Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by ebiederm on Thu, 27 Sep 2007 19:25:48 GMT
View Forum Message <> Reply to Message

I still need to look at the code in detail but I have some concerns
I want to inject into this conversation of future sysfs architecture.

- If we want to carefully limit sysfs from going to wild code review
  is clearly not enough. We need some technological measures to
  assist us. As the experience with sysctl has shown.

  I discovered that something like 10% of the sysctl entries were
  buggy and had been for years when I added basic runtime sanity
  checks.

  I had also found one instance in the kernel and had one instance
  from outside the kernel where people had created files under
  /proc/sys not as sysctls but as using the infrastructure from
  proc_generic.c because it happened to work.

  So while it very well may be we don't want to use the kobject
  interface anymore. I expect that we want to have the sysfs_dirent
  interface not exported to modules, and only allow direct
  access from code compiled into the kernel.

  Mostly I am thinking that any non-object model users should have
  their own dedicated wrapper layer. To help keep things consistent
  and to make it hard enough to abuse the system that people will
  find that it is usually easier to do it the right way.

- The network namespace work scheduled to be merged in 2.6.24 is
  currently has a dependency in Kconfig that is "&& !SYSFS"
  because sysfs is currently very much a moving target.

  Does it look like we can resolve Tejun's work for 2.6.24?
  If not does it make sense to push my patches that allow
  multiple mounts of sysfs for 2.6.24? So I can allow
  network namespaces in the presence of sysfs.

  Outside of sysfs and the device model I'm only talk maybe 30 lines
  of code... So I could easily merge that patch later in the
  merge window after the other pieces have gone in.

- Farther down the road we have the device namespace.
  The bounding requirements are:
  - We want to restrict which set of devices a subset of process
    can access.
  - When we migrate an application we want to preserve the device

numbers of all devices that show up in the new location.
So filesystems whose block devices reside on a SAN, ramdisks,
ttys, etc.
Other devices that really are different we can handle with
hotplug remove and add events, during the migration.

So while there is lower hanging fruit the requirements for a
device namespace are becoming clear, and don't look like something
we will ultimately be able to dodge.

For sysfs the implication is that we will need to filter the
hotplug events based upon the device namespace of the recipient, and
we will need to restrict the set of devices that show up in sysfs
based on who mounts it (as the prototype patches with the network
namespace are doing).

Also fun is that the dev file implementation needs to be able to
report different major:minor numbers based on which mount of
sysfs we are dealing with.

Eric

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by Tejun Heo on Sat, 29 Sep 2007 22:06:23 GMT
View Forum Message <> Reply to Message

Hello, Eric.

Eric W. Biederman wrote:
> Mostly I am thinking that any non-object model users should have
> their own dedicated wrapper layer. To help keep things consistent
> and to make it hard enough to abuse the system that people will
> find that it is usually easier to do it the right way.

Hmmm... I think most current non-driver-model sysfs users are deep in
kernel anyway, but I think not exporting sysfs interface at all might be
a bit too restrictive. I think we need to examine the current
non-driver-model sysfs users thoroughly to determine what to do about
this. But, yes, I do agree that we need to put restrictions one way or
the other.

> Does it look like we can resolve Tejun's work for 2.6.24?
> If not does it make sense to push my patches that allow

> multiple mounts of sysfs for 2.6.24? So I can allow
> network namespaces in the presence of sysfs.
>
> Outside of sysfs and the device model I'm only talk maybe 30 lines
> of code... So I could easily merge that patch later in the
> merge window after the other pieces have gone in.

I think it would be better if namespace comes after interface update and other new features, especially symlink renaming, but, under the current circumstance, it might delay namespace unnecessarily and I have no problem with your patches going in first. My concerns are...

* Do you think you can use new rename implementation contained in this series? It borrows basic ideas from the implementation you used for namespace but is more generic. It would be great if you can use it without too much modification.

* I'm still against using callbacks to determine namespace tags because callbacks need to be coupled with sysfs internals more tightly and are more difficult to grasp interface-wise.

