Subject: [PATCH 5/7] SUNRPC: pass network namespace to service registering routines
Posted by Stanislav Kinsbursky on Thu, 15 Dec 2011 16:01:22 GMT
View Forum Message <> Reply to Message

Lockd and NFSd services will handle requests from and to many network nsamespaces. And thus have to be registered and unregistered per network namespace.

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

---
include/linux/sunrpc/svc.h | 2 +-
net/sunrpc/svc.c | 42 ++++++++++++++++++++++++++----------
net/sunrpc/svcsock.c | 3 ++-
3 files changed, 26 insertions(+), 21 deletions(-)

```
diff --git a/include/linux/sunrpc/svc.h b/include/linux/sunrpc/svc.h
index 35b37b1..d3563c2 100644
--- a/include/linux/sunrpc/svc.h
+++ b/include/linux/sunrpc/svc.h
@@ -428,7 +428,7 @@ void     svc_destroy(struct svc_serv *);
 int     svc_process(struct svc_rqst *);
 int     bc_svc_process(struct svc_serv *, struct rpc_rqst *,
    struct svc_rqst *);
-int     svc_register(const struct svc_serv *, const int,
+int     svc_register(const struct svc_serv *, struct net *, const int,
     const unsigned short, const unsigned short);

 void     svc_wake_up(struct svc_serv *);
diff --git a/net/sunrpc/svc.c b/net/sunrpc/svc.c
index 03e9f04..137475a 100644
--- a/net/sunrpc/svc.c
+++ b/net/sunrpc/svc.c
@@ -30,7 +30,7 @@

 #define RPCDBG_FACILITY RPCDBG_SVCDSP

-static void svc_unregister(const struct svc_serv *serv);
+static void svc_unregister(const struct svc_serv *serv, struct net *net);

 #define svc_serv_is_pooled(serv)    ((serv)->sv_function)

@@ -375,13 +375,13 @@ static int svc_rpcb_setup(struct svc_serv *serv)
   return err;

  /* Remove any stale portmap registrations */
- svc_unregister(serv);
```

```
+ svc_unregister(serv, &init_net);
  return 0;
 }

 void svc_rpcb_cleanup(struct svc_serv *serv)
 {
- svc_unregister(serv);
+ svc_unregister(serv, &init_net);
  rpcb_put_local(&init_net);
 }
 EXPORT_SYMBOL_GPL(svc_rpcb_cleanup);
@@ -790,7 +790,8 @@ EXPORT_SYMBOL_GPL(svc_exit_thread);
  * Returns zero on success; a negative errno value is returned
  * if any error occurs.
  */
-static int __svc_rpcb_register4(const u32 program, const u32 version,
+static int __svc_rpcb_register4(struct net *net, const u32 program,
+    const u32 version,
     const unsigned short protocol,
     const unsigned short port)
 {
@@ -813,7 +814,7 @@ static int __svc_rpcb_register4(const u32 program, const u32 version,
   return -ENOPROTOOPT;
  }

- error = rpcb_v4_register(&init_net, program, version,
+ error = rpcb_v4_register(net, program, version,
      (const struct sockaddr *)&sin, netid);

  /*
@@ -821,7 +822,7 @@ static int __svc_rpcb_register4(const u32 program, const u32 version,
   * registration request with the legacy rpcbind v2 protocol.
   */
  if (error == -EPROTONOSUPPORT)
-  error = rpcb_register(&init_net, program, version, protocol, port);
+  error = rpcb_register(net, program, version, protocol, port);

  return error;
 }
@@ -837,7 +838,8 @@ static int __svc_rpcb_register4(const u32 program, const u32 version,
  * Returns zero on success; a negative errno value is returned
  * if any error occurs.
  */
-static int __svc_rpcb_register6(const u32 program, const u32 version,
+static int __svc_rpcb_register6(struct net *net, const u32 program,
+    const u32 version,
     const unsigned short protocol,
     const unsigned short port)
```

```
 {
@@ -860,7 +862,7 @@ static int __svc_rpcb_register6(const u32 program, const u32 version,
   return -ENOPROTOOPT;
  }

- error = rpcb_v4_register(&init_net, program, version,
+ error = rpcb_v4_register(net, program, version,
      (const struct sockaddr *)&sin6, netid);

  /*
@@ -880,7 +882,7 @@ static int __svc_rpcb_register6(const u32 program, const u32 version,
  * Returns zero on success; a negative errno value is returned
  * if any error occurs.
  */
-static int __svc_register(const char *progname,
+static int __svc_register(struct net *net, const char *progname,
