Subject: Re: [PATCH 06/11] sysfs: Implement sysfs tagged directory support.
Posted by Tejun Heo on Wed, 02 Jul 2008 04:37:32 GMT
View Forum Message <> Reply to Message

Hello,

Eric W. Biederman wrote:
>>> A directory displaying only a single tag is an necessary constraint for
>>> a large number of reasons.
>> Okay, that isn't exactly the impression I get but... well. Let's see.
>
> Well one of those reasons is not having duplicate entries in your directory listing.
> That is much harder otherwise.

Agreed.

>> For netns, yes. I just think it would be better if the sysfs mechanism
>> to support that concept is more generic especially because it doesn't
>> seem too difficult to make it that way.
>
> Well the envisioned use is for other namespaces and they all are similar
> to the network namespace in that way.

Something I've been curious about is a directory which contains both the
untagged entries and tagged ones. I can definitely imagine something
like that to be useful for block device namespace.

>>>>> Cause you to view an the tags as dynamic?
>>>> The thing is that I don't really see why there's tagged_dir_ops at all.
>>> We need callbacks for interfacing with the kobject layer, and for
>>> selecting our set of tags at mount time. Not tagged_dir_ops so much
>>> as tagged_type_ops.
>> The kobject op seems a bit strange way to interface to me. For mount,
>> yeah, we'll need a hook somewhere or pass it via mount option maybe.
>
> I will look how if there is a place in the kobject layer to put it. With
> a second but noticeably different user I can compare and see how hard that will be.

Great, thanks.

>>> Further the abstraction is logically exactly one tag on a
>>> (sb,directory) pair.
>> I'm not so sure here. As a policy, maybe but I don't really see a
>> fundamental reason that the mechanism should enforce this.
>
> Well in the first implementation.

This pretty much defines the interface and is likely to force future

users to fit themselves into it.

>>> 4. Interface with the kobject layer.
>>>   kobject_add calls sysfs_create_dir
>>>   kboject_rename calls sysfs_rename_dir
>>>   kobject_del calls sysfs_remove_dir
>>>
>>>   For the first two operations we need a helper function to go from a
>>>   kobject to a tag.
>> Why not just add a parameter to sysfs_create_dir()? It's just twisted.
>
> I added it where it was easiest. Adding a parameter to sysfs_create_dir
> simply means I have to add the function to the kobject layer. It is certainly
> worth a second look though.

Is it difficult to just export it via kobject and device layer? If
changing the default function is too much of a hassle (and I'm sure it
would be), just add an extended version which takes @tag. The current
implementation feels like it tried too hard to not add intermediate
interfaces and ended up shooting outside from the innermost layer.

>>> We need helper functions for interfacing with the rest of the kernel.
>> Yes, that's why I view it as strange. These can be done in forward way
>> (by passing in mount options and/or arguments) but it's done by first
>> going into the sysfs and then calling back out to outer layer.
>
> Well in the case of mount the default parameter at least is current, and
> there are good reasons for that.

I was imagining something like...

 mount -t sysfs -o ns=0,4,5 /my/sys

And let the userland control which ns's are visible in the particular
mount. I'm not sure how useful that will be tho.

> On the other side I can't pass a tag through from the device layer to
> the kobject layer. It isn't a concept the kobject layer supports.

I think it's best to make kobject layer support it.

> At least though the conversation is in relative agreement. I will refresh
> the patches shortly and see where we are at.

Thanks a lot for the patience. :-)

--
tejun

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