
Subject: [PATCH 0/4 net-2.6.15][UNIX] Make unix sysctls per-namespace

Posted by [Pavel Emelianov](#) on Fri, 30 Nov 2007 16:23:53 GMT

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

Hi, Herbert, guys.

Since you have accepted some net-namespaces-related work from Eric (sysctl namespaces), I hope, that you can pay some attention to further work in this direction.

This set makes the unix-sockets sysctls (currently this includes the sys/net/unix/max_dgram_qlen only) per net namespace.

I splitted it into four patches, to make the review simpler. Hope that this split will help.

This set resembles the one Eric has in his netns tree, but differs in some (maybe minor) ways.

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

Containers mailing list

Containers@lists.linux-foundation.org

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

Subject: [PATCH 1/4 net-2.6.25][UNIX] Extend unix_sysctl_(un)register prototypes

Posted by [Pavel Emelianov](#) on Fri, 30 Nov 2007 16:26:51 GMT

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

Add the struct net * argument to both of them to use in the future. Also make the register one return an error code.

It is useless right now, but will make the future patches much simpler.

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

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

```
index a1c805d..e0fba26 100644
```

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

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

```
@@ -60,11 +60,11 @@ struct unix_sock {
```

```
#ifdef CONFIG_SYSCTL
```

```

extern int sysctl_unix_max_dgram_qlen;
-extern void unix_sysctl_register(void);
-extern void unix_sysctl_unregister(void);
+extern int unix_sysctl_register(struct net *net);
+extern void unix_sysctl_unregister(struct net *net);
#else
-static inline void unix_sysctl_register(void) {}
-static inline void unix_sysctl_unregister(void) {}
+static inline int unix_sysctl_register(struct net *net) { return 0; }
+static inline void unix_sysctl_unregister(struct net *net) {}
#endif
#endif
#endif
diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 393197a..a0aa6d3 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -2175,7 +2175,7 @@ static int __init af_unix_init(void)

```

```

    sock_register(&unix_family_ops);
    register_pernet_subsys(&unix_net_ops);
- unix_sysctl_register();
+ unix_sysctl_register(&init_net);
out:
    return rc;
}
@@ -2183,7 +2183,7 @@ out:
static void __exit af_unix_exit(void)
{
    sock_unregister(PF_UNIX);
- unix_sysctl_unregister();
+ unix_sysctl_unregister(&init_net);
    proto_unregister(&unix_proto);
    unregister_pernet_subsys(&unix_net_ops);
}
diff --git a/net/unix/sysctl_net_unix.c b/net/unix/sysctl_net_unix.c
index eb0bd57..b2e0407 100644
--- a/net/unix/sysctl_net_unix.c
+++ b/net/unix/sysctl_net_unix.c
@@ -48,12 +48,13 @@ static ctl_table unix_root_table[] = {

```

```

static struct ctl_table_header * unix_sysctl_header;

-void unix_sysctl_register(void)
+int unix_sysctl_register(struct net *net)
{
    unix_sysctl_header = register_sysctl_table(unix_root_table);
+ return unix_sysctl_header == NULL ? -ENOMEM : 0;

```

```

}

-void unix_sysctl_unregister(void)
+void unix_sysctl_unregister(struct net *net)
{
    unregister_sysctl_table(unix_sysctl_header);
}
--
1.5.3.4

```

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

Subject: [PATCH 2/4 net-2.6.25][UNIX] Move the sysctl_unix_max_dgram_qlen on struct net
Posted by [Pavel Emelianov](#) on Fri, 30 Nov 2007 16:29:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

This will make all the sub-namespaces always use the default value (10) and leave the tuning via sysctl to the init namespace only.

Per-namespace tuning is coming.

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

```

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index 235214c..b0cf075 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -38,6 +38,9 @@ struct net {
    /* List of all packet sockets. */
    rwlock_t packet_sklist_lock;
    struct hlist_head packet_sklist;
+
+ /* unix sockets */
+ int sysctl_unix_max_dgram_qlen;
};

```

```

#ifdef CONFIG_NET
diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index a0aa6d3..73620d6 100644
--- a/net/unix/af_unix.c