> - Farther down the road we have the device namespace.
> The bounding requirements are:
> - We want to restrict which set of devices a subset of process
> can access.
> - When we migrate an application we want to preserve the device
> numbers of all devices that show up in the new location.
> So filesystems whose block devices reside on a SAN, ramdisks,
> ttys, etc.
> Other devices that really are different we can handle with
> hotplug remove and add events, during the migration.
>
> So while there is lower hanging fruit the requirements for a
> device namespace are becoming clear, and don't look like something
> we will ultimately be able to dodge.
>
> For sysfs the implication is that we will need to filter the
> hotplug events based upon the device namespace of the recipient, and
> we will need to restrict the set of devices that show up in sysfs
> based on who mounts it (as the prototype patches with the network
> namespace are doing).
>
> Also fun is that the dev file implementation needs to be able to
> report different major:minor numbers based on which mount of
> sysfs we are dealing with.

Ah... Coming few months will be fun, won't they? :-)

--
tejun

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

---

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by Greg KH on Fri, 05 Oct 2007 06:23:02 GMT
View Forum Message <> Reply to Message

On Thu, Sep 27, 2007 at 01:25:48PM -0600, Eric W. Biederman wrote:
>
> I still need to look at the code in detail but I have some concerns
> I want to inject into this conversation of future sysfs architecture.
>
> - If we want to carefully limit sysfs from going to wild code review
> is clearly not enough. We need some technological measures to
> assist us. As the experience with sysctl has shown.

I totally agree. You should see the ways that people have tried to
circumvent the current kobject/sysfs code over the past years. It's so
scary it's not even funny...

> - The network namespace work scheduled to be merged in 2.6.24 is
> currently has a dependency in Kconfig that is "&& !SYSFS"
> because sysfs is currently very much a moving target.
>
> Does it look like we can resolve Tejun's work for 2.6.24?
> If not does it make sense to push my patches that allow
> multiple mounts of sysfs for 2.6.24? So I can allow
> network namespaces in the presence of sysfs.
>
> Outside of sysfs and the device model I'm only talk maybe 30 lines
> of code... So I could easily merge that patch later in the
> merge window after the other pieces have gone in.

I would be interested in seeing what your patches look like. I don't
think that we should take any more sysfs changes for 2.6.24 as we do
have a lot of them right now, and I don't think that Tejun and I agree
on the future direction of the outstanding ones just yet.

But I don't think that your multiple-mount patches could make it into
.24, unless .23 is still weeks away.

> - Farther down the road we have the device namespace.
> The bounding requirements are:

> - We want to restrict which set of devices a subset of process
>   can access.

That's reasonable.

> - When we migrate an application we want to preserve the device
>   numbers of all devices that show up in the new location.
>   So filesystems whose block devices reside on a SAN, ramdisks,
>   ttys, etc.
>   Other devices that really are different we can handle with
>   hotplug remove and add events, during the migration.
>
> So while there is lower hanging fruit the requirements for a
> device namespace are becoming clear, and don't look like something
> we will ultimately be able to dodge.
>
> For sysfs the implication is that we will need to filter the
> hotplug events based upon the device namespace of the recipient, and
> we will need to restrict the set of devices that show up in sysfs
> based on who mounts it (as the prototype patches with the network
> namespace are doing).

That is going to be interesting to see how you come up with a way to do
hat.

> Also fun is that the dev file implementation needs to be able to
> report different major:minor numbers based on which mount of
> sysfs we are dealing with.

