
Subject: Re: [PATCH v3 03/11] NFS: move per-net callback thread initialization to
nfs_callback_up_net()

Posted by [Stanislav Kinsbursky](#) on Wed, 25 Jul 2012 10:46:12 GMT

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

```
>> On Tue, Jul 03, 2012 at 08:19:46PM +0400, Stanislav Kinsbursky wrote:
>>> This new function is now called before nfs_minorversion_callback_svc_setup()).
>>>
>>> Also few small changes:
>>> 1) current network namespace in nfs_callback_up() was replaced by transport net.
>>> 2) svc_shutdown_net() was moved prior to callback usage counter decrement
>>> (because in case of per-net data allocation failure svc_shutdown_net() have to
>>> be skipped).
>>>
>>> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>>> ---
>>> fs/nfs/callback.c | 125 ++++++
>>> fs/nfs/client.c   | 2 -
>>> 2 files changed, 79 insertions(+), 48 deletions(-)
>>>
>>> diff --git a/fs/nfs/callback.c b/fs/nfs/callback.c
>>> index f8d0c21..b61ed61 100644
>>> --- a/fs/nfs/callback.c
>>> +++ b/fs/nfs/callback.c
>>> @@ -63,6 +63,32 @@ static struct kernel_param_ops param_ops_portnr = {
>>>
>>> module_param_named(callback_tcpport, nfs_callback_set_tcpport, portnr, 0644);
>>>
>>> +static int nfs4_callback_up_net(struct svc_serv *serv, struct net *net)
>>> +{
>>> + int ret;
>>> +
>>> + ret = svc_create_xprt(serv, "tcp", net, PF_INET,
>>> + nfs_callback_set_tcpport, SVC_SOCKET_ANONYMOUS);
>>> + if (ret <= 0)
>>> + goto out_err;
>>> + nfs_callback_tcpport = ret;
>>> + dprintk("NFS: Callback listener port = %u (af %u, net %p)\n",
>>> + nfs_callback_tcpport, PF_INET, net);
>>> +
>>> + ret = svc_create_xprt(serv, "tcp", net, PF_INET6,
>>> + nfs_callback_set_tcpport, SVC_SOCKET_ANONYMOUS);
>>> + if (ret > 0) {
>>> + nfs_callback_tcpport6 = ret;
>>> + dprintk("NFS: Callback listener port = %u (af %u, net %p)\n",
>>> + nfs_callback_tcpport6, PF_INET6, net);
```

```

>>> + } else if (ret != -EAFNOSUPPORT)
>>> + goto out_err;
>>> + return 0;
>>> +
>>> +out_err:
>>> + return (ret) ? ret : -ENOMEM;
>>> +}
>>> +
>>> + /*
>>> +  * This is the NFSv4 callback kernel thread.
>>> +  */
>>> @@ -104,36 +130,21 @@ nfs4_callback_svc(void *vrqstp)
>>> static struct svc_rqst *
>>> nfs4_callback_up(struct svc_serv *serv, struct rpc_xprt *xprt)
>>> {
>>> - int ret;
>>> -
>>> - ret = svc_create_xprt(serv, "tcp", &init_net, PF_INET,
>>> - nfs_callback_set_tcpport, SVC_SOCKET_ANONYMOUS);
>>> - if (ret <= 0)
>>> - goto out_err;
>>> - nfs_callback_tcpport = ret;
>>> - dprintk("NFS: Callback listener port = %u (af %u)\n",
>>> - nfs_callback_tcpport, PF_INET);
>>> -
>>> - ret = svc_create_xprt(serv, "tcp", &init_net, PF_INET6,
>>> - nfs_callback_set_tcpport, SVC_SOCKET_ANONYMOUS);
>>> - if (ret > 0) {
>>> - nfs_callback_tcpport6 = ret;
>>> - dprintk("NFS: Callback listener port = %u (af %u)\n",
>>> - nfs_callback_tcpport6, PF_INET6);
>>> - } else if (ret == -EAFNOSUPPORT)
>>> - ret = 0;
>>> - else
>>> - goto out_err;
>>> -
>>> return svc_prepare_thread(serv, &serv->sv_pools[0], NUMA_NO_NODE);
>>> -
>>> -out_err:
>>> - if (ret == 0)
>>> - ret = -ENOMEM;
>>> - return ERR_PTR(ret);
>>> }
>>>
>>> #if defined(CONFIG_NFS_V4_1)
>>> +static int nfs41_callback_up_net(struct svc_serv *serv, struct net *net)
>>> +{
>>> + /*