```

```

+++ b/net/unix/af_unix.c
@@ -117,8 +117,6 @@
#include <net/checksum.h>
#include <linux/security.h>

-int sysctl_unix_max_dgram_qlen __read_mostly = 10;
-
static struct hlist_head unix_socket_table[UNIX_HASH_SIZE + 1];
static DEFINE_SPINLOCK(unix_table_lock);
static atomic_t unix_nr_socks = ATOMIC_INIT(0);
@@ -594,7 +592,7 @@ static struct sock * unix_create1(struct net *net, struct socket *sock)
    &af_unix_sk_receive_queue_lock_key);

    sk->sk_write_space = unix_write_space;
- sk->sk_max_ack_backlog = sysctl_unix_max_dgram_qlen;
+ sk->sk_max_ack_backlog = net->sysctl_unix_max_dgram_qlen;
    sk->sk_destruct = unix_sock_destructor;
    u = unix_sk(sk);
    u->dentry = NULL;
@@ -2140,6 +2138,8 @@ static int unix_net_init(struct net *net)
{
    int error = -ENOMEM;

+ net->sysctl_unix_max_dgram_qlen = 10;
+
#ifdef CONFIG_PROC_FS
    if (!proc_net_fops_create(net, "unix", 0, &unix_seq_fops))
        goto out;
diff --git a/net/unix/sysctl_net_unix.c b/net/unix/sysctl_net_unix.c
index b2e0407..c46cec0 100644
--- a/net/unix/sysctl_net_unix.c
+++ b/net/unix/sysctl_net_unix.c
@@ -18,7 +18,7 @@ static ctl_table unix_table[] = {
{
    .ctl_name = NET_UNIX_MAX_DGRAM_QLEN,
    .procname = "max_dgram_qlen",
- .data = &sysctl_unix_max_dgram_qlen,
+ .data = &init_net.sysctl_unix_max_dgram_qlen,
    .maxlen = sizeof(int),
    .mode = 0644,
    .proc_handler = &proc_dointvec
--
1.5.3.4

```

Subject: [PATCH 3/4 net-2.6.25][UNIX] Use ctl paths to register unix ctl tables
Posted by [Pavel Emelianov](#) on Fri, 30 Nov 2007 16:30:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Unlike previous ones, this patch is useful by its own,
as it decreases the vmlinux size :)

But it will be used later, when the per-namespace sysctl
is added.

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

```
diff --git a/net/unix/sysctl_net_unix.c b/net/unix/sysctl_net_unix.c
```

```
index c46cec0..43dd356 100644
```

```
--- a/net/unix/sysctl_net_unix.c
```

```
+++ b/net/unix/sysctl_net_unix.c
```

```
@@ -26,31 +26,17 @@ static ctl_table unix_table[] = {  
    { .ctl_name = 0 }  
};
```

```
-static ctl_table unix_net_table[] = {
```

```
- {  
-    .ctl_name = NET_UNIX,  
-    .procname = "unix",  
-    .mode = 0555,  
-    .child = unix_table  
- },
```

```
- { .ctl_name = 0 }
```

```
-};
```

```
-
```

```
-static ctl_table unix_root_table[] = {
```

```
- {  
-    .ctl_name = CTL_NET,  
-    .procname = "net",  
-    .mode = 0555,  
-    .child = unix_net_table  
- },
```

```
- { .ctl_name = 0 }
```

```
+static struct ctl_path unix_path[] = {
```

```
+ { .procname = "net", .ctl_name = CTL_NET, },
```

```
+ { .procname = "unix", .ctl_name = NET_UNIX, },
```

```
+ { },
```

```
};
```

```
static struct ctl_table_header * unix_sysctl_header;
```

```
int unix_sysctl_register(struct net *net)
```

```

{
- unix_sysctl_header = register_sysctl_table(unix_root_table);
+ unix_sysctl_header = register_sysctl_paths(unix_path, unix_table);
  return unix_sysctl_header == NULL ? -ENOMEM : 0;
}

```

--

1.5.3.4

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

Subject: [PATCH 4/4 net-2.6.25][UNIX] Make the unix sysctl tables per-namespace
Posted by [Pavel Emelianov](#) on Fri, 30 Nov 2007 16:34:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is the core.

- * add the ctl_table_header on the struct net;
- * make the unix_sysctl_register and _unregister clone the table;
- * moves calls to them into per-net init and exit callbacks;
- * move the .data pointer in the proper place.

