Subject: [patch -mm 0/4] mqueue namespace
Posted by Cedric Le Goater on Wed, 28 Nov 2007 16:37:28 GMT
View Forum Message <> Reply to Message

Hello !

Here's a small patchset introducing a new namespace for POSIX
message queues.

Nothing really complex a part from the mqueue filesystem which
needed some special care

Thanks for reviewing !

C.

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

Subject: Re: [patch -mm 0/4] mqueue namespace
Posted by Pavel Emelianov on Wed, 28 Nov 2007 17:28:41 GMT
View Forum Message <> Reply to Message

Cedric Le Goater wrote:
> Hello !
>
> Here's a small patchset introducing a new namespace for POSIX
> message queues.
>
> Nothing really complex a part from the mqueue filesystem which
> needed some special care

Hm... Why did you decided to make it separately from the
IPC namespace?

> Thanks for reviewing !
>
> C.
>

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

Subject: Re: [patch -mm 0/4] mqueue namespace
Posted by Cedric Le Goater on Thu, 29 Nov 2007 09:52:28 GMT

View Forum Message <> Reply to Message

Pavel Emelyanov wrote:
> Cedric Le Goater wrote:
>> Hello !
>>
>> Here's a small patchset introducing a new namespace for POSIX
>> message queues.
>>
>> Nothing really complex a part from the mqueue filesystem which
>> needed some special care
>
> Hm... Why did you decided to make it separately from the
> IPC namespace?

Mostly because it has its own configuration option and filesystem
which requires to clone also the mnt namespace.

but yes they could probably be merged. Let's see what the others
have to say about it.

C.

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

Subject: Re: [patch -mm 0/4] mqueue namespace
Posted by ebiederm on Fri, 20 Jun 2008 03:00:51 GMT

View Forum Message <> Reply to Message

Cedric Le Goater <clg@fr.ibm.com> writes:

> Hello !
>
> Here's a small patchset introducing a new namespace for POSIX
> message queues.
>
> Nothing really complex a part from the mqueue filesystem which
> needed some special care

This looks stalled. I have a brainstorm that might takes a totally
different perspective on things.

The only reason we don't just allow multiple mounts of mqueuefs to solve this problem is because there is a kernel syscall on the path.

If we just hard coded a mount point into the kernel and required user
space to always mount mqueuefs there the problem would be solved.

hard coding a mount point is unfortunately violates the unix rule
of separating mechanism and policy.

One way to fix that is to add a hidden directory to the mnt namespace.
Where magic in kernel filesystems can be mounted. Only visible
with a magic openat flag. Then:

fd = openat(AT_FDKERN, ".", O_DIRECTORY)
fchdir(fd);
umount("./mqueue", MNT_DETACH);
mount(("none", "./mqueue", "mqueue", 0, NULL);

Would unshare the mqueue namespace.

Implemented for plan9 this would solve a problem of how do you get
access to all of it's special filesystems. As only bind mounts
and remote filesystem mounts are available. For linux thinking about
it might shake the conversation up a bit.

Eric

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

---

Subject: Re: [patch -mm 0/4] mqueue namespace
Posted by ebiederm on Fri, 20 Jun 2008 03:39:44 GMT

View Forum Message <> Reply to Message

ebiederm@xmission.com (Eric W. Biederman) writes:

> One way to fix that is to add a hidden directory to the mnt namespace.
> Where magic in kernel filesystems can be mounted. Only visible
> with a magic openat flag. Then:
>
> fd = openat(AT_FDKERN, ".", O_DIRECTORY)
> fchdir(fd);
> umount("./mqueue", MNT_DETACH);
> mount(("none", "./mqueue", "mqueue", 0, NULL);
>
> Would unshare the mqueue namespace.
>

> Implemented for plan9 this would solve a problem of how do you get
> access to all of it's special filesystems. As only bind mounts
> and remote filesystem mounts are available. For linux thinking about
> it might shake the conversation up a bit.

Thinking about this some more. What is especially attractive if we do all namespaces this way is that it solves two lurking problems.
1) How do you keep a namespace around without a process in it.
2) How do you enter a container.

If we could land the namespaces in the filesystem we could easily persist them past the point where a process is present in one if we so choose.

Entering a container would be a matter of replacing your current namespaces mounts with namespace mounts take from the filesystem.

I expect performance would degrade in practice, but it is tempting to implement it and run a benchmark and see if we can measure anything.