Um, no, that's not going to happen. /dev/sda will _always_ have the
same major:minor number, as defined by the LSB spec. You can not break
that at all. So while you might not want to show all mounts
/sys/devices/block/sda/ the ones that you do, will all have the LSB
defined major:minor number assigned to it.

thanks,

greg k-h

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

---

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by ebiederm on Fri, 05 Oct 2007 12:12:41 GMT
View Forum Message <> Reply to Message

Greg KH <greg@kroah.com> writes:
>
>> Also fun is that the dev file implementation needs to be able to
>> report different major:minor numbers based on which mount of
>> sysfs we are dealing with.
>
> Um, no, that's not going to happen. /dev/sda will _always_ have the
> same major:minor number, as defined by the LSB spec. You can not break
> that at all. So while you might not want to show all mounts
> /sys/devices/block/sda/ the ones that you do, will all have the LSB
> defined major:minor number assigned to it.

Hmm. If that is in the LSB it must come from
Documentation/devices.txt I'm not after changing the user
visible major/minor assignments.

Let me see if a concrete example will help. Suppose I have
have a SAN with two disks: disk-1 and disk-2. I have
two machines A and B. On machine A I get the mapping:
sda -> disk-1, sdb ->disk-2. On machine B I wind up with
a different probe order so I get the mapping: sda -> disk-2
sdb ->disk-1.

To be very clear by sda I mean the block device with major 8 and
minor 0, and by sdb I mean the block device with major 8 and minor
16.

So I decide I want an environment on machine B that looks just
like the environment on machine A, so I can bring transfer over
a running program or whatever. So I run around looking at UUID
labels and what not and I discover that the machine B knows disk-1 as
sdb and that machine A knows disk-1 as sda. So I want to say:
/sys/devices/block/sdb show up in this other device namespace as
/sys/devices/block/sda.

In that instance a running program won't notice the difference.

Eric

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

---

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by ebiederm on Fri, 05 Oct 2007 12:44:04 GMT
View Forum Message <> Reply to Message

Greg KH <greg@kroah.com> writes:

> I would be interested in seeing what your patches look like.

Sure.

> I don't
> think that we should take any more sysfs changes for 2.6.24 as we do
> have a lot of them right now, and I don't think that Tejun and I agree
> on the future direction of the outstanding ones just yet.

Sounds reasonable.

> But I don't think that your multiple-mount patches could make it into
> .24, unless .23 is still weeks away.

Well I have posted them all earlier. At this point I it makes most
sense to wait until after the big merge happen and every rebases on
top of that. Then everyone will have network namespace support and
it is easier to look through all of the patches. Especially since
it looks like the merge window will open any day now.

I will quickly recap the essence of what I am looking at:
On directories of interest I tag all of their directory
entries with which namespace they belong to. On a mount
of sysfs I remember which namespace we were in when we
mounted sysfs. The I filter readdir and lookup based upon
the namespace I captured at mount time. I do my best
to generalize it so that the logic can work for different
namespaces.

Currently the heart of the patch from the network namespace
is below (I sniped the part that does the capture at mount time).
Basically the interface to users of this functionality is just
providing some way to go from a super block or a kobject to
the tag sysfs is using to filter things.

So I get one sysfs_dirent tree, but each super_block has it's
own tree of dcache entries.

Everything else is pretty much details in checking and propagating
the tags into the appropriate places.

Eric

diff --git a/net/core/net-sysfs.c b/net/core/net-sysfs.c
index 5adfdc2..a300f6e 100644
--- a/net/core/net-sysfs.c

```
+++ b/net/core/net-sysfs.c
@@ -435,6 +437,23 @@ static void netdev_release(struct device *d)
  kfree((char *)dev - dev->padded);
 }

+static const void *net_sb_tag(struct sysfs_tag_info *info)
+{
+ return info->net_ns;
+}
+
+static const void *net_kobject_tag(struct kobject *kobj)
+{
+ struct net_device *dev;
+ dev = container_of(kobj, struct net_device, dev.kobj);
+ return dev->nd_net;
+}
+
+static const struct sysfs_tagged_dir_operations net_tagged_dir_operations = {
+ .sb_tag = net_sb_tag,
+ .kobject_tag = net_kobject_tag,
+};
+
 static struct class net_class = {
  .name = "net",
  .dev_release = netdev_release,
@@ -444,6 +463,7 @@ static struct class net_class = {
 #ifdef CONFIG_HOTPLUG
  .dev_uevent = netdev_uevent,
 #endif
+ .tag_ops = &net_tagged_dir_operations,
 };

 /* Delete sysfs entries but hold kobject reference until after all
--
1.5.3.rc6.17.g1911
```

