
Subject: [PATCH] SUNRPC: create svc_xprt in proper network namespace

Posted by [Stanislav Kinsbursky](#) on Tue, 06 Dec 2011 10:23:04 GMT

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

This patch makes svc_xprt inherit network namespace link from it's socket.

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

```
---
include/linux/sunrpc/svc_xprt.h      |  2 +-
net/sunrpc/svc_xprt.c                |  6 +++---
net/sunrpc/svcsock.c                 |  8 +++++---
net/sunrpc/xprttdma/svc_rdma_transport.c |  2 +-
4 files changed, 10 insertions(+), 8 deletions(-)

diff --git a/include/linux/sunrpc/svc_xprt.h b/include/linux/sunrpc/svc_xprt.h
index 8620f79..18df516 100644
--- a/include/linux/sunrpc/svc_xprt.h
+++ b/include/linux/sunrpc/svc_xprt.h
@@ -109,7 +109,7 @@ static inline int register_xprt_user(struct svc_xprt *xprt, struct svc_xprt_user
 *u

int svc_reg_xprt_class(struct svc_xprt_class *);
void svc_unreg_xprt_class(struct svc_xprt_class *);
-void svc_xprt_init(struct svc_xprt_class *, struct svc_xprt *,
+void svc_xprt_init(struct net *, struct svc_xprt_class *, struct svc_xprt *,
    struct svc_serv *);
int svc_create_xprt(struct svc_serv *, const char *, struct net *,
    const int, const unsigned short, int);
diff --git a/net/sunrpc/svc_xprt.c b/net/sunrpc/svc_xprt.c
index 447cd0e..f8f0627 100644
--- a/net/sunrpc/svc_xprt.c
+++ b/net/sunrpc/svc_xprt.c
@@ -147,8 +147,8 @@ EXPORT_SYMBOL_GPL(svc_xprt_put);
 * Called by transport drivers to initialize the transport independent
 * portion of the transport instance.
 */
-void svc_xprt_init(struct svc_xprt_class *xcl, struct svc_xprt *xprt,
-    struct svc_serv *serv)
+void svc_xprt_init(struct net *net, struct svc_xprt_class *xcl,
+    struct svc_xprt *xprt, struct svc_serv *serv)
{
    memset(xprt, 0, sizeof(*xprt));
    xprt->xprt_class = xcl;
@@ -163,7 +163,7 @@ void svc_xprt_init(struct svc_xprt_class *xcl, struct svc_xprt *xprt,
    spin_lock_init(&xprt->xprt_lock);
    set_bit(XPT_BUSY, &xprt->xprt_flags);
    rpc_init_wait_queue(&xprt->xprt_bc_pending, "xprt_bc_pending");
```

```
- xprt->xpt_net = get_net(&init_net);
+ xprt->xpt_net = get_net(net);
}
EXPORT_SYMBOL_GPL(svc_xprt_init);
```

```
diff --git a/net/sunrpc/svcsock.c b/net/sunrpc/svcsock.c
index 71bed1c..277909e 100644
```

```
--- a/net/sunrpc/svcsock.c
+++ b/net/sunrpc/svcsock.c
@@ -739,7 +739,8 @@ static void svc_udp_init(struct svc_sock *svsk, struct svc_serv *serv)
{
    int err, level, optname, one = 1;

- svc_xprt_init(&svc_udp_class, &svsk->sk_xprt, serv);
+ svc_xprt_init(sock_net(svsk->sk_sock->sk), &svc_udp_class,
+    &svsk->sk_xprt, serv);
    clear_bit(XPT_CACHE_AUTH, &svsk->sk_xprt.xpt_flags);
    svsk->sk_sk->sk_data_ready = svc_udp_data_ready;
    svsk->sk_sk->sk_write_space = svc_write_space;
@@ -1343,7 +1344,8 @@ static void svc_tcp_init(struct svc_sock *svsk, struct svc_serv *serv)
{
    struct sock *sk = svsk->sk_sk;

- svc_xprt_init(&svc_tcp_class, &svsk->sk_xprt, serv);
+ svc_xprt_init(sock_net(svsk->sk_sock->sk), &svc_tcp_class,
+    &svsk->sk_xprt, serv);
    set_bit(XPT_CACHE_AUTH, &svsk->sk_xprt.xpt_flags);
    if (sk->sk_state == TCP_LISTEN) {
        dprintk("setting up TCP socket for listening\n");
@@ -1659,7 +1661,7 @@ static struct svc_xprt *svc_bc_create_socket(struct svc_serv *serv,
    return ERR_PTR(-ENOMEM);

    xprt = &svsk->sk_xprt;
- svc_xprt_init(&svc_tcp_bc_class, xprt, serv);
+ svc_xprt_init(net, &svc_tcp_bc_class, xprt, serv);

    serv->sv_bc_xprt = xprt;
```

```
diff --git a/net/sunrpc/xprtrdma/svc_rdma_transport.c b/net/sunrpc/xprtrdma/svc_rdma_transport.c
index ba1296d..894cb42 100644
```

```
--- a/net/sunrpc/xprtrdma/svc_rdma_transport.c
+++ b/net/sunrpc/xprtrdma/svc_rdma_transport.c
@@ -453,7 +453,7 @@ static struct svcxprt_rdma *rdma_create_xprt(struct svc_serv *serv,

    if (!cma_xprt)
        return NULL;
- svc_xprt_init(&svc_rdma_class, &cma_xprt->sc_xprt, serv);
+ svc_xprt_init(&init_net, &svc_rdma_class, &cma_xprt->sc_xprt, serv);
```

```
INIT_LIST_HEAD(&cma_xprt->sc_accept_q);
INIT_LIST_HEAD(&cma_xprt->sc_dto_q);
INIT_LIST_HEAD(&cma_xprt->sc_rq_dto_q);
```

