
Subject: [PATCH v4 05/10] NFS: callback service start function introduced
Posted by [Stanislav Kinsbursky](#) on Mon, 20 Aug 2012 14:00:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is just a code move, which from my POW makes code looks better.

I.e. now on start we have 3 different stages:

- 1) Service creation.
- 2) Service per-net data allocation.
- 3) Service start.

Patch also renames goto label "out_err:" into "err_start:" to reflect new changes.

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

```
---
fs/nfs/callback.c | 77 ++++++++++++++++++++++++++++++++++++++-----
1 files changed, 45 insertions(+), 32 deletions(-)

diff --git a/fs/nfs/callback.c b/fs/nfs/callback.c
index a528cb7..5d5f9d1 100644
--- a/fs/nfs/callback.c
+++ b/fs/nfs/callback.c
@@ -217,6 +217,46 @@ static inline void nfs_callback_bc_serv(u32 minorversion, struct rpc_xprt
 *xprt,
 }
#endif /* CONFIG_NFS_V4_1 */

+static int nfs_callback_start_svc(int minorversion, struct rpc_xprt *xprt,
+    struct svc_serv *serv)
+{
+    struct svc_rqst *rqstp;
+    int (*callback_svc)(void *vrqstp);
+    struct nfs_callback_data *cb_info = &nfs_callback_info[minorversion];
+    char svc_name[12];
+    int ret;
+    int minorversion_setup;
+
+    + nfs_callback_bc_serv(minorversion, xprt, serv);
+
+    + minorversion_setup = nfs_minorversion_callback_svc_setup(minorversion,
+    +    serv, &rqstp, &callback_svc);
+    + if (!minorversion_setup) {
+    +     /* v4.0 callback setup */
+    +     rqstp = nfs4_callback_up(serv);
+    +     callback_svc = nfs4_callback_svc;
+    + }
+
+    + if (IS_ERR(rqstp))
```

```

+ return PTR_ERR(rqstp);
+
+ svc_sock_update_bufs(serv);
+
+ sprintf(svc_name, "nfsv4.%u-svc", minorversion);
+ cb_info->serv = serv;
+ cb_info->rqst = rqstp;
+ cb_info->task = kthread_run(callback_svc, cb_info->rqst, svc_name);
+ if (IS_ERR(cb_info->task)) {
+   ret = PTR_ERR(cb_info->task);
+   svc_exit_thread(cb_info->rqst);
+   cb_info->rqst = NULL;
+   cb_info->task = NULL;
+   return PTR_ERR(cb_info->task);
+ }
+ dprintk("nfs_callback_up: service started\n");
+ return 0;
+}
+
static int nfs_callback_up_net(int minorversion, struct svc_serv *serv, struct net *net)
{
    int ret;
@@ -299,12 +339,8 @@ static struct svc_serv *nfs_callback_create_svc(int minorversion)
int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt)
{
    struct svc_serv *serv;
- struct svc_rqst *rqstp;
- int (*callback_svc)(void *vrqstp);
    struct nfs_callback_data *cb_info = &nfs_callback_info[minorversion];
- char svc_name[12];
    int ret = 0;
- int minorversion_setup;
    struct net *net = xprt->xprt_net;

    mutex_lock(&nfs_callback_mutex);
@@ -324,34 +360,10 @@ int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt)
    if (ret < 0)
        goto err_net;

- nfs_callback_bc_serv(minorversion, xprt, serv);
-
- minorversion_setup = nfs_minversion_callback_svc_setup(minorversion,
-   serv, &rqstp, &callback_svc);
- if (!minorversion_setup) {
-   /* v4.0 callback setup */
-   rqstp = nfs4_callback_up(serv);
-   callback_svc = nfs4_callback_svc;
- }

```

```

-
- if (IS_ERR(rqstp)) {
-   ret = PTR_ERR(rqstp);
-   goto out_err;
- }
-
- svc_sock_update_bufs(serv);
+ ret = nfs_callback_start_svc(minorversion, xprt, serv);
+ if (ret < 0)
+   goto err_start;

- sprintf(svc_name, "nfsv4.%u-svc", minorversion);
- cb_info->serv = serv;
- cb_info->rqst = rqstp;
- cb_info->task = kthread_run(callback_svc, cb_info->rqst, svc_name);
- if (IS_ERR(cb_info->task)) {
-   ret = PTR_ERR(cb_info->task);
-   svc_exit_thread(cb_info->rqst);
-   cb_info->rqst = NULL;
-   cb_info->task = NULL;
-   goto out_err;
- }
out:
/*
 * svc_create creates the svc_serv with sv_nrthreads == 1, and then
@@ -363,7 +375,8 @@ out:
err_create:
    mutex_unlock(&nfs_callback_mutex);
    return ret;
-out_err:
+
+err_start:
    svc_shutdown_net(serv, net);
err_net:
    dprintk("NFS: Couldn't create callback socket or server thread; "

```
