
Subject: [PATCH net-2.6.25 2/6][IPVS] Switch to using ctl_paths.
Posted by [Pavel Emelianov](#) on Tue, 08 Jan 2008 15:55:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

The feature of ipvs ctls is that the net/ipv4/vs path is common for core ipvs ctls and for two schedulers, so I make it exported and re-use it in modules.

Two other .c files required linux/sysctl.h to make the extern declaration of this path compile well.

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

```
include/net/ip_vs.h      | 1 +
net/ipv4/ipvs/ip_vs_ctl.c | 35 ++++++-----
net/ipv4/ipvs/ip_vs_est.c | 1 +
net/ipv4/ipvs/ip_vs_lblc.c | 31 +-----
net/ipv4/ipvs/ip_vs_lblcr.c | 31 +-----
net/ipv4/ipvs/ip_vs_sched.c | 1 +
6 files changed, 12 insertions(+), 88 deletions(-)
```

```
diff --git a/include/net/ip_vs.h b/include/net/ip_vs.h
index 3de6d1e..02ab7ca 100644
```

```
--- a/include/net/ip_vs.h
```

```
+++ b/include/net/ip_vs.h
```

```
@@ -854,6 +854,7 @@ extern int sysctl_ip_vs_expire_quiescent_template;
extern int sysctl_ip_vs_sync_threshold[2];
extern int sysctl_ip_vs_nat_icmp_send;
extern struct ip_vs_stats ip_vs_stats;
+extern struct ctl_path net_vs_ctl_path[];
```

```
extern struct ip_vs_service *
```

```
ip_vs_service_get(__u32 fwmark, __u16 protocol, __be32 vaddr, __be16 vport);
```

```
diff --git a/net/ipv4/ipvs/ip_vs_ctl.c b/net/ipv4/ipvs/ip_vs_ctl.c
```

```
index 693d924..9fecfe7 100644
```

```
--- a/net/ipv4/ipvs/ip_vs_ctl.c
```

```
+++ b/net/ipv4/ipvs/ip_vs_ctl.c
```

```
@@ -1591,34 +1591,13 @@ static struct ctl_table vs_vars[] = {
    { .ctl_name = 0 }
};
```

```
-static ctl_table vs_table[] = {
```

```
- {
```

```
- .procname = "vs",
```

```
- .mode = 0555,
```

```
- .child = vs_vars
```

```

- },
- { .ctl_name = 0 }
-};
-
-static ctl_table ipvs_ipv4_table[] = {
- {
- .ctl_name = NET_IPV4,
- .procname = "ipv4",
- .mode = 0555,
- .child = vs_table,
- },
- { .ctl_name = 0 }
-};
-
-static ctl_table vs_root_table[] = {
- {
- .ctl_name = CTL_NET,
- .procname = "net",
- .mode = 0555,
- .child = ipvs_ipv4_table,
- },
- { .ctl_name = 0 }
+struct ctl_path net_vs_ctl_path[] = {
+ { .procname = "net", .ctl_name = CTL_NET, },
+ { .procname = "ipv4", .ctl_name = NET_IPV4, },
+ { .procname = "vs", },
+ { }
};
+EXPORT_SYMBOL_GPL(net_vs_ctl_path);

static struct ctl_table_header * sysctl_header;

@@ -2345,7 +2324,7 @@ int ip_vs_control_init(void)
    proc_net_fops_create(&init_net, "ip_vs", 0, &ip_vs_info_fops);
    proc_net_fops_create(&init_net, "ip_vs_stats", 0, &ip_vs_stats_fops);

- sysctl_header = register_sysctl_table(vs_root_table);
+ sysctl_header = register_sysctl_paths(net_vs_ctl_path, vs_vars);

/* Initialize ip_vs_svc_table, ip_vs_svc_fwm_table, ip_vs_rtable */
for(idx = 0; idx < IP_VS_SVC_TAB_SIZE; idx++) {
diff --git a/net/ipv4/ipvs/ip_vs_est.c b/net/ipv4/ipvs/ip_vs_est.c
index efdd74e..dfa0d71 100644
--- a/net/ipv4/ipvs/ip_vs_est.c
+++ b/net/ipv4/ipvs/ip_vs_est.c
@@ -18,6 +18,7 @@
#include <linux/slab.h>
#include <linux/types.h>

