posted by ebiederm on Tue, 31 Jul 2007 03:24:03 GMT
View Forum Message <> Reply to Message

Ugh. I need to step back and carefully define what I'm seeing but it
looks like the current sysfs locking is wrong.

I'm starting to find little inconsistencies all over the place
such as:

Which lock actually protects sd->s_children?
- It isn't sysfs_mutex. (see sysfs_lookup)
- It isn't inode->i_mutex (we only get it if we happen to have the inode
  in core)

At first glance sysfs_assoc_lock looks just as bad.

Eric

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

---

Subject: Re: [PATCH 2/4] sysfs: Implement sysfs manged shadow directory support.
Posted by Tejun Heo on Tue, 31 Jul 2007 03:41:45 GMT
View Forum Message <> Reply to Message

Hello,

Eric W. Biederman wrote:
> Ugh. I need to step back and carefully define what I'm seeing but it
> looks like the current sysfs locking is wrong.
>

> I'm starting to find little inconsistencies all over the place
> such as:
>
> Which lock actually protects sd->s_children?
> - It isn't sysfs_mutex. (see sysfs_lookup)
> - It isn't inode->i_mutex (we only get it if we happen to have the inode
>   in core)

Yeah, I missed two places while converting to sysfs_mutex.
sysfs_lookup() and rename(). I'm about to post patch to fix it.

> At first glance sysfs_assoc_lock looks just as bad.

I think sysfs_assoc_lock is okay. It's tricky tho. Why do you think
it's bad?

--
tejun

Subject: Re: [PATCH 2/4] sysfs: Implement sysfs manged shadow directory support.
Posted by Tejun Heo on Tue, 31 Jul 2007 03:51:57 GMT
View Forum Message <> Reply to Message

Hello,

Eric W. Biederman wrote:
> How close are we to the point where we can get mount sysfs multiple
> times and get multiple dentry trees with different super blocks?

Yeah, that sounds much better. We only have to pay attention to getting
sysfs_dirent tree correct. The rest can be done by just looking up the
correct sysfs_dirent in sysfs_lookup(). We would still need to pin all
shadows to keep sysfs_get_dentry() working.

> That really does sound like the right way to go. Especially as it
> simplifies the monitoring of containers. If you want to watch what
> the view looks like in some container your bind mount his sysfs and
> look at that.
>
> If we can do that the dcache side at least will be beautiful. And
> with a little care we may be able to reduce the work to a special case
> in lookup, some extra handling to mark directories as belonging only

> to a certain mount of sysfs.
>
> If we can find something that is stupid and simple I'm all for that.

Amen.

> To reach the no-kobj utopia we may also need a special device_migrate
> that is a super set of device_rename (because sometimes we need to
> rename devices when we move them between namespaces).

One thing I'm curious about is which semantic is appropriate behavior
when a node is migrated from one namespace to another - renaming or
deactivation followed by activation in new name space. I guess it
doesn't really matter.

> So are we close to having a sysfs that we can have multiple super
> blocks for?

Sorry but I dunno. It sounds much more appealing than other approaches tho.

--
tejun