
Subject: [PATCH 0/4] SUNRPC: pass network namespace context to sockaddr construction routines

Posted by [Stanislav Kinsbursky](#) on Wed, 07 Dec 2011 11:21:16 GMT

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

This is another precursor patch set aimed to remove hard-coded "init_net" reference from SUNRPC sockaddr conversion and construction routines.

The following series consists of:

Stanislav Kinsbursky (4):

SUNRPC: use passed network namespace context in rpc_parse_scope_id()

SUNRPC: use passed network namespace context in rpc_pton6()

SUNRPC: use passed network namespace context in rpc_pton()

SUNRPC: use passed network namespace context in rpc_uaddr2sockaddr()

```
fs/nfs/dns_resolve.c      | 4 ++--
fs/nfs/nfs4filelayoutdev.c | 2 +-
fs/nfs/nfs4namespace.c    | 2 +-
fs/nfs/super.c            | 4 ++--
fs/nfsd/nfs4state.c       | 2 +-
fs/nfsd/nfsctl.c          | 2 +-
include/linux/sunrpc/clnt.h | 4 ++--
net/sunrpc/addr.c         | 24 ++++++++-----
net/sunrpc/rpcb_clnt.c    | 2 +-
net/sunrpc/svcauth_unix.c | 2 +-
10 files changed, 25 insertions(+), 23 deletions(-)
```

--

Signature

Subject: [PATCH 1/4] SUNRPC: use passed network namespace context in rpc_parse_scope_id()

Posted by [Stanislav Kinsbursky](#) on Wed, 07 Dec 2011 11:22:51 GMT

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

Use incoming network context in rpc_parse_scope_id() instead of hard-coded "init_net".

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

```
net/sunrpc/addr.c | 9 +++++--
1 files changed, 5 insertions(+), 4 deletions(-)
```

```

diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c
index 67a655e..35eb188 100644
--- a/net/sunrpc/addr.c
+++ b/net/sunrpc/addr.c
@@ -156,8 +156,9 @@ static size_t rpc_pton4(const char *buf, const size_t buflen,
 }

#ifdef CONFIG_IPV6 || defined(CONFIG_IPV6_MODULE)
-static int rpc_parse_scope_id(const char *buf, const size_t buflen,
-    const char *delim, struct sockaddr_in6 *sin6)
+static int rpc_parse_scope_id(struct net *net, const char *buf,
+    const size_t buflen, const char *delim,
+    struct sockaddr_in6 *sin6)
{
    char *p;
    size_t len;
@@ -177,7 +178,7 @@ static int rpc_parse_scope_id(const char *buf, const size_t buflen,
    unsigned long scope_id = 0;
    struct net_device *dev;

-    dev = dev_get_by_name(&init_net, p);
+    dev = dev_get_by_name(net, p);
    if (dev != NULL) {
        scope_id = dev->ifindex;
        dev_put(dev);
@@ -213,7 +214,7 @@ static size_t rpc_pton6(const char *buf, const size_t buflen,
    if (in6_pton(buf, buflen, addr, IPV6_SCOPE_DELIMITER, &delim) == 0)
        return 0;

-    if (!rpc_parse_scope_id(buf, buflen, delim, sin6))
+    if (!rpc_parse_scope_id(&init_net, buf, buflen, delim, sin6))
        return 0;

    sin6->sin6_family = AF_INET6;

```

Subject: [PATCH 2/4] SUNRPC: use passed network namespace context in
rpc_pton6()

Posted by [Stanislav Kinsbursky](#) on Wed, 07 Dec 2011 11:24:03 GMT

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

Use incoming network context in rpc_pton6() instead of hard-coded "init_net".

