
Subject: [PATCH 6/6] Lockd: shutdown NLM hosts in network namespace context
Posted by Stanislav Kinsbursky on Tue, 31 Jan 2012 11:08:29 GMT
[View Forum Message](#) <[Reply to Message](#)

Lockd now managed in network namespace context. And this patch introduces network namespace related NLM hosts shutdown in case of releasing per-net Lockd resources.

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

```
---  
fs/lockd/host.c      | 26 ++++++-----  
fs/lockd/svc.c       |  4 +-+-  
include/linux/lockd/lockd.h |  1 +  
3 files changed, 23 insertions(+), 8 deletions(-)
```

```
diff --git a/fs/lockd/host.c b/fs/lockd/host.c  
index 9ebd91d..eb75ca7 100644  
--- a/fs/lockd/host.c  
+++ b/fs/lockd/host.c  
@@ -565,12 +565,8 @@ void nlm_host_rebooted(const struct nlm_reboot *info)  
    nsm_release(nsm);  
}  
  
-/*  
- * Shut down the hosts module.  
- * Note that this routine is called only at server shutdown time.  
- */  
void  
-nlm_shutdown_hosts(void)  
+nlm_shutdown_hosts_net(struct net *net)  
{  
    struct hlist_head *chain;  
    struct hlist_node *pos;  
@@ -582,6 +578,8 @@ nlm_shutdown_hosts(void)  
    /* First, make all hosts eligible for gc */  
    dprintk("lockd: nuking all hosts...\n");  
    for_each_host(host, pos, chain, nlm_server_hosts) {  
+     if (net && host->net != net)  
+         continue;  
    host->h_expires = jiffies - 1;  
    if (host->h_rpcclnt) {  
        rpc_shutdown_client(host->h_rpcclnt);  
@@ -592,15 +590,29 @@ nlm_shutdown_hosts(void)  
    /* Then, perform a garbage collection pass */  
    nlm_gc_hosts();  
    mutex_unlock(&nlm_host_mutex);  
+}
```

```

+
+/*
+ * Shut down the hosts module.
+ * Note that this routine is called only at server shutdown time.
+ */
+void
+nlm_shutdown_hosts(void)
+{
+ struct hlist_head *chain;
+ struct hlist_node *pos;
+ struct nlm_host *host;
+
+ nlm_shutdown_hosts_net(NULL);

/* complain if any hosts are left */
if (nrhosts != 0) {
    printk(KERN_WARNING "lockd: couldn't shutdown host module!\n");
    dprintk("lockd: %lu hosts left:\n", nrhosts);
    for_each_host(host, pos, chain, nlm_server_hosts) {
-    dprintk("      %s (cnt %d use %d exp %ld)\n",
+    dprintk("      %s (cnt %d use %d exp %ld net %p)\n",
            host->h_name, atomic_read(&host->h_count),
-    host->h_inuse, host->h_expires);
+    host->h_inuse, host->h_expires, host->net);
        }
    }
}

diff --git a/fs/lockd/svc.c b/fs/lockd/svc.c
index 86e17e8..b34100e 100644
--- a/fs/lockd/svc.c
+++ b/fs/lockd/svc.c
@@ -281,8 +281,10 @@ static void lockd_down_net(struct net *net)
    struct svc_serv *serv = nlmsvc_rqst->rq_server;

    if (ln->nlmsvc_users) {
-    if (--ln->nlmsvc_users == 0)
+    if (--ln->nlmsvc_users == 0) {
+        nlm_shutdown_hosts_net(net);
        svc_shutdown_net(serv, net);
+    }
    } else {
        printk(KERN_ERR "lockd_down_net: no users! task=%p, net=%p\n",
              nlmsvc_task, net);
    }
}

diff --git a/include/linux/lockd/lockd.h b/include/linux/lockd/lockd.h
index 94b3d13..f04ce6a 100644
--- a/include/linux/lockd/lockd.h
+++ b/include/linux/lockd/lockd.h
@@ -234,6 +234,7 @@ struct rpc_clnt * nlm_bind_host(struct nlm_host *);

```

```
void    nlm_rebind_host(struct nlm_host *);  
struct nlm_host * nlm_get_host(struct nlm_host *);  
void    nlm_shutdown_hosts(void);  
+void   nlm_shutdown_hosts_net(struct net *net);  
void    nlm_host_rebooted(const struct nlm_reboot *);  
  
/*
```