Subject: Re: [PATCH] SUNRPC: create svc_xprt in proper network namespace
Posted by [Stanislav Kinsbursky](#) on Tue, 06 Dec 2011 12:44:25 GMT

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

> Stanislav Kinsbursky wrote:

>

> This patch makes svc_xprt inherit network namespace link from it's socket.

>

> Should be "its socket."

Yep, sure. Silly mistake.

Thanks.

--

Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH] SUNRPC: create svc_xprt in proper network namespace
Posted by [Jim Rees](#) on Tue, 06 Dec 2011 12:44:27 GMT

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

Stanislav Kinsbursky wrote:

This patch makes svc_xprt inherit network namespace link from it's socket.

Should be "its socket."

Subject: Re: [PATCH] SUNRPC: create svc_xprt in proper network namespace
Posted by [bfields](#) on Tue, 06 Dec 2011 20:48:51 GMT

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

On Tue, Dec 06, 2011 at 02:19:10PM +0300, Stanislav Kinsbursky wrote:

> This patch makes svc_xprt inherit network namespace link from it's socket.

And poor rdma gets left out. OK. Applying for 3.3 (with changelog typo fixed).

--b.

```

>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>
> ---
> include/linux/sunrpc/svc_xprt.h      | 2 +-
> net/sunrpc/svc_xprt.c                | 6 +++---
> net/sunrpc/svcsock.c                 | 8 +++++---
> net/sunrpc/xprtrdma/svc_rdma_transport.c | 2 +-
> 4 files changed, 10 insertions(+), 8 deletions(-)
>
> diff --git a/include/linux/sunrpc/svc_xprt.h b/include/linux/sunrpc/svc_xprt.h
> index 8620f79..18df516 100644
> --- a/include/linux/sunrpc/svc_xprt.h
> +++ b/include/linux/sunrpc/svc_xprt.h
> @@ -109,7 +109,7 @@ static inline int register_xprt_user(struct svc_xprt *xprt, struct
> svc_xprt_user *u
>
> int svc_reg_xprt_class(struct svc_xprt_class *);
> void svc_unreg_xprt_class(struct svc_xprt_class *);
> -void svc_xprt_init(struct svc_xprt_class *, struct svc_xprt *,
> +void svc_xprt_init(struct net *, struct svc_xprt_class *, struct svc_xprt *,
> struct svc_serv *);
> int svc_create_xprt(struct svc_serv *, const char *, struct net *,
> const int, const unsigned short, int);
> diff --git a/net/sunrpc/svc_xprt.c b/net/sunrpc/svc_xprt.c
> index 447cd0e..f8f0627 100644
> --- a/net/sunrpc/svc_xprt.c
> +++ b/net/sunrpc/svc_xprt.c
> @@ -147,8 +147,8 @@ EXPORT_SYMBOL_GPL(svc_xprt_put);
> * Called by transport drivers to initialize the transport independent
> * portion of the transport instance.
> */
> -void svc_xprt_init(struct svc_xprt_class *xcl, struct svc_xprt *xprt,
> - struct svc_serv *serv)
> +void svc_xprt_init(struct net *net, struct svc_xprt_class *xcl,
> + struct svc_xprt *xprt, struct svc_serv *serv)
> {
>     memset(xprt, 0, sizeof(*xprt));
>     xprt->xprt_class = xcl;
> @@ -163,7 +163,7 @@ void svc_xprt_init(struct svc_xprt_class *xcl, struct svc_xprt *xprt,
> spin_lock_init(&xprt->xprt_lock);
> set_bit(XPT_BUSY, &xprt->xprt_flags);
> rpc_init_wait_queue(&xprt->xprt_bc_pending, "xprt_bc_pending");
> - xprt->xprt_net = get_net(&init_net);
> + xprt->xprt_net = get_net(net);
> }
> EXPORT_SYMBOL_GPL(svc_xprt_init);
>

