Subject: [PATCH 2/6] Lockd: pernet usage counter introduced
Posted by Stanislav Kinsbursky on Tue, 31 Jan 2012 11:07:57 GMT
View Forum Message <> Reply to Message

Lockd is going to be shared between network namespaces - i.e. going to be able
to handle lock requests from different network namespaces. This means, that
network namespace related resources have to be allocated not once (like now),
but for every network namespace context, from which service is requested to
operate.
This patch implements Lockd per-net users accounting. New per-net counter is
used to determine, when per-net resources have to be freed.

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

---
 fs/lockd/netns.h | 12 ++++++++++
 fs/lockd/svc.c | 45 +++---
 2 files changed, 54 insertions(+), 3 deletions(-)
 create mode 100644 fs/lockd/netns.h

diff --git a/fs/lockd/netns.h b/fs/lockd/netns.h
new file mode 100644
index 0000000..ce227e0
--- /dev/null
+++ b/fs/lockd/netns.h
@@ -0,0 +1,12 @@
+#ifndef __LOCKD_NETNS_H__
+#define __LOCKD_NETNS_H__
+
+#include <net/netns/generic.h>
+
+struct lockd_net {
+ unsigned int nlmsvc_users;
+};
+
+extern int lockd_net_id;
+
+#endif
diff --git a/fs/lockd/svc.c b/fs/lockd/svc.c
index 26d8b78..b461733 100644
--- a/fs/lockd/svc.c
+++ b/fs/lockd/svc.c
@@ -35,6 +35,8 @@
 #include <linux/lockd/lockd.h>
 #include <linux/nfs.h>

+#include "netns.h"
+

```
 #define NLMDBG_FACILITY  NLMDBG_SVC
 #define LOCKD_BUFSIZE  (1024 + NLMSVC_XDRSIZE)
 #define ALLOWED_SIGS  (sigmask(SIGKILL))
@@ -50,6 +52,8 @@ static struct task_struct *nlmsvc_task;
 static struct svc_rqst  *nlmsvc_rqst;
 unsigned long   nlmsvc_timeout;

+int lockd_net_id;
+
 /*
  * These can be set at insmod time (useful for NFS as root filesystem),
  * and also changed through the sysctl interface.  -- Jamie Lokier, Aug 2003
@@ -315,8 +319,12 @@ int lockd_up(void)
 destroy_and_out:
  svc_destroy(serv);
 out:
- if (!error)
+ if (!error) {
+  struct lockd_net *ln = net_generic(net, lockd_net_id);
+
+  ln->nlmsvc_users++;
   nlmsvc_users++;
+ }
  mutex_unlock(&nlmsvc_mutex);
  return error;
 }
@@ -499,24 +507,55 @@ module_param_call(nlm_tcpport, param_set_port, param_get_int,
 module_param(nsm_use_hostnames, bool, 0644);
 module_param(nlm_max_connections, uint, 0644);

+static int lockd_init_net(struct net *net)
+{
+ return 0;
+}
+
+static void lockd_exit_net(struct net *net)
+{
+}
+
+static struct pernet_operations lockd_net_ops = {
+ .init = lockd_init_net,
+ .exit = lockd_exit_net,
+ .id = &lockd_net_id,
+ .size = sizeof(struct lockd_net),
+};
+
+
 /*
```

```
 * Initialising and terminating the module.
 */

static int __init init_nlm(void)
{
+ int err;
+
#ifdef CONFIG_SYSCTL
+ err = -ENOMEM;
 nlm_sysctl_table = register_sysctl_table(nlm_sysctl_root);
- return nlm_sysctl_table ? 0 : -ENOMEM;
-#else
+ if (nlm_sysctl_table == NULL)
+  goto err_sysctl;
+#endif
+ err = register_pernet_subsys(&lockd_net_ops);
+ if (err)
+  goto err_pernet;
 return 0;
+
+err_pernet:
+#ifdef CONFIG_SYSCTL
+ unregister_sysctl_table(nlm_sysctl_table);
#endif
+err_sysctl:
+ return err;
}

static void __exit exit_nlm(void)
{
 /* FIXME: delete all NLM clients */
 nlm_shutdown_hosts();
+ unregister_pernet_subsys(&lockd_net_ops);
#ifdef CONFIG_SYSCTL
 unregister_sysctl_table(nlm_sysctl_table);
#endif
```