
Subject: [PATCH v2 02/12] NFS: callback service creation function introduced
Posted by [Stanislav Kinsbursky](#) on Tue, 22 May 2012 07:36:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

This function creates service if it's not exist, or increase usage counter of the existent, and returns pointer to it.
Usage counter will be dropped by `svc_destroy()` later in `nfs_callback_up()`.

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

fs/nfs/callback.c | 55 ++++++-----
1 files changed, 45 insertions(+), 10 deletions(-)

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

index f3d7ee2..b34ccc6 100644

--- a/fs/nfs/callback.c

+++ b/fs/nfs/callback.c

```
@@ -242,12 +242,46 @@ static inline void nfs_callback_bc_serv(u32 minorversion, struct  
rpc_xprt *xprt,  
{  
#endif /* CONFIG_NFS_V4_1 */
```

```
+static struct svc_serv *nfs_callback_create_svc(int minorversion)
```

```
+{
```

```
+ struct nfs_callback_data *cb_info = &nfs_callback_info[minorversion];
```

```
+ struct svc_serv *serv;
```

```
+
```

```
+ /*
```

```
+ * Check whether we're already up and running.
```

```
+ */
```

```
+ if (cb_info->task) {
```

```
+ /*
```

```
+ * Note: increase service usage, because later in case of error
```

```
+ * svc_destroy() will be called.
```

```
+ */
```

```
+ svc_get(cb_info->serv);
```

```
+ return cb_info->serv;
```

```
+ }
```

```
+
```

```
+ /*
```

```
+ * Sanity check: if there's no task,
```

```
+ * we should be the first user ...
```

```
+ */
```

```
+ if (cb_info->users)
```

```
+ printk(KERN_WARNING "nfs_callback_up: no kthread, %d users??\n",
```

```
+ cb_info->users);
```

```
+
```

```
+ serv = svc_create(&nfs4_callback_program, NFS4_CALLBACK_BUFSIZE, NULL);
```

```

+ if (!serv) {
+   printk(KERN_ERR "lockd_up: create service failed\n");
+   return ERR_PTR(-ENOMEM);
+ }
+ dprintk("nfs_callback_up: service created\n");
+ return serv;
+}
+
/*
 * Bring up the callback thread if it is not already up.
 */
int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt)
{
- struct svc_serv *serv = NULL;
+ struct svc_serv *serv;
  struct svc_rqst *rqstp;
  int (*callback_svc)(void *vrqstp);
  struct nfs_callback_data *cb_info = &nfs_callback_info[minorversion];
@@ -257,15 +291,17 @@ int nfs_callback_up(u32 minorversion, struct rpc_xprt *xprt)
  struct net *net = current->nsproxy->net_ns;

  mutex_lock(&nfs_callback_mutex);
+
+ serv = nfs_callback_create_svc(minorversion);
+ if (IS_ERR(serv)) {
+   ret = PTR_ERR(serv);
+   goto err_create;
+ }
+
  if (cb_info->users++ || cb_info->task != NULL) {
    nfs_callback_bc_serv(minorversion, xprt, cb_info);
    goto out;
  }
- serv = svc_create(&nfs4_callback_program, NFS4_CALLBACK_BUFSIZE, NULL);
- if (!serv) {
-   ret = -ENOMEM;
-   goto out_err;
- }

  ret = svc_bind(serv, net);
  if (ret < 0) {
@@ -306,16 +342,15 @@ out:
  * on both success and failure so that the refcount is 1 when the
  * thread exits.
  */
- if (serv)
-   svc_destroy(serv);
+ svc_destroy(serv);

```

```
+err_create:
    mutex_unlock(&nfs_callback_mutex);
    return ret;
out_err:
    dprintk("NFS: Couldn't create callback socket or server thread; "
        "err = %d\n", ret);
    cb_info->users--;
- if (serv)
-     svc_shutdown_net(serv, net);
+ svc_shutdown_net(serv, net);
    goto out;
}
```
