Subject: Re: [Containers] Q: Do systems using containers user more process ids?
Posted by Dave Hansen on Mon, 14 Aug 2006 21:18:34 GMT
View Forum Message <> Reply to Message

On Mon, 2006-08-14 at 15:01 -0600, Eric W. Biederman wrote:
> The practical question is if systems using containers are using noticeably
> more pids than anyone else.   So far the responses I have gotten indicate
> that users aren't. So at least until we descend into multi-core madness
> it sounds like the current structures are fine, but it might be worth moving
> the cap on the number of pid hash table entries at some point in the future.

Since it is already resized at boot-time, I can't imagine this be a real
problem to fix. I assume you're just trying to see if anybody has run
into it as of yet.

Perhaps a one-time-per-boot warning in find_pid() if the chains get too
long would be nice to have. It wouldn't give us detailed performance
measurements, but it would be a nice canary in the mine in case
something goes horribly wrong.

What about something like this?

```
---

 lxc-dave/kernel/pid.c |    9 +++++++++
 1 files changed, 9 insertions(+)

diff -puN Makefile~warn-on-long-pidhash-chains Makefile
diff -puN kernel/pid.c~warn-on-long-pidhash-chains kernel/pid.c
--- lxc/kernel/pid.c~warn-on-long-pidhash-chains 2006-08-14 14:13:39.000000000 -0700
+++ lxc-dave/kernel/pid.c 2006-08-14 14:17:49.000000000 -0700
@@ -209,9 +209,18 @@ struct pid * fastcall find_pid(int nr)
 {
  struct hlist_node *elem;
  struct pid *pid;