
Subject: [PATCH 06/11] SUNRPC: register xs_tunables sysctl table per network namespace

Posted by [Stanislav Kinsbursky](#) on Wed, 14 Dec 2011 10:47:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

This patch introduces per-net sysctl header for xs_tunables table and moves xs_tunables table registration to per-net operations.

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

```
---
include/linux/sunrpc/xprtsock.h | 3 ++
net/sunrpc/netns.h              | 1 +
net/sunrpc/sunrpc_syms.c        | 7 +++++
net/sunrpc/xprtsock.c           | 52 ++++++++++++++++++++++++++++++++++++++-----
4 files changed, 49 insertions(+), 14 deletions(-)
```

```
diff --git a/include/linux/sunrpc/xprtsock.h b/include/linux/sunrpc/xprtsock.h
```

```
index 3f14a02..3ba4920 100644
```

```
--- a/include/linux/sunrpc/xprtsock.h
```

```
+++ b/include/linux/sunrpc/xprtsock.h
```

```
@@ -9,6 +9,9 @@
```

```
#ifdef __KERNEL__
```

```
+int socket_sysctl_init(struct net *);
```

```
+void socket_sysctl_exit(struct net *);
```

```
+
```

```
int init_socket_xprt(void);
```

```
void cleanup_socket_xprt(void);
```

```
diff --git a/net/sunrpc/netns.h b/net/sunrpc/netns.h
```

```
index 05ce3b8..da825e5 100644
```

```
--- a/net/sunrpc/netns.h
```

```
+++ b/net/sunrpc/netns.h
```

```
@@ -15,6 +15,7 @@ struct sunrpc_net {
```

```
#ifdef RPC_DEBUG
```

```
struct ctl_table_set sysctls;
```

```
struct ctl_table_header *debug_ctl_header;
```

```
+ struct ctl_table_header *xs_tunables_header;
```

```
#endif
```

```
};
```

```
diff --git a/net/sunrpc/sunrpc_syms.c b/net/sunrpc/sunrpc_syms.c
```

```
index 0fd156a..2b76e18 100644
```

```
--- a/net/sunrpc/sunrpc_syms.c
```

```
+++ b/net/sunrpc/sunrpc_syms.c
```

```
@@ -42,12 +42,18 @@ static __net_init int sunrpc_init_net(struct net *net)
```

```

    err = debug_sysctl_init(net);
    if (err)
        goto err_sysctl;
+
+ err = socket_sysctl_init(net);
+ if (err)
+     goto err_sock_ctl;
#endif
    INIT_LIST_HEAD(&sn->all_clients);
    spin_lock_init(&sn->rpc_client_lock);
    return 0;

#ifdef RPC_DEBUG
+err_sock_ctl:
+ debug_sysctl_exit(net);
err_sysctl:
    ip_map_cache_destroy(net);
#endif
@@ -60,6 +66,7 @@ err_proc:
static __net_exit void sunrpc_exit_net(struct net *net)
{
#ifdef RPC_DEBUG
+ socket_sysctl_exit(net);
    debug_sysctl_exit(net);
#endif
    ip_map_cache_destroy(net);
diff --git a/net/sunrpc/xprtsock.c b/net/sunrpc/xprtsock.c
index 1acc9a7..9e42db6 100644
--- a/net/sunrpc/xprtsock.c
+++ b/net/sunrpc/xprtsock.c
@@ -74,14 +74,14 @@ static unsigned int xs_tcp_fin_timeout __read_mostly =
XS_TCP_LINGER_TO;

#ifdef RPC_DEBUG

+#include "netns.h"
+
static unsigned int min_slot_table_size = RPC_MIN_SLOT_TABLE;
static unsigned int max_slot_table_size = RPC_MAX_SLOT_TABLE;
static unsigned int max_tcp_slot_table_limit = RPC_MAX_SLOT_TABLE_LIMIT;
static unsigned int xprt_min_resvport_limit = RPC_MIN_RESVPORT;
static unsigned int xprt_max_resvport_limit = RPC_MAX_RESVPORT;

-static struct ctl_table_header *sunrpc_table_header;
-
/*
 * FIXME: changing the UDP slot table size should also resize the UDP
 *        socket buffers for existing UDP transports

```

```

@@ -2870,17 +2870,48 @@ static struct xprt_class xs_bc_tcp_transport = {
    .setup = xs_setup_bc_tcp,
};

#ifdef RPC_DEBUG
+int create_xs_tunables_table(struct net *net)
+{
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
+
+ sn->xs_tunables_header = register_sunrpc_sysctl(net, xs_tunables_table);
+ if (sn->xs_tunables_header == NULL)
+ return -ENOMEM;
+ return 0;
+
+}
+void destroy_xs_tunables_table(struct net *net)
+{
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
+ unregister_sysctl_table(sn->xs_tunables_header);
+ sn->xs_tunables_header = NULL;
+}
#endif
+
+
+int socket_sysctl_init(struct net *net)
+{
+ #ifdef RPC_DEBUG
+ return create_xs_tunables_table(net);
+ #else
+ return 0;
+ #endif
+}
+
+void socket_sysctl_exit(struct net *net)
+{
+ #ifdef RPC_DEBUG
+ destroy_xs_tunables_table(net);
+ #endif
+}
+
+/**
+ * init_socket_xprt - set up xprtsock's sysctls, register with RPC client
+ *
+ */
+int init_socket_xprt(void)
+{
+ #ifdef RPC_DEBUG
+ if (!sunrpc_table_header)

```

```

- sunrpc_table_header = register_sunrpc_sysctl(&init_net, xs_tunables_table);
-#endif
-
  xprt_register_transport(&xs_local_transport);
  xprt_register_transport(&xs_udp_transport);
  xprt_register_transport(&xs_tcp_transport);
@@ -2895,13 +2926,6 @@ int init_socket_xprt(void)
  */
  void cleanup_socket_xprt(void)
  {
-#ifdef RPC_DEBUG
- if (sunrpc_table_header) {
-  unregister_sysctl_table(sunrpc_table_header);
-  sunrpc_table_header = NULL;
- }
-#endif
-
  xprt_unregister_transport(&xs_local_transport);
  xprt_unregister_transport(&xs_udp_transport);
  xprt_unregister_transport(&xs_tcp_transport);

```