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

```

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index b0cf075..f97b2a4 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -11,6 +11,8 @@
 struct proc_dir_entry;
 struct net_device;
 struct sock;
+struct ctl_table_header;
+
 struct net {
  atomic_t count; /* To decided when the network
    * namespace should be freed.
@@ -41,6 +43,7 @@ struct net {

  /* unix sockets */
  int sysctl_unix_max_dgram_qlen;
+ struct ctl_table_header *unix_ctl;

```

```

};

#ifdef CONFIG_NET
diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 73620d6..b8a2189 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -2139,10 +2139,14 @@ static int unix_net_init(struct net *net)
    int error = -ENOMEM;

    net->sysctl_unix_max_dgram_qlen = 10;
+ if (unix_sysctl_register(net))
+ goto out;

#ifdef CONFIG_PROC_FS
- if (!proc_net_fops_create(net, "unix", 0, &unix_seq_fops))
+ if (!proc_net_fops_create(net, "unix", 0, &unix_seq_fops)) {
+ unix_sysctl_unregister(net);
    goto out;
+ }
#endif
    error = 0;
out:
@@ -2151,6 +2155,7 @@ out:

static void unix_net_exit(struct net *net)
{
+ unix_sysctl_unregister(net);
    proc_net_remove(net, "unix");
}

@@ -2175,7 +2180,6 @@ static int __init af_unix_init(void)

    sock_register(&unix_family_ops);
    register_pernet_subsys(&unix_net_ops);
- unix_sysctl_register(&init_net);
out:
    return rc;
}
@@ -2183,7 +2187,6 @@ out:
static void __exit af_unix_exit(void)
{
    sock_unregister(PF_UNIX);
- unix_sysctl_unregister(&init_net);
    proto_unregister(&unix_proto);
    unregister_pernet_subsys(&unix_net_ops);
}
diff --git a/net/unix/sysctl_net_unix.c b/net/unix/sysctl_net_unix.c

```

index 43dd356..1c57f8f 100644

--- a/net/unix/sysctl_net_unix.c

+++ b/net/unix/sysctl_net_unix.c

```
@@ -32,16 +32,33 @@ static struct ctl_path unix_path[] = {
    {},
};
```

```
-static struct ctl_table_header * unix_sysctl_header;
```

```
-
```

```
int unix_sysctl_register(struct net *net)
{
- unix_sysctl_header = register_sysctl_paths(unix_path, unix_table);
- return unix_sysctl_header == NULL ? -ENOMEM : 0;
+ struct ctl_table *table;
+
+ table = kmemdup(unix_table, sizeof(unix_table), GFP_KERNEL);
+ if (table == NULL)
+ goto err_alloc;
+
+ table[0].data = &net->sysctl_unix_max_dgram_qlen;
+ net->unix_ctl = register_net_sysctl_table(net, unix_path, table);
+ if (net->unix_ctl != NULL)
+ goto err_reg;
+
+ return 0;
+
+err_reg:
+ kfree(table);
+err_alloc:
+ return -ENOMEM;
}
```

```
void unix_sysctl_unregister(struct net *net)
{
- unregister_sysctl_table(unix_sysctl_header);
+ struct ctl_table *table;
+
+ table = net->unix_ctl->ctl_table_arg;
+ unregister_sysctl_table(net->unix_ctl);
+ kfree(table);
}
```

```
--
```

1.5.3.4

Containers mailing list

Containers@lists.linux-foundation.org

Subject: [PATCH 4/4 (resent) net-2.6.25][UNIX] Make the unix sysctl tables per-namespace

Posted by [Pavel Emelianov](#) on Fri, 30 Nov 2007 16:37:28 GMT

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

I'm awfully sorry, but I noticed, that I sent the wrong patch right after I pressed the "Send" button. The first version contains a fatal error - the return code in register function should be inverted (the if (net->unix_ctl != NULL) one). Otherwise the registered table will be freed :(

Sorry. This is the correct patch.

This is the core.

- * add the ctl_table_header on the struct net;
- * make the unix_sysctl_register and _unregister clone the table;
- * moves calls to them into per-net init and exit callbacks;
- * move the .data pointer in the proper place.

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

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h

index b0cf075..f97b2a4 100644

--- a/include/net/net_namespace.h

+++ b/include/net/net_namespace.h

@@ -11,6 +11,8 @@

struct proc_dir_entry;

struct net_device;

struct sock;

+struct ctl_table_header;

+

struct net {

atomic_t count; /* To decided when the network

* namespace should be freed.

@@ -41,6 +43,7 @@ struct net {

/* unix sockets */

int sysctl_unix_max_dgram_qlen;

+ struct ctl_table_header *unix_ctl;

};

