
Subject: [PATCH v2 6/6] SUNRPC: remove RPC PipeFS mount point reference from RPC client

Posted by [Stanislav Kinsbursky](#) on Wed, 11 Jan 2012 15:18:42 GMT

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

This is a cleanup patch. We don't need this reference anymore.

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

```
fs/nfs/idmap.c          | 4 +--
include/linux/sunrpc/clnt.h | 2 +-
net/sunrpc/auth_gss/auth_gss.c | 8 +++++-
net/sunrpc/clnt.c        | 21 ++++++++-----
4 files changed, 17 insertions(+), 18 deletions(-)
```

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

index 096e374..9c701b5 100644

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

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

```
@ @ -368,8 +368,8 @ @ nfs_idmap_new(struct nfs_client *clp)
    return error;
}
```

```
- if (clp->cl_rpcclient->cl_path.dentry)
```

```
- pipe->dentry = rpc_mkpipe_dentry(clp->cl_rpcclient->cl_path.dentry,
```

```
+ 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);
```

diff --git a/include/linux/sunrpc/clnt.h b/include/linux/sunrpc/clnt.h

index a4f237c..1e56469 100644

--- a/include/linux/sunrpc/clnt.h

+++ b/include/linux/sunrpc/clnt.h

```
@ @ -57,7 +57,7 @ @ struct rpc_clnt {
```

```
    int cl_nodelen; /* nodename length */
    char cl_nodename[UNIX_MAXNODENAME];
```

```
- struct path cl_path;
```

```
+ struct dentry * cl_dentry;
```

```
    struct rpc_clnt * cl_parent; /* Points to parent of clones */
```

```
    struct rpc_rtt cl_rtt_default;
```

```
    struct rpc_timeout cl_timeout_default;
```

diff --git a/net/sunrpc/auth_gss/auth_gss.c b/net/sunrpc/auth_gss/auth_gss.c

index d7a5719..1fd4abe 100644

--- a/net/sunrpc/auth_gss/auth_gss.c

+++ b/net/sunrpc/auth_gss/auth_gss.c

```

@@ -779,12 +779,12 @@ static int gss_pipes_dentries_create(struct rpc_auth *auth)
    gss_auth = container_of(auth, struct gss_auth, rpc_auth);
    clnt = gss_auth->client;

- gss_auth->pipe[1]->dentry = rpc_mkpipe_dentry(clnt->cl_path.dentry,
+ gss_auth->pipe[1]->dentry = rpc_mkpipe_dentry(clnt->cl_dentry,
    "gssd",
    clnt, gss_auth->pipe[1]);
    if (IS_ERR(gss_auth->pipe[1]->dentry))
        return PTR_ERR(gss_auth->pipe[1]->dentry);
- gss_auth->pipe[0]->dentry = rpc_mkpipe_dentry(clnt->cl_path.dentry,
+ gss_auth->pipe[0]->dentry = rpc_mkpipe_dentry(clnt->cl_dentry,
    gss_auth->mech->gm_name,
    clnt, gss_auth->pipe[0]);
    if (IS_ERR(gss_auth->pipe[0]->dentry)) {
@@ -806,7 +806,7 @@ static void gss_pipes_dentries_destroy_net(struct rpc_clnt *clnt,

    sb = rpc_get_sb_net(net);
    if (sb) {
- if (clnt->cl_path.dentry)
+ if (clnt->cl_dentry)
        gss_pipes_dentries_destroy(auth);
        rpc_put_sb_net(net);
    }
@@ -821,7 +821,7 @@ static int gss_pipes_dentries_create_net(struct rpc_clnt *clnt,

    sb = rpc_get_sb_net(net);
    if (sb) {
- if (clnt->cl_path.dentry)
+ if (clnt->cl_dentry)
        err = gss_pipes_dentries_create(auth);
        rpc_put_sb_net(net);
    }
diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
index e3ced30..ed7c22d 100644
--- a/net/sunrpc/clnt.c
+++ b/net/sunrpc/clnt.c
@@ -98,12 +98,12 @@ static void rpc_unregister_client(struct rpc_clnt *clnt)

static void __rpc_clnt_remove_pipedir(struct rpc_clnt *clnt)
{
- if (clnt->cl_path.dentry) {
+ if (clnt->cl_dentry) {
    if (clnt->cl_auth && clnt->cl_auth->au_ops->pipes_destroy)
        clnt->cl_auth->au_ops->pipes_destroy(clnt->cl_auth);
- rpc_remove_client_dir(clnt->cl_path.dentry);
+ rpc_remove_client_dir(clnt->cl_dentry);
    }
}

```

```

- clnt->cl_path.dentry = NULL;
+ clnt->cl_dentry = NULL;
}

static void rpc_clnt_remove_pipedir(struct rpc_clnt *clnt)
@@ -154,20 +154,19 @@ static int
rpc_setup_pipedir(struct rpc_clnt *clnt, char *dir_name)
{
    struct super_block *pipefs_sb;
- struct path path;
+ struct dentry *dentry;

- clnt->cl_path.mnt = ERR_PTR(-ENOENT);
- clnt->cl_path.dentry = NULL;
+ clnt->cl_dentry = NULL;
    if (dir_name == NULL)
        return 0;
    pipefs_sb = rpc_get_sb_net(clnt->cl_xprt->xprt_net);
    if (!pipefs_sb)
        return 0;
- path.dentry = rpc_setup_pipedir_sb(pipefs_sb, clnt, dir_name);
+ dentry = rpc_setup_pipedir_sb(pipefs_sb, clnt, dir_name);
    rpc_put_sb_net(clnt->cl_xprt->xprt_net);
- if (IS_ERR(path.dentry))
-     return PTR_ERR(path.dentry);
- clnt->cl_path = path;
+ if (IS_ERR(dentry))
+     return PTR_ERR(dentry);
+ clnt->cl_dentry = dentry;
    return 0;
}

@@ -186,7 +185,7 @@ static int __rpc_pipefs_event(struct rpc_clnt *clnt, unsigned long event,
    BUG_ON(dentry == NULL);
    if (IS_ERR(dentry))
        return PTR_ERR(dentry);
- clnt->cl_path.dentry = dentry;
+ clnt->cl_dentry = dentry;
    if (clnt->cl_auth->au_ops->pipes_create) {
        err = clnt->cl_auth->au_ops->pipes_create(clnt->cl_auth);
        if (err)

```