      const u32 program, const u32 version,
      const int family,
      const unsigned short protocol,
@@ -890,12 +892,12 @@ static int __svc_register(const char *progname,

  switch (family) {
  case PF_INET:
-  error = __svc_rpcb_register4(program, version,
+  error = __svc_rpcb_register4(net, program, version,
       protocol, port);
   break;
 #if defined(CONFIG_IPV6) || defined(CONFIG_IPV6_MODULE)
  case PF_INET6:
-  error = __svc_rpcb_register6(program, version,
+  error = __svc_rpcb_register6(net, program, version,
       protocol, port);
 #endif /* defined(CONFIG_IPV6) || defined(CONFIG_IPV6_MODULE) */
  }
@@ -909,14 +911,16 @@ static int __svc_register(const char *progname,
 /**
  * svc_register - register an RPC service with the local portmapper
  * @serv: svc_serv struct for the service to register
+ * @net: net namespace for the service to register
  * @family: protocol family of service's listener socket
  * @proto: transport protocol number to advertise
  * @port: port to advertise
  *
  * Service is registered for any address in the passed-in protocol family
  */
-int svc_register(const struct svc_serv *serv, const int family,
-   const unsigned short proto, const unsigned short port)
+int svc_register(const struct svc_serv *serv, struct net *net,
```

```
+   const int family, const unsigned short proto,
+   const unsigned short port)
 {
  struct svc_program *progp;
  unsigned int  i;
@@ -941,7 +945,7 @@ int svc_register(const struct svc_serv *serv, const int family,
    if (progp->pg_vers[i]->vs_hidden)
     continue;

-   error = __svc_register(progp->pg_name, progp->pg_prog,
+   error = __svc_register(net, progp->pg_name, progp->pg_prog,
       i, family, proto, port);
    if (error < 0)
     break;
@@ -958,19 +962,19 @@ int svc_register(const struct svc_serv *serv, const int family,
  * any "inet6" entries anyway.  So a PMAP_UNSET should be sufficient
  * in this case to clear all existing entries for [program, version].
  */
-static void __svc_unregister(const u32 program, const u32 version,
+static void __svc_unregister(struct net *net, const u32 program, const u32 version,
         const char *progname)
 {
  int error;

- error = rpcb_v4_register(&init_net, program, version, NULL, "");
+ error = rpcb_v4_register(net, program, version, NULL, "");

  /*
   * User space didn't support rpcbind v4, so retry this
   * request with the legacy rpcbind v2 protocol.
   */
  if (error == -EPROTONOSUPPORT)
-  error = rpcb_register(&init_net, program, version, 0, 0);
+  error = rpcb_register(net, program, version, 0, 0);

  dprintk("svc: %s(%sv%u), error %d\n",
    __func__, progname, version, error);
@@ -984,7 +988,7 @@ static void __svc_unregister(const u32 program, const u32 version,
  * The result of unregistration is reported via dprintk for those who want
  * verification of the result, but is otherwise not important.
  */
-static void svc_unregister(const struct svc_serv *serv)
+static void svc_unregister(const struct svc_serv *serv, struct net *net)
 {
  struct svc_program *progp;
  unsigned long flags;
@@ -1001,7 +1005,7 @@ static void svc_unregister(const struct svc_serv *serv)
```

```
    dprintk("svc: attempting to unregister %sv%u\n",
     progp->pg_name, i);
-   __svc_unregister(progp->pg_prog, i, progp->pg_name);
+   __svc_unregister(net, progp->pg_prog, i, progp->pg_name);
   }
  }

diff --git a/net/sunrpc/svcsock.c b/net/sunrpc/svcsock.c
index 277909e..110735f 100644
--- a/net/sunrpc/svcsock.c
+++ b/net/sunrpc/svcsock.c
@@ -1409,7 +1409,8 @@ static struct svc_sock *svc_setup_socket(struct svc_serv *serv,

  /* Register socket with portmapper */
  if (*errp >= 0 && pmap_register)
-  *errp = svc_register(serv, inet->sk_family, inet->sk_protocol,
+  *errp = svc_register(serv, sock->sk->sk_net, inet->sk_family,
+         inet->sk_protocol,
          ntohs(inet_sk(inet)->inet_sport));

  if (*errp < 0) {
```