Subject: [PATCH 1/6] Lockd: create permanent lockd sockets in current network namespace
Posted by Stanislav Kinsbursky on Tue, 31 Jan 2012 11:07:48 GMT
View Forum Message <> Reply to Message

This patch parametrizes Lockd permanent sockets creation routine by network namespace context.
It also replaces hard-coded init_net with current network namespace context in Lockd sockets creation routines.
This approach looks safe, because Lockd is created during NFS mount (or NFS server start) and thus socket is required exactly in current network namespace context. But in the same time it means, that Lockd sockets inherits first Lockd requester network namespace. This issue will be fixed in further patches of the series.

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

---
 fs/lockd/svc.c | 23 +++++++++++++++----------
 1 files changed, 13 insertions(+), 10 deletions(-)

```
diff --git a/fs/lockd/svc.c b/fs/lockd/svc.c
index 55fea92..26d8b78 100644
--- a/fs/lockd/svc.c
+++ b/fs/lockd/svc.c
@@ -189,27 +189,29 @@ lockd(void *vrqstp)
 }

 static int create_lockd_listener(struct svc_serv *serv, const char *name,
-     const int family, const unsigned short port)
+     struct net *net, const int family,
+     const unsigned short port)
 {
  struct svc_xprt *xprt;

- xprt = svc_find_xprt(serv, name, &init_net, family, 0);
+ xprt = svc_find_xprt(serv, name, net, family, 0);
  if (xprt == NULL)
-  return svc_create_xprt(serv, name, &init_net, family, port,
+  return svc_create_xprt(serv, name, net, family, port,
        SVC_SOCK_DEFAULTS);
  svc_xprt_put(xprt);
  return 0;
 }

-static int create_lockd_family(struct svc_serv *serv, const int family)
+static int create_lockd_family(struct svc_serv *serv, struct net *net,
+          const int family)
```

```
 {
  int err;

- err = create_lockd_listener(serv, "udp", family, nlm_udpport);
+ err = create_lockd_listener(serv, "udp", net, family, nlm_udpport);
  if (err < 0)
   return err;

- return create_lockd_listener(serv, "tcp", family, nlm_tcpport);
+ return create_lockd_listener(serv, "tcp", net, family, nlm_tcpport);
 }

 /*
@@ -222,16 +224,16 @@ static int create_lockd_family(struct svc_serv *serv, const int family)
  * Returns zero if all listeners are available; otherwise a
  * negative errno value is returned.
  */
-static int make_socks(struct svc_serv *serv)
+static int make_socks(struct svc_serv *serv, struct net *net)
 {
  static int warned;
  int err;

- err = create_lockd_family(serv, PF_INET);
+ err = create_lockd_family(serv, net, PF_INET);
  if (err < 0)
   goto out_err;

- err = create_lockd_family(serv, PF_INET6);
+ err = create_lockd_family(serv, net, PF_INET6);
  if (err < 0 && err != -EAFNOSUPPORT)
   goto out_err;

@@ -252,6 +254,7 @@ int lockd_up(void)
 {
  struct svc_serv *serv;
  int  error = 0;
+ struct net *net = current->nsproxy->net_ns;

  mutex_lock(&nlmsvc_mutex);
  /*
@@ -275,7 +278,7 @@ int lockd_up(void)
   goto out;
  }

- error = make_socks(serv);
+ error = make_socks(serv, net);
  if (error < 0)
```

goto destroy_and_out;