Subject: [PATCH v2 5/7] SUNRPC: put pipefs superblock link on network namespace
Posted by Stanislav Kinsbursky on Tue, 08 Nov 2011 11:16:12 GMT
View Forum Message <> Reply to Message

We have modules (like, pNFS blocklayout module) which creates pipes on rpc_pipefs. Thus we need per-net operations for them. To make it possible we require appropriate super block. So we have to put sb link on network namespace context. Note, that it's not strongly required to create pipes in per-net operations. IOW, if pipefs wasn't mounted yet, that no sb link reference will present on network namespace and in this case we need just need to pass through pipe creation. Pipe dentry will be created during pipefs mount notification.

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

```
---
 net/sunrpc/netns.h    |    2 ++
 net/sunrpc/rpc_pipe.c |    4 ++++
 2 files changed, 6 insertions(+), 0 deletions(-)

diff --git a/net/sunrpc/netns.h b/net/sunrpc/netns.h
index d013bf2..b384252 100644
--- a/net/sunrpc/netns.h
+++ b/net/sunrpc/netns.h
@@ -9,6 +9,8 @@ struct cache_detail;
 struct sunrpc_net {
  struct proc_dir_entry *proc_net_rpc;
  struct cache_detail *ip_map_cache;
+
+ struct super_block *pipefs_sb;
 };

 extern int sunrpc_net_id;
diff --git a/net/sunrpc/rpc_pipe.c b/net/sunrpc/rpc_pipe.c
index 1de840d..831e81d 100644
--- a/net/sunrpc/rpc_pipe.c
+++ b/net/sunrpc/rpc_pipe.c
@@ -1030,6 +1030,7 @@ rpc_fill_super(struct super_block *sb, void *data, int silent)
  struct inode *inode;
  struct dentry *root;
  struct net *net = data;
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
  int err;

  sb->s_blocksize = PAGE_CACHE_SIZE;
@@ -1054,6 +1055,7 @@ rpc_fill_super(struct super_block *sb, void *data, int silent)
  if (err)
   goto err_depopulate;
```

```
  sb->s_fs_info = get_net(net);
+ sn->pipefs_sb = sb;
  return 0;

 err_depopulate:
@@ -1074,7 +1076,9 @@ rpc_mount(struct file_system_type *fs_type,
 void rpc_kill_sb(struct super_block *sb)
 {
  struct net *net = sb->s_fs_info;
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);

+ sn->pipefs_sb = NULL;
  put_net(net);
  blocking_notifier_call_chain(&rpc_pipefs_notifier_list,
        RPC_PIPEFS_UMOUNT,
```