Subject: Re: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by dev on Fri, 05 Oct 2007 13:02:29 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:

> Greg KH <greg@kroah.com> writes:
>
>>> Also fun is that the dev file implementation needs to be able to
>>> report different major:minor numbers based on which mount of
>>> sysfs we are dealing with.
>>
>>Um, no, that's not going to happen. /dev/sda will _always_ have the
>>same major:minor number, as defined by the LSB spec. You can not break
>>that at all. So while you might not want to show all mounts
>>/sys/devices/block/sda/ the ones that you do, will all have the LSB
>>defined major:minor number assigned to it.
>
>
> Hmm. If that is in the LSB it must come from
> Documentation/devices.txt I'm not after changing the user
> visible major/minor assignments.
>
> Let me see if a concrete example will help. Suppose I have
> have a SAN with two disks: disk-1 and disk-2. I have
> two machines A and B. On machine A I get the mapping:
> sda -> disk-1, sdb ->disk-2. On machine B I wind up with
> a different probe order so I get the mapping: sda -> disk-2
> sdb ->disk-1.
>
> To be very clear by sda I mean the block device with major 8 and
> minor 0, and by sdb I mean the block device with major 8 and minor
> 16.
>
> So I decide I want an environment on machine B that looks just
> like the environment on machine A, so I can bring transfer over
> a running program or whatever. So I run around looking at UUID
> labels and what not and I discover that the machine B knows disk-1 as
> sdb and that machine A knows disk-1 as sda. So I want to say:
> /sys/devices/block/sdb show up in this other device namespace as
> /sys/devices/block/sda.

Imho environments to be migratable should have no direct access to the devices.
You can use any of stacked virtual filesystems to hide real
device from container.
You will have problems much bigger than this one otherwise
(imagine access to video, sound etc.)

Kirill

Subject: Re: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by ebiederm on Fri, 05 Oct 2007 13:24:55 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

> Imho environments to be migratable should have no direct access to the devices.
> You can use any of stacked virtual filesystems to hide real
> device from container.
> You will have problems much bigger than this one otherwise
> (imagine access to video, sound etc.)

What I am primarily concern about is when you can make the case that
the hardware we are talking is present before and after the migration.

When you are directly accessing a device. For times when it makes
sense to directly access hardware in a container (think infiniband
OS-bypass NICs). We need to tell user space that the device was
unplugged and another one was plugged in. If user space can cope with
that things should continue to work.

There are some very specific cases that we can support:
- Stateless devices like /dev/zero and dev/random.
- Virtual devices like ttys, ramdisks, loop devices
- Remote block devices like SCSI disks on a san, iSCSI, nbd, ATAoE.
- Local pseudo block devices like the backing devices for virtual
  filesystems.

There are very specific limits in which this can work and be useable,
and I don't claim to have looked at all of the details, but for the
block device case in particular we export the block device number
to user space in stat. There are some common applications which do
memorize stat data for files things like: git, incremental backup
software, and intrusion detection software.

Frankly the times when this matters is rare enough I don't put a
big priority on getting this done quickly. However a strong case
has been made for all of this filtering so we can run things like
udev in a container. Initially I only expect stateless character
devices and ttys to be allowed in a namespace, and I don't have
a clue what device number we will use in st_dev for stat in the
case our block device isn't in the user space interface. I just know
that it sounds like where we want to be eventually and thinking
about it now isn't a problem.

