Subject: [PATCH v2 1/5] NFS: handle blocklayout pipe PipeFS dentry by network namespace aware routines
Posted by Stanislav Kinsbursky on Tue, 10 Jan 2012 13:04:16 GMT
View Forum Message <> Reply to Message

This patch makes blocklayout pipe dentry 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/blocklayout/blocklayout.c | 61 ++++++++++++++++++++++++++++++-------
 1 files changed, 49 insertions(+), 12 deletions(-)

```
diff --git a/fs/nfs/blocklayout/blocklayout.c b/fs/nfs/blocklayout/blocklayout.c
index 2f4ede1..489f95c 100644
--- a/fs/nfs/blocklayout/blocklayout.c
+++ b/fs/nfs/blocklayout/blocklayout.c
@@ -965,10 +965,55 @@ static const struct rpc_pipe_ops bl_upcall_ops = {
  .destroy_msg = bl_pipe_destroy_msg,
 };

+static struct dentry *nfs4blocklayout_register_sb(struct super_block *sb,
+         struct rpc_pipe *pipe)
+{
+ struct dentry *dir, *dentry;
+
+ dir = rpc_d_lookup_sb(sb, NFS_PIPE_DIRNAME);
+ if (dir == NULL)
+  return ERR_PTR(-ENOENT);
+ dentry = rpc_mkpipe_dentry(dir, "blocklayout", NULL, pipe);
+ dput(dir);
+ return dentry;
+}
+
+static void nfs4blocklayout_unregister_sb(struct super_block *sb,
+       struct rpc_pipe *pipe)
+{
+ if (pipe->dentry)
+  rpc_unlink(pipe->dentry);
+}
+
+static struct dentry *nfs4blocklayout_register_net(struct net *net,
+         struct rpc_pipe *pipe)
+{
+ struct super_block *pipefs_sb;
+ struct dentry *dentry;
```

```
+
+ pipefs_sb = rpc_get_sb_net(net);
+ if (!pipefs_sb)
+  return ERR_PTR(-ENOENT);
+ dentry = nfs4blocklayout_register_sb(pipefs_sb, pipe);
+ rpc_put_sb_net(net);
+ return dentry;
+}
+
+static void nfs4blocklayout_unregister_net(struct net *net,
+        struct rpc_pipe *pipe)
+{
+ struct super_block *pipefs_sb;
+
+ pipefs_sb = rpc_get_sb_net(net);
+ if (pipefs_sb) {
+  nfs4blocklayout_unregister_sb(pipefs_sb, pipe);
+  rpc_put_sb_net(net);
+ }
+}
+
 static int __init nfs4blocklayout_init(void)
 {
  struct vfsmount *mnt;
- struct path path;
  int ret;

  dprintk("%s: NFSv4 Block Layout Driver Registering...\n", __func__);
@@ -984,21 +1029,13 @@ static int __init nfs4blocklayout_init(void)
   ret = PTR_ERR(mnt);
   goto out_remove;
  }
-
- ret = vfs_path_lookup(mnt->mnt_root,
-        mnt,
-        NFS_PIPE_DIRNAME, 0, &path);
- if (ret)
-  goto out_putrpc;
-
  bl_device_pipe = rpc_mkpipe_data(&bl_upcall_ops, 0);
- path_put(&path);
  if (IS_ERR(bl_device_pipe)) {
   ret = PTR_ERR(bl_device_pipe);
   goto out_putrpc;
  }
- bl_device_pipe->dentry = rpc_mkpipe_dentry(path.dentry, "blocklayout",
-        NULL, bl_device_pipe);
+ bl_device_pipe->dentry = nfs4blocklayout_register_net(&init_net,
```

```
+            bl_device_pipe);
  if (IS_ERR(bl_device_pipe->dentry)) {
   ret = PTR_ERR(bl_device_pipe->dentry);
   goto out_destroy_pipe;
@@ -1021,7 +1058,7 @@ static void __exit nfs4blocklayout_exit(void)
         __func__);

  pnfs_unregister_layoutdriver(&blocklayout_type);
- rpc_unlink(bl_device_pipe->dentry);
+ nfs4blocklayout_unregister_net(&init_net, bl_device_pipe);
  rpc_destroy_pipe_data(bl_device_pipe);
  rpc_put_mount();
 }
```