Subject: [PATCH v3 2/4] NFS: release per-net clients lock before calling PipeFS dentries creation
Posted by Stanislav Kinsbursky on Mon, 27 Feb 2012 18:05:37 GMT
View Forum Message <> Reply to Message

v3:
1) Lookup for client is performed from the beginning of the list on each PipeFS
event handling operation.

Lockdep is sad otherwise, because inode mutex is taken on PipeFS dentry
creation, which can be called on mount notification, where this per-net client
lock is taken on clients list walk.

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

---
 fs/nfs/idmap.c | 30 ++++++++++++++++++++++++------
 1 files changed, 24 insertions(+), 6 deletions(-)

diff --git a/fs/nfs/idmap.c b/fs/nfs/idmap.c
index b5c6d8e..d4db3b6 100644
--- a/fs/nfs/idmap.c
+++ b/fs/nfs/idmap.c
@@ -553,23 +553,41 @@ static int __rpc_pipefs_event(struct nfs_client *clp, unsigned long event,
  return err;
 }

-static int rpc_pipefs_event(struct notifier_block *nb, unsigned long event,
-       void *ptr)
+static struct nfs_client *nfs_get_client_for_event(struct net *net, int event)
 {
- struct super_block *sb = ptr;
- struct nfs_net *nn = net_generic(sb->s_fs_info, nfs_net_id);
+ struct nfs_net *nn = net_generic(net, nfs_net_id);
+ struct dentry *cl_dentry;
  struct nfs_client *clp;
- int error = 0;

  spin_lock(&nn->nfs_client_lock);
  list_for_each_entry(clp, &nn->nfs_client_list, cl_share_link) {
   if (clp->rpc_ops != &nfs_v4_clientops)
    continue;
+ cl_dentry = clp->cl_idmap->idmap_pipe->dentry;
+ if (((event == RPC_PIPEFS_MOUNT) && cl_dentry) ||
+   ((event == RPC_PIPEFS_UMOUNT) && !cl_dentry))
+   continue;
+ atomic_inc(&clp->cl_count);

```
+  spin_unlock(&nn->nfs_client_lock);
+  return clp;
+ }
+ spin_unlock(&nn->nfs_client_lock);
+ return NULL;
+}
+
+static int rpc_pipefs_event(struct notifier_block *nb, unsigned long event,
+       void *ptr)
+{
+ struct super_block *sb = ptr;
+ struct nfs_client *clp;
+ int error = 0;
+
+ while ((clp = nfs_get_client_for_event(sb->s_fs_info, event))) {
   error = __rpc_pipefs_event(clp, event, sb);
+  nfs_put_client(clp);
   if (error)
    break;
  }
- spin_unlock(&nn->nfs_client_lock);
  return error;
 }
```