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

net/sunrpc/addr.c | 8 ++++----

1 files changed, 4 insertions(+), 4 deletions(-)

diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c

index 35eb188..c13a521 100644

--- a/net/sunrpc/addr.c

+++ b/net/sunrpc/addr.c

```
@@ -198,7 +198,7 @@ static int rpc_parse_scope_id(struct net *net, const char *buf,
    return 0;
}
```

```
-static size_t rpc_pton6(const char *buf, const size_t buflen,
```

```
+static size_t rpc_pton6(struct net *net, const char *buf, const size_t buflen,
    struct sockaddr *sap, const size_t salen)
```

```
{
    struct sockaddr_in6 *sin6 = (struct sockaddr_in6 *)sap;
```

```
@@ -214,14 +214,14 @@ static size_t rpc_pton6(const char *buf, const size_t buflen,
    if (in6_pton(buf, buflen, addr, IPV6_SCOPE_DELIMITER, &delim) == 0)
        return 0;
```

```
- if (!rpc_parse_scope_id(&init_net, buf, buflen, delim, sin6))
```

```
+ if (!rpc_parse_scope_id(net, buf, buflen, delim, sin6))
    return 0;
```

```
    sin6->sin6_family = AF_INET6;
    return sizeof(struct sockaddr_in6);
}
```

```
#else
```

```
-static size_t rpc_pton6(const char *buf, const size_t buflen,
```

```
+static size_t rpc_pton6(struct net *net, const char *buf, const size_t buflen,
    struct sockaddr *sap, const size_t salen)
```

```
{
    return 0;
```

```
@@ -249,7 +249,7 @@ size_t rpc_pton(const char *buf, const size_t buflen,
```

```
    for (i = 0; i < buflen; i++)
        if (buff[i] == ':')
```

```
-    return rpc_pton6(buf, buflen, sap, salen);
```

```
+    return rpc_pton6(&init_net, buf, buflen, sap, salen);
    return rpc_pton4(buf, buflen, sap, salen);
}
```

```
EXPORT_SYMBOL_GPL(rpc_pton);
```

Subject: [PATCH 3/4] SUNRPC: use passed network namespace context in
rpc_pton()

Posted by [Stanislav Kinsbursky](#) on Wed, 07 Dec 2011 11:25:09 GMT

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

Use incoming network context in `rpc_pton()` instead of hard-coded "init_net".

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

```
fs/nfs/dns_resolve.c      | 4 ++--
fs/nfs/nfs4filelayoutdev.c | 2 +-
fs/nfs/nfs4namespace.c    | 2 +-
fs/nfs/super.c            | 4 ++--
fs/nfsd/nfsctl.c          | 2 +-
include/linux/sunrpc/clnt.h | 2 +-
net/sunrpc/addr.c         | 7 ++++---
net/sunrpc/svcauth_unix.c | 2 +-
8 files changed, 13 insertions(+), 12 deletions(-)
```

diff --git a/fs/nfs/dns_resolve.c b/fs/nfs/dns_resolve.c

index a6e711a..e698242 100644

--- a/fs/nfs/dns_resolve.c

+++ b/fs/nfs/dns_resolve.c

@@ -20,7 +20,7 @@ ssize_t nfs_dns_resolve_name(char *name, size_t namelen,

ip_len = dns_query(NULL, name, namelen, NULL, &ip_addr, NULL);

if (ip_len > 0)

- ret = rpc_pton(ip_addr, ip_len, sa, salen);

+ ret = rpc_pton(&init_net, ip_addr, ip_len, sa, salen);

else

ret = -ESRCH;

kfree(ip_addr);

@@ -224,7 +224,7 @@ static int nfs_dns_parse(struct cache_detail *cd, char *buf, int buflen)

len = qword_get(&buf, buf1, sizeof(buf1));

if (len <= 0)

goto out;

- key.addrlen = rpc_pton(buf1, len,

+ key.addrlen = rpc_pton(&init_net, buf1, len,

(struct sockaddr *)&key.addr,

sizeof(key.addr));

diff --git a/fs/nfs/nfs4filelayoutdev.c b/fs/nfs/nfs4filelayoutdev.c

index ed388aa..d6b5409 100644

--- a/fs/nfs/nfs4filelayoutdev.c

+++ b/fs/nfs/nfs4filelayoutdev.c

@@ -457,7 +457,7 @@ decode_ds_addr(struct xdr_stream *stream, gfp_t gfp_flags)

INIT_LIST_HEAD(&da->da_node);

