Subject: [PATCH v2 3/5] NFS: blocklayout PipeFS notifier introduced
Posted by Stanislav Kinsbursky on Tue, 10 Jan 2012 13:04:32 GMT
View Forum Message <> Reply to Message

This patch subscribes blocklayout pipes to RPC pipefs notifications. Notifier
is registering on blocklayout module load. This notifier callback is
responsible for creation/destruction of PipeFS blocklayout pipe dentry.
Note that no locking required in notifier callback because PipeFS superblock
pointer is passed as an argument from it's creation or destruction routine and
thus we can be sure about it's validity.

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

---
 fs/nfs/blocklayout/blocklayout.c | 50 ++-
 1 files changed, 49 insertions(+), 1 deletions(-)

```
diff --git a/fs/nfs/blocklayout/blocklayout.c b/fs/nfs/blocklayout/blocklayout.c
index ce76d05..322d920 100644
--- a/fs/nfs/blocklayout/blocklayout.c
+++ b/fs/nfs/blocklayout/blocklayout.c
@@ -984,6 +984,48 @@ static void nfs4blocklayout_unregister_sb(struct super_block *sb,
   rpc_unlink(pipe->dentry);
 }

+static int rpc_pipefs_event(struct notifier_block *nb, unsigned long event,
+      void *ptr)
+{
+ struct super_block *sb = ptr;
+ struct net *net = sb->s_fs_info;
+ struct nfs_net *nn = net_generic(net, nfs_net_id);
+ struct dentry *dentry;
+ int ret = 0;
+
+ if (!try_module_get(THIS_MODULE))
+  return 0;
+
+ if (nn->bl_device_pipe == NULL) {
+  module_put(THIS_MODULE);
+  return 0;
+ }
+
+ switch (event) {
+ case RPC_PIPEFS_MOUNT:
+  dentry = nfs4blocklayout_register_sb(sb, nn->bl_device_pipe);
+  if (IS_ERR(dentry)) {
+   ret = PTR_ERR(dentry);
+   break;
```

```
+  }
+  nn->bl_device_pipe->dentry = dentry;
+  break;
+ case RPC_PIPEFS_UMOUNT:
+  if (nn->bl_device_pipe->dentry)
+   nfs4blocklayout_unregister_sb(sb, nn->bl_device_pipe);
+  break;
+ default:
+  ret = -ENOTSUPP;
+  break;
+ }
+ module_put(THIS_MODULE);
+ return ret;
+}
+
+static struct notifier_block nfs4blocklayout_block = {
+ .notifier_call = rpc_pipefs_event,
+};
+
 static struct dentry *nfs4blocklayout_register_net(struct net *net,
         struct rpc_pipe *pipe)
 {
@@ -1059,12 +1101,17 @@ static int __init nfs4blocklayout_init(void)
   ret = PTR_ERR(mnt);
   goto out_remove;
  }
- ret = register_pernet_subsys(&nfs4blocklayout_net_ops);
+ ret = rpc_pipefs_notifier_register(&nfs4blocklayout_block);
  if (ret)
   goto out_remove;
+ ret = register_pernet_subsys(&nfs4blocklayout_net_ops);
+ if (ret)
+  goto out_notifier;
 out:
  return ret;

+out_notifier:
+ rpc_pipefs_notifier_unregister(&nfs4blocklayout_block);
 out_remove:
  pnfs_unregister_layoutdriver(&blocklayout_type);
  return ret;
@@ -1075,6 +1122,7 @@ static void __exit nfs4blocklayout_exit(void)
  dprintk("%s: NFSv4 Block Layout Driver Unregistering...\n",
         __func__);

+ rpc_pipefs_notifier_unregister(&nfs4blocklayout_block);
  unregister_pernet_subsys(&nfs4blocklayout_net_ops);
  pnfs_unregister_layoutdriver(&blocklayout_type);
```

```
}
```