Subject: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by Pavel Emelianov on Fri, 14 Sep 2007 07:39:32 GMT
View Forum Message <> Reply to Message

I proposed introducing a list_for_each_entry_continue_reverse
macro to be used in setup_net() when unrolling the failed
->init callback.

Here is the macro and some more cleanup in the setup_net() itself
to remove one variable from the stack :) Minor, but the code
looks nicer.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

I have problems with cloning repos from git.openvz.org, so
this patch comes to the yesterdays net-2.6.24 David's tree.

```
diff --git a/include/linux/list.h b/include/linux/list.h
index f29fc9c..ad9dcb9 100644
--- a/include/linux/list.h
+++ b/include/linux/list.h
@@ -525,6 +525,20 @@ static inline void list_splice_init_rcu(
       pos = list_entry(pos->member.next, typeof(*pos), member))

 /**
+ * list_for_each_entry_continue_reverse - iterate backwards from the given point
+ * @pos: the type * to use as a loop cursor.
+ * @head: the head for your list.
+ * @member: the name of the list_struct within the struct.
+ *
+ * Start to iterate over list of given type backwards, continuing after
+ * the current position.
+ */
+#define list_for_each_entry_continue_reverse(pos, head, member)  \
+ for (pos = list_entry(pos->member.prev, typeof(*pos), member); \
+      prefetch(pos->member.prev), &pos->member != (head); \
+      pos = list_entry(pos->member.prev, typeof(*pos), member))
+
+/**
  * list_for_each_entry_from - iterate over list of given type from the current point
  * @pos: the type * to use as a loop cursor.
  * @head: the head for your list.
diff --git a/net/core/net_namespace.c b/net/core/net_namespace.c
index 1fc513c..a9dd261 100644
--- a/net/core/net_namespace.c
+++ b/net/core/net_namespace.c
```

```
@@ -102,7 +102,6 @@ static int setup_net(struct net *net)
 {
  /* Must be called with net_mutex held */
  struct pernet_operations *ops;
- struct list_head *ptr;
  int error;

  memset(net, 0, sizeof(struct net));
@@ -110,8 +109,7 @@ static int setup_net(struct net *net)
  atomic_set(&net->use_count, 0);

  error = 0;
- list_for_each(ptr, &pernet_list) {
-  ops = list_entry(ptr, struct pernet_operations, list);
+ list_for_each_entry(ops, &pernet_list, list) {
   if (ops->init) {
    error = ops->init(net);
    if (error < 0)
@@ -120,12 +118,12 @@ static int setup_net(struct net *net)
  }
 out:
  return error;
+
 out_undo:
  /* Walk through the list backwards calling the exit functions
   * for the pernet modules whose init functions did not fail.
   */
- for (ptr = ptr->prev; ptr != &pernet_list; ptr = ptr->prev) {
-  ops = list_entry(ptr, struct pernet_operations, list);
+ list_for_each_entry_continue_reverse(ops, &pernet_list, list) {
   if (ops->exit)
    ops->exit(net);
  }
```

Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by Stephen Hemminger on Fri, 14 Sep 2007 12:49:38 GMT
View Forum Message <> Reply to Message

On Fri, 14 Sep 2007 11:39:32 +0400
Pavel Emelyanov <xemul@openvz.org> wrote:

> I proposed introducing a list_for_each_entry_continue_reverse
> macro to be used in setup_net() when unrolling the failed
> ->init callback.
>
> Here is the macro and some more cleanup in the setup_net() itself

> to remove one variable from the stack :) Minor, but the code
> looks nicer.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Maybe it is time to just eliminate the init hook from the API.
It has very few users, and there is no reason the setup needed
could be done before or after registering in most cases.

Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by ebiederm on Fri, 14 Sep 2007 14:41:07 GMT
View Forum Message <> Reply to Message

Stephen Hemminger <shemminger@linux-foundation.org> writes:

> On Fri, 14 Sep 2007 11:39:32 +0400
> Pavel Emelyanov <xemul@openvz.org> wrote:
>
>> I proposed introducing a list_for_each_entry_continue_reverse
>> macro to be used in setup_net() when unrolling the failed
>> ->init callback.
>>
>> Here is the macro and some more cleanup in the setup_net() itself
>> to remove one variable from the stack :) Minor, but the code
>> looks nicer.
>>
>> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>
> Maybe it is time to just eliminate the init hook from the API.
> It has very few users, and there is no reason the setup needed
> could be done before or after registering in most cases.

I guess only have 5 out of the 29 users I have in my full patchset
is few. But that is to be expected because so far only the core
has been converted.

I looked again at the initialization to see if you had a point about
the initialization but in every instance I looked at the function
was performing work that needed to happen during the creation of
each network namespace. So the work very much needs to be done there.

Ok looking some more I can see why this isn't obvious yet. copy_net_ns
hasn't been merged yet, and that is where we create new network namespaces.
And call setup_net on each new network namespace.