- if (!rpc_pton(buf, portstr-buf, (struct sockaddr *)&da->da_addr,

+ if (!rpc_pton(&init_net, buf, portstr-buf, (struct sockaddr *)&da->da_addr,
sizeof(da->da_addr))) {

```

    dprintf("%s: error parsing address %s\n", __func__, buf);
    goto out_free_da;
diff --git a/fs/nfs/nfs4namespace.c b/fs/nfs/nfs4namespace.c
index bb80c49..85aa3da 100644
--- a/fs/nfs/nfs4namespace.c
+++ b/fs/nfs/nfs4namespace.c
@@ -98,7 +98,7 @@ static ssize_t nfs_parse_server_name(char *string, size_t len,
{
    ssize_t ret;

- ret = rpc_pton(string, len, sa, salen);
+ ret = rpc_pton(&init_net, string, len, sa, salen);
    if (ret == 0) {
        ret = nfs_dns_resolve_name(string, len, sa, salen);
        if (ret < 0)
diff --git a/fs/nfs/super.c b/fs/nfs/super.c
index 1347774..b31bdf4 100644
--- a/fs/nfs/super.c
+++ b/fs/nfs/super.c
@@ -1392,7 +1392,7 @@ static int nfs_parse_mount_options(char *raw,
    if (string == NULL)
        goto out_nomem;
    mnt->nfs_server.addrlen =
-    rpc_pton(string, strlen(string),
+    rpc_pton(&init_net, string, strlen(string),
        (struct sockaddr *)
        &mnt->nfs_server.address,
        sizeof(mnt->nfs_server.address));
@@ -1414,7 +1414,7 @@ static int nfs_parse_mount_options(char *raw,
    if (string == NULL)
        goto out_nomem;
    mnt->mount_server.addrlen =
-    rpc_pton(string, strlen(string),
+    rpc_pton(&init_net, string, strlen(string),
        (struct sockaddr *)
        &mnt->mount_server.address,
        sizeof(mnt->mount_server.address));
diff --git a/fs/nfsd/nfsctl.c b/fs/nfsd/nfsctl.c
index c45a2ea..f9d0599 100644
--- a/fs/nfsd/nfsctl.c
+++ b/fs/nfsd/nfsctl.c
@@ -222,7 +222,7 @@ static ssize_t write_unlock_ip(struct file *file, char *buf, size_t size)
    if (qword_get(&buf, fo_path, size) < 0)
        return -EINVAL;

- if (rpc_pton(fo_path, size, sap, salen) == 0)
+ if (rpc_pton(&init_net, fo_path, size, sap, salen) == 0)
        return -EINVAL;

```

```

    return nlmsvc_unlock_all_by_ip(sap);
diff --git a/include/linux/sunrpc/clnt.h b/include/linux/sunrpc/clnt.h
index 3d8f9c4..877a466 100644
--- a/include/linux/sunrpc/clnt.h
+++ b/include/linux/sunrpc/clnt.h
@@ -162,7 +162,7 @@ size_t rpc_peeraddr(struct rpc_clnt *, struct sockaddr *, size_t);
const char *rpc_peeraddr2str(struct rpc_clnt *, enum rpc_display_format_t);

size_t rpc_ntop(const struct sockaddr *, char *, const size_t);
-size_t rpc_pton(const char *, const size_t,
+size_t rpc_pton(struct net *, const char *, const size_t,
    struct sockaddr *, const size_t);
char * rpc_sockaddr2uaddr(const struct sockaddr *, gfp_t);
size_t rpc_uaddr2sockaddr(const char *, const size_t,
diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c
index c13a521..982fac7 100644
--- a/net/sunrpc/addr.c
+++ b/net/sunrpc/addr.c
@@ -230,6 +230,6 @@ static size_t rpc_pton6(struct net *net, const char *buf, const size_t
buflen,

/**
 * rpc_pton - Construct a sockaddr in @sap
+ * @net: applicable network namespace
 * @buf: C string containing presentation format IP address
 * @buflen: length of presentation address in bytes
 * @sap: buffer into which to plant socket address
@@ -242,14 +242,14 @@ static size_t rpc_pton6(struct net *net, const char *buf, const size_t
buflen,
 * socket address, if successful. Returns zero if an error
 * occurred.
 */
-size_t rpc_pton(const char *buf, const size_t buflen,
+size_t rpc_pton(struct net *net, const char *buf, const size_t buflen,
    struct sockaddr *sap, const size_t salen)
{
    unsigned int i;

    for (i = 0; i < buflen; i++)
        if (buf[i] == ':')
-    return rpc_pton6(&init_net, buf, buflen, sap, salen);
+    return rpc_pton6(net, buf, buflen, sap, salen);
    return rpc_pton4(buf, buflen, sap, salen);
}
EXPORT_SYMBOL_GPL(rpc_pton);
@@ -340,7 +341,7 @@ size_t rpc_uaddr2sockaddr(const char *uaddr, const size_t uaddr_len,
    port = (unsigned short)((porthi << 8) | portlo);

