
Subject: [PATCH 04/11] SUNRPC: per-net sysctl's set introduced
Posted by [Stanislav Kinsbursky](#) on Wed, 14 Dec 2011 10:47:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch introduces per-net sysctl set and it's initialization routines. Also it modifies sysctl's registering helpers to make them use these sets.

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

```
---
include/linux/sunrpc/debug.h | 7 ++++++-
net/sunrpc/netns.h           | 3 +++
net/sunrpc/sunrpc_syms.c     | 13 ++++++++--
net/sunrpc/sysctl.c          | 38 ++++++++-----
net/sunrpc/xprtrdma/transport.c | 2 +-
net/sunrpc/xprtsock.c        | 2 +-
6 files changed, 56 insertions(+), 9 deletions(-)
```

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

```
index b5e0c46..f2d825c 100644
```

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

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

```
@@ -67,7 +67,12 @@ extern unsigned int nlm_debug;
```

```
void rpc_register_sysctl(void);
```

```
void rpc_unregister_sysctl(void);
```

```
-struct ctl_table_header *register_sunrpc_sysctl(struct ctl_table *);
```

```
+struct net;
```

```
+struct ctl_table;
```

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

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

```
+struct ctl_table_header *register_sunrpc_sysctl(struct net *,
```

```
+ struct ctl_table *);
```

```
#endif
```

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

```
index 6010c46..a949ddb 100644
```

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

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

```
@@ -12,6 +12,9 @@ struct sunrpc_net {
```

```
    struct list_head all_clients;
```

```
    spinlock_t rpc_client_lock;
```

```
+#ifdef RPC_DEBUG
```

```
+ struct ctl_table_set sysctls;
```

```
+#endif
```

```
};
```

```

extern int sunrpc_net_id;
diff --git a/net/sunrpc/sunrpc_syms.c b/net/sunrpc/sunrpc_syms.c
index e57aa10..44759e9 100644
--- a/net/sunrpc/sunrpc_syms.c
+++ b/net/sunrpc/sunrpc_syms.c
@@ -38,11 +38,19 @@ static __net_init int sunrpc_init_net(struct net *net)
    err = ip_map_cache_create(net);
    if (err)
        goto err_ipmap;
-
+
+#ifdef RPC_DEBUG
+ err = debug_sysctl_init(net);
+ if (err)
+ goto err_sysctl;
+#endif
    INIT_LIST_HEAD(&sn->all_clients);
    spin_lock_init(&sn->rpc_client_lock);
    return 0;

+#ifdef RPC_DEBUG
+err_sysctl:
+ ip_map_cache_destroy(net);
+#endif
err_ipmap:
    rpc_proc_exit(net);
err_proc:
@@ -51,6 +59,9 @@ err_proc:

static __net_exit void sunrpc_exit_net(struct net *net)
{
+#ifdef RPC_DEBUG
+ debug_sysctl_exit(net);
+#endif
    ip_map_cache_destroy(net);
    rpc_proc_exit(net);
}
diff --git a/net/sunrpc/sysctl.c b/net/sunrpc/sysctl.c
index f6e7da2..adebf0a 100644
--- a/net/sunrpc/sysctl.c
+++ b/net/sunrpc/sysctl.c
@@ -40,7 +40,7 @@ EXPORT_SYMBOL_GPL(nlm_debug);
#ifdef RPC_DEBUG

#include <linux/nsproxy.h>
-#include <net/net_namespace.h>
+#include "netns.h"

```

```

static struct ctl_table_header *sunrpc_table_header;
static ctl_table debug_table[];
@@ -50,23 +50,51 @@ struct ctl_path sunrpc_path[] = {
    { },
};

+static struct ctl_table_set *
+sunrpc_debug_lookup(struct ctl_table_root *root, struct nsproxy *namespaces)
+{
+ struct sunrpc_net *sn = net_generic(namespaces->net_ns, sunrpc_net_id);
+
+ return &sn->sysctls;
+}
+
static struct ctl_table_root sunrpc_debug_root = {
+ .lookup = sunrpc_debug_lookup,
};

-struct ctl_table_header *register_sunrpc_sysctl(struct ctl_table *table)
+struct ctl_table_header *register_sunrpc_sysctl(struct net *net,
+ struct ctl_table *table)
{
- return __register_sysctl_paths(&sunrpc_debug_root, current->nsproxy,
+ struct nsproxy namespaces;
+
+ namespaces = *current->nsproxy;
+ namespaces.net_ns = net;
+ return __register_sysctl_paths(&sunrpc_debug_root, &namespaces,
+ sunrpc_path, table);
}
EXPORT_SYMBOL_GPL(register_sunrpc_sysctl);

+int debug_sysctl_init(struct net *net)
+{
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
+
+ setup_sysctl_set(&sn->sysctls, NULL, NULL);
+ return 0;
+}
+
+void debug_sysctl_exit(struct net *net)
+{
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
+
+ WARN_ON(!list_empty(&sn->sysctls.list));
+}
+
void

```

```

rpc_register_sysctl(void)
{
    if (!sunrpc_table_header) {
- setup_sysctl_set(&sunrpc_debug_root.default_set, NULL, NULL);
    register_sysctl_root(&sunrpc_debug_root);
- sunrpc_table_header = register_sunrpc_sysctl(debug_table);
+ sunrpc_table_header = register_sunrpc_sysctl(&init_net, debug_table);
    }
}

```

diff --git a/net/sunrpc/xprtrdma/transport.c b/net/sunrpc/xprtrdma/transport.c

index 53a8622..4a63b35 100644

--- a/net/sunrpc/xprtrdma/transport.c

+++ b/net/sunrpc/xprtrdma/transport.c

@@ -762,7 +762,7 @@ static int __init xprt_rdma_init(void)

```

#ifdef RPC_DEBUG
    if (!sunrpc_table_header)
- sunrpc_table_header = register_sunrpc_sysctl(xr_tunables_table);
+ sunrpc_table_header = register_sunrpc_sysctl(&init_net, xr_tunables_table);
#endif
    return 0;
}

```

diff --git a/net/sunrpc/xprtsock.c b/net/sunrpc/xprtsock.c

index 6f1be96..1acc9a7 100644

--- a/net/sunrpc/xprtsock.c

+++ b/net/sunrpc/xprtsock.c

@@ -2878,7 +2878,7 @@ int init_socket_xprt(void)

```

{
#ifdef RPC_DEBUG
    if (!sunrpc_table_header)
- sunrpc_table_header = register_sunrpc_sysctl(xs_tunables_table);
+ sunrpc_table_header = register_sunrpc_sysctl(&init_net, xs_tunables_table);
#endif

```

```

    xprt_register_transport(&xs_local_transport);

```