```

```

> diff --git a/net/sunrpc/svcsock.c b/net/sunrpc/svcsock.c
> index 71bed1c..277909e 100644
> --- a/net/sunrpc/svcsock.c
> +++ b/net/sunrpc/svcsock.c
> @@ -739,7 +739,8 @@ static void svc_udp_init(struct svc_sock *svsk, struct svc_serv *serv)
> {
>     int err, level, optname, one = 1;
>
>     - svc_xprt_init(&svc_udp_class, &svsk->sk_xprt, serv);
> + svc_xprt_init(sock_net(svsk->sk_socket->sk), &svc_udp_class,
> +     &svsk->sk_xprt, serv);
>     clear_bit(XPT_CACHE_AUTH, &svsk->sk_xprt.xpt_flags);
>     svsk->sk_sk->sk_data_ready = svc_udp_data_ready;
>     svsk->sk_sk->sk_write_space = svc_write_space;
> @@ -1343,7 +1344,8 @@ static void svc_tcp_init(struct svc_sock *svsk, struct svc_serv *serv)
> {
>     struct sock *sk = svsk->sk_sk;
>
>     - svc_xprt_init(&svc_tcp_class, &svsk->sk_xprt, serv);
> + svc_xprt_init(sock_net(svsk->sk_socket->sk), &svc_tcp_class,
> +     &svsk->sk_xprt, serv);
>     set_bit(XPT_CACHE_AUTH, &svsk->sk_xprt.xpt_flags);
>     if (sk->sk_state == TCP_LISTEN) {
>         dprintk("setting up TCP socket for listening\n");
> @@ -1659,7 +1661,7 @@ static struct svc_xprt *svc_bc_create_socket(struct svc_serv *serv,
>     return ERR_PTR(-ENOMEM);
>
>     xprt = &svsk->sk_xprt;
>     - svc_xprt_init(&svc_tcp_bc_class, xprt, serv);
> + svc_xprt_init(net, &svc_tcp_bc_class, xprt, serv);
>
>     serv->sv_bc_xprt = xprt;
>
> diff --git a/net/sunrpc/xprtrdma/svc_rdma_transport.c
b/net/sunrpc/xprtrdma/svc_rdma_transport.c
> index ba1296d..894cb42 100644
> --- a/net/sunrpc/xprtrdma/svc_rdma_transport.c
> +++ b/net/sunrpc/xprtrdma/svc_rdma_transport.c
> @@ -453,7 +453,7 @@ static struct svcxprt_rdma *rdma_create_xprt(struct svc_serv *serv,
>
>     if (!cma_xprt)
>         return NULL;
>     - svc_xprt_init(&svc_rdma_class, &cma_xprt->sc_xprt, serv);
> + svc_xprt_init(&init_net, &svc_rdma_class, &cma_xprt->sc_xprt, serv);
>     INIT_LIST_HEAD(&cma_xprt->sc_accept_q);
>     INIT_LIST_HEAD(&cma_xprt->sc_dto_q);
>     INIT_LIST_HEAD(&cma_xprt->sc_rq_dto_q);
>

```

Subject: Re: [PATCH] SUNRPC: create svc_xprt in proper network namespace
Posted by [Stanislav Kinsbursky](#) on Wed, 07 Dec 2011 08:45:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

> On Tue, Dec 06, 2011 at 02:19:10PM +0300, Stanislav Kinsbursky wrote:
>> This patch makes svc_xprt inherit network namespace link from it's socket.
>
> And poor rdma gets left out.

Sorry, but this is not a part of my interest.

> OK. Applying for 3.3 (with changelog typo
> fixed).
>

Good, thanks.

--

Best regards,
Stanislav Kinsbursky