I will take a look at that patch and see if I can come up with a

safe version of it to merge to allow for a little more transparency.

```
> struct net *copy_net_ns(unsigned long flags, struct net *old_net)
> {
>         struct net *new_net = NULL;
>         int err;
>
>         get_net(old_net);
>
>         if (!(flags & CLONE_NEWNET))
>                 return old_net;
>
>         err = -EPERM;
>         if (!capable(CAP_SYS_ADMIN))
>                 goto out;
>
>         err = -ENOMEM;
>         new_net = net_alloc();
>         if (!new_net)
>                 goto out;
>
>         mutex_lock(&net_mutex);
>         err = setup_net(new_net);
>         if (err)
>                 goto out_unlock;
>
>         net_lock();
>         list_add_tail(&new_net->list, &net_namespace_list);
>         net_unlock();
>
>
> out_unlock:
>         mutex_unlock(&net_mutex);
> out:
>         put_net(old_net);
>         if (err) {
>                 net_free(new_net);
>                 new_net = ERR_PTR(err);
>         }
>         return new_net;
> }
```

Eric

Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net

Posted by ebiederm on Fri, 14 Sep 2007 17:33:00 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov <xemul@openvz.org> writes:

> I proposed introducing a list_for_each_entry_continue_reverse
> macro to be used in setup_net() when unrolling the failed
> ->init callback.
>
> Here is the macro and some more cleanup in the setup_net() itself
> to remove one variable from the stack :) Minor, but the code
> looks nicer.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Acked-by: "Eric W. Biederman" <ebiederm@xmission.com>

Eric

Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by Stephen Hemminger on Fri, 14 Sep 2007 20:06:56 GMT
View Forum Message <> Reply to Message

On Fri, 14 Sep 2007 08:41:07 -0600
ebiederm@xmission.com (Eric W. Biederman) wrote:

> Stephen Hemminger <shemminger@linux-foundation.org> writes:
>
> > On Fri, 14 Sep 2007 11:39:32 +0400
> > Pavel Emelyanov <xemul@openvz.org> wrote:
> >
> >> I proposed introducing a list_for_each_entry_continue_reverse
> >> macro to be used in setup_net() when unrolling the failed
> >> ->init callback.
> >>
> >> Here is the macro and some more cleanup in the setup_net() itself
> >> to remove one variable from the stack :) Minor, but the code
> >> looks nicer.
> >>
> >> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
> >
> > Maybe it is time to just eliminate the init hook from the API.
> > It has very few users, and there is no reason the setup needed
> > could be done before or after registering in most cases.
>
> I guess only have 5 out of the 29 users I have in my full patchset

> is few. But that is to be expected because so far only the core
> has been converted.
>
> I looked again at the initialization to see if you had a point about
> the initialization but in every instance I looked at the function
> was performing work that needed to happen during the creation of
> each network namespace. So the work very much needs to be done there.
>
> Ok looking some more I can see why this isn't obvious yet. copy_net_ns
> hasn't been merged yet, and that is where we create new network namespaces.
> And call setup_net on each new network namespace.
>
> I will take a look at that patch and see if I can come up with a
> safe version of it to merge to allow for a little more transparency.

Could we just make it so dev->init is not allowed to fail? Then it
can be a void function and the nasty unwind code can go?

Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by ebiederm on Fri, 14 Sep 2007 21:53:21 GMT
View Forum Message <> Reply to Message

Stephen Hemminger <shemminger@linux-foundation.org> writes:

> Could we just make it so dev->init is not allowed to fail? Then it
> can be a void function and the nasty unwind code can go?

Unfortunately we need to allocate memory, and perform other operations
that can fail. That's the nature of the problem.

So I think not allowing init to fail would be optimizing for the wrong
the case. Allowing init to fail makes the rest of the code simpler
because we don't have to perform the impossible when the highly
unlikely happens.

The ugly unwind is only about 5 lines of code that never need to
change (except for beautification). So I don't think the cost
is prohibitive.

Eric

Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by davem on Sun, 16 Sep 2007 23:49:14 GMT

From: Stephen Hemminger <shemminger@linux-foundation.org>
Date: Fri, 14 Sep 2007 22:07:14 +0200

> Could we just make it so dev->init is not allowed to fail? Then it
> can be a void function and the nasty unwind code can go?

Someone (not me :-) need to do an audit to find all current
users of this function and determine if they all can live
without returning errors.

If so, sure let's make the change and simplify things.

Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by ebiederm on Mon, 17 Sep 2007 00:06:00 GMT

David Miller <davem@davemloft.net> writes:

> From: Stephen Hemminger <shemminger@linux-foundation.org>
> Date: Fri, 14 Sep 2007 22:07:14 +0200
>
>> Could we just make it so dev->init is not allowed to fail? Then it
>> can be a void function and the nasty unwind code can go?
>
> Someone (not me :-) need to do an audit to find all current
> users of this function and determine if they all can live
> without returning errors.
>
> If so, sure let's make the change and simplify things.

I did that audit when I replied to Stephen the first time and I just
redid it to verify myself. We are calling functions that can fail
from the init function (kmalloc in the most common). So the
init function can fail.

So short of adding a bunch of BUG_ON's to the kernel to trap those
failure cases we can't remove the backwards list walk. Especially
since I can initiate this code path as root by calling
"clone(CLONE_NEWNET...)".

Eric

Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by davem on Mon, 17 Sep 2007 00:07:00 GMT
View Forum Message <> Reply to Message

From: ebiederm@xmission.com (Eric W. Biederman)
Date: Sun, 16 Sep 2007 18:06:00 -0600

> I did that audit when I replied to Stephen the first time and I just
> redid it to verify myself. We are calling functions that can fail
> from the init function (kmalloc in the most common). So the
> init function can fail.
>
> So short of adding a bunch of BUG_ON's to the kernel to trap those
> failure cases we can't remove the backwards list walk. Especially
> since I can initiate this code path as root by calling
> "clone(CLONE_NEWNET...)".

I just noticed that posting and thanks for reiterating.