Subject: [PATCH 3/6] NFS: handle NFS caches dentries by network namespace aware routines
Posted by Stanislav Kinsbursky on Fri, 25 Nov 2011 13:18:22 GMT
View Forum Message <> Reply to Message

This patch makes NFS caches PipeFS dentries allocated and destroyed in network namespace context by PipeFS network namespace aware routines.

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

```
---
 fs/nfs/cache_lib.c   |   44 ++++++++++++++++++++++++++++++++++---------
 fs/nfs/cache_lib.h   |    4 ++--
 fs/nfs/dns_resolve.c |    4 ++--
 3 files changed, 39 insertions(+), 13 deletions(-)

diff --git a/fs/nfs/cache_lib.c b/fs/nfs/cache_lib.c
index 9d79a2e..5dd017b 100644
--- a/fs/nfs/cache_lib.c
+++ b/fs/nfs/cache_lib.c
@@ -13,6 +13,7 @@
 #include <linux/slab.h>
 #include <linux/sunrpc/cache.h>
 #include <linux/sunrpc/rpc_pipe_fs.h>
+#include <net/net_namespace.h>

 #include "cache_lib.h"

@@ -111,20 +112,34 @@ int nfs_cache_wait_for_upcall(struct nfs_cache_defer_req *dreq)
  return 0;
 }

-int nfs_cache_register(struct cache_detail *cd)
+static int nfs_cache_register_sb(struct super_block *sb, struct cache_detail *cd)
+{
+ int ret;
+ struct dentry *dir;
+
+ dir = rpc_d_lookup_sb(sb, "cache");
+ BUG_ON(dir == NULL);
+ ret = sunrpc_cache_register_pipefs(dir, cd->name, 0600, cd);
+ dput(dir);
+ return ret;
+}
+
+int nfs_cache_register_net(struct net *net, struct cache_detail *cd)
 {
  struct vfsmount *mnt;
```

```
- struct path path;
+ struct super_block *pipefs_sb;
  int ret;

  mnt = rpc_get_mount();
  if (IS_ERR(mnt))
   return PTR_ERR(mnt);
- ret = vfs_path_lookup(mnt->mnt_root, mnt, "/cache", 0, &path);
- if (ret)
+ pipefs_sb = rpc_get_sb_net(net);
+ if (!pipefs_sb) {
+  ret = -ENOENT;
   goto err;
- ret = sunrpc_cache_register_pipefs(path.dentry, cd->name, 0600, cd);
- path_put(&path);
+ }
+ ret = nfs_cache_register_sb(pipefs_sb, cd);
+ rpc_put_sb_net(net);
  if (!ret)
   return ret;
 err:
@@ -132,10 +147,21 @@ err:
  return ret;
 }

-void nfs_cache_unregister(struct cache_detail *cd)
+static void nfs_cache_unregister_sb(struct super_block *sb, struct cache_detail *cd)
 {
- sunrpc_cache_unregister_pipefs(cd);
- sunrpc_destroy_cache_detail(cd);
+ if (cd->u.pipefs.dir)
+  sunrpc_cache_unregister_pipefs(cd);
+}
+
+void nfs_cache_unregister_net(struct net *net, struct cache_detail *cd)
+{
+ struct super_block *pipefs_sb;
+
+ pipefs_sb = rpc_get_sb_net(net);
+ if (pipefs_sb) {
+  nfs_cache_unregister_sb(pipefs_sb, cd);
+  rpc_put_sb_net(net);
+ }
  rpc_put_mount();
 }

diff --git a/fs/nfs/cache_lib.h b/fs/nfs/cache_lib.h
index 815dd66..e0a6cc4 100644
```

```
--- a/fs/nfs/cache_lib.h
+++ b/fs/nfs/cache_lib.h
@@ -25,5 +25,5 @@ extern int nfs_cache_wait_for_upcall(struct nfs_cache_defer_req *dreq);

 extern void nfs_cache_init(struct cache_detail *cd);
 extern void nfs_cache_destroy(struct cache_detail *cd);
-extern int nfs_cache_register(struct cache_detail *cd);
-extern void nfs_cache_unregister(struct cache_detail *cd);
+extern int nfs_cache_register_net(struct net *net, struct cache_detail *cd);
+extern void nfs_cache_unregister_net(struct net *net, struct cache_detail *cd);
diff --git a/fs/nfs/dns_resolve.c b/fs/nfs/dns_resolve.c
index 619dea6..3cbf4b8 100644
--- a/fs/nfs/dns_resolve.c
+++ b/fs/nfs/dns_resolve.c
@@ -364,7 +364,7 @@ int nfs_dns_resolver_init(void)
  int err;

  nfs_cache_init(&nfs_dns_resolve);
- err = nfs_cache_register(&nfs_dns_resolve);
+ err = nfs_cache_register_net(&init_net, &nfs_dns_resolve);
  if (err) {
   nfs_cache_destroy(&nfs_dns_resolve);
   return err;
@@ -374,7 +374,7 @@ int nfs_dns_resolver_init(void)

 void nfs_dns_resolver_destroy(void)
 {
- nfs_cache_unregister(&nfs_dns_resolve);
+ nfs_cache_unregister_net(&init_net, &nfs_dns_resolve);
  nfs_cache_destroy(&nfs_dns_resolve);
 }
```