Subject: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by ebiederm on Mon, 11 Dec 2006 12:11:16 GMT
View Forum Message <> Reply to Message

This reverts commit 373beb35cd6b625e0ba4ad98baace12310a26aa8.

No one is using this identifier yet. The purpose of this identifier
is to export nsproxy to user space which is wrong. nsproxy is
an internal implementation optimization, which should keep our
fork times from getting slower as we increase the number of global
namespaces you don't have to share.

Adding a global identifier like this is inappropriate because it makes
namespaces inherently non-recursive, greatly limiting what we can
do with them in the future.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
include/linux/init_task.h | 1 -
include/linux/nsproxy.h | 1 -
kernel/nsproxy.c | 4 +---
3 files changed, 1 insertions(+), 5 deletions(-)

```
diff --git a/include/linux/init_task.h b/include/linux/init_task.h
index b531515..6383d2d 100644
--- a/include/linux/init_task.h
+++ b/include/linux/init_task.h
@@ -75,7 +75,6 @@ extern struct nsproxy init_nsproxy;
  .pid_ns  = &init_pid_ns,    \
  .count  = ATOMIC_INIT(1),   \
  .nslock  = __SPIN_LOCK_UNLOCKED(nsproxy.nslock),  \
- .id  = 0,     \
  .uts_ns  = &init_uts_ns,    \
  .mnt_ns  = NULL,     \
  INIT_IPC_NS(ipc_ns)     \
diff --git a/include/linux/nsproxy.h b/include/linux/nsproxy.h
index fdfb0e4..0b9f0dc 100644
--- a/include/linux/nsproxy.h
+++ b/include/linux/nsproxy.h
@@ -24,7 +24,6 @@ struct pid_namespace;
 struct nsproxy {
  atomic_t count;
  spinlock_t nslock;
- unsigned long id;
  struct uts_namespace *uts_ns;
  struct ipc_namespace *ipc_ns;
  struct mnt_namespace *mnt_ns;
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
```

```
index e2ce748..f5b9ee6 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -46,10 +46,8 @@ static inline struct nsproxy *clone_namespaces(struct nsproxy *orig)
  struct nsproxy *ns;

  ns = kmemdup(orig, sizeof(struct nsproxy), GFP_KERNEL);
- if (ns) {
+ if (ns)
   atomic_set(&ns->count, 1);
-  ns->id = -1;
- }
  return ns;
 }

--
1.4.4.1.g278f
```

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by Cedric Le Goater on Mon, 11 Dec 2006 15:46:08 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> This reverts commit 373beb35cd6b625e0ba4ad98baace12310a26aa8.
>
> No one is using this identifier yet. The purpose of this identifier
> is to export nsproxy to user space which is wrong. nsproxy is
> an internal implementation optimization, which should keep our
> fork times from getting slower as we increase the number of global
> namespaces you don't have to share.
>
> Adding a global identifier like this is inappropriate because it makes
> namespaces inherently non-recursive, greatly limiting what we can
> do with them in the future.

Future will tell us, until then, let's see how useless and buggy this non
feature is.

So for the moment, I would keep it and let people experiment.

thanks,

C.

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by ebiederm on Mon, 11 Dec 2006 19:10:36 GMT
View Forum Message <> Reply to Message

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

> Eric W. Biederman wrote:
>> This reverts commit 373beb35cd6b625e0ba4ad98baace12310a26aa8.
>>
>> No one is using this identifier yet. The purpose of this identifier
>> is to export nsproxy to user space which is wrong. nsproxy is
>> an internal implementation optimization, which should keep our
>> fork times from getting slower as we increase the number of global
>> namespaces you don't have to share.
>>
>> Adding a global identifier like this is inappropriate because it makes
>> namespaces inherently non-recursive, greatly limiting what we can
>> do with them in the future.
>
> Future will tell us, until then, let's see how useless and buggy this non
> feature is.
>
> So for the moment, I would keep it and let people experiment.

Even if the id is a sane idea nsproxy is very much the wrong place to
put it. nsproxy is an optimization so we don't bloat task struct with
several additional pointers, and it keeps fork times under control because
in the normal case we only have a single increment instead of several.
I'm not fully convinced it isn't a pessimization because it adds an
extra indirection. It is fully inappropriate to export that to user
space.

Now I don't mind a little experimentation but not in the stable kernel
when several people disagree.

To a very large degree adding an id to struct nsproxy violates the compromise
we came to when we agreed to add nsproxy.

