Subject: [PATCH v3 16/16] Rework get_rpc_pipefs and introduce put_rpc_pipefs()
Posted by Kirill A. Shutemov on Fri, 14 Jan 2011 13:49:14 GMT
View Forum Message <> Reply to Message

Now sunrpc module can be removed normally.

Signed-off-by: Kirill A. Shutemov <kas@openvz.org>
---
 fs/nfs/client.c | 4 +-
 fs/nfs/super.c | 2 +-
 fs/nfsd/nfs4callback.c | 2 +-
 fs/nfsd/nfssvc.c | 4 +-
 include/linux/sunrpc/rpc_pipe_fs.h | 1 +
 net/sunrpc/rpc_pipe.c | 51 ++++++++++++++++++++++++++----------
 6 files changed, 40 insertions(+), 24 deletions(-)

```
diff --git a/fs/nfs/client.c b/fs/nfs/client.c
index 8dd0e99..0b8ac12 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -838,7 +838,7 @@ static int nfs_init_server(struct nfs_server *server,

  /* Allocate or find a client reference we can use */
  clp = nfs_get_client(&cl_init);
- mntput(cl_init.rpcmount);
+ put_rpc_pipefs(cl_init.rpcmount);
  if (IS_ERR(clp)) {
   dprintk("<-- nfs_init_server() = error %ld\n", PTR_ERR(clp));
   return PTR_ERR(clp);
@@ -1533,7 +1533,7 @@ static int nfs4_init_server(struct nfs_server *server,
    &timeparms,
    data->minorversion,
    rpcmount);
- mntput(rpcmount);
+ put_rpc_pipefs(rpcmount);
  if (error < 0)
   goto error;

diff --git a/fs/nfs/super.c b/fs/nfs/super.c
index be4852b..148843e 100644
--- a/fs/nfs/super.c
+++ b/fs/nfs/super.c
@@ -1668,7 +1668,7 @@ static int nfs_try_mount(struct nfs_parsed_mount_data *args,
   * to a file handle.
   */
  status = nfs_mount(&request);
- mntput(request.rpcmount);
+ put_rpc_pipefs(request.rpcmount);
```

```
  if (status != 0) {
   dfprintk(MOUNT, "NFS: unable to mount server %s, error %d\n",
     request.hostname, status);
diff --git a/fs/nfsd/nfs4callback.c b/fs/nfsd/nfs4callback.c
index 61210b6..4d8383b 100644
--- a/fs/nfsd/nfs4callback.c
+++ b/fs/nfsd/nfs4callback.c
@@ -667,7 +667,7 @@ int setup_callback_client(struct nfs4_client *clp, struct nfs4_cb_conn
*conn)
   return PTR_ERR(args.rpcmount);
  /* Create RPC client */
  client = rpc_create(&args);
- mntput(args.rpcmount);
+ put_rpc_pipefs(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 7353420..cff49fc 100644
--- a/fs/nfsd/nfssvc.c
+++ b/fs/nfsd/nfssvc.c
@@ -227,7 +227,7 @@ static int nfsd_startup(unsigned short port, int nrservs)
   goto out_racache;
  }
  ret = lockd_up(rpcmount);
- mntput(rpcmount);
+ put_rpc_pipefs(rpcmount);
  if (ret)
   goto out_racache;
  ret = nfs4_state_start();
@@ -346,7 +346,7 @@ int nfsd_create_serv(void)
  nfsd_serv = svc_create_pooled(&nfsd_program, rpcmount,
          nfsd_max_blksize, nfsd_last_thread, nfsd,
          THIS_MODULE);
- mntput(rpcmount);
+ put_rpc_pipefs(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 a0b9c46..328b3da 100644
--- a/include/linux/sunrpc/rpc_pipe_fs.h
+++ b/include/linux/sunrpc/rpc_pipe_fs.h
@@ -45,6 +45,7 @@ RPC_I(struct inode *inode)
 }

 extern struct vfsmount *get_rpc_pipefs(const char *path);
+extern void put_rpc_pipefs(struct vfsmount *rpcmount);
```

```
 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 96973c0..82feacd 100644
--- a/net/sunrpc/rpc_pipe.c
+++ b/net/sunrpc/rpc_pipe.c
@@ -33,6 +33,7 @@
 #include <linux/sunrpc/cache.h>

 struct vfsmount *init_rpc_pipefs __read_mostly;
+static int init_rpc_pipefs_count;

 static struct file_system_type rpc_pipe_fs_type;

@@ -1005,22 +1006,35 @@ static int check_rpc_pipefs(struct vfsmount *mnt, void *arg)
  return 1;
 }

-struct vfsmount *get_rpc_pipefs(const char *p)
+static struct vfsmount *find_rpc_pipefs(void)
 {
- int error;
  struct vfsmount *rpcmount = ERR_PTR(-EINVAL);
- struct path path;
+ int err;

- if (!p) {
-  iterate_mounts(check_rpc_pipefs, &rpcmount,
-    current->nsproxy->mnt_ns->root);
+ iterate_mounts(check_rpc_pipefs, &rpcmount,
+   current->nsproxy->mnt_ns->root);

-  if (IS_ERR(rpcmount) && (current->nsproxy->mnt_ns ==
-     init_task.nsproxy->mnt_ns))
-   return mntget(init_rpc_pipefs);
+ if (!IS_ERR(rpcmount))
+  return rpcmount;

+ if (current->nsproxy->mnt_ns != init_task.nsproxy->mnt_ns)
   return rpcmount;
- }
+
+ err = simple_pin_fs(&rpc_pipe_fs_type, &init_rpc_pipefs,
+   &init_rpc_pipefs_count);
+ if (err)
+  return ERR_PTR(err);
+ return init_rpc_pipefs;
```

```
+}
+
+struct vfsmount *get_rpc_pipefs(const char *p)
+{
+ int error;
+ struct vfsmount *rpcmount = ERR_PTR(-EINVAL);
+ struct path path;
+
+ if (!p)
+  return find_rpc_pipefs();

  error = kern_path(p, LOOKUP_FOLLOW | LOOKUP_DIRECTORY, &path);
  if (error)
@@ -1033,6 +1047,15 @@ struct vfsmount *get_rpc_pipefs(const char *p)
 }
 EXPORT_SYMBOL_GPL(get_rpc_pipefs);

+void put_rpc_pipefs(struct vfsmount *rpcmount)
+{
+ if (rpcmount == init_rpc_pipefs)
+  simple_release_fs(&init_rpc_pipefs, &init_rpc_pipefs_count);
+ else
+  mntput(rpcmount);
+}
+EXPORT_SYMBOL_GPL(put_rpc_pipefs);
+
 struct destroy_cb {
  struct list_head list;
  void (*callback)(void *data);
@@ -1232,16 +1255,8 @@ int register_rpc_pipefs(void)
  if (err)
   goto destroy_cache;

- init_rpc_pipefs = kern_mount(&rpc_pipe_fs_type);
- if (IS_ERR(init_rpc_pipefs)) {
-  err = PTR_ERR(init_rpc_pipefs);
-  goto unregister_fs;
- }
-
  return 0;

-unregister_fs:
- unregister_filesystem(&rpc_pipe_fs_type);
 destroy_cache:
  kmem_cache_destroy(rpc_inode_cachep);
  return err;
--
1.7.3.4
```

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containe rs