Subject: [PATCH 1/4] net: Dynamically allocate the per cpu counters for the loopback device.
Posted by ebiederm on Wed, 26 Sep 2007 23:53:40 GMT
View Forum Message <> Reply to Message

This patch add support for dynamically allocating the statistics counters
for the loopback device and adds appropriate device methods for allocating
and freeing the loopback device.

This completes support for creating multiple instances of the loopback
device,  in preparation for creating per network namespace instances.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/net/loopback.c |   32 ++++++++++++++++++++++++++++----
 1 files changed, 28 insertions(+), 4 deletions(-)

```
diff --git a/drivers/net/loopback.c b/drivers/net/loopback.c
index 4b6f7b2..f3018bb 100644
--- a/drivers/net/loopback.c
+++ b/drivers/net/loopback.c
@@ -62,7 +62,6 @@ struct pcpu_lstats {
  unsigned long packets;
  unsigned long bytes;
 };
-static DEFINE_PER_CPU(struct pcpu_lstats, pcpu_lstats);

 #define LOOPBACK_OVERHEAD (128 + MAX_HEADER + 16 + 16)

@@ -134,7 +133,7 @@ static void emulate_large_send_offload(struct sk_buff *skb)
  */
 static int loopback_xmit(struct sk_buff *skb, struct net_device *dev)
 {
- struct pcpu_lstats *lb_stats;
+ struct pcpu_lstats *pcpu_lstats, *lb_stats;

  skb_orphan(skb);

@@ -155,7 +154,8 @@ static int loopback_xmit(struct sk_buff *skb, struct net_device *dev)
  dev->last_rx = jiffies;

  /* it's OK to use __get_cpu_var() because BHs are off */
- lb_stats = &__get_cpu_var(pcpu_lstats);
+ pcpu_lstats = netdev_priv(dev);
+ lb_stats = per_cpu_ptr(pcpu_lstats, smp_processor_id());
  lb_stats->bytes += skb->len;
  lb_stats->packets++;
```

```
@@ -166,15 +166,17 @@ static int loopback_xmit(struct sk_buff *skb, struct net_device *dev)

 static struct net_device_stats *get_stats(struct net_device *dev)
 {
+ const struct pcpu_lstats *pcpu_lstats;
  struct net_device_stats *stats = &dev->stats;
  unsigned long bytes = 0;
  unsigned long packets = 0;
  int i;

+ pcpu_lstats = netdev_priv(dev);
  for_each_possible_cpu(i) {
   const struct pcpu_lstats *lb_stats;

-  lb_stats = &per_cpu(pcpu_lstats, i);
+  lb_stats = per_cpu_ptr(pcpu_lstats, i);
   bytes   += lb_stats->bytes;
   packets += lb_stats->packets;
  }
@@ -198,6 +200,26 @@ static const struct ethtool_ops loopback_ethtool_ops = {
  .get_rx_csum  = always_on,
 };

+static int loopback_dev_init(struct net_device *dev)
+{
+ struct pcpu_lstats *lstats;
+
+ lstats = alloc_percpu(struct pcpu_lstats);
+ if (!lstats)
+  return -ENOMEM;
+
+ dev->priv = lstats;
+ return 0;
+}
+
+static void loopback_dev_free(struct net_device *dev)
+{
+ struct pcpu_lstats *lstats = netdev_priv(dev);
+
+ free_percpu(lstats);
+ free_netdev(dev);
+}
+
 /*
  * The loopback device is special. There is only one instance and
  * it is statically allocated. Don't do this for other devices.
@@ -225,6 +247,8 @@ static void loopback_setup(struct net_device *dev)
   | NETIF_F_LLTX
```

```
   | NETIF_F_NETNS_LOCAL,
  dev->ethtool_ops = &loopback_ethtool_ops;
+ dev->init = loopback_dev_init;
+ dev->destructor = loopback_dev_free;
 }

 /* Setup and register the loopback device. */
--
1.5.3.rc6.17.g1911
```

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers