Subject: [PATCH net-2.6.26 3/6][SOCK][NETNS]: Add the percpu prot_inuse counter in the struct net.
Posted by Pavel Emelianov on Mon, 31 Mar 2008 12:15:47 GMT
View Forum Message <> Reply to Message

Such an accounting would cost us two more dereferences to get the
percpu variable from the struct net, so I make sock_prot_inuse_get
and _add calls work differently depending on CONFIG_NET_NS - without
it old optimized routines are used.

The per-cpu counter for init_net is prepared in core_initcall, so
that even af_inet, that starts as fs_initcall, will already have the
init_net prepared.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---
 include/net/netns/core.h | 3 ++
 net/core/sock.c | 48 ++
 2 files changed, 51 insertions(+), 0 deletions(-)

diff --git a/include/net/netns/core.h b/include/net/netns/core.h
index 0e8c0f8..24d4be7 100644
--- a/include/net/netns/core.h
+++ b/include/net/netns/core.h
@@ -2,12 +2,15 @@
 #define __NETNS_CORE_H__

 struct ctl_table_header;
+struct prot_inuse;

 struct netns_core {
  /* core sysctls */
  struct ctl_table_header *sysctl_hdr;

  int sysctl_somaxconn;
+
+ struct prot_inuse *inuse;
 };

 #endif
diff --git a/net/core/sock.c b/net/core/sock.c
index 6f36ab9..83e11f7 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -1947,6 +1947,53 @@ struct prot_inuse {
 };

```
 static DECLARE_BITMAP(proto_inuse_idx, PROTO_INUSE_NR);
+
+#ifdef CONFIG_NET_NS
+void sock_prot_inuse_add(struct net *net, struct proto *prot, int val)
+{
+ int cpu = smp_processor_id();
+ per_cpu_ptr(net->core.inuse, cpu)->val[prot->inuse_idx] += val;
+}
+EXPORT_SYMBOL_GPL(sock_prot_inuse_add);
+
+int sock_prot_inuse_get(struct net *net, struct proto *prot)
+{
+ int cpu, idx = prot->inuse_idx;
+ int res = 0;
+
+ for_each_possible_cpu(cpu)
+  res += per_cpu_ptr(net->core.inuse, cpu)->val[idx];
+
+ return res >= 0 ? res : 0;
+}
+EXPORT_SYMBOL_GPL(sock_prot_inuse_get);
+
+static int sock_inuse_init_net(struct net *net)
+{
+ net->core.inuse = alloc_percpu(struct prot_inuse);
+ return net->core.inuse ? 0 : -ENOMEM;
+}
+
+static void sock_inuse_exit_net(struct net *net)
+{
+ free_percpu(net->core.inuse);
+}
+
+static struct pernet_operations net_inuse_ops = {
+ .init = sock_inuse_init_net,
+ .exit = sock_inuse_exit_net,
+};
+
+static __init int net_inuse_init(void)
+{
+ if (register_pernet_subsys(&net_inuse_ops))
+  panic("Cannot initialize net inuse counters");
+
+ return 0;
+}
+
+core_initcall(net_inuse_init);
+#else
```

```
 static DEFINE_PER_CPU(struct prot_inuse, prot_inuse);

 void sock_prot_inuse_add(struct net *net, struct proto *prot, int val)
@@ -1966,6 +2013,7 @@ int sock_prot_inuse_get(struct net *net, struct proto *prot)
  return res >= 0 ? res : 0;
 }
 EXPORT_SYMBOL_GPL(sock_prot_inuse_get);
+#endif

 static void assign_proto_idx(struct proto *prot)
 {
--
1.5.3.4
```