Eric

_____

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by Greg KH on Tue, 09 Oct 2007 22:51:39 GMT
View Forum Message <> Reply to Message

On Fri, Oct 05, 2007 at 06:12:41AM -0600, Eric W. Biederman wrote:
> Greg KH <greg@kroah.com> writes:
> >
> >> Also fun is that the dev file implementation needs to be able to
> >> report different major:minor numbers based on which mount of
> >> sysfs we are dealing with.
> >
> > Um, no, that's not going to happen. /dev/sda will _always_ have the
> > same major:minor number, as defined by the LSB spec. You can not break
> > that at all. So while you might not want to show all mounts
> > /sys/devices/block/sda/ the ones that you do, will all have the LSB
> > defined major:minor number assigned to it.
>
> Hmm. If that is in the LSB it must come from
> Documentation/devices.txt

Yes, that is the requirement.

> I'm not after changing the user visible major/minor assignments.

Oh, I misunderstood what you wrote above then.

> Let me see if a concrete example will help. Suppose I have
> have a SAN with two disks: disk-1 and disk-2. I have
> two machines A and B. On machine A I get the mapping:
> sda -> disk-1, sdb ->disk-2. On machine B I wind up with
> a different probe order so I get the mapping: sda -> disk-2
> sdb ->disk-1.

Ok.

> To be very clear by sda I mean the block device with major 8 and
> minor 0, and by sdb I mean the block device with major 8 and minor
> 16.

Ok.

> So I decide I want an environment on machine B that looks just
> like the environment on machine A, so I can bring transfer over

> a running program or whatever. So I run around looking at UUID
> labels and what not and I discover that the machine B knows disk-1 as
> sdb and that machine A knows disk-1 as sda. So I want to say:
> /sys/devices/block/sdb show up in this other device namespace as
> /sys/devices/block/sda.

Ah, but if you do that then the "other" device namespace would have
/sys/devices/block/sda/dev be 8:16, right? And that is not valid as sda
for that namespace must always map to the device with a 8:0 major:minor
as per the LSB spec.

So, no, that's not going to be allowed, sorry.

thanks,

greg k-h

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by Greg KH on Tue, 09 Oct 2007 22:53:06 GMT
View Forum Message <> Reply to Message

On Fri, Oct 05, 2007 at 06:44:04AM -0600, Eric W. Biederman wrote:
> Greg KH <greg@kroah.com> writes:
>
> > I would be interested in seeing what your patches look like.
>
> Sure.
>
> > I don't
> > think that we should take any more sysfs changes for 2.6.24 as we do
> > have a lot of them right now, and I don't think that Tejun and I agree
> > on the future direction of the outstanding ones just yet.
>
> Sounds reasonable.
>
> > But I don't think that your multiple-mount patches could make it into
> > .24, unless .23 is still weeks away.
>
> Well I have posted them all earlier. At this point I it makes most
> sense to wait until after the big merge happen and every rebases on
> top of that. Then everyone will have network namespace support and
> it is easier to look through all of the patches. Especially since
> it looks like the merge window will open any day now.

>
> I will quickly recap the essence of what I am looking at:
> On directories of interest I tag all of their directory
> entries with which namespace they belong to. On a mount
> of sysfs I remember which namespace we were in when we
> mounted sysfs. The I filter readdir and lookup based upon
> the namespace I captured at mount time. I do my best
> to generalize it so that the logic can work for different
> namespaces.

Ok, I have no objection to that. Let's wait for 2.6.24 to settle down
:)

thanks,

greg k-h

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

---

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by ebiederm on Wed, 10 Oct 2007 13:16:48 GMT
View Forum Message <> Reply to Message

Greg KH <greg@kroah.com> writes:

> On Fri, Oct 05, 2007 at 06:12:41AM -0600, Eric W. Biederman wrote:
>> Greg KH <greg@kroah.com> writes:
>> >
>> >>   Also fun is that the dev file implementation needs to be able to
>> >>   report different major:minor numbers based on which mount of
>> >>   sysfs we are dealing with.
>> >
>> > Um, no, that's not going to happen. /dev/sda will _always_ have the
>> > same major:minor number, as defined by the LSB spec. You can not break
>> > that at all. So while you might not want to show all mounts
>> > /sys/devices/block/sda/ the ones that you do, will all have the LSB
>> > defined major:minor number assigned to it.
>>
>> Hmm. If that is in the LSB it must come from
>> Documentation/devices.txt
>
> Yes, that is the requirement.
>
>> I'm not after changing the user visible major/minor assignments.
>

> Oh, I misunderstood what you wrote above then.