```

```

*c = '\0';
- if (rpc_pton(buf, strlen(buf), sap, salen) == 0)
+ if (rpc_pton(&init_net, buf, strlen(buf), sap, salen) == 0)
    return 0;

    switch (sap->sa_family) {
diff --git a/net/sunrpc/svcauth_unix.c b/net/sunrpc/svcauth_unix.c
index ce13632..a7d0141 100644
--- a/net/sunrpc/svcauth_unix.c
+++ b/net/sunrpc/svcauth_unix.c
@@ -211,7 +211,7 @@ static int ip_map_parse(struct cache_detail *cd,
    len = qword_get(&mesg, buf, mlen);
    if (len <= 0) return -EINVAL;

- if (rpc_pton(buf, len, &address.sa, sizeof(address)) == 0)
+ if (rpc_pton(&init_net, buf, len, &address.sa, sizeof(address)) == 0)
    return -EINVAL;
    switch (address.sa.sa_family) {
    case AF_INET:

```

Subject: [PATCH 4/4] SUNRPC: use passed network namespace context in
rpc_uaddr2sockaddr()

Posted by [Stanislav Kinsbursky](#) on Wed, 07 Dec 2011 11:27:19 GMT

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

Use incoming network context in rpc_uaddr2sockaddr() instead of hard-coded
"init_net".

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

```

---
fs/nfsd/nfs4state.c      | 2 +-
include/linux/sunrpc/clnt.h | 2 +-
net/sunrpc/addr.c        | 8 +++++--
net/sunrpc/rpcb_clnt.c    | 2 +-
4 files changed, 8 insertions(+), 6 deletions(-)

diff --git a/fs/nfsd/nfs4state.c b/fs/nfsd/nfs4state.c
index 47e94e3..e18f46f 100644
--- a/fs/nfsd/nfs4state.c
+++ b/fs/nfsd/nfs4state.c
@@ -1301,7 +1301,7 @@ gen_callback(struct nfs4_client *clp, struct nfsd4_setclientid *se, struct
svc_r
    else
        goto out_err;

