Subject: Re: [PATCH] NFS: hard-code init_net for NFS callback transports
Posted by Stanislav Kinsbursky on Wed, 13 Jun 2012 08:16:07 GMT
View Forum Message <> Reply to Message

> On Mon, Jun 11, 2012 at 08:37:18AM -0400, J. Bruce Fields wrote:
>> On Fri, Jun 01, 2012 at 01:55:47PM +0400, Stanislav Kinsbursky wrote:
>>> This patch set is a bug fix for Bruce's 3.5 branch.
>>> I'll rebase this for 3.4 and resend.
>>
>> Apologies for neglecting this, I'm done with the bakeathon and should be
>> back to this soon.
>>
>> But, I'm a bit confused: you say you'll rebase this, but I've only seen
>> two rebased patches for 3.4, and neither looks like a simple rebase of
>> this one.
>
> Oh, I see; the backport of "SUNRPC: new svc_bind() routine introduced"
> is a combination of the patch by the same name upstream with this one.
>
> In future for backports I'd rather keep a one-to-one correspondance
> between the original patches and the backported patches. (Even if that
> means 4.2 might not be as "bisectable".)
>
> Also, get this patch into stable by just adding a "cc:
> stable@vger.kernel.org" with the signed-off at the end of this patch.
>
> I've gone ahead and fixed that up; results in the "for-3.4" and
> "for-3.5" branches of
>
> git://linux-nfs.org/~bfields/linux-topics.git
>
> Let me know if anything looks wrong.
>

Hi, Bruce. Thanks for you time and guidance with this patch set.
But I'm a bit confused - what I have to do now? Take the patches from your 3.4
branch and send them to "stable"?

> --b.
>
>>
>> OK, I'll look more closely and probably I'll figure out. But in
>> general, a little more detail in changelogs would help me process
>> patches more quickly:
>>

>> - Please make sure it's clear whether a given patch is a bug
>>   fix, a new feature, or code reshuffling that isn't intended to
>>   change behavior. If it's a bug fix, tell me whether we've had
>>   the bug forever or whether it's a recent regression. And
>>   (relevant here) if it's a recent regression, reference the ID
>>   of the commit that introduced it.
>> - If something is a backport, please reference in the changelog
>>   the commit ID of the backported patch, especially if it's not
>>   obviously the same.
>>
>> --b.
>>
>>
>>>
>>> In case of destroying mount namespace on child reaper exit, nsproxy is zeroed
>>> to the point already. So, dereferencing of it is invalid.
>>> This patch hard-code "init_net" for all network namespace references for NFS
>>> callback services. This will be fixed with proper NFS callback
>>> containerization.
>>>
>>> Signed-off-by: Stanislav Kinsbursky<skinsbursky@parallels.com>
>>> ---
>>>   fs/nfs/callback.c | 11 +++++------
>>>   1 files changed, 5 insertions(+), 6 deletions(-)
>>>
>>> diff --git a/fs/nfs/callback.c b/fs/nfs/callback.c
>>> index 970659d..23ff18f 100644
>>> --- a/fs/nfs/callback.c
>>> +++ b/fs/nfs/callback.c
>>> @@ -17,7 +17,6 @@
>>>   #include<linux/kthread.h>
>>>   #include<linux/sunrpc/svcauth_gss.h>
>>>   #include<linux/sunrpc/bc_xprt.h>
>>> -#include<linux/nsproxy.h>
>>>
>>>   #include<net/inet_sock.h>
>>>
>>> @@ -107,7 +106,7 @@ nfs4_callback_up(struct svc_serv *serv, struct rpc_xprt *xprt)
>>>   {
>>>   int ret;
>>>
>>> - ret = svc_create_xprt(serv, "tcp", xprt->xprt_net, PF_INET,
>>> + ret = svc_create_xprt(serv, "tcp",&init_net, PF_INET,
>>>   nfs_callback_set_tcpport, SVC_SOCK_ANONYMOUS);
>>>   if (ret<= 0)
>>>   goto out_err;
>>> @@ -115,7 +114,7 @@ nfs4_callback_up(struct svc_serv *serv, struct rpc_xprt *xprt)
>>>   dprintk("NFS: Callback listener port = %u (af %u)\n",

```
>>>     nfs_callback_tcpport, PF_INET);
>>>
>>> - ret = svc_create_xprt(serv, "tcp", xprt->xprt_net, PF_INET6,
>>> + ret = svc_create_xprt(serv, "tcp",&init_net, PF_INET6,
>>>     nfs_callback_set_tcpport, SVC_SOCK_ANONYMOUS);
>>>   if (ret> 0) {
>>>     nfs_callback_tcpport6 = ret;
>>> @@ -184,7 +183,7 @@ nfs41_callback_up(struct svc_serv *serv, struct rpc_xprt *xprt)
>>>     * fore channel connection.
>>>     * Returns the input port (0) and sets the svc_serv bc_xprt on success
>>>     */
>>> - ret = svc_create_xprt(serv, "tcp-bc", xprt->xprt_net, PF_INET, 0,
>>> + ret = svc_create_xprt(serv, "tcp-bc",&init_net, PF_INET, 0,
>>>         SVC_SOCK_ANONYMOUS);
>>>   if (ret< 0) {
>>>     rqstp = ERR_PTR(ret);
>>> @@ -254,7 +253,7 @@ int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt)
>>>   char svc_name[12];
>>>   int ret = 0;
>>>   int minorversion_setup;
>>> - struct net *net = current->nsproxy->net_ns;
>>> + struct net *net =&init_net;
>>>
>>>   mutex_lock(&nfs_callback_mutex);
>>>   if (cb_info->users++ || cb_info->task != NULL) {
>>> @@ -330,7 +329,7 @@ void nfs_callback_down(int minorversion)
>>>   cb_info->users--;
>>>   if (cb_info->users == 0&&  cb_info->task != NULL) {
>>>     kthread_stop(cb_info->task);
>>> -  svc_shutdown_net(cb_info->serv, current->nsproxy->net_ns);
>>> +  svc_shutdown_net(cb_info->serv,&init_net);
>>>     svc_exit_thread(cb_info->rqst);
>>>     cb_info->serv = NULL;
>>>     cb_info->rqst = NULL;
>>>
```

--
Best regards,
Stanislav Kinsbursky