My above sentence is slightly misleading. That should have been.
I am not after changing the device name to major:minor assignments
as specified in Documentation/devices.txt.

So within a single device namespace everything is normal and as it
always has been. Weirdness only ensues when you look across device
namespaces.

>> Let me see if a concrete example will help. Suppose I have
>> have a SAN with two disks: disk-1 and disk-2. I have
>> two machines A and B. On machine A I get the mapping:
>> sda -> disk-1, sdb ->disk-2. On machine B I wind up with
>> a different probe order so I get the mapping: sda -> disk-2
>> sdb ->disk-1.
>
> Ok.
>
>> To be very clear by sda I mean the block device with major 8 and
>> minor 0, and by sdb I mean the block device with major 8 and minor
>> 16.
>
> Ok.
>
>> So I decide I want an environment on machine B that looks just
>> like the environment on machine A, so I can bring transfer over
>> a running program or whatever. So I run around looking at UUID
>> labels and what not and I discover that the machine B knows disk-1 as
>> sdb and that machine A knows disk-1 as sda. So I want to say:
>> /sys/devices/block/sdb show up in this other device namespace as
>> /sys/devices/block/sda.

>
> Ah, but if you do that then the "other" device namespace would have
> /sys/devices/block/sda/dev be 8:16, right?

No. The "other" device namespace I would construct on machine B to
look just like the device namespace that existed on machine A.
Making /sys/devices/block/sda would still be 8:0.

So to be very clear on machine B when talking about disk-1 I would have.
initial device namespace:
  /sys/devices/block/sdb
  /sys/devices/block/sdb/dev 8:16

"other" device namespace:
  /sys/devices/block/sda

/sys/devices/block/sda/dev 8:0

Similarly on machine B when talking about disk-2 I would have.
initial device namespace:
/sys/devices/block/sda
/sys/devices/block/sda/dev 8:0

"other" device namespace:
/sys/devices/block/sdb
/sys/devices/block/sdb/dev 8:16

So between the two devices namespaces on machine B the two disks
would exchange their user visible identities.

Eric

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

---

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by Greg KH on Wed, 10 Oct 2007 20:44:49 GMT
View Forum Message <> Reply to Message

On Wed, Oct 10, 2007 at 07:16:48AM -0600, Eric W. Biederman wrote:
> Greg KH <greg@kroah.com> writes:
>
> > On Fri, Oct 05, 2007 at 06:12:41AM -0600, Eric W. Biederman wrote:
> >> Greg KH <greg@kroah.com> writes:
> >> >
> >> >> Also fun is that the dev file implementation needs to be able to
> >> >> report different major:minor numbers based on which mount of
> >> >> sysfs we are dealing with.
> >> >
> >> > Um, no, that's not going to happen. /dev/sda will _always_ have the
> >> > same major:minor number, as defined by the LSB spec. You can not break
> >> > that at all. So while you might not want to show all mounts
> >> > /sys/devices/block/sda/ the ones that you do, will all have the LSB
> >> > defined major:minor number assigned to it.
> >>
> >> Hmm. If that is in the LSB it must come from
> >> Documentation/devices.txt
> >
> > Yes, that is the requirement.
> >
> >> I'm not after changing the user visible major/minor assignments.
> >

