
Subject: [PATCH v2 5/6] NFS: handle NFS idmap pipe PipeFS dentries by network namespace aware routines

Posted by [Stanislav Kinsbursky](#) on Tue, 10 Jan 2012 12:13:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

This patch makes NFS idmap pipes dentries allocated and destroyed in network namespace context by PipeFS network namespace aware routines.
Network namespace context is obtained from nfs_client structure.

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

fs/nfs/idmap.c | 61 +++-----
1 files changed, 53 insertions(+), 8 deletions(-)

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

index 9c701b5..2bab772 100644

--- a/fs/nfs/idmap.c

+++ b/fs/nfs/idmap.c

@@ -348,6 +348,56 @@ static const struct rpc_pipe_ops idmap_upcall_ops = {
 .destroy_msg = idmap_pipe_destroy_msg,
};

+static void __nfs_idmap_unregister(struct rpc_pipe *pipe)

+{
+ if (pipe->dentry)
+ rpc_unlink(pipe->dentry);
+}

+
+static int __nfs_idmap_register(struct dentry *dir,

+ struct idmap *idmap,
+ struct rpc_pipe *pipe)

+{
+ struct dentry *dentry;
+
+ dentry = rpc_mkpipe_dentry(dir, "idmap", idmap, pipe);
+ if (IS_ERR(dentry))
+ return PTR_ERR(dentry);
+ pipe->dentry = dentry;
+ return 0;
+}

+
+static void nfs_idmap_unregister(struct nfs_client *clp,

+ struct rpc_pipe *pipe)
+{
+ struct net *net = clp->net;
+ struct super_block *pipefs_sb;
+
+}

```

+ pipefs_sb = rpc_get_sb_net(net);
+ if (pipefs_sb) {
+   __nfs_idmap_unregister(pipe);
+   rpc_put_sb_net(net);
+ }
+}
+
+static int nfs_idmap_register(struct nfs_client *clp,
+    struct idmap *idmap,
+    struct rpc_pipe *pipe)
+{
+   struct net *net = clp->net;
+   struct super_block *pipefs_sb;
+   int err = 0;
+
+   pipefs_sb = rpc_get_sb_net(net);
+   if (pipefs_sb) {
+     if (clp->cl_rpcclient->cl_dentry)
+       err = __nfs_idmap_register(clp->cl_rpcclient->cl_dentry,
+           idmap, pipe);
+     rpc_put_sb_net(net);
+   }
+   return err;
+}
+
+int
+nfs_idmap_new(struct nfs_client *clp)
+{
@@ -367,12 +417,8 @@ nfs_idmap_new(struct nfs_client *clp)
    kfree(idmap);
    return error;
  }
-
- if (clp->cl_rpcclient->cl_dentry)
-   pipe->dentry = rpc_mkpipe_dentry(clp->cl_rpcclient->cl_dentry,
-       "idmap", idmap, pipe);
- if (IS_ERR(pipe->dentry)) {
-   error = PTR_ERR(pipe->dentry);
+ error = nfs_idmap_register(clp, idmap, pipe);
+ if (error) {
+   rpc_destroy_pipe_data(pipe);
+   kfree(idmap);
+   return error;
@@ -395,8 +441,7 @@ nfs_idmap_delete(struct nfs_client *clp)

  if (!idmap)
    return;
- if (idmap->idmap_pipe->dentry)

```

```
- rpc_unlink(idmap->idmap_pipe->dentry);  
+ nfs_idmap_unregister(clp, idmap->idmap_pipe);  
  rpc_destroy_pipe_data(idmap->idmap_pipe);  
  clp->cl_idmap = NULL;  
  kfree(idmap);
```
