Subject: [RFC][PATCH 02/14] Move alloc_pid call to copy_process
Posted by Sukadev Bhattiprolu on Wed, 21 Mar 2007 03:18:40 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: [RFC][PATCH 02/14] Move alloc_pid call to copy_process

Move alloc_pid() into copy_process(). This will help in
code to support cloning of pid namespace.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
---
 kernel/fork.c | 31 +++++++++++++++++++++++++---------
 1 file changed, 22 insertions(+), 9 deletions(-)

Index: lx26-21-rc3-mm2/kernel/fork.c
==========
--- lx26-21-rc3-mm2.orig/kernel/fork.c 2007-03-20 15:55:52.000000000 -0700
+++ lx26-21-rc3-mm2/kernel/fork.c 2007-03-20 20:17:05.000000000 -0700
@@ -952,6 +952,11 @@ static inline void rcu_task_init(struct
 static inline void rcu_task_init(struct task_struct *p) {}
 #endif

+enum idle_process {
+ COPY_NON_IDLE_PROCESS,
+ COPY_IDLE_PROCESS,
+};
+
 /*
  * This creates a new process as a copy of the old one,
  * but does not actually start it yet.
@@ -966,10 +971,11 @@ static struct task_struct *copy_process(
      unsigned long stack_size,
      int __user *parent_tidptr,
      int __user *child_tidptr,
-     struct pid *pid)
+   enum idle_process type)
 {
  int retval;
  struct task_struct *p = NULL;
+ struct pid *pid;

  if ((clone_flags & (CLONE_NEWNS|CLONE_FS)) == (CLONE_NEWNS|CLONE_FS))
   return ERR_PTR(-EINVAL);
@@ -1030,6 +1036,13 @@ static struct task_struct *copy_process(
  if (p->binfmt && !try_module_get(p->binfmt->module))
   goto bad_fork_cleanup_put_domain;

```
+ if (unlikely(type == COPY_IDLE_PROCESS))
+  pid = &init_struct_pid;
+ else {
+  pid = alloc_pid();
+  if (!pid)
+   goto bad_fork_put_binfmt_module;
+ }
  p->did_exec = 0;
  delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
  copy_flags(clone_flags, p);
@@ -1306,6 +1319,9 @@ bad_fork_cleanup_cpuset:
 #endif
  cpuset_exit(p);
  delayacct_tsk_free(p);
+ if (pid != &init_struct_pid)
+  free_pid(pid);
+bad_fork_put_binfmt_module:
  if (p->binfmt)
   module_put(p->binfmt->module);
 bad_fork_cleanup_put_domain:
@@ -1332,7 +1348,7 @@ struct task_struct * __cpuinit fork_idle
  struct pt_regs regs;

  task = copy_process(CLONE_VM, 0, idle_regs(&regs), 0, NULL, NULL,
-    &init_struct_pid);
+    COPY_IDLE_PROCESS);
  if (!IS_ERR(task))
   init_idle(task, cpu);

@@ -1353,14 +1369,10 @@ long do_fork(unsigned long clone_flags,
        int __user *child_tidptr)
 {
  struct task_struct *p;
- struct pid *pid = alloc_pid();
  long nr;

- if (!pid)
-  return -EAGAIN;
- nr = pid->nr;
-
- p = copy_process(clone_flags, stack_start, regs, stack_size, parent_tidptr, child_tidptr, pid);
+ p = copy_process(clone_flags, stack_start, regs, stack_size,
+   parent_tidptr, child_tidptr, COPY_NON_IDLE_PROCESS);
  /*
   * Do this prior waking up the new thread - the thread pointer
   * might get invalid after that point, if the thread exits quickly.
@@ -1368,6 +1380,8 @@ long do_fork(unsigned long clone_flags,
  if (!IS_ERR(p)) {
```

```
  struct completion vfork;

+  nr = pid_nr(task_pid(p));
+
  if (clone_flags & CLONE_VFORK) {
   p->vfork_done = &vfork;
   init_completion(&vfork);
@@ -1397,7 +1411,6 @@ long do_fork(unsigned long clone_flags,
   tracehook_report_vfork_done(p, nr);
  }
 } else {
-  free_pid(pid);
  nr = PTR_ERR(p);
 }
 return nr;
```

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