Subject: [PATCH 2.6.25] net: move trie_local and trie_main into the proc iterator
Posted by den on Thu, 06 Dec 2007 14:59:25 GMT
View Forum Message <> Reply to Message

From: Eric W. Biederman <ebiederm@xmission.com>

We only use these variables when displaying the trie in proc so
place them into the iterator to make this explicit. We should
probably do something smarter to handle the CONFIG_IP_MULTIPLE_TABLES
case but at least this makes it clear that the silliness is limited
to the display in /proc.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
Signed-off-by: Denis V. Lunev <den@openvz.org>
---
net/ipv4/fib_trie.c | 47 ++---------
1 files changed, 34 insertions(+), 13 deletions(-)

```
diff --git a/net/ipv4/fib_trie.c b/net/ipv4/fib_trie.c
index 8d8c291..6385cca 100644
--- a/net/ipv4/fib_trie.c
+++ b/net/ipv4/fib_trie.c
@@ -164,7 +164,6 @@ static struct tnode *halve(struct trie *t, struct tnode *tn);
 static void tnode_free(struct tnode *tn);

 static struct kmem_cache *fn_alias_kmem __read_mostly;
-static struct trie *trie_local = NULL, *trie_main = NULL;

 static inline struct tnode *node_parent(struct node *node)
 {
@@ -2000,11 +1999,6 @@ struct fib_table * __init fib_hash_init(u32 id)
  trie_init(t);

  if (id == RT_TABLE_LOCAL)
-  trie_local = t;
- else if (id == RT_TABLE_MAIN)
-  trie_main = t;
-
- if (id == RT_TABLE_LOCAL)
   printk(KERN_INFO "IPv4 FIB: Using LC-trie version %s\n", VERSION);

  return tb;
@@ -2013,6 +2007,7 @@ struct fib_table * __init fib_hash_init(u32 id)
 #ifdef CONFIG_PROC_FS
 /* Depth first Trie walk iterator */
 struct fib_trie_iter {
+ struct trie *trie_local, *trie_main;
  struct tnode *tnode;
```

```
  struct trie *trie;
  unsigned index;
@@ -2179,7 +2174,20 @@ static void trie_show_stats(struct seq_file *seq, struct trie_stat *stat)

 static int fib_triestat_seq_show(struct seq_file *seq, void *v)
 {
+ struct trie *trie_local, *trie_main;
  struct trie_stat *stat;
+ struct fib_table *tb;
+
+ trie_local = NULL;
+ tb = fib_get_table(RT_TABLE_LOCAL);
+ if (tb)
+  trie_local = (struct trie *) tb->tb_data;
+
+ trie_main = NULL;
+ tb = fib_get_table(RT_TABLE_MAIN);
+ if (tb)
+  trie_main = (struct trie *) tb->tb_data;
+

  stat = kmalloc(sizeof(*stat), GFP_KERNEL);
  if (!stat)
@@ -2223,13 +2231,13 @@ static struct node *fib_trie_get_idx(struct fib_trie_iter *iter,
  loff_t idx = 0;
  struct node *n;

- for (n = fib_trie_get_first(iter, trie_local);
+ for (n = fib_trie_get_first(iter, iter->trie_local);
       n; ++idx, n = fib_trie_get_next(iter)) {
   if (pos == idx)
    return n;
  }

- for (n = fib_trie_get_first(iter, trie_main);
+ for (n = fib_trie_get_first(iter, iter->trie_main);
       n; ++idx, n = fib_trie_get_next(iter)) {
   if (pos == idx)
    return n;
@@ -2239,10 +2247,23 @@ static struct node *fib_trie_get_idx(struct fib_trie_iter *iter,

 static void *fib_trie_seq_start(struct seq_file *seq, loff_t *pos)
 {
+ struct fib_trie_iter *iter = seq->private;
+ struct fib_table *tb;
+
+ if (!iter->trie_local) {
+  tb = fib_get_table(RT_TABLE_LOCAL);
```

```
+  if (tb)
+   iter->trie_local = (struct trie *) tb->tb_data;
+ }
+ if (!iter->trie_main) {
+  tb = fib_get_table(RT_TABLE_MAIN);
+  if (tb)
+   iter->trie_main = (struct trie *) tb->tb_data;
+ }
  rcu_read_lock();
  if (*pos == 0)
   return SEQ_START_TOKEN;
- return fib_trie_get_idx(seq->private, *pos - 1);
+ return fib_trie_get_idx(iter, *pos - 1);
 }

 static void *fib_trie_seq_next(struct seq_file *seq, void *v, loff_t *pos)
@@ -2260,8 +2281,8 @@ static void *fib_trie_seq_next(struct seq_file *seq, void *v, loff_t *pos)
   return v;

  /* continue scan in next trie */
- if (iter->trie == trie_local)
-  return fib_trie_get_first(iter, trie_main);
+ if (iter->trie == iter->trie_local)
+  return fib_trie_get_first(iter, iter->trie_main);

  return NULL;
 }
@@ -2327,7 +2348,7 @@ static int fib_trie_seq_show(struct seq_file *seq, void *v)
   return 0;

  if (!node_parent(n)) {
-  if (iter->trie == trie_local)
+  if (iter->trie == iter->trie_local)
    seq_puts(seq, "<local>:\n");
   else
    seq_puts(seq, "<main>:\n");
@@ -2426,7 +2447,7 @@ static int fib_route_seq_show(struct seq_file *seq, void *v)
   return 0;
  }

- if (iter->trie == trie_local)
+ if (iter->trie == iter->trie_local)
   return 0;
  if (IS_TNODE(l))
   return 0;
--
1.5.3.rc5
```

Subject: Re: [PATCH 2.6.25] net: move trie_local and trie_main into the proc iterator
Posted by davem on Fri, 07 Dec 2007 08:48:11 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Thu, 6 Dec 2007 18:00:12 +0300

> From: Eric W. Biederman <ebiederm@xmission.com>
>
> We only use these variables when displaying the trie in proc so
> place them into the iterator to make this explicit. We should
> probably do something smarter to handle the CONFIG_IP_MULTIPLE_TABLES
> case but at least this makes it clear that the silliness is limited
> to the display in /proc.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> Signed-off-by: Denis V. Lunev <den@openvz.org>

Applied.