Subject: [PATCH 11/11] SUNRPC: sysctl table for rpc_debug introduced
Posted by Stanislav Kinsbursky on Wed, 14 Dec 2011 10:47:15 GMT
View Forum Message <> Reply to Message

This patch provides showing of pending RPC tasks for right namespace in case of
write oparation to "rpc_debug" sysctl.

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

---
 include/linux/sunrpc/sched.h | 1
 net/sunrpc/netns.h | 1
 net/sunrpc/sysctl.c | 87 ++-----
 3 files changed, 77 insertions(+), 12 deletions(-)

diff --git a/include/linux/sunrpc/sched.h b/include/linux/sunrpc/sched.h
index b16243a..02f5fce 100644
--- a/include/linux/sunrpc/sched.h
+++ b/include/linux/sunrpc/sched.h
@@ -244,7 +244,6 @@ int rpciod_up(void);
 void rpciod_down(void);
 int __rpc_wait_for_completion_task(struct rpc_task *task, int (*)(void *));
 #ifdef RPC_DEBUG
-struct net;
 void rpc_show_tasks(struct net *);
 #endif
 int rpc_init_mempool(void);
diff --git a/net/sunrpc/netns.h b/net/sunrpc/netns.h
index 23d110d..0926705 100644
--- a/net/sunrpc/netns.h
+++ b/net/sunrpc/netns.h
@@ -16,6 +16,7 @@ struct sunrpc_net {
  struct ctl_table_set sysctls;
  struct ctl_table_header *debug_ctl_header;
  struct ctl_table_header *xs_tunables_header;
+ struct ctl_table_header *rpc_debug_ctl_header;
 #endif
  unsigned int xprt_udp_slot_table_entries;
  unsigned int xprt_tcp_slot_table_entries;
diff --git a/net/sunrpc/sysctl.c b/net/sunrpc/sysctl.c
index 224b075..52f6fb5 100644
--- a/net/sunrpc/sysctl.c
+++ b/net/sunrpc/sysctl.c
@@ -44,6 +44,15 @@ EXPORT_SYMBOL_GPL(nlm_debug);

 static ctl_table debug_table[];

+static ctl_table rpc_debug_table[] = {

```
+ {
+  .procname = "rpc_debug",
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+ },
+ { }
+};
+
 struct ctl_path sunrpc_path[] = {
  { .procname = "sunrpc", },
  { },
@@ -73,6 +82,48 @@ struct ctl_table_header *register_sunrpc_sysctl(struct net *net,
 }
 EXPORT_SYMBOL_GPL(register_sunrpc_sysctl);

+static int proc_rpcdebug(ctl_table *table, int write,
+    void __user *buffer, size_t *lenp, loff_t *ppos);
+
+static int register_rpc_debug_table(struct net *net)
+{
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
+ struct ctl_table *table;
+
+ table = rpc_debug_table;
+ if (!net_eq(net, &init_net)) {
+  table = kmemdup(rpc_debug_table, sizeof(rpc_debug_table),
+    GFP_KERNEL);
+  if (table == NULL)
+   goto err_alloc;
+ }
+ table[0].data = net;
+ table[0].proc_handler = proc_rpcdebug;
+
+ sn->rpc_debug_ctl_header = register_sunrpc_sysctl(net, table);
+ if (sn->rpc_debug_ctl_header == NULL)
+  goto err_reg;
+ return 0;
+
+err_reg:
+ if (!net_eq(net, &init_net))
+  kfree(table);
+err_alloc:
+ return -ENOMEM;
+}
+
+static void unregister_rpc_debug_table(struct net *net)
+{
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
```

```
+ struct ctl_table *table;
+
+ table = sn->rpc_debug_ctl_header->ctl_table_arg;
+ unregister_sysctl_table(sn->rpc_debug_ctl_header);
+ sn->rpc_debug_ctl_header = NULL;
+ if (!net_eq(net, &init_net))
+  kfree(table);
+}
+
 int debug_sysctl_init(struct net *net)
 {
  struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
@@ -80,14 +131,23 @@ int debug_sysctl_init(struct net *net)
  setup_sysctl_set(&sn->sysctls, NULL, NULL);
  sn->debug_ctl_header = register_sunrpc_sysctl(net, debug_table);
  if (sn->debug_ctl_header == NULL)
-  return -ENOMEM;
+  goto err_debug;
+ if (register_rpc_debug_table(net) < 0)
+  goto err_rpc_debug;
  return 0;
+
+err_rpc_debug:
+ unregister_sysctl_table(sn->debug_ctl_header);
+ sn->debug_ctl_header = NULL;
+err_debug:
+ return -ENOMEM;
 }

 void debug_sysctl_exit(struct net *net)
 {
  struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);

+ unregister_rpc_debug_table(net);
  unregister_sysctl_table(sn->debug_ctl_header);
  sn->debug_ctl_header = NULL;
  WARN_ON(!list_empty(&sn->sysctls.list));
@@ -158,9 +218,6 @@ proc_dodebug(ctl_table *table, int write,
    left--, s++;
   if (net_eq(current->nsproxy->net_ns, &init_net))
    *(unsigned int *) table->data = value;
-  /* Display the RPC tasks on writing to rpc_debug */
-  if (strcmp(table->procname, "rpc_debug") == 0)
-   rpc_show_tasks(&init_net);
  } else {
   if (!access_ok(VERIFY_WRITE, buffer, left))
    return -EFAULT;
@@ -183,15 +240,23 @@ done:
```

```
 }


+static int
+proc_rpcdebug(ctl_table *table, int write,
+    void __user *buffer, size_t *lenp, loff_t *ppos)
+{
+ int err;
+ struct ctl_table tmp = *table;
+
+ tmp.data = &rpc_debug;
+ err = proc_dodebug(&tmp, write, buffer, lenp, ppos);
+ if (!err && write)
+  /* Display the RPC tasks on writing to rpc_debug */
+  rpc_show_tasks(table->data);
+ return err;
+}
+
 static ctl_table debug_table[] = {
  {
-  .procname = "rpc_debug",
-  .data  = &rpc_debug,
-  .maxlen  = sizeof(int),
-  .mode  = 0644,
-  .proc_handler = proc_dodebug
- },
- {
   .procname = "nfs_debug",
   .data  = &nfs_debug,
   .maxlen  = sizeof(int),
```