#ifdef CONFIG_NET

```

diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 73620d6..b8a2189 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -2139,10 +2139,14 @@ static int unix_net_init(struct net *net)
    int error = -ENOMEM;

    net->sysctl_unix_max_dgram_qlen = 10;
+ if (unix_sysctl_register(net))
+ goto out;

#ifdef CONFIG_PROC_FS
- if (!proc_net_fops_create(net, "unix", 0, &unix_seq_fops))
+ if (!proc_net_fops_create(net, "unix", 0, &unix_seq_fops)) {
+ unix_sysctl_unregister(net);
    goto out;
+ }
#endif
    error = 0;
out:
@@ -2151,6 +2155,7 @@ out:

static void unix_net_exit(struct net *net)
{
+ unix_sysctl_unregister(net);
    proc_net_remove(net, "unix");
}

@@ -2175,7 +2180,6 @@ static int __init af_unix_init(void)

    sock_register(&unix_family_ops);
    register_pernet_subsys(&unix_net_ops);
- unix_sysctl_register(&init_net);
out:
    return rc;
}
@@ -2183,7 +2187,6 @@ out:
static void __exit af_unix_exit(void)
{
    sock_unregister(PF_UNIX);
- unix_sysctl_unregister(&init_net);
    proto_unregister(&unix_proto);
    unregister_pernet_subsys(&unix_net_ops);
}
diff --git a/net/unix/sysctl_net_unix.c b/net/unix/sysctl_net_unix.c
index 43dd356..1c57f8f 100644
--- a/net/unix/sysctl_net_unix.c
+++ b/net/unix/sysctl_net_unix.c

```

```

@@ -32,16 +32,33 @@ static struct ctl_path unix_path[] = {
    {},
};

-static struct ctl_table_header * unix_sysctl_header;
-
int unix_sysctl_register(struct net *net)
{
- unix_sysctl_header = register_sysctl_paths(unix_path, unix_table);
- return unix_sysctl_header == NULL ? -ENOMEM : 0;
+ struct ctl_table *table;
+
+ table = kmemdup(unix_table, sizeof(unix_table), GFP_KERNEL);
+ if (table == NULL)
+ goto err_alloc;
+
+ table[0].data = &net->sysctl_unix_max_dgram_qlen;
+ net->unix_ctl = register_net_sysctl_table(net, unix_path, table);
+ if (net->unix_ctl == NULL)
+ goto err_reg;
+
+ return 0;
+
+err_reg:
+ kfree(table);
+err_alloc:
+ return -ENOMEM;
}

void unix_sysctl_unregister(struct net *net)
{
- unregister_sysctl_table(unix_sysctl_header);
+ struct ctl_table *table;
+
+ table = net->unix_ctl->ctl_table_arg;
+ unregister_sysctl_table(net->unix_ctl);
+ kfree(table);
}

```

--
1.5.3.4

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

Subject: Re: [PATCH 0/4 net-2.6.15][UNIX] Make unix sysctls per-namespaces
Posted by [ebiederm](#) on Fri, 30 Nov 2007 22:23:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Pavel Emelyanov <xemul@openvz.org> writes:

> Hi, Herbert, guys.
>
> Since you have accepted some net-namespaces-related work
> from Eric (sysctl namespaces), I hope, that you can pay
> some attention to further work in this direction.
>
> This set makes the unix-sockets sysctls (currently this
> includes the sys/net/unix/max_dgram_qlen only) per net
> namespace.
>
> I splitted it into four patches, to make the review simpler.
> Hope that this split will help.
>
> This set resembles the one Eric has in his netns tree, but
> differs in some (maybe minor) ways.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Looks sane skimming through the patches.

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

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

Subject: Re: [PATCH 4/4 (resent) net-2.6.25][UNIX] Make the unix sysctl tables
per-namespaces
Posted by [Herbert Xu](#) on Sat, 01 Dec 2007 12:57:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, Nov 30, 2007 at 07:37:28PM +0300, Pavel Emelyanov wrote:

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

All applied to net-2.6.25.

> diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
> index b0cf075..f97b2a4 100644
> --- a/include/net/net_namespace.h

```
> +++ b/include/net/net_namespace.h
> @@ -41,6 +43,7 @@ struct net {
>
> /* unix sockets */
> int sysctl_unix_max_dgram_qlen;
> + struct ctl_table_header *unix_ctl;
> };
```

But I gotta say this struct/file is going to be enormous. It's also one of those files that causes everything to get recompiled. Maybe we ought to make a rule that each subsystem only gets to have at most one entry in it :)

Thanks,

--

Visit Openswan at <http://www.openswan.org/>

Email: Herbert Xu ~{PmV>Hl~} <herbert@gondor.apana.org.au>

Home Page: <http://gondor.apana.org.au/~herbert/>

PGP Key: <http://gondor.apana.org.au/~herbert/pubkey.txt>

Containers mailing list

Containers@lists.linux-foundation.org

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

Subject: Re: [PATCH 4/4 (resent) net-2.6.25][UNIX] Make the unix sysctl tables per-namespace

Posted by [den](#) on Sat, 01 Dec 2007 13:07:05 GMT

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

Herbert Xu wrote:

> On Fri, Nov 30, 2007 at 07:37:28PM +0300, Pavel Emelyanov wrote:

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

>

> All applied to net-2.6.25.