I am willing to discuss this but not while it is silently being added
to the user interface and being exported to userspace in a way we have
to support for the forseeable future. To that I strongly object.

The fact that it is simply dead code for 2.6.20 is probably sufficient
justification to revert it until we can agree.

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by Vlad Yasevich on Mon, 11 Dec 2006 19:29:05 GMT
View Forum Message <> Reply to Message

Hi Eric

This is just a note that namespace identification, whatever form
it takes eventually, is a very useful idea.

The concept of being able to identify the namespaces running, and
manage them as your needs evolve without restarting is very useful
(even if not used yet).

-vlad

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by ebiederm on Mon, 11 Dec 2006 19:47:04 GMT
View Forum Message <> Reply to Message

Vlad Yasevich <vladislav.yasevich@hp.com> writes:

> Hi Eric
>
> This is just a note that namespace identification, whatever form
> it takes eventually, is a very useful idea.
>
> The concept of being able to identify the namespaces running, and
> manage them as your needs evolve without restarting is very useful
> (even if not used yet).

I completely agree we need to be able to identify the namespaces and
manage them. This is in the area of adding new syscalls and new semantics
and we have to very careful so we don't introduce security holes etc.

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by serue on Mon, 11 Dec 2006 20:24:37 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Cedric Le Goater <clg@fr.ibm.com> writes:
>
> > Eric W. Biederman wrote:
> >> This reverts commit 373beb35cd6b625e0ba4ad98baace12310a26aa8.
> >>
> >> No one is using this identifier yet. The purpose of this identifier
> >> is to export nsproxy to user space which is wrong. nsproxy is
> >> an internal implementation optimization, which should keep our
> >> fork times from getting slower as we increase the number of global
> >> namespaces you don't have to share.
> >>
> >> Adding a global identifier like this is inappropriate because it makes
> >> namespaces inherently non-recursive, greatly limiting what we can
> >> do with them in the future.
> >
> > Future will tell us, until then, let's see how useless and buggy this non
> > feature is.
> >
> > So for the moment, I would keep it and let people experiment.
>
> Even if the id is a sane idea nsproxy is very much the wrong place to
> put it. nsproxy is an optimization so we don't bloat task struct with
> several additional pointers, and it keeps fork times under control because
> in the normal case we only have a single increment instead of several.
> I'm not fully convinced it isn't a pessimization because it adds an
> extra indirection. It is fully inappropriate to export that to user
> space.
>
> Now I don't mind a little experimentation but not in the stable kernel
> when several people disagree.
>
> To a very large degree adding an id to struct nsproxy violates the compromise

> we came to when we agreed to add nsproxy.
>
> I am willing to discuss this but not while it is silently being added

Now now, it's not being silently added, it was a very clearly commented
part of a proposed patchset sent to all interested parties for review,
and now being argued over. Sounds kosher to me.

I think the problem is that some people wnat to see an answer to the
namespace entering problem right now, but the alternate solution ased on
using pids as implicit identifiers can't be used until the pidspaces are
fully implemented.

> to the user interface and being exported to userspace in a way we have
> to support for the forseeable future. To that I strongly object.
>
> The fact that it is simply dead code for 2.6.20 is probably sufficient
> justification to revert it until we can agree.

Cedric, do you mind moving these patches to the end of the set, so
that we can continue to develop patches against the -lxc tree without
being depending on these patches? I'd hate to have this become a reason
for people not to develop against -lxc.

-serge

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by Cedric Le Goater on Mon, 11 Dec 2006 21:47:26 GMT
View Forum Message <> Reply to Message

> Even if the id is a sane idea nsproxy is very much the wrong place to
> put it. nsproxy is an optimization so we don't bloat task struct with
> several additional pointers, and it keeps fork times under control because
> in the normal case we only have a single increment instead of several.

yes and so ?

> I'm not fully convinced it isn't a pessimization because it adds an
> extra indirection. It is fully inappropriate to export that to user
> space.

this is not exported to user space yet.

> Now I don't mind a little experimentation but not in the stable kernel
> when several people disagree.

yeah, i'm not sure how to understand that "several".

> To a very large degree adding an id to struct nsproxy violates the compromise
> we came to when we agreed to add nsproxy.

compromise ... you should say eric's capitulation ;)

> I am willing to discuss this but not while it is silently being added

you're in cc:

> to the user interface and being exported to userspace in a way we have
> to support for the forseeable future. To that I strongly object.

again : this is not exported to user space yet.

> The fact that it is simply dead code for 2.6.20 is probably sufficient
> justification to revert it until we can agree.

ok. i'll keep adding it to the patchset.

thanks for your positive contribution,

C, lightly upset but will not surrender.