```

```

- conn->cb_addrln = rpc_uaddr2sockaddr(se->se_callback_addr_val,
+ conn->cb_addrln = rpc_uaddr2sockaddr(&init_net, se->se_callback_addr_val,
    se->se_callback_addr_len,
    (struct sockaddr *)&conn->cb_addr,
    sizeof(conn->cb_addr));
diff --git a/include/linux/sunrpc/clnt.h b/include/linux/sunrpc/clnt.h
index 877a466..029fb48 100644
--- a/include/linux/sunrpc/clnt.h
+++ b/include/linux/sunrpc/clnt.h
@@ -165,7 +165,7 @@ size_t rpc_ntop(const struct sockaddr *, char *, const size_t);
size_t rpc_pton(struct net *, const char *, const size_t,
    struct sockaddr *, const size_t);
char * rpc_sockaddr2uaddr(const struct sockaddr *, gfp_t);
-size_t rpc_uaddr2sockaddr(const char *, const size_t,
+size_t rpc_uaddr2sockaddr(struct net *, const char *, const size_t,
    struct sockaddr *, const size_t);

static inline unsigned short rpc_get_port(const struct sockaddr *sap)
diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c
index 982fac7..bad31da 100644
--- a/net/sunrpc/addr.c
+++ b/net/sunrpc/addr.c
@@ -297,6 +297,7 @@ char *rpc_sockaddr2uaddr(const struct sockaddr *sap, gfp_t gfp_flags)

/**
 * rpc_uaddr2sockaddr - convert a universal address to a socket address.
+ * @net: applicable network namespace
 * @uaddr: C string containing universal address to convert
 * @uaddr_len: length of universal address string
 * @sap: buffer into which to plant socket address
@@ -308,9 +308,9 @@ char *rpc_sockaddr2uaddr(const struct sockaddr *sap, gfp_t gfp_flags)
 * Returns the size of the socket address if successful; otherwise
 * zero is returned.
 */
-size_t rpc_uaddr2sockaddr(const char *uaddr, const size_t uaddr_len,
-    struct sockaddr *sap, const size_t salen)
+size_t rpc_uaddr2sockaddr(struct net *net, const char *uaddr,
+    const size_t uaddr_len, struct sockaddr *sap,
+    const size_t salen)
{
    char *c, buf[RPCBIND_MAXUADDRLEN + sizeof('\0')];
    unsigned long portlo, porthi;
@@ -341,7 +341,7 @@ size_t rpc_uaddr2sockaddr(const char *uaddr, const size_t uaddr_len,
    port = (unsigned short)((porthi << 8) | portlo);

    *c = '\0';
- if (rpc_pton(&init_net, buf, strlen(buf), sap, salen) == 0)
+ if (rpc_pton(net, buf, strlen(buf), sap, salen) == 0)

```

```

return 0;

switch (sap->sa_family) {
diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
index 8761bf8..42e1b2e 100644
--- a/net/sunrpc/rpcb_clnt.c
+++ b/net/sunrpc/rpcb_clnt.c
@@ -925,7 +925,7 @@ static int rpcb_dec_getaddr(struct rpc_rqst *req, struct xdr_stream *xdr,
    dprintk("RPC: %5u RPCB_%s reply: %s\n", task->tk_pid,
        task->tk_msg.rpc_proc->p_name, (char *)p);

- if (rpc_uaddr2sockaddr((char *)p, len, sap, sizeof(address)) == 0)
+ if (rpc_uaddr2sockaddr(&init_net, (char *)p, len, sap, sizeof(address)) == 0)
    goto out_fail;
    rpcb->r_port = rpc_get_port(sap);

```

Subject: Re: [PATCH 0/4] SUNRPC: pass network namespace context to sockaddr construction routines
Posted by [bfields](#) on Thu, 08 Dec 2011 20:46:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, Dec 07, 2011 at 03:20:07PM +0300, Stanislav Kinsbursky wrote:
> This is another precursor patch set aimed to remove hard-coded "init_net"
> reference from SUNRPC sockaddr conversion and construction routines.

ACK--all look straightforward. (Only complaint would be that this isn't terribly interesting till we see how these are used.)

--b.

```

>
> The following series consists of:
>
> ---
>
> Stanislav Kinsbursky (4):
>   SUNRPC: use passed network namespace context in rpc_parse_scope_id()
>   SUNRPC: use passed network namespace context in rpc_pton6()
>   SUNRPC: use passed network namespace context in rpc_pton()
>   SUNRPC: use passed network namespace context in rpc_uaddr2sockaddr()
>
>
> fs/nfs/dns_resolve.c      | 4 ++--
> fs/nfs/nfs4filelayoutdev.c | 2 +-
> fs/nfs/nfs4namespace.c    | 2 +-
> fs/nfs/super.c            | 4 ++--

```

```
> fs/nfsd/nfs4state.c      | 2 +-
> fs/nfsd/nfsctl.c         | 2 +-
> include/linux/sunrpc/clnt.h | 4 +++-
> net/sunrpc/addr.c        | 24 ++++++++-----
> net/sunrpc/rpcb_clnt.c   | 2 +-
> net/sunrpc/svcauth_unix.c | 2 +-
> 10 files changed, 25 insertions(+), 23 deletions(-)
>
> --
> Signature
```

Subject: Re: [PATCH 1/4] SUNRPC: use passed network namespace context in
rpc_parse_scope_id()

Posted by [bfields](#) on Thu, 08 Dec 2011 20:47:49 GMT

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

On Wed, Dec 07, 2011 at 03:20:16PM +0300, Stanislav Kinsbursky wrote:

> Use incoming network context in rpc_parse_scope_id() instead of hard-coded

Changelogs are a little confusing; I might have said "allow
rpc_parse_scope_id() caller to pass in network context instead of using
hard-code "init_net"."

