

---

Subject: [PATCH 0/8 net-2.6.26] [NETNS]: namespace refcounting cleanup

Posted by [den](#) on Tue, 15 Apr 2008 12:35:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Network namespace has two reference counters:

- count
- use\_count.

The namespace is scheduled to destruction automatically when the count becomes 0.

There are several SLAB objects with a pointer to a namespace on them. These objects are cleaned up during namespace stop. Some of them increment use\_count, some don't. This set fixes this discrepancy, i.e. now all such objects increment the use\_count.

Though, the use\_count itself is used in a very debug manner and checked only during namespace stop. So, I have placed it under NETNS\_REFCNT\_DEBUG definition exactly like this is done for socket refcounting code to remove extra atomic from any possible fast paths.

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

---

Containers mailing list

[Containers@lists.linux-foundation.org](mailto:Containers@lists.linux-foundation.org)

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---

---

Subject: [PATCH 1/8 net-2.6.26] [NETNS]: Make netns refcounting debug like a socket one.

Posted by [den](#) on Tue, 15 Apr 2008 12:37:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Make release\_net/hold\_net noop for performance-hungry people. This is a debug staff and should be used in the debug mode only.

Add check for net != NULL in hold/release calls. This will be required later on.

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

---  
include/net/net\_namespace.h | 44 ++++++-----  
net/core/net\_namespace.c | 4 +++  
2 files changed, 30 insertions(+), 16 deletions(-)

diff --git a/include/net/net\_namespace.h b/include/net/net\_namespace.h  
index e2aee26..269a681 100644  
--- a/include/net/net\_namespace.h

```

+++ b/include/net/net_namespace.h
@@ -21,9 +21,11 @@ struct net_device;
    atomic_t count; /* To decided when the network
        * namespace should be freed.
        */
+#ifdef NETNS_REFCNT_DEBUG
    atomic_t use_count; /* To track references we
        * destroy on demand
        */
+#endif
    struct list_head list; /* list of network namespaces */
    struct work_struct work; /* work struct for freeing */

@@ -115,17 +119,6 @@ static inline void put_net(struct net *net)
    __put_net(net);
}

-static inline struct net *hold_net(struct net *net)
-{-
- atomic_inc(&net->use_count);
- return net;
-}
-
-static inline void release_net(struct net *net)
-{-
- atomic_dec(&net->use_count);
-}
-
static inline
int net_eq(const struct net *net1, const struct net *net2)
{
@@ -141,27 +134,46 @@ static inline void put_net(struct net *net)
{
}

+static inline struct net *maybe_get_net(struct net *net)
+{-
+ return net;
+}
+
+static inline
+int net_eq(const struct net *net1, const struct net *net2)
+{-
+ return 1;
+}
+#endif
+
+

```

```

+#ifdef NETNS_REFCNT_DEBUG
static inline struct net *hold_net(struct net *net)
{
+ if (net == NULL)
+ return NULL;
+ atomic_inc(&net->use_count);
return net;
}

static inline void release_net(struct net *net)
{
+ if (net == NULL)
+ return;
+ atomic_dec(&net->use_count);
}
-
-static inline struct net *maybe_get_net(struct net *net)
+#else
+static inline struct net *hold_net(struct net *net)
{
return net;
}

-static inline
-int net_eq(const struct net *net1, const struct net *net2)
+static inline void release_net(struct net *net)
{
- return 1;
}
#endif

+
#define for_each_net(VAR) \
list_for_each_entry(VAR, &net_namespace_list, list)

diff --git a/net/core/net_namespace.c b/net/core/net_namespace.c
index 7b66083..f310ef3 100644
--- a/net/core/net_namespace.c
+++ b/net/core/net_namespace.c
@@ -30,7 +30,9 @@ static __net_init int setup_net(struct net *net)
int error;

atomic_set(&net->count, 1);
+#ifdef NETNS_REFCNT_DEBUG
atomic_set(&net->use_count, 0);
+#endif

error = 0;

