Subject: [PATCH 1/4] SUNRPC: replace per-net client lock by rw mutex
Posted by Stanislav Kinsbursky on Mon, 27 Feb 2012 13:48:52 GMT
View Forum Message <> Reply to Message

Lockdep is sad otherwise, because inode mutex is taken on PipeFS dentry
creation, which can be called on mount notification, where this per-net client
lock is taken on clients list walk.

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

---
 net/sunrpc/clnt.c | 16 ++++++++--------
 net/sunrpc/netns.h | 2 +-
 net/sunrpc/sunrpc_syms.c | 2 +-
 3 files changed, 10 insertions(+), 10 deletions(-)

diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
index bb7ed2f3..2d3a547 100644
--- a/net/sunrpc/clnt.c
+++ b/net/sunrpc/clnt.c
@@ -83,18 +83,18 @@ static void rpc_register_client(struct rpc_clnt *clnt)
 {
  struct sunrpc_net *sn = net_generic(clnt->cl_xprt->xprt_net, sunrpc_net_id);

- spin_lock(&sn->rpc_client_lock);
+ down_write(&sn->rpc_client_lock);
  list_add(&clnt->cl_clients, &sn->all_clients);
- spin_unlock(&sn->rpc_client_lock);
+ up_write(&sn->rpc_client_lock);
 }

 static void rpc_unregister_client(struct rpc_clnt *clnt)
 {
  struct sunrpc_net *sn = net_generic(clnt->cl_xprt->xprt_net, sunrpc_net_id);

- spin_lock(&sn->rpc_client_lock);
+ down_write(&sn->rpc_client_lock);
  list_del(&clnt->cl_clients);
- spin_unlock(&sn->rpc_client_lock);
+ up_write(&sn->rpc_client_lock);
 }

 static void __rpc_clnt_remove_pipedir(struct rpc_clnt *clnt)
@@ -212,13 +212,13 @@ static int rpc_pipefs_event(struct notifier_block *nb, unsigned long
event,
  int error = 0;
  struct sunrpc_net *sn = net_generic(sb->s_fs_info, sunrpc_net_id);

```
- spin_lock(&sn->rpc_client_lock);
+ down_read(&sn->rpc_client_lock);
  list_for_each_entry(clnt, &sn->all_clients, cl_clients) {
   error = __rpc_pipefs_event(clnt, event, sb);
   if (error)
    break;
  }
- spin_unlock(&sn->rpc_client_lock);
+ up_read(&sn->rpc_client_lock);
  return error;
 }

@@ -1947,7 +1947,7 @@ void rpc_show_tasks(struct net *net)
  int header = 0;
  struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);

- spin_lock(&sn->rpc_client_lock);
+ down_read(&sn->rpc_client_lock);
  list_for_each_entry(clnt, &sn->all_clients, cl_clients) {
   spin_lock(&clnt->cl_lock);
   list_for_each_entry(task, &clnt->cl_tasks, tk_task) {
@@ -1959,6 +1959,6 @@ void rpc_show_tasks(struct net *net)
   }
   spin_unlock(&clnt->cl_lock);
  }
- spin_unlock(&sn->rpc_client_lock);
+ up_read(&sn->rpc_client_lock);
 }
 #endif
diff --git a/net/sunrpc/netns.h b/net/sunrpc/netns.h
index ce7bd44..88f4b2e 100644
--- a/net/sunrpc/netns.h
+++ b/net/sunrpc/netns.h
@@ -17,7 +17,7 @@ struct sunrpc_net {
  struct mutex pipefs_sb_lock;

  struct list_head all_clients;
- spinlock_t rpc_client_lock;
+ struct rw_semaphore rpc_client_lock;

  struct rpc_clnt *rpcb_local_clnt;
  struct rpc_clnt *rpcb_local_clnt4;
diff --git a/net/sunrpc/sunrpc_syms.c b/net/sunrpc/sunrpc_syms.c
index 21d106e..57fa75f 100644
--- a/net/sunrpc/sunrpc_syms.c
+++ b/net/sunrpc/sunrpc_syms.c
@@ -49,7 +49,7 @@ static __net_init int sunrpc_init_net(struct net *net)
```

```
  rpc_pipefs_init_net(net);
  INIT_LIST_HEAD(&sn->all_clients);
- spin_lock_init(&sn->rpc_client_lock);
+ init_rwsem(&sn->rpc_client_lock);
  spin_lock_init(&sn->rpcb_clnt_lock);
  return 0;
```