>

>> diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h

>> index b0cf075..f97b2a4 100644

>> --- a/include/net/net_namespace.h

>> +++ b/include/net/net_namespace.h

>> @@ -41,6 +43,7 @@ struct net {

>>

>> /* unix sockets */

>> int sysctl_unix_max_dgram_qlen;

>> + struct ctl_table_header *unix_ctl;

>> };

>

> But I gotta say this struct/file is going to be enormous. It's also

> one of those files that causes everything to get recompiled. Maybe
> we ought to make a rule that each subsystem only gets to have at most
> one entry in it :)
>
> Thanks,

Good point, thanks. We'll start thinking in that direction. Right now it
is not finally cursed with all staff around.

Regards,
Den

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

Subject: Re: [PATCH 4/4 (resent) net-2.6.25][UNIX] Make the unix sysctl tables
per-namespace
Posted by [Pavel Emelianov](#) on Sat, 01 Dec 2007 13:19:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Denis V. Lunev wrote:

> Herbert Xu wrote:
>> On Fri, Nov 30, 2007 at 07:37:28PM +0300, Pavel Emelyanov wrote:
>>> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>> All applied to net-2.6.25.
>>
>>> diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
>>> index b0cf075..f97b2a4 100644
>>> --- a/include/net/net_namespace.h
>>> +++ b/include/net/net_namespace.h
>>> @@ -41,6 +43,7 @@ struct net {
>>>
>>> /* unix sockets */
>>> int sysctl_unix_max_dgram_qlen;
>>> + struct ctl_table_header *unix_ctl;
>>> };
>> But I gotta say this struct/file is going to be enormous. It's also
>> one of those files that causes everything to get recompiled. Maybe
>> we ought to make a rule that each subsystem only gets to have at most
>> one entry in it :)
>>
>> Thanks,
>
> Good point, thanks. We'll start thinking in that direction. Right now it
> is not finally cursed with all staff around.

Agree, the point is good :) but it has one pitfall :(

Look, now we make `_one_` dereference to get any `net->xxx` variable (`sysctl`, `list head`, `lock`, etc). When we force each subsystem has it's "private" pointer on this, we'll make them take `_two_` dereferences. Before the whole net namespace stuff started we made `_zero_` dereferences :) This may tell upon the performance.

I'm not claiming that this is the major case against this idea, but when developing this idea, I think we should keep that fact in mind and pay good attention to performance regressions.

> Regards,
> Den

Thanks,
Pavel

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

Subject: Re: [PATCH 4/4 (resent) net-2.6.25][UNIX] Make the unix sysctl tables per-namespace

Posted by [ebiederm](#) on Sat, 01 Dec 2007 19:32:02 GMT

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

Pavel Emelyanov <xemul@openvz.org> writes:

>>> But I gotta say this struct/file is going to be enormous. It's also
>>> one of those files that causes everything to get recompiled. Maybe
>>> we ought to make a rule that each subsystem only gets to have at most
>>> one entry in it :)

>>>

>>> Thanks,

>>

>> Good point, thanks. We'll start thinking in that direction. Right now it
>> is not finally cursed with all stuff around.

>

> Agree, the point is good :) but it has one pitfall :(

>

> Look, now we make `_one_` dereference to get any `net->xxx` variable
> (`sysctl`, `list head`, `lock`, etc). When we force each subsystem
> has it's "private" pointer on this, we'll make them take `_two_`
> dereferences. Before the whole net namespace stuff started we
> made `_zero_` dereferences :) This may tell upon the performance.

>
> I'm not claiming that this is the major case against this idea,
> but when developing this idea, I think we should keep that fact
> in mind and pay good attention to performance regressions.

Currently in my proof of concept tree I am at 65 variables and 648 bytes. This includes patches that are largely complete for ipv4. In number of variables this is about half of the current struct net_device, so I think the usage looks manageable.

I agree that both performance and size are significant concerns, and that is essentially why struct net has the structure it does today.

I print the size of struct net out at boot, we have to actually look at struct net when we make changes, so I don't think size bloat is going to happen unnoticed.

By keeping the size below PAGE_SIZE, and keeping the number of variables per network subsystem few and small we should be ok.

The fact that changing struct net causes the core of the networking stack to recompile is an added bonus that should also discourage people from playing with it too much.

My recommendation is to keep an eye on struct net and if what we are doing there becomes a problem address it then.

Eric

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