Subject: [PATCH 21/28] [FLAT 6/6] Moving the pid into the namespace
Posted by Pavel Emelianov on Fri, 15 Jun 2007 16:21:31 GMT
View Forum Message <> Reply to Message

This is the implementation of move_pid_to_ns (see [PATCH 15/28])

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

```
---

 pid.c |   58 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 files changed, 58 insertions(+)

--- ./kernel/pid.c.flatclone 2007-06-15 15:26:23.000000000 +0400
+++ ./kernel/pid.c 2007-06-15 15:28:22.000000000 +0400
@@ -331,6 +331,48 @@ struct pid * fastcall find_pid_ns(int nr
  return (ns == &init_pid_ns ?
    find_global_pid(nr) : find_virtual_pid(nr, ns));
 }
+
+static inline int move_pid_to_ns(struct pid *pid, struct pid_namespace *ns)
+{
+ int vnr;
+
+ /*
+  * the pid is in this ns already. e.g. this may happen if
+  * the task has equal pid and pgid
+  */
+ if (pid->ns == ns)
+  return 0;
+
+ BUG_ON(pid->ns != &init_pid_ns);
+
+ vnr = alloc_pidmap(ns);
+ if (vnr < 0)
+  return -ENOMEM;
+
+ get_pid_ns(ns);
+ pid->vnr = vnr;
+ pid->ns = ns;
+ spin_lock_irq(&pidmap_lock);
+ hlist_add_head_rcu(&pid->vpid_chain,
+   &vpid_hash[vpid_hashfn(vnr, ns)]);
+ spin_unlock_irq(&pidmap_lock);
+ return 0;
+}
+
+static inline void del_pid_from_ns(struct pid *pid, struct pid_namespace *ns)
```

```
+{
+ if (pid->ns == &init_pid_ns)
+  return;
+
+ spin_lock_irq(&pidmap_lock);
+ hlist_del_rcu(&pid->vpid_chain);
+ spin_unlock_irq(&pidmap_lock);
+
+ free_pidmap(ns, pid->vnr);
+ put_pid_ns(pid->ns);
+ pid->vnr = pid->nr;
+ pid->ns = &init_pid_ns;
+}
 #endif
 #endif

@@ -588,6 +630,15 @@ struct pid_namespace *copy_pid_ns(int fl
  if (!(flags & CLONE_NEWPIDS))
   goto out;

+#ifdef CONFIG_PID_NS_FLAT
+ /*
+  * flat model doen't allow to create the nested namespaces
+  */
+ new_ns = ERR_PTR(-EINVAL);
+ if (old_ns != &init_pid_ns)
+  goto out_put;
+#endif
+
  new_ns = ERR_PTR(-EBUSY);
  if (!alone_in_pgrp(current))
   goto out_put;
@@ -646,6 +697,13 @@ out_pgid:
 out_sid:
  del_pid_from_ns(task_pid(tsk), ns);
 out_pid:
+#ifdef CONFIG_PID_NS_FLAT
+ /*
+  * this cannot happen if we use flat pid namespaces as we try to
+  * allocate the very first pid from the pidmap with one page in it
+  */
+ BUG();
+#endif
  return err;
 }
 #else
```

Containers mailing list

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