```

```

list_for_each_entry(ops, &pernet_list, list) {
@@ -70,11 +72,13 @@ static void net_free(struct net *net)
if (!net)
return;

+#ifdef NETNS_REFCNT_DEBUG
if (unlikely(atomic_read(&net->use_count) != 0)) {
printk(KERN_EMERG "network namespace not free! Usage: %d\n",
atomic_read(&net->use_count));
return;
}
+#endif

kmem_cache_free(net_cachep, net);
}
--
1.5.3.rc5

```

---

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

---



---

Subject: [PATCH 2/8 net-2.6.26] [NETNS]: Add netns refcnt debug for kernel sockets.

Posted by [den](#) on Tue, 15 Apr 2008 12:37:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Protocol control sockets and netlink kernel sockets should not prevent the namespace stop request. They are initialized and disposed in a special way by sk\_change\_net/sk\_release\_kernel.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```

---
include/net/sock.h | 2 +-
net/core/sock.c | 1 +
2 files changed, 2 insertions(+), 1 deletions(-)

```

```

diff --git a/include/net/sock.h b/include/net/sock.h
index 09255ea..dc42b44 100644
--- a/include/net/sock.h
+++ b/include/net/sock.h
@@ -1314,7 +1314,7 @@ void sock_net_set(struct sock *sk, struct net *net)
static inline void sk_change_net(struct sock *sk, struct net *net)
{
put_net(sock_net(sk));
- sock_net_set(sk, net);

```

```
+ sock_net_set(sk, hold_net(net));
}

extern void sock_enable_timestamp(struct sock *sk);
diff --git a/net/core/sock.c b/net/core/sock.c
index f2ccb16..015ec69 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -1001,6 +1001,7 @@ void sk_release_kernel(struct sock *sk)

    sock_hold(sk);
    sock_release(sk->sk_socket);
+ release_net(sock_net(sk));
    sock_net_set(sk, get_net(&init_net));
    sock_put(sk);
}
--
1.5.3.rc5
```

---

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

---

---

Subject: [PATCH 3/8 net-2.6.26] [NETNS]: Add netns refcnt debug for timewait buckets.

Posted by [den](#) on Tue, 15 Apr 2008 12:37:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

---  
net/ipv4/inet\_timewait\_sock.c | 3 +-  
1 files changed, 2 insertions(+), 1 deletions(-)

```
diff --git a/net/ipv4/inet_timewait_sock.c b/net/ipv4/inet_timewait_sock.c
index a741378..ce16e9a 100644
--- a/net/ipv4/inet_timewait_sock.c
+++ b/net/ipv4/inet_timewait_sock.c
@@ -57,6 +57,7 @@ void inet_twsks_put(struct inet_timewait_sock *tw)
    printk(KERN_DEBUG "%s timewait_sock %p released\n",
           tw->tw_prot->name, tw);
    #endif
+ release_net(twsk_net(tw));
    kmem_cache_free(tw->tw_prot->twsks_prot->twsks_slab, tw);
    module_put(owner);
}
@@ -124,7 +125,7 @@ struct inet_timewait_sock *inet_twsks_alloc(const struct sock *sk, const int
```