--b.

```
> "init_net".
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>
> ---
> net/sunrpc/addr.c | 9 +++++--
> 1 files changed, 5 insertions(+), 4 deletions(-)
>
> diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c
> index 67a655e..35eb188 100644
> --- a/net/sunrpc/addr.c
> +++ b/net/sunrpc/addr.c
> @@ -156,8 +156,9 @@ static size_t rpc_pton4(const char *buf, const size_t buflen,
> }
>
> #if defined(CONFIG_IPV6) || defined(CONFIG_IPV6_MODULE)
> -static int rpc_parse_scope_id(const char *buf, const size_t buflen,
> -    const char *delim, struct sockaddr_in6 *sin6)
> +static int rpc_parse_scope_id(struct net *net, const char *buf,
> +    const size_t buflen, const char *delim,
> +    struct sockaddr_in6 *sin6)
> {
```

```

> char *p;
> size_t len;
> @@ -177,7 +178,7 @@ static int rpc_parse_scope_id(const char *buf, const size_t buflen,
> unsigned long scope_id = 0;
> struct net_device *dev;
>
> - dev = dev_get_by_name(&init_net, p);
> + dev = dev_get_by_name(net, p);
> if (dev != NULL) {
>     scope_id = dev->ifindex;
>     dev_put(dev);
> @@ -213,7 +214,7 @@ static size_t rpc_pton6(const char *buf, const size_t buflen,
> if (in6_pton(buf, buflen, addr, IPV6_SCOPE_DELIMITER, &delim) == 0)
>     return 0;
>
> - if (!rpc_parse_scope_id(buf, buflen, delim, sin6))
> + if (!rpc_parse_scope_id(&init_net, buf, buflen, delim, sin6))
>     return 0;
>
> sin6->sin6_family = AF_INET6;
>

```

Subject: Re: [PATCH 1/4] SUNRPC: use passed network namespace context in
rpc_parse_scope_id()

Posted by [Stanislav Kinsbursky](#) on Mon, 12 Dec 2011 16:03:45 GMT

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

```

> On Wed, Dec 07, 2011 at 03:20:16PM +0300, Stanislav Kinsbursky wrote:
>> Use incoming network context in rpc_parse_scope_id() instead of hard-coded
>
> Changelogs are a little confusing; I might have said "allow
> rpc_parse_scope_id() caller to pass in network context instead of using
> hard-code "init_net"."
>

```

Hi, Bruce.

Probably, your variant is better. My English is not good enough to write
descriptive and, in the same time, clear and short comments to patches.

```

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

```

```

>> net/sunrpc/addr.c | 9 +++++----
>> 1 files changed, 5 insertions(+), 4 deletions(-)
>>
>> diff --git a/net/sunrpc/addr.c b/net/sunrpc/addr.c
>> index 67a655e..35eb188 100644
>> --- a/net/sunrpc/addr.c
>> +++ b/net/sunrpc/addr.c
>> @@ -156,8 +156,9 @@ static size_t rpc_pton4(const char *buf, const size_t buflen,
>> }
>>
>> #if defined(CONFIG_IPV6) || defined(CONFIG_IPV6_MODULE)
>> -static int rpc_parse_scope_id(const char *buf, const size_t buflen,
>> -    const char *delim, struct sockaddr_in6 *sin6)
>> +static int rpc_parse_scope_id(struct net *net, const char *buf,
>> +    const size_t buflen, const char *delim,
>> +    struct sockaddr_in6 *sin6)
>> {
>>     char *p;
>>     size_t len;
>> @@ -177,7 +178,7 @@ static int rpc_parse_scope_id(const char *buf, const size_t buflen,
>>     unsigned long scope_id = 0;
>>     struct net_device *dev;
>>
>> - dev = dev_get_by_name(&init_net, p);
>> + dev = dev_get_by_name(net, p);
>>     if (dev != NULL) {
>>         scope_id = dev->ifindex;
>>         dev_put(dev);
>> @@ -213,7 +214,7 @@ static size_t rpc_pton6(const char *buf, const size_t buflen,
>>     if (in6_pton(buf, buflen, addr, IPV6_SCOPE_DELIMITER, &delim) == 0)
>>         return 0;
>>
>> - if (!rpc_parse_scope_id(buf, buflen, delim, sin6))
>> + if (!rpc_parse_scope_id(&init_net, buf, buflen, delim, sin6))
>>     return 0;
>>
>>     sin6->sin6_family = AF_INET6;
>>

