Subject: [PATCH v2] SUNRPC: make SUNPRC clients list per network namespace context
Posted by Stanislav Kinsbursky on Thu, 12 Jan 2012 18:07:51 GMT
View Forum Message <> Reply to Message

v2:
1) rebased on current repo ("devel" head of "trondmy") state. Have to be
inserted between patches
"SUNRPC: handle GSS AUTH pipes by network namespace aware routines"
and
"SUNRPC: subscribe RPC clients to pipefs notifications".
Currently "devel" head is not compilable without this patch.

This patch moves static SUNRPC clients list and it's lock to sunrpc_net
structure.
Currently this list is used only for debug purposes. But later it will be used
also for selecting clients by networks namespace on PipeFS mount/umount events.
Per-network namespace lists will make this faster and simplier.

Note: client list is taken from "init_net" network namespace context in
rpc_show_tasks(). This will be changed some day later with making SUNRPC
sysctl's per network namespace context.

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

---
 include/linux/sunrpc/sched.h | 3 ++-
 net/sunrpc/clnt.c | 26 ++++++++++++++++++----------
 net/sunrpc/netns.h | 3 +++
 net/sunrpc/sunrpc_syms.c | 3 +++
 net/sunrpc/sysctl.c | 4 +++-
 5 files changed, 26 insertions(+), 13 deletions(-)

diff --git a/include/linux/sunrpc/sched.h b/include/linux/sunrpc/sched.h
index e775689..b16243a 100644
--- a/include/linux/sunrpc/sched.h
+++ b/include/linux/sunrpc/sched.h
@@ -244,7 +244,8 @@ int rpciod_up(void);
 void rpciod_down(void);
 int __rpc_wait_for_completion_task(struct rpc_task *task, int (*)(void *));
 #ifdef RPC_DEBUG
-void rpc_show_tasks(void);
+struct net;
+void rpc_show_tasks(struct net *);
 #endif
 int rpc_init_mempool(void);
 void rpc_destroy_mempool(void);
diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c

```
index 5ef192c..90e82c5 100644
--- a/net/sunrpc/clnt.c
+++ b/net/sunrpc/clnt.c
@@ -38,6 +38,7 @@
 #include <linux/sunrpc/bc_xprt.h>

 #include "sunrpc.h"
+#include "netns.h"

 #ifdef RPC_DEBUG
 # define RPCDBG_FACILITY RPCDBG_CALL
@@ -50,8 +51,6 @@
 /*
  * All RPC clients are linked into this list
  */
-static LIST_HEAD(all_clients);
-static DEFINE_SPINLOCK(rpc_client_lock);

 static DECLARE_WAIT_QUEUE_HEAD(destroy_wait);

@@ -81,16 +80,20 @@ static int rpc_ping(struct rpc_clnt *clnt);

 static void rpc_register_client(struct rpc_clnt *clnt)
 {
- spin_lock(&rpc_client_lock);
- list_add(&clnt->cl_clients, &all_clients);
- spin_unlock(&rpc_client_lock);
+ struct sunrpc_net *sn = net_generic(clnt->cl_xprt->xprt_net, sunrpc_net_id);
+
+ spin_lock(&sn->rpc_client_lock);
+ list_add(&clnt->cl_clients, &sn->all_clients);
+ spin_unlock(&sn->rpc_client_lock);
 }

 static void rpc_unregister_client(struct rpc_clnt *clnt)
 {
- spin_lock(&rpc_client_lock);
+ struct sunrpc_net *sn = net_generic(clnt->cl_xprt->xprt_net, sunrpc_net_id);
+
+ spin_lock(&sn->rpc_client_lock);
  list_del(&clnt->cl_clients);
- spin_unlock(&rpc_client_lock);
+ spin_unlock(&sn->rpc_client_lock);
 }

 static void __rpc_clnt_remove_pipedir(struct rpc_clnt *clnt)
@@ -1883,14 +1886,15 @@ static void rpc_show_task(const struct rpc_clnt *clnt,
   task->tk_action, rpc_waitq);
```

```
 }

-void rpc_show_tasks(void)
+void rpc_show_tasks(struct net *net)
 {
  struct rpc_clnt *clnt;
  struct rpc_task *task;
  int header = 0;
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);

- spin_lock(&rpc_client_lock);
- list_for_each_entry(clnt, &all_clients, cl_clients) {
+ spin_lock(&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) {
     if (!header) {
@@ -1901,6 +1905,6 @@ void rpc_show_tasks(void)
    }
    spin_unlock(&clnt->cl_lock);
  }
- spin_unlock(&rpc_client_lock);
+ spin_unlock(&sn->rpc_client_lock);
 }
 #endif
diff --git a/net/sunrpc/netns.h b/net/sunrpc/netns.h
index 11d2f48..0f3af34 100644
--- a/net/sunrpc/netns.h
+++ b/net/sunrpc/netns.h
@@ -12,6 +12,9 @@ struct sunrpc_net {

  struct super_block *pipefs_sb;
  struct mutex pipefs_sb_lock;
+
+ struct list_head all_clients;
+ spinlock_t rpc_client_lock;
 };

 extern int sunrpc_net_id;
diff --git a/net/sunrpc/sunrpc_syms.c b/net/sunrpc/sunrpc_syms.c
index 7086d11..b4217dc 100644
--- a/net/sunrpc/sunrpc_syms.c
+++ b/net/sunrpc/sunrpc_syms.c
@@ -29,6 +29,7 @@ int sunrpc_net_id;
 static __net_init int sunrpc_init_net(struct net *net)
 {
  int err;
+ struct sunrpc_net *sn = net_generic(net, sunrpc_net_id);
```

```
  err = rpc_proc_init(net);
  if (err)
@@ -39,6 +40,8 @@ static __net_init int sunrpc_init_net(struct net *net)
   goto err_ipmap;

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

 err_ipmap:
diff --git a/net/sunrpc/sysctl.c b/net/sunrpc/sysctl.c
index e65dcc6..af7d339 100644
--- a/net/sunrpc/sysctl.c
+++ b/net/sunrpc/sysctl.c
@@ -20,6 +20,8 @@
 #include <linux/sunrpc/stats.h>
 #include <linux/sunrpc/svc_xprt.h>

+#include "netns.h"
+
 /*
  * Declare the debug flags here
  */
@@ -110,7 +112,7 @@ proc_dodebug(ctl_table *table, int write,
   *(unsigned int *) table->data = value;
   /* Display the RPC tasks on writing to rpc_debug */
   if (strcmp(table->procname, "rpc_debug") == 0)
-   rpc_show_tasks();
+   rpc_show_tasks(&init_net);
  } else {
   if (!access_ok(VERIFY_WRITE, buffer, left))
    return -EFAULT;
```