```
stat
tw->tw_hash = sk->sk_hash;
tw->tw_ipv6only = 0;
tw->tw_prot = sk->sk_prot_creator;
- twsk_net_set(tw, sock_net(sk));
+ twsk_net_set(tw, hold_net(sock_net(sk)));
  atomic_set(&tw->tw_refcnt, 1);
  inet_twsks_dead_node_init(tw);
  __module_get(tw->tw_prot->owner);
--
1.5.3.rc5
```

---

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

---

---

Subject: [PATCH 4/8 net-2.6.26] [NETNS]: Add netns refcnt debug into fib\_info.  
Posted by [den](#) on Tue, 15 Apr 2008 12:37:51 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

---  
net/ipv4/fib\_semantics.c | 3 ++-  
1 files changed, 2 insertions(+), 1 deletions(-)

diff --git a/net/ipv4/fib\_semantics.c b/net/ipv4/fib\_semantics.c

index a13c847..3b83c34 100644

--- a/net/ipv4/fib\_semantics.c

+++ b/net/ipv4/fib\_semantics.c

@@ -152,6 +152,7 @@ void free\_fib\_info(struct fib\_info \*fi)

nh->nh\_dev = NULL;

} endfor\_nexthops(fi);

fib\_info\_cnt--;

+ release\_net(fi->fib\_net);

kfree(fi);

}

@@ -730,7 +731,7 @@ struct fib\_info \*fib\_create\_info(struct fib\_config \*cfg)

goto failure;

fib\_info\_cnt++;

- fi->fib\_net = net;

+ fi->fib\_net = hold\_net(net);

fi->fib\_protocol = cfg->fc\_protocol;

fi->fib\_flags = cfg->fc\_flags;

fi->fib\_priority = cfg->fc\_priority;

--  
1.5.3.rc5

---

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

---

---

Subject: [PATCH 5/8 net-2.6.26] [NETNS]: Add netns refcnt debug for inet bind buckets.

Posted by [den](#) on Tue, 15 Apr 2008 12:37:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

---

net/ipv4/inet\_hashtables.c | 3 ++-  
1 files changed, 2 insertions(+), 1 deletions(-)

diff --git a/net/ipv4/inet\_hashtables.c b/net/ipv4/inet\_hashtables.c

index 32ca2f8..1612184 100644

--- a/net/ipv4/inet\_hashtables.c

+++ b/net/ipv4/inet\_hashtables.c

@@ -35,7 +35,7 @@ struct inet\_bind\_bucket \*inet\_bind\_bucket\_create(struct kmem\_cache  
\*cachep,  
struct inet\_bind\_bucket \*tb = kmem\_cache\_alloc(cachep, GFP\_ATOMIC);

if (tb != NULL) {

- tb->ib\_net = net;

+ tb->ib\_net = hold\_net(net);

tb->port = snum;

tb->fastreuse = 0;

INIT\_HLIST\_HEAD(&tb->owners);

@@ -51,6 +51,7 @@ void inet\_bind\_bucket\_destroy(struct kmem\_cache \*cachep, struct  
inet\_bind\_bucket

{  
if (hlist\_empty(&tb->owners)) {

\_\_hlist\_del(&tb->node);

+ release\_net(tb->ib\_net);

kmem\_cache\_free(cachep, tb);

}

}

--

1.5.3.rc5

---

Containers mailing list  
Containers@lists.linux-foundation.org

Subject: [PATCH 6/8 net-2.6.26] [NETNS]: Add netns refcnt debug for dst ops.

Posted by [den](#) on Tue, 15 Apr 2008 12:37:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

---

net/ipv6/route.c | 4 +++-

1 files changed, 3 insertions(+), 1 deletions(-)

diff --git a/net/ipv6/route.c b/net/ipv6/route.c

index 6293cb9..210a079 100644

--- a/net/ipv6/route.c

+++ b/net/ipv6/route.c

@@ -2622,7 +2622,7 @@ static int ip6\_route\_net\_init(struct net \*net)  
 GFP\_KERNEL);

if (!net->ipv6.ip6\_dst\_ops)

goto out;

- net->ipv6.ip6\_dst\_ops->dst\_net = net;

+ net->ipv6.ip6\_dst\_ops->dst\_net = hold\_net(net);

net->ipv6.ip6\_null\_entry = kmemdup(&ip6\_null\_entry\_template,  
 sizeof(\*net->ipv6.ip6\_null\_entry),

@@ -2669,6 +2669,7 @@ out:

return ret;

out\_ip6\_dst\_ops:

+ release\_net(net->ipv6.ip6\_dst\_ops->dst\_net);

kfree(net->ipv6.ip6\_dst\_ops);

goto out;

}

@@ -2684,6 +2685,7 @@ static void ip6\_route\_net\_exit(struct net \*net)

kfree(net->ipv6.ip6\_prohibit\_entry);

kfree(net->ipv6.ip6\_blk\_hole\_entry);

#endif

+ release\_net(net->ipv6.ip6\_dst\_ops->dst\_net);

kfree(net->ipv6.ip6\_dst\_ops);

}

--

1.5.3.rc5

---

Containers mailing list

[Containers@lists.linux-foundation.org](mailto:Containers@lists.linux-foundation.org)

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---



Subject: [PATCH 7/8 net-2.6.26] [NETNS]: Add netns refcnt debug to fib rules.

Posted by [den](#) on Tue, 15 Apr 2008 12:37:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Signed-off-by: Denis V. Lunev <den@openvz.org>

---