```

```

>>> + * Create an svc_sock for the back channel service that shares the
>>> + * fore channel connection.
>>> + * Returns the input port (0) and sets the svc_serv bc_xprt on success
>>> + */
>>> + return svc_create_xprt(serv, "tcp-bc", net, PF_INET, 0,
>>> +     SVC_SOCKET_ANONYMOUS);
>>> +}
>>> +
>>> /*
>>>  * The callback service for NFSv4.1 callbacks
>>>  */
>>> @@ -176,19 +187,6 @@ static struct svc_rqst *
>>> nfs41_callback_up(struct svc_serv *serv, struct rpc_xprt *xprt)
>>> {
>>>     struct svc_rqst *rqstp;
>>> - int ret;
>>> -
>>> - /*
>>> - * Create an svc_sock for the back channel service that shares the
>>> - * fore channel connection.
>>> - * Returns the input port (0) and sets the svc_serv bc_xprt on success
>>> - */
>>> - ret = svc_create_xprt(serv, "tcp-bc", &init_net, PF_INET, 0,
>>> -     SVC_SOCKET_ANONYMOUS);
>>> - if (ret < 0) {
>>> -     rqstp = ERR_PTR(ret);
>>> -     goto out;
>>> - }
>>>
>>> /*
>>>  * Save the svc_serv in the transport so that it can
>>> @@ -204,7 +202,6 @@ nfs41_callback_up(struct svc_serv *serv, struct rpc_xprt *xprt)
>>>     svc_xprt_put(serv->sv_bc_xprt);
>>>     serv->sv_bc_xprt = NULL;
>>> }
>>> -out:
>>>     dprintk("--> %s return %ld\n", __func__,
>>>         IS_ERR(rqstp) ? PTR_ERR(rqstp) : 0);
>>>     return rqstp;
>>> @@ -228,6 +225,11 @@ static inline void nfs_callback_bc_serv(u32 minorversion, struct
>>> rpc_xprt *xprt,
>>>     xprt->bc_serv = cb_info->serv;
>>> }
>>> #else
>>> +static int nfs41_callback_up_net(struct svc_serv *serv, struct net *net)
>>> +{
>>> + return 0;
>>> +}

```

```

>>> +
>>> static inline int nfs_minorversion_callback_svc_setup(u32 minorversion,
>>> struct svc_serv *serv, struct rpc_xprt *xprt,
>>> struct svc_rqst **rqstpp, int (**callback_svc)(void *vrqstp))
>>> @@ -241,6 +243,36 @@ static inline void nfs_callback_bc_serv(u32 minorversion, struct
rpc_xprt *xprt,
>>> }
>>> #endif /* CONFIG_NFS_V4_1 */
>>>
>>> +static int nfs_callback_up_net(int minorversion, struct svc_serv *serv, struct net *net)
>>> +{
>>> + int ret;
>>> +
>>> + dprintk("NFS: create per-net callback data; net=%p\n", net);
>>> +
>>> + ret = svc_bind(serv, net);
>>> + if (ret < 0) {
>>> + printk(KERN_WARNING "NFS: bind callback service failed\n");
>>> + goto err_bind;
>>> + }
>>> +
>>> + if (minorversion) {
>>> + ret = nfs41_callback_up_net(serv, net);
>>> + if (ret < 0)
>>> + goto err_socks;
>>> + }
>>> +
>>> + if (ret == 0)
>>> + ret = nfs4_callback_up_net(serv, net);
>>>
>> So in the minorversion 1 case, and in the absence of errors, you set up
>> both the 4.1 and 4.0 callback xprts?
>>
>> That doesn't look right. You should need only one or the other.
>>
>
> Thanks for the catch, Bruce.
> I'll fix it and resend.
> But currently I'm thinking, that for 3.6 kernel probably it's better to push
> this code through Trond's tree.
>

```

No, Bruce. It works fine. nfs41_callback_up_net() will return either negative value of port number. I.e. nfs4_callback_up_net() won't be called. But this looks confusing. I'll rewrite this part.

--

Best regards,
Stanislav Kinsbursky