```

```

#include <linux/interrupt.h>
#include <linux/sysctl.h>

#include <net/ip_vs.h>

diff --git a/net/ipv4/ipvs/ip_vs_lblc.c b/net/ipv4/ipvs/ip_vs_lblc.c
index bf8c04a..3888642 100644
--- a/net/ipv4/ipvs/ip_vs_lblc.c
+++ b/net/ipv4/ipvs/ip_vs_lblc.c
@@ -123,35 +123,6 @@ static ctl_table vs_vars_table[] = {
     { .ctl_name = 0 }
 };

-static ctl_table vs_table[] = {
- {
-     .procname = "vs",
-     .mode = 0555,
-     .child = vs_vars_table
- },
- { .ctl_name = 0 }
-};
-
-static ctl_table ipvs_ipv4_table[] = {
- {
-     .ctl_name = NET_IPV4,
-     .procname = "ipv4",
-     .mode = 0555,
-     .child = vs_table
- },
- { .ctl_name = 0 }
-};
-
-static ctl_table lblc_root_table[] = {
- {
-     .ctl_name = CTL_NET,
-     .procname = "net",
-     .mode = 0555,
-     .child = ipvs_ipv4_table
- },
- { .ctl_name = 0 }
-};
-
static struct ctl_table_header * sysctl_header;

/*
@@ -582,7 +553,7 @@ static int __init ip_vs_lblc_init(void)
int ret;

```

```

INIT_LIST_HEAD(&ip_vs_lblcr_scheduler.n_list);
- sysctl_header = register_sysctl_table(lblcr_root_table);
+ sysctl_header = register_sysctl_paths(net_vs_ctl_path, vs_vars_table);
ret = register_ip_vs_scheduler(&ip_vs_lblcr_scheduler);
if (ret)
    unregister_sysctl_table(sysctl_header);
diff --git a/net/ipv4/ipvs/ip_vs_lblcr.c b/net/ipv4/ipvs/ip_vs_lblcr.c
index f50da64..daa260e 100644
--- a/net/ipv4/ipvs/ip_vs_lblcr.c
+++ b/net/ipv4/ipvs/ip_vs_lblcr.c
@@ -311,35 +311,6 @@ static ctl_table vs_vars_table[] = {
    { .ctl_name = 0 }
};

-static ctl_table vs_table[] = {
- {
-  .procname = "vs",
-  .mode = 0555,
-  .child = vs_vars_table
- },
- { .ctl_name = 0 }
-};
-
-static ctl_table ipvs_ipv4_table[] = {
- {
-  .ctl_name = NET_IPV4,
-  .procname = "ipv4",
-  .mode = 0555,
-  .child = vs_table
- },
- { .ctl_name = 0 }
-};
-
-static ctl_table lblcr_root_table[] = {
- {
-  .ctl_name = CTL_NET,
-  .procname = "net",
-  .mode = 0555,
-  .child = ipvs_ipv4_table
- },
- { .ctl_name = 0 }
-};
-
static struct ctl_table_header * sysctl_header;

/*
@@ -771,7 +742,7 @@ static int __init ip_vs_lblcr_init(void)
int ret;

```

```
INIT_LIST_HEAD(&ip_vs_lbcr_scheduler.n_list);
- sysctl_header = register_sysctl_table(lbcr_root_table);
+ sysctl_header = register_sysctl_paths(net_vs_ctl_path, vs_vars_table);
ret = register_ip_vs_scheduler(&ip_vs_lbcr_scheduler);
if (ret)
    unregister_sysctl_table(sysctl_header);
diff --git a/net/ipv4/ipvs/ip_vs_sched.c b/net/ipv4/ipvs/ip_vs_sched.c
index 4322358..121a32b 100644
--- a/net/ipv4/ipvs/ip_vs_sched.c
+++ b/net/ipv4/ipvs/ip_vs_sched.c
@@ -24,6 +24,7 @@
#include <linux/interrupt.h>
#include <asm/string.h>
#include <linux/kmod.h>
+#include <linux/sysctl.h>

#include <net/ip_vs.h>
```