> > Oh, I misunderstood what you wrote above then.
>
> My above sentence is slightly misleading. That should have been.
> I am not after changing the device name to major:minor assignments
> as specified in Documentation/devices.txt.
>
> So within a single device namespace everything is normal and as it
> always has been. Weirdness only ensues when you look across device
> namespaces.
>
> >> Let me see if a concrete example will help. Suppose I have
> >> have a SAN with two disks: disk-1 and disk-2. I have
> >> two machines A and B. On machine A I get the mapping:
> >> sda -> disk-1, sdb ->disk-2. On machine B I wind up with
> >> a different probe order so I get the mapping: sda -> disk-2
> >> sdb ->disk-1.
> >
> > Ok.
> >
> >> To be very clear by sda I mean the block device with major 8 and
> >> minor 0, and by sdb I mean the block device with major 8 and minor
> >> 16.
> >
> > Ok.
> >
> >> So I decide I want an environment on machine B that looks just
> >> like the environment on machine A, so I can bring transfer over
> >> a running program or whatever. So I run around looking at UUID
> >> labels and what not and I discover that the machine B knows disk-1 as
> >> sdb and that machine A knows disk-1 as sda. So I want to say:
> >> /sys/devices/block/sdb show up in this other device namespace as
> >> /sys/devices/block/sda.
>
> >
> > Ah, but if you do that then the "other" device namespace would have
> > /sys/devices/block/sda/dev be 8:16, right?
>
> No. The "other" device namespace I would construct on machine B to
> look just like the device namespace that existed on machine A.
> Making /sys/devices/block/sda would still be 8:0.
>
> So to be very clear on machine B when talking about disk-1 I would have.
> initial device namespace:
>   /sys/devices/block/sdb
>   /sys/devices/block/sdb/dev 8:16
>
> "other" device namespace:
>   /sys/devices/block/sda

> /sys/devices/block/sda/dev 8:0
>
> Similarly on machine B when talking about disk-2 I would have.
> initial device namespace:
> /sys/devices/block/sda
> /sys/devices/block/sda/dev 8:0
>
> "other" device namespace:
> /sys/devices/block/sdb
> /sys/devices/block/sdb/dev 8:16
>
> So between the two devices namespaces on machine B the two disks
> would exchange their user visible identities.

Ah, ok, that makes more sense.

And seems quite difficult to do, good luck with that :)

greg k-h

_____

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

---

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by ebiederm on Wed, 10 Oct 2007 21:16:57 GMT
View Forum Message <> Reply to Message

Greg KH <greg@kroah.com> writes:

> Ah, ok, that makes more sense.
>
> And seems quite difficult to do, good luck with that :)

Thanks. At least now all I have to do is worry about the
details when we get that far instead of selling the big picture...

My gut feel is that sysfs is probably the hardest part to deal with,
and maybe half of the problem. Just intercepting the lookup by
device number is fairly simple, I think there is one spot
for block devices and another for character devices.

Anyway once the network namespace support is in with any luck that
will have solved half the sysfs problem.

Eric

_____

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by Sukadev Bhattiprolu on Tue, 16 Oct 2007 22:18:16 GMT
View Forum Message <> Reply to Message

Eric W. Biederman [ebiederm@xmission.com] wrote:
| Greg KH <greg@kroah.com> writes:
|
| > On Fri, Oct 05, 2007 at 06:12:41AM -0600, Eric W. Biederman wrote:
| >> Greg KH <greg@kroah.com> writes:
| >> >
| >> >> Also fun is that the dev file implementation needs to be able to
| >> >> report different major:minor numbers based on which mount of
| >> >> sysfs we are dealing with.
| >> >
| >> > Um, no, that's not going to happen. /dev/sda will _always_ have the
| >> > same major:minor number, as defined by the LSB spec. You can not break
| >> > that at all. So while you might not want to show all mounts
| >> > /sys/devices/block/sda/ the ones that you do, will all have the LSB
| >> > defined major:minor number assigned to it.
| >>
| >> Hmm. If that is in the LSB it must come from
| >> Documentation/devices.txt
| >
| > Yes, that is the requirement.
| >
| >> I'm not after changing the user visible major/minor assignments.
| >
| > Oh, I misunderstood what you wrote above then.
|
| My above sentence is slightly misleading. That should have been.
| I am not after changing the device name to major:minor assignments
| as specified in Documentation/devices.txt.
|
| So within a single device namespace everything is normal and as it
| always has been. Weirdness only ensues when you look across device
| namespaces.
|
| >> Let me see if a concrete example will help. Suppose I have
| >> have a SAN with two disks: disk-1 and disk-2. I have
| >> two machines A and B. On machine A I get the mapping:
| >> sda -> disk-1, sdb ->disk-2. On machine B I wind up with
| >> a different probe order so I get the mapping: sda -> disk-2
| >> sdb ->disk-1.

