
Subject: [PATCH 5/6] LockD: service start function introduced
Posted by [Stanislav Kinsbursky](#) on Wed, 25 Apr 2012 14:23:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is just a code move, which from my POW makes code looks better.

I.e. now on start we have 3 different stages:

- 1) Service creation.
- 2) Service per-net data allocation.
- 3) Service start.

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

fs/lockd/svc.c | 67 ++++++-----
1 files changed, 42 insertions(+), 25 deletions(-)

diff --git a/fs/lockd/svc.c b/fs/lockd/svc.c

index 53cd69e..b47bf77 100644

--- a/fs/lockd/svc.c

+++ b/fs/lockd/svc.c

```
@@ -291,6 +291,46 @@ static void lockd_down_net(struct svc_serv *serv, struct net *net)
 }
 }
```

```
+static int lockd_start_svc(struct svc_serv *serv)
```

```
+{
```

```
+ int error;
```

```
+
```

```
+ if (nlmsvc_rqst)
```

```
+ return 0;
```

```
+
```

```
+ /*
```

```
+ * Create the kernel thread and wait for it to start.
```

```
+ */
```

```
+ nlmsvc_rqst = svc_prepare_thread(serv, &serv->sv_pools[0], NUMA_NO_NODE);
```

```
+ if (IS_ERR(nlmsvc_rqst)) {
```

```
+ error = PTR_ERR(nlmsvc_rqst);
```

```
+ printk(KERN_WARNING
```

```
+ "lockd_up: svc_rqst allocation failed, error=%d\n",
```

```
+ error);
```

```
+ goto out_rqst;
```

```
+ }
```

```
+
```

```
+ svc_sock_update_bufs(serv);
```

```
+ serv->sv_maxconn = nlm_max_connections;
```

```
+
```

```
+ nlmsvc_task = kthread_run(lockd, nlmsvc_rqst, serv->sv_name);
```

```
+ if (IS_ERR(nlmsvc_task)) {
```

```

+ error = PTR_ERR(nlmsvc_task);
+ printk(KERN_WARNING
+ "lockd_up: kthread_run failed, error=%d\n", error);
+ goto out_task;
+ }
+ dprintk("lockd_up: service started\n");
+ return 0;
+
+out_task:
+ svc_exit_thread(nlmsvc_rqst);
+ nlmsvc_task = NULL;
+out_rqst:
+ nlmsvc_rqst = NULL;
+ return error;
+}
+
static struct svc_serv *lockd_create_svc(void)
{
    struct svc_serv *serv;
@@ -343,32 +383,9 @@ int lockd_up(struct net *net)
    if (error < 0)
        goto err_net;

- /*
-  * Create the kernel thread and wait for it to start.
-  */
- nlmsvc_rqst = svc_prepare_thread(serv, &serv->sv_pools[0], NUMA_NO_NODE);
- if (IS_ERR(nlmsvc_rqst)) {
-     error = PTR_ERR(nlmsvc_rqst);
-     nlmsvc_rqst = NULL;
-     printk(KERN_WARNING
- "lockd_up: svc_rqst allocation failed, error=%d\n",
-     error);
-     goto err_start;
- }
-
- svc_sock_update_bufs(serv);
- serv->sv_maxconn = nlm_max_connections;
-
- nlmsvc_task = kthread_run(lockd, nlmsvc_rqst, serv->sv_name);
- if (IS_ERR(nlmsvc_task)) {
-     error = PTR_ERR(nlmsvc_task);
-     svc_exit_thread(nlmsvc_rqst);
-     nlmsvc_task = NULL;
-     nlmsvc_rqst = NULL;
-     printk(KERN_WARNING
- "lockd_up: kthread_run failed, error=%d\n", error);
+ error = lockd_start_svc(serv);

```

```
+ if (error < 0)
  goto err_start;
- }

nlmsvc_users++;
/*
```