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by Daniel Lezcano on Mon, 11 Dec 2006 22:51:40 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
>> Quoting Eric W. Biederman (ebiederm@xmission.com):
>>
>> Now now, it's not being silently added, it was a very clearly commented
>> part of a proposed patchset sent to all interested parties for review,
>> and now being argued over. Sounds kosher to me.
>
> Yes. I guess the part that was moderately silent was the fact that it
> was intended to be exported to user space. If you couldn't see the implication

> that part was not explicit. But I do agree that I missed this patch
> in the first round of review, and my apologies for that.
>
>> I think the problem is that some people wnat to see an answer to the
>> namespace entering problem right now, but the alternate solution ased on
>> using pids as implicit identifiers can't be used until the pidspaces are
>> fully implemented.
>
> I agree with that and that is a worthy discussion.
>
> One of the reasons I'm not too concerned is that sys_ptrace completely
> solves that problem today. The syscall interface completely sucks for
> handling that case but it works.
>
> The one instance where we clearly need a way to talk about namespaces
> besides enter is for moving network interfaces between network
> namespaces and I haven't looked close yet but I don't think either
> Dmitry or Daniel in their network namespace patches was using this id.
>

Well, I don't do that for the moment, but I was planning to use the
namespace id.

  -- Daniel

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by ebiederm on Tue, 12 Dec 2006 03:51:46 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <daniel.lezcano@free.fr> writes:

>>
>> I agree with that and that is a worthy discussion.
>>
>> One of the reasons I'm not too concerned is that sys_ptrace completely
>> solves that problem today. The syscall interface completely sucks for
>> handling that case but it works.
>>
>> The one instance where we clearly need a way to talk about namespaces
>> besides enter is for moving network interfaces between network
>> namespaces and I haven't looked close yet but I don't think either
>> Dmitry or Daniel in their network namespace patches was using this id.
>>

>
> Well, I don't do that for the moment, but I was planning to use the namespace
> id.

To be very clear.
- I completely agree we need an identifier for namespaces.
  So far my vision is one per namespace not one per nsproxy.
- I believe the identifier should be in one of the namespaces,
  so we don't have problems with recursion.

I want to be able to use things like the pam_namespace module in a guest.

My suggestion is that we name our process groups in the traditional pid
namespace.

Then we can have a name for each process group in each namespace.

For debugging I think this could be quite helpful if someone gets
their reference counting wrong.

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by Cedric Le Goater on Wed, 13 Dec 2006 17:21:38 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Daniel Lezcano <daniel.lezcano@free.fr> writes:
>
>>> I agree with that and that is a worthy discussion.
>>>
>>> One of the reasons I'm not too concerned is that sys_ptrace completely
>>> solves that problem today. The syscall interface completely sucks for
>>> handling that case but it works.
>>>
>>> The one instance where we clearly need a way to talk about namespaces
>>> besides enter is for moving network interfaces between network
>>> namespaces and I haven't looked close yet but I don't think either
>>> Dmitry or Daniel in their network namespace patches was using this id.
>>>
>> Well, I don't do that for the moment, but I was planning to use the namespace
>> id.
>

> To be very clear.
> - I completely agree we need an identifier for namespaces.
> So far my vision is one per namespace not one per nsproxy.
> - I believe the identifier should be in one of the namespaces,
> so we don't have problems with recursion.

a new nsproxy is created each time any namespace is unshared, so it's
basically the same to use the nsproxy id. today you can identify
any namespace by its nsproxy. that's how the bind_ns syscalls works.

but he, let's see where the discussion brings us.

at least, we *agreed* that we need an id. now, let's find a location
for it and a way to bind to it.

> I want to be able to use things like the pam_namespace module in a
> guest.

It should be possible.

C.

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH 1/1] Revert "[PATCH] identifier to nsproxy"
Posted by Cedric Le Goater on Wed, 13 Dec 2006 17:34:58 GMT
View Forum Message <> Reply to Message

>> The fact that it is simply dead code for 2.6.20 is probably sufficient
>> justification to revert it until we can agree.
>
> Cedric, do you mind moving these patches to the end of the set, so
> that we can continue to develop patches against the -lxc tree without
> being depending on these patches? I'd hate to have this become a reason
> for people not to develop against -lxc.

sure, in next -lxc drop.

I've merged the openvz patches in 2.6.19-mm1-lxc1 and will move these id
patches down the stack.

C.

_____
Containers mailing list
Containers@lists.osdl.org