```

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH 1/4] SUNRPC: use passed network namespace context in
rpc_parse_scope_id()

Posted by [bfields](#) on Mon, 12 Dec 2011 16:07:22 GMT

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

On Mon, Dec 12, 2011 at 08:03:45PM +0400, Stanislav Kinsbursky wrote:

> >On Wed, Dec 07, 2011 at 03:20:16PM +0300, Stanislav Kinsbursky wrote:
> >>Use incoming network context in `rpc_parse_scope_id()` instead of hard-coded
> >
> >Changelogs are a little confusing; I might have said "allow
> >`rpc_parse_scope_id()` caller to pass in network context instead of using
> >hard-code "init_net".
>
> Probably, your variant is better. My English is not good enough to
> write descriptive and, in the same time, clear and short comments to
> patches.

Uh, and mine's got a problem or two as well. ("hard-code" should be "hard-coded" in the above.)

My main complaint was just that "incoming network context" sounds like it refers to the network context associated with an incoming rpc request. But actually all you're doing is using the network context passed in by the caller.

--b.

Subject: Re: [PATCH 0/4] SUNRPC: pass network namespace context to sockaddr construction routines

Posted by [Stanislav Kinsbursky](#) on Mon, 12 Dec 2011 16:13:24 GMT

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

> On Wed, Dec 07, 2011 at 03:20:07PM +0300, Stanislav Kinsbursky wrote:
>> This is another precursor patch set aimed to remove hard-coded "init_net"
>> reference from SUNRPC sockaddr conversion and construction routines.
>
> ACK--all look straightforward. (Only complaint would be that this isn't
> terribly interesting till we see how these are used.)
>

Actually, all is simple. These `init_net` pointers will be replaced later by "transports, clients and others" network namespace contexts. I have some more patches for NFS layer. But I don't see any sense to send them now since no "virtualization" patches were committed yet. So, currently, I'm trying to finalize "SUNRPC per network namespace" task, and soon I'll send a patch set making SUNRPC `sysctl`'s network-namespace-aware. It is a little bit tricky and ugly from my point of view, so any ideas are welcome.

```

> --b.
>
>
>>
>> The following series consists of:
>>
>> ---
>>
>> Stanislav Kinsbursky (4):
>>     SUNRPC: use passed network namespace context in rpc_parse_scope_id()
>>     SUNRPC: use passed network namespace context in rpc_pton6()
>>     SUNRPC: use passed network namespace context in rpc_pton()
>>     SUNRPC: use passed network namespace context in rpc_uaddr2sockaddr()
>>
>>
>> fs/nfs/dns_resolve.c      |  4 ++--
>> fs/nfs/nfs4filelayoutdev.c |  2 +-
>> fs/nfs/nfs4namespace.c   |  2 +-
>> fs/nfs/super.c           |  4 ++--
>> fs/nfsd/nfs4state.c      |  2 +-
>> fs/nfsd/nfsctl.c         |  2 +-
>> include/linux/sunrpc/clnt.h |  4 ++--
>> net/sunrpc/addr.c        | 24 ++++++++-----
>> net/sunrpc/rpcb_clnt.c   |  2 +-
>> net/sunrpc/svcauth_unix.c |  2 +-
>> 10 files changed, 25 insertions(+), 23 deletions(-)
>>
>> --
>> Signature

```

--

Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH 0/4] SUNRPC: pass network namespace context to sockaddr construction routines