| >
| > Ok.
| >
| >> To be very clear by sda I mean the block device with major 8 and
| >> minor 0, and by sdb I mean the block device with major 8 and minor
| >> 16.
| >
| > Ok.
| >
| >> So I decide I want an environment on machine B that looks just
| >> like the environment on machine A, so I can bring transfer over
| >> a running program or whatever. So I run around looking at UUID
| >> labels and what not and I discover that the machine B knows disk-1 as
| >> sdb and that machine A knows disk-1 as sda. So I want to say:
| >> /sys/devices/block/sdb show up in this other device namespace as
| >> /sys/devices/block/sda.
|
| >
| > Ah, but if you do that then the "other" device namespace would have
| > /sys/devices/block/sda/dev be 8:16, right?
|
| No. The "other" device namespace I would construct on machine B to
| look just like the device namespace that existed on machine A.
| Making /sys/devices/block/sda would still be 8:0.
|
| So to be very clear on machine B when talking about disk-1 I would have.
| initial device namespace:
| /sys/devices/block/sdb
| /sys/devices/block/sdb/dev 8:16
|
| "other" device namespace:
| /sys/devices/block/sda
| /sys/devices/block/sda/dev 8:0
|
| Similarly on machine B when talking about disk-2 I would have.
| initial device namespace:
| /sys/devices/block/sda
| /sys/devices/block/sda/dev 8:0
|
| "other" device namespace:
| /sys/devices/block/sdb
| /sys/devices/block/sdb/dev 8:16
|
| So between the two devices namespaces on machine B the two disks
| would exchange their user visible identities.

So an application that would migrate from machine A to B has to
use virtual names (like "disk-1" and "disk-2") to access the disk

right ?

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

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

---

Subject: Re: [PATCHSET 3/4] sysfs: divorce sysfs from kobject and driver model
Posted by ebiederm on Tue, 16 Oct 2007 23:54:06 GMT
View Forum Message <> Reply to Message

sukadev@us.ibm.com writes:

> | No. The "other" device namespace I would construct on machine B to
> | look just like the device namespace that existed on machine A.
> | Making /sys/devices/block/sda would still be 8:0.
> |
> | So to be very clear on machine B when talking about disk-1 I would have.
> | initial device namespace:
> | /sys/devices/block/sdb
> | /sys/devices/block/sdb/dev 8:16
> |
> | "other" device namespace:
> | /sys/devices/block/sda
> | /sys/devices/block/sda/dev 8:0
> |
> | Similarly on machine B when talking about disk-2 I would have.
> | initial device namespace:
> | /sys/devices/block/sda
> | /sys/devices/block/sda/dev 8:0
> |
> | "other" device namespace:
> | /sys/devices/block/sdb
> | /sys/devices/block/sdb/dev 8:16
> |
> | So between the two devices namespaces on machine B the two disks
> | would exchange their user visible identities.
>
> So an application that would migrate from machine A to B has to
> use virtual names (like "disk-1" and "disk-2") to access the disk
> right ?

No. It is worse you need to access a filesystem and probably
a block device that is available on both machine A and machine B.
With care we can introduce appropriate namespaces and namespace semantics
so we can make the names be what we need.

For a classic tricky case think what it would require to migrate
a git archive with checked out files and not need to say
"git-update-index --refresh" before you work with the files.

I used names like disk-1 and disk-2 instead of UUIDs because it
was easier for me to type and think about. You do need some
kind of absolute disk or filesystem identity you can refer back to.

Eric

_____

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