Subject: Re: [PATCH 06/11] sysfs: Implement sysfs tagged directory support.
Posted by ebiederm on Wed, 02 Jul 2008 03:53:00 GMT
View Forum Message <> Reply to Message

Tejun Heo <htejun@gmail.com> writes:

> Hello,
>
> Eric W. Biederman wrote:
>> What we are implementing is not, a sb with a set of tags that are displayed,
>> but directories with a single tag that is displayed. The sb just happens
>> to hold the state for the directories.
>>
>> 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.

> A few things...
>
> 1. The lookup is currently done linearly and is fast enough for now.
> Also, most lookup ops are cached by vfs layer. I'm not sure how
> probable it is that we're gonna need hash or tree based sd lookup.

I don't know how bad sysfs is. On the sysctl side I have people complaining
because I am doing a lookup during insert and that lookup is linear. Sysfs
appears to have the same complexity as sysctl but just smaller constants.

>> That is just one important aspect of it.   We need a way to describe
>> which tag a sb,directory pair displays. It is a fundamental concept.
>
> 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.

>>>> 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.

>>> 3. enable / disable tag on a sb
>>   Disagree that is too flexible. Tags on a sb need to be
>>   unchanging or else we get vfs layer issues.
>
> Yeah, this really should be something which can't change once it's mounted.
The VFS chokes otherwise because it can't cache things properly.

>>   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.

>> 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.

>> 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.

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.

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

Eric