
Subject: [PATCH 4/6] LockD: make nlm hosts network namespace aware
Posted by Stanislav Kinsbursky on Tue, 31 Jan 2012 11:08:13 GMT
[View Forum Message](#) <[Reply to Message](#)

This object depends on RPC client, and thus on network namespace.
So let's make it's allocation and lookup in network namespace context.

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

```
fs/lockd/clntlock.c      |  3 ++
fs/lockd/host.c          | 16 ++++++
fs/nfs/client.c          |  1 +
include/linux/lockd/bind.h |  1 +
include/linux/lockd/lockd.h|  4 +++
5 files changed, 21 insertions(+), 4 deletions(-)
```

```
diff --git a/fs/lockd/clntlock.c b/fs/lockd/clntlock.c
index 8d4ea83..ba1dc2e 100644
--- a/fs/lockd/clntlock.c
+++ b/fs/lockd/clntlock.c
@@ -62,7 +62,8 @@ struct nlm_host *nlmclnt_init(const struct nlmclnt_initdata *nlm_init)
```

```
host = nlmclnt_lookup_host(nlm_init->address, nlm_init->addrlen,
    nlm_init->protocol, nlm_version,
-   nlm_init->hostname, nlm_init->noresvport);
+   nlm_init->hostname, nlm_init->noresvport,
+   nlm_init->net);
```

```
if (host == NULL) {
    lockd_down();
    return ERR_PTR(-ENOLCK);
```

```
diff --git a/fs/lockd/host.c b/fs/lockd/host.c
index 6f29836..9ebd91d 100644
--- a/fs/lockd/host.c
+++ b/fs/lockd/host.c
```

```
@@ -17,6 +17,8 @@
#include <linux/lockd/lockd.h>
#include <linux/mutex.h>
```

```
+#include <linux/sunrpc/svc_xprt.h>
```

```
+
#include <net/ipv6.h>
```

```
#define NLMDBG_FACILITY NLMDBG_HOSTCACHE
@@ -54,6 +56,7 @@ struct nlm_lookup_host_info {
    const char *hostname; /* remote's hostname */
    const size_t hostname_len; /* it's length */
    const int noresvport; /* use non-priv port */
```

```

+ struct net *net; /* network namespace to bind */
};

/*
@@ -155,6 +158,7 @@ static struct nlm_host *nlm_alloc_host(struct nlm_lookup_host_info *ni,
INIT_LIST_HEAD(&host->h_reclaim);
host->h_nsmhandle = nsm;
host->h_addrbuf = nsm->sm_addrbuf;
+ host->net = ni->net;

out:
return host;
@@ -206,7 +210,8 @@ struct nlm_host *nlmclnt_lookup_host(const struct sockaddr *sap,
const unsigned short protocol,
const u32 version,
const char *hostname,
- int noresvport)
+ int noresvport,
+ struct net *net)
{
struct nlm_lookup_host_info ni = {
.server = 0,
@@ -217,6 +222,7 @@ struct nlm_host *nlmclnt_lookup_host(const struct sockaddr *sap,
.hostname = hostname,
.hostname_len = strlen(hostname),
.noresvport = noresvport,
+ .net = net,
};
struct hlist_head *chain;
struct hlist_node *pos;
@@ -231,6 +237,8 @@ struct nlm_host *nlmclnt_lookup_host(const struct sockaddr *sap,
chain = &nlm_client_hosts[nlm_hash_address(sap)];
hlist_for_each_entry(host, pos, chain, h_hash) {
+ if (host->net != net)
+ continue;
if (!rpc_cmp_addr(nlm_addr(host), sap))
continue;

@@ -318,6 +326,7 @@ struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,
struct nsm_handle *nsm = NULL;
struct sockaddr *src_sap = svc_daddr(rqstp);
size_t src_len = rqstp->rq_daddrlen;
+ struct net *net = rqstp->rq_xprt->xpt_net;
struct nlm_lookup_host_info ni = {
.server = 1,
.sap = svc_addr(rqstp),
@@ -326,6 +335,7 @@ struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,

```

```

.version = rqstp->rq_vers,
.hostname = hostname,
.hostname_len = hostname_len,
+ .net = net,
};

dprintk("lockd: %s(host='%'s', vers=%u, proto=%s)\n", __func__,
@@ -339,6 +349,8 @@ struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,
chain = &nlm_server_hosts[nlm_hash_address(ni.sap)];
hlist_for_each_entry(host, pos, chain, h_hash) {
+ if (host->net != net)
+ continue;
if (!rpc_cmp_addr(nlm_addr(host), ni.sap))
continue;

@@ -431,7 +443,7 @@ nlm_bind_host(struct nlm_host *host)
.to_retries = 5U,
};
struct rpc_create_args args = {
- .net = &init_net,
+ .net = host->net,
.protocol = host->h_proto,
.address = nlm_addr(host),
.addrlen = host->h_addrlen,
diff --git a/fs/nfs/client.c b/fs/nfs/client.c
index 2328dc4..1a5cd49 100644
--- a/fs/nfs/client.c
+++ b/fs/nfs/client.c
@@ -707,6 +707,7 @@ static int nfs_start_lockd(struct nfs_server *server)
.nfs_version = clp->rpc_ops->version,
.noressvport = server->flags & NFS_MOUNT_NORESSVPORT ?
1 : 0,
+ .net = clp->net,
};

if (nlm_init.nfs_version > 3)
diff --git a/include/linux/lockd/bind.h b/include/linux/lockd/bind.h
index fbc48f8..11a966e 100644
--- a/include/linux/lockd/bind.h
+++ b/include/linux/lockd/bind.h
@@ -42,6 +42,7 @@ struct nlmcnt_initdata {
unsigned short protocol;
u32 nfs_version;
int noressvport;
+ struct net *net;
};

```

```
/*
diff --git a/include/linux/lockd/lockd.h b/include/linux/lockd/lockd.h
index 8949167..94b3d13 100644
--- a/include/linux/lockd/lockd.h
+++ b/include/linux/lockd/lockd.h
@@ -67,6 +67,7 @@ struct nlm_host {
    struct list_head h_reclaim; /* Locks in RECLAIM state */
    struct nsm_handle *h_nsmhandle; /* NSM status handle */
    char *h_addrbuf; /* address eyecatcher */
+   struct net *net; /* host net */
};

/*
@@ -222,7 +223,8 @@ struct nlm_host *nlmclnt_lookup_host(const struct sockaddr *sap,
    const unsigned short protocol,
    const u32 version,
    const char *hostname,
-   int noresvport);
+   int noresvport,
+   struct net *net);
void nlmclnt_release_host(struct nlm_host *);
struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,
    const char *hostname,
```
