Subject: [PATCH 4/4] SUNRPC: service shutdown function in network namespace context introduced
Posted by Stanislav Kinsbursky on Wed, 25 Jan 2012 13:47:34 GMT
View Forum Message <> Reply to Message

This function is enough for releasing resources, allocated for network
namespace context, in case of sharing service between them.
IOW, each service "user" (LockD, NFSd, etc), which wants to share service
between network namespaces, have to release related resources by the function,
introduced in this patch, instead of performing service shutdown (of course in
case the service is shared already to the moment of release).

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

---
 net/sunrpc/svc.c | 33 ++++++++++++++++++++++++++----------
 1 files changed, 20 insertions(+), 13 deletions(-)

diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
index 279bbd8..3adfd26 100644
--- a/net/sunrpc/svc.c
+++ b/net/sunrpc/svc.c
@@ -510,6 +510,24 @@ svc_create_pooled(struct svc_program *prog, unsigned int bufsize,
 }
 EXPORT_SYMBOL_GPL(svc_create_pooled);

+void svc_shutdown_net(struct svc_serv *serv, struct net *net)
+{
+ /*
+ * The set of xprts (contained in the sv_tempsocks and
+ * sv_permsocks lists) is now constant, since it is modified
+ * only by accepting new sockets (done by service threads in
+ * svc_recv) or aging old ones (done by sv_temptimer), or
+ * configuration changes (excluded by whatever locking the
+ * caller is using--nfsd_mutex in the case of nfsd). So it's
+ * safe to traverse those lists and shut everything down:
+ */
+ svc_close_net(serv, net);
+
+ if (serv->sv_shutdown)
+ serv->sv_shutdown(serv, net);
+}
+EXPORT_SYMBOL_GPL(svc_shutdown_net);
+
 /*
  * Destroy an RPC service. Should be called with appropriate locking to
  * protect the sv_nrthreads, sv_permsocks and sv_tempsocks.
@@ -532,23 +550,12 @@ svc_destroy(struct svc_serv *serv)

```
   printk("svc_destroy: no threads for serv=%p!\n", serv);

  del_timer_sync(&serv->sv_temptimer);
- /*
-  * The set of xprts (contained in the sv_tempsocks and
-  * sv_permsocks lists) is now constant, since it is modified
-  * only by accepting new sockets (done by service threads in
-  * svc_recv) or aging old ones (done by sv_temptimer), or
-  * configuration changes (excluded by whatever locking the
-  * caller is using--nfsd_mutex in the case of nfsd).  So it's
-  * safe to traverse those lists and shut everything down:
-  */
- svc_close_net(serv, net);
+
+ svc_shutdown_net(serv, net);

  BUG_ON(!list_empty(&serv->sv_permsocks));
  BUG_ON(!list_empty(&serv->sv_tempsocks));

- if (serv->sv_shutdown)
-  serv->sv_shutdown(serv, net);
-
  cache_clean_deferred(serv);

  if (svc_serv_is_pooled(serv))
```