Eric

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

---

Subject: Re: [patch -mm 0/4] mqueue namespace
Posted by serue on Fri, 20 Jun 2008 14:50:31 GMT

View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Cedric Le Goater <clg@fr.ibm.com> writes:
>
> > Hello !
> >
> > Here's a small patchset introducing a new namespace for POSIX
> > message queues.
> >
> > Nothing really complex a part from the mqueue filesystem which
> > needed some special care
>
> This looks stalled.

It actually isn't really - Cedric had resent it a few weeks ago but had troubles with the mail server so it never hit the lists. I think Dave made a few more changes from there and was getting ready to resend again. Dave?

> I have a brainstorm that might takes a totally
> different perspective on things.
>
> The only reason we don't just allow multiple mounts of mqueuefs to
> solve this problem is because there is a kernel syscall on the path.
>
> If we just hard coded a mount point into the kernel and required user
> space to always mount mqueuefs there the problem would be solved.
>
> hard coding a mount point is unfortunately violates the unix rule
> of separating mechanism and policy.
>
> One way to fix that is to add a hidden directory to the mnt namespace.
> Where magic in kernel filesystems can be mounted. Only visible
> with a magic openat flag. Then:
>
> fd = openat(AT_FDKERN, ".", O_DIRECTORY)
> fchdir(fd);
> umount("./mqueue", MNT_DETACH);
> mount(("none", "./mqueue", "mqueue", 0, NULL);
>
> Would unshare the mqueue namespace.
>
> Implemented for plan9 this would solve a problem of how do you get
> access to all of it's special filesystems. As only bind mounts
> and remote filesystem mounts are available. For linux thinking about
> it might shake the conversation up a bit.

It is unfortunate that two actions are needed to properly complete the
unshare, and we had definately talked about just using the mount before.
I forget why we decided it wasn't practical, so maybe what you describe
solves it...

But at least the current patch reuses CLONE_NEWIPC for posix ipc, which
also seems to make sense.

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

Subject: Re: [patch -mm 0/4] mqueue namespace
Posted by serue on Fri, 20 Jun 2008 14:53:25 GMT

View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> ebiederm@xmission.com (Eric W. Biederman) writes:
>
> > One way to fix that is to add a hidden directory to the mnt namespace.
> > Where magic in kernel filesystems can be mounted. Only visible
> > with a magic openat flag. Then:
> >
> > fd = openat(AT_FDKERN, ".", O_DIRECTORY)
> > fchdir(fd);
> > umount("./mqueue", MNT_DETACH);
> > mount(("none", "./mqueue", "mqueue", 0, NULL);
> >
> > Would unshare the mqueue namespace.
> >
> > Implemented for plan9 this would solve a problem of how do you get
> > access to all of it's special filesystems. As only bind mounts
> > and remote filesystem mounts are available. For linux thinking about
> > it might shake the conversation up a bit.
>
> Thinking about this some more. What is especially attractive if we do
> all namespaces this way is that it solves two lurking problems.
> 1) How do you keep a namespace around without a process in it.
> 2) How do you enter a container.
>
> If we could land the namespaces in the filesystem we could easily
> persist them past the point where a process is present in one if we so
> choose.
>
> Entering a container would be a matter of replacing your current
> namespaces mounts with namespace mounts take from the filesystem.
>
> I expect performance would degrade in practice, but it is tempting
> to implement it and run a benchmark and see if we can measure anything.

The device ns could be a mount of an fs with the devices created in it,
while mknod becomes a symlink from that fs. And once a network
namespace is a filesystem, we can aim for the plan9 NAT solution of
mounting a remote /net onto ours. Neat.

But bye-bye posix?

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

Subject: Re: [patch -mm 0/4] mqueue namespace
Posted by ebiederm on Fri, 20 Jun 2008 19:11:26 GMT

View Forum Message <> Reply to Message

"Serge E. Hallyn" <serue@us.ibm.com> writes:

>
> It is unfortunate that two actions are needed to properly complete the
> unshare, and we had definately talked about just using the mount before.
> I forget why we decided it wasn't practical, so maybe what you describe
> solves it...

What is worse, and I don't see a way around it: Is that we don't have any callbacks to check where things are mounted. So we can't ensure the proper kind of filesystem is mounted in the right place.

That is there is too much freedom in the mount apis to allow for reliable operation.

> But at least the current patch reuses CLONE_NEWIPC for posix ipc, which
> also seems to make sense.

Sort of. I'm really annoyed with whoever did the posix mqueue support. Adding the magic syscall that has to know the internal mount instead of requiring the thing be mounted somewhere and just rejecting filedescriptors for the wrong sorts of files.

Eric

____

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