```
include/net/fib_rules.h | 1 +
net/core/fib_rules.c   | 5 +++--
2 files changed, 4 insertions(+), 2 deletions(-)
```

```
diff --git a/include/net/fib_rules.h b/include/net/fib_rules.h
```

```
index 34349f9..a5c6ccc 100644
```

```
--- a/include/net/fib_rules.h
```

```
+++ b/include/net/fib_rules.h
```

```
@@ -87,6 +87,7 @@ static inline void fib_rule_get(struct fib_rule *rule)
static inline void fib_rule_put_rcu(struct rcu_head *head)
{
    struct fib_rule *rule = container_of(head, struct fib_rule, rcu);
+ release_net(rule->fr_net);
    kfree(rule);
}
```

```
diff --git a/net/core/fib_rules.c b/net/core/fib_rules.c
```

```
index 540c072..e3e9ab0 100644
```

```
--- a/net/core/fib_rules.c
```

```
+++ b/net/core/fib_rules.c
```

```
@@ -29,7 +29,7 @@ int fib_default_rule_add(struct fib_rules_ops *ops,
    r->pref = pref;
    r->table = table;
    r->flags = flags;
- r->fr_net = ops->fro_net;
+ r->fr_net = hold_net(ops->fro_net);
```

```
/* The lock is not required here, the list in unreachable
```

```
* at the moment this function is called */
```

```
@@ -243,7 +243,7 @@ static int fib_nl_newrule(struct sk_buff *skb, struct nlmsg_hdr* nlh, void
*arg)
    err = -ENOMEM;
    goto errout;
}
```

```
- rule->fr_net = net;
```

```
+ rule->fr_net = hold_net(net);
```

```
if (tb[FRA_PRIORITY])
```

```
    rule->pref = nla_get_u32(tb[FRA_PRIORITY]);
```

```
@@ -344,6 +344,7 @@ static int fib_nl_newrule(struct sk_buff *skb, struct nlmsg_hdr* nlh, void
*arg)
    return 0;
```

```
errout_free:
+ release_net(rule->fr_net);
  kfree(rule);
errout:
  rules_ops_put(ops);
--
1.5.3.rc5
```

---

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

---

---

Subject: [PATCH 8/8 net-2.6.26] [NETNS]: Add netns refcnt debug for network devices.

Posted by [den](#) on Tue, 15 Apr 2008 12:37:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

dev\_set\_net is called for

- just allocated devices
- devices moving from one namespace to another

release\_net has proper check inside to distinguish these cases.

Signed-off-by: Denis V. Lunev <[den@openvz.org](mailto:den@openvz.org)>

---  
include/linux/netdevice.h | 3 +-  
net/core/dev.c | 2 ++  
2 files changed, 4 insertions(+), 1 deletions(-)

