
Subject: [PATCH v3, RESEND 15/16] sunrpc: remove global init_rpc_pipefs
Posted by [Kirill A. Shutemov](#) on Tue, 08 Feb 2011 18:42:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Replace remaining init_rpc_pipefs references with get_rpc_pipefs(NULL)
and make init_rpc_pipefs static.

Signed-off-by: Kirill A. Shutemov <kas@openvz.org>
Reviewed-by: Rob Landley <rlandley@parallels.com>

fs/nfsd/nfs4callback.c | 5 ++++-
fs/nfsd/nfssvc.c | 16 ++++++++
include/linux/sunrpc/rpc_pipe_fs.h | 2 --
net/sunrpc/rpc_pipe.c | 1 -
4 files changed, 18 insertions(+), 6 deletions(-)

diff --git a/fs/nfsd/nfs4callback.c b/fs/nfsd/nfs4callback.c
index 6636ab2..3048988 100644

--- a/fs/nfsd/nfs4callback.c
+++ b/fs/nfsd/nfs4callback.c
@@ -646,7 +646,6 @@ static int setup_callback_client(struct nfs4_client *clp, struct
nfs4_cb_conn *c
 .version = 0,
 .authflavor = clp->cl_flavor,
 .flags = (RPC_CLNT_CREATE_NOPING | RPC_CLNT_CREATE_QUIET),
- .rpcmount = init_rpc_pipefs,
};
struct rpc_clnt *client;

@@ -666,8 +665,12 @@ static int setup_callback_client(struct nfs4_client *clp, struct
nfs4_cb_conn *c
 args.prognumber = clp->cl_cb_session->se_cb_prog;
 args.protocol = XPRT_TRANSPORT_BC_TCP;
}
+ args.rpcmount = get_rpc_pipefs(NULL);
+ if (IS_ERR(args.rpcmount))
+ return PTR_ERR(args.rpcmount);
/* Create RPC client */
client = rpc_create(&args);
+ mntput(args.rpcmount);
if (IS_ERR(client)) {
 dprintk("NFSD: couldn't create callback client: %ld\n",
PTR_ERR(client));

diff --git a/fs/nfsd/nfssvc.c b/fs/nfsd/nfssvc.c
index dc11012..a10b1439 100644

--- a/fs/nfsd/nfssvc.c
+++ b/fs/nfsd/nfssvc.c
@@ -206,6 +206,7 @@ static bool nfsd_up = false;

```

static int nfsd_startup(unsigned short port, int nrservs)
{
    int ret;
+ struct vfsmount *rpcmount;

    if (nfsd_up)
        return 0;
@@ -220,7 +221,13 @@ static int nfsd_startup(unsigned short port, int nrservs)
    ret = nfsd_init_socks(port);
    if (ret)
        goto out_racache;
- ret = lockd_up(init_rpc_pipefs);
+ rpcmount = get_rpc_pipefs(NULL);
+ if (IS_ERR(rpcmount)) {
+     ret = PTR_ERR(rpcmount);
+     goto out_racache;
+ }
+ ret = lockd_up(rpcmount);
+ mntput(rpcmount);
    if (ret)
        goto out_racache;
    ret = nfs4_state_start();
@@ -308,6 +315,7 @@ static void set_max_drc(void)
int nfsd_create_serv(void)
{
    int err = 0;
+ struct vfsmount *rpcmount;

    WARN_ON(!mutex_is_locked(&nfsd_mutex));
    if (nfsd_serv) {
@@ -332,9 +340,13 @@ int nfsd_create_serv(void)
    }
    nfsd_reset_versions();

- nfsd_serv = svc_create_pooled(&nfsd_program, init_rpc_pipefs,
+ rpcmount = get_rpc_pipefs(NULL);
+ if (IS_ERR(rpcmount))
+     return PTR_ERR(rpcmount);
+ nfsd_serv = svc_create_pooled(&nfsd_program, rpcmount,
                                nfsd_max_blksize, nfsd_last_thread, nfsd,
                                THIS_MODULE);
+ mntput(rpcmount);
    if (nfsd_serv == NULL)
        return -ENOMEM;

diff --git a/include/linux/sunrpc/rpc_pipe_fs.h b/include/linux/sunrpc/rpc_pipe_fs.h
index 4a8830a..a0b9c46 100644
--- a/include/linux/sunrpc/rpc_pipe_fs.h

```

```

+++ b/include/linux/sunrpc/rpc_pipe_fs.h
@@ -44,8 +44,6 @@ RPC_I(struct inode *inode)
    return container_of(inode, struct rpc_inode, vfs_inode);
}

-extern struct vfsmount *init_rpc_pipefs;
-
extern struct vfsmount *get_rpc_pipefs(const char *path);
extern int rpc_pipefs_add_destroy_cb(struct super_block *sb,
    void (*destroy_cb)(void *data), void *data);
diff --git a/net/sunrpc/rpc_pipe.c b/net/sunrpc/rpc_pipe.c
index f824f33..7c16261 100644
--- a/net/sunrpc/rpc_pipe.c
+++ b/net/sunrpc/rpc_pipe.c
@@ -33,7 +33,6 @@
#include <linux/sunrpc/cache.h>

struct vfsmount *init_rpc_pipefs __read_mostly;
-EXPORT_SYMBOL_GPL(init_rpc_pipefs);

static struct file_system_type rpc_pipe_fs_type;

--
1.7.4

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