Posted by [bfields](#) on Mon, 12 Dec 2011 16:15:42 GMT

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

On Mon, Dec 12, 2011 at 08:13:24PM +0400, Stanislav Kinsbursky wrote:

```

> > On Wed, Dec 07, 2011 at 03:20:07PM +0300, Stanislav Kinsbursky wrote:
> >> This is another precursor patch set aimed to remove hard-coded "init_net"
> >> reference from SUNRPC sockaddr conversion and construction routines.
> >

```

> >ACK--all look straightforward. (Only complaint would be that this isn't
> >terribly interesting till we see how these are used.)
> >
>
> Actually, all is simple. These init_net pointers will be replaced
> later by "transports, clients and others" network namespace
> contexts.
> I have some more patches for NFS layer. But I don't see any sense to
> sent them now since no "virtualization" patches were committed yet.
> So, currently, I'm trying to finalize "SUNRPC per network namespace"
> task, and soon I'll send a patch set making SUNRPC sysctl's
> network-namespace-aware. It is a little bit tricky and ugly from my
> pow, so any ideas are welcome.

No ideas.... What's the ugly part?

--b.

Subject: Re: [PATCH 1/4] SUNRPC: use passed network namespace context in
rpc_parse_scope_id()

Posted by [Stanislav Kinsbursky](#) on Mon, 12 Dec 2011 16:18:05 GMT

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

> On Mon, Dec 12, 2011 at 08:03:45PM +0400, Stanislav Kinsbursky wrote:

>>> On Wed, Dec 07, 2011 at 03:20:16PM +0300, Stanislav Kinsbursky wrote:
>>>> Use incoming network context in rpc_parse_scope_id() instead of hard-coded
>>>
>>> Changelogs are a little confusing; I might have said "allow
>>> rpc_parse_scope_id() caller to pass in network context instead of using
>>> hard-code "init_net"."
>>
>> Probably, your variant is better. My English is not good enough to
>> write descriptive and, in the same time, clear and short comments to
>> patches.
>
> Uh, and mine's got a problem or two as well. ("hard-code" should be
> "hard-coded" in the above.)
>
> My main complaint was just that "incoming network context" sounds like
> it refers to the network context associated with an incoming rpc
> request. But actually all you're doing is using the network context
> passed in by the caller.
>
> --b.

Yep, agreed.
I'll resend next patch set version with better comments.

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH 0/4] SUNRPC: pass network namespace context to sockaddr construction routines

Posted by [Stanislav Kinsbursky](#) on Mon, 12 Dec 2011 16:27:36 GMT

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

> On Mon, Dec 12, 2011 at 08:13:24PM +0400, Stanislav Kinsbursky wrote:

>>> On Wed, Dec 07, 2011 at 03:20:07PM +0300, Stanislav Kinsbursky wrote:
>>>> This is another precursor patch set aimed to remove hard-coded "init_net"
>>>> reference from SUNRPC sockaddr conversion and construction routines.
>>>
>>> ACK--all look straightforward. (Only complaint would be that this isn't
>>> terribly interesting till we see how these are used.)
>>>
>>
>> Actually, all is simple. These init_net pointers will be replaced
>> later by "transports, clients and others" network namespace
>> contexts.
>> I have some more patches for NFS layer. But I don't see any sense to
>> sent them now since no "virtualization" patches were committed yet.
>> So, currently, I'm trying to finalize "SUNRPC per network namespace"
>> task, and soon I'll send a patch set making SUNRPC sysctl's
>> network-namespace-aware. It is a little bit tricky and ugly from my
>> pow, so any ideas are welcome.
>
> No ideas.... What's the ugly part?
>
> --b.

I have some "idée fixe", that all this file system parts of SUNRPC (and NFS) layers have to made per network namespace context in the same manner as sysfs done. I.e. it's content should depends on mount point creator network namespace context instead of "current" one.

But proc (and sysctl's as well) doesn't allow to make it so simple like sysfs does. So, currently sysctl's code is about 30 lines. With my patches in will be approximately 3-4 times longer.

But this "dependence on mount point creator network namespace" goal is achieved. Anyway, I'll cleanup this code and send path set soon.

--

Best regards,
Stanislav Kinsbursky