```
diff --git a/include/linux/netdevice.h b/include/linux/netdevice.h
index 8b17ed4..7c1d446 100644
--- a/include/linux/netdevice.h
+++ b/include/linux/netdevice.h
@@ -758,7 +758,8 @@ static inline
void dev_net_set(struct net_device *dev, struct net *net)
{
#ifdef CONFIG_NET_NS
- dev->nd_net = net;
+ release_net(dev->nd_net);
+ dev->nd_net = hold_net(net);
#endif
}
```

```
diff --git a/net/core/dev.c b/net/core/dev.c
index 7aa0112..77530e9 100644
--- a/net/core/dev.c
+++ b/net/core/dev.c
```

```
@@ -4042,6 +4042,8 @@ EXPORT_SYMBOL(alloc_netdev_mq);
*/
void free_netdev(struct net_device *dev)
{
+ release_net(dev_net(dev));
+
/* Compatibility with error handling in drivers */
if (dev->reg_state == NETREG_UNINITIALIZED) {
    kfree((char *)dev - dev->padded);
}
--
1.5.3.rc5
```

---

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

---

---

Subject: Re: [PATCH 1/8 net-2.6.26] [NETNS]: Make netns refcounting debug like a socket one.

Posted by [Brian Haley](#) on Tue, 15 Apr 2008 14:55:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Denis V. Lunev wrote:

```
> +#ifdef NETNS_REFCNT_DEBUG
> static inline struct net *hold_net(struct net *net)
> {
> + if (net == NULL)
> + return NULL;
> + atomic_inc(&net->use_count);
> return net;
> }
```

This could be shrunk to:

```
if (net)
    atomic_inc(&net->use_count);
return net;
```

```
> static inline void release_net(struct net *net)
> {
> + if (net == NULL)
> + return;
> + atomic_dec(&net->use_count);
> }
```

This one too:

```
if (net)
    atomic_dec(&net->use_count);
```

-Brian

---

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

---

---

Subject: Re: [PATCH 1/8 net-2.6.26] [NETNS]: Make netns refcounting debug like a socket one.

Posted by [davem](#) on Wed, 16 Apr 2008 08:58:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

From: Brian Haley <brian.haley@hp.com>

Date: Tue, 15 Apr 2008 10:55:20 -0400

```
> Denis V. Lunev wrote:
> > +#ifdef NETNS_REFCNT_DEBUG
> > static inline struct net *hold_net(struct net *net)
> > {
> > + if (net == NULL)
> > + return NULL;
> > + atomic_inc(&net->use_count);
> > return net;
> > }
```

```
>
> This could be shrunk to:
```

```
...
> This one too:
```

I've checked in Denis's patch with those simplifications added.  
Thanks everyone.

---

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

---

---

Subject: Re: [PATCH 0/8 net-2.6.26] [NETNS]: namespace refcounting cleanup

Posted by [davem](#) on Wed, 16 Apr 2008 09:07:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

From: "Denis V. Lunev" <den@openvz.org>

Date: Tue, 15 Apr 2008 16:35:59 +0400

> Network namespace has two reference counters:  
> - count  
> - use\_count.  
> The namespace is scheduled to destruction automatically when the count  
> becomes 0.  
>  
> There are several SLAB objects with a pointer to a namespace on them.  
> These objects are cleaned up during namespace stop. Some of them  
> increment use\_count, some don't. This set fixes this discrepancy,  
> i.e. now all such objects increment the use\_count.  
>  
> Though, the use\_count itself is used in a very debug manner and checked  
> only during namespace stop. So, I have placed it under  
> NETNS\_REFCNT\_DEBUG definition exactly like this is done for socket  
> refcounting code to remove extra atomic from any possible fast paths.  
>  
> Signed-off-by: Denis V. Lunev <den@openvz.org>

This is a notification that the Linux NETDEV Patch Robot has used all automated tools known to man to validate your patches, and has decided to apply them to the net-2.6.26 GIT tree.

\*beep\* \*beep\*

You may continue hacking.

\*beep\* \*beep\*

Have a pleasant day.

---

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

---