Subject: Re: [PATCH] NFS: init client before declaration
Posted by Myklebust, Trond on Tue, 22 May 2012 15:51:44 GMT
View Forum Message <> Reply to Message

On Tue, 2012-05-22 at 19:29 +0400, Stanislav Kinsbursky wrote:
> On 22.05.2012 19:00, Myklebust, Trond wrote:
> > On Tue, 2012-05-22 at 10:29 -0400, Trond Myklebust wrote:
> >> On Tue, 2012-05-22 at 16:40 +0400, Stanislav Kinsbursky wrote:
> >>> Client have to be initialized prior to adding it to per-net clients list,
> >>> because otherwise there are races, shown below:
> >>>
> >>> CPU#0 CPU#1
> >>> _____ _____
> >>>
> >>> nfs_get_client
> >>> nfs_alloc_client
> >>> list_add(..., nfs_client_list)
> >>> rpc_fill_super
> >>> rpc_pipefs_event
> >>> nfs_get_client_for_event
> >>> __rpc_pipefs_event
> >>> (clp->cl_rpcclient is uninitialized)
> >>> BUG()
> >>> init_client
> >>> clp->cl_rpcclient = ...
> >>>
> >>
> >> Why not simply change nfs_get_client_for_event() so that it doesn't
> >> touch nfs_clients that have clp->cl_cons_state!=NFS_CS_READY?
> >>
> >> That should ensure that it doesn't touch nfs_clients that failed to
> >> initialise and/or are still in the process of being initialised.
> >
> > ...actually, come to think of it. Why not just add a helper function
> > "bool nfs_client_active(const struct nfs_client *clp)" to
> > fs/nfs/client.c that does a call to
> > wait_event_killable(nfs_client_active_wq, clp->cl_cons_state< NFS_CS_INITING);
> > and checks the resulting value of clp->cl_cons_state?
> >
>
> Sorry, but I don't understand the idea...
> Where are you proposing to call this function?
> In __rpc_pipefs_event() prior to dentries creatios?

See below:

8< ---------- ----------
>From f5b90df6381a20395d9f88a199e9e52f44267457 Mon Sep 17 00:00:00 2001

From: Trond Myklebust <Trond.Myklebust@netapp.com>
Date: Tue, 22 May 2012 11:49:55 -0400
Subject: [PATCH] NFSv4: Fix a race in the net namespace mount notification

Since the struct nfs_client gets added to the global nfs_client_list
before it is initialised, it is possible that rpc_pipefs_event can
end up trying to create idmapper entries for such a thing.

The solution is to have the mount notification wait for the
nfs_client initialisation to complete.

Reported-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
Signed-off-by: Trond Myklebust <Trond.Myklebust@netapp.com>
---
 fs/nfs/client.c | 14 ++++++++++++++
 fs/nfs/idmap.c | 3 ++-
 fs/nfs/internal.h | 1 +
 3 files changed, 17 insertions(+), 1 deletions(-)

diff --git a/fs/nfs/client.c b/fs/nfs/client.c
index 60f7e4e..3fa44ef 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -592,6 +592,20 @@ void nfs_mark_client_ready(struct nfs_client *clp, int state)
  wake_up_all(&nfs_client_active_wq);
 }

+static bool nfs_client_ready(struct nfs_client *clp)
+{
+ return clp->cl_cons_state <= NFS_CS_READY;
+}
+
+int nfs_wait_client_ready(struct nfs_client *clp)
+{
+ if (wait_event_killable(nfs_client_active_wq, nfs_client_ready(clp)) < 0)
+ return -ERESTARTSYS;
+ if (clp->cl_cons_state < 0)
+ return clp->cl_cons_state;
+ return 0;
+}
+
 /*
  * With sessions, the client is not marked ready until after a
  * successful EXCHANGE_ID and CREATE_SESSION.
diff --git a/fs/nfs/idmap.c b/fs/nfs/idmap.c
index 3e8edbe..67962c8 100644
--- a/fs/nfs/idmap.c
+++ b/fs/nfs/idmap.c

```
@@ -558,7 +558,8 @@ static int rpc_pipefs_event(struct notifier_block *nb, unsigned long event,
   return 0;

  while ((clp = nfs_get_client_for_event(sb->s_fs_info, event))) {
-  error = __rpc_pipefs_event(clp, event, sb);
+  if (nfs_wait_client_ready(clp) == 0)
+   error = __rpc_pipefs_event(clp, event, sb);
   nfs_put_client(clp);
   if (error)
    break;
diff --git a/fs/nfs/internal.h b/fs/nfs/internal.h
index b777bda..3be00a0 100644
--- a/fs/nfs/internal.h
+++ b/fs/nfs/internal.h
@@ -168,6 +168,7 @@ extern struct nfs_server *nfs_clone_server(struct nfs_server *,
         struct nfs_fattr *,
         rpc_authflavor_t);
 extern void nfs_mark_client_ready(struct nfs_client *clp, int state);
+extern int nfs_wait_client_ready(struct nfs_client *clp);
 extern int nfs4_check_client_ready(struct nfs_client *clp);
 extern struct nfs_client *nfs4_set_ds_client(struct nfs_client* mds_clp,
          const struct sockaddr *ds_addr,
--
1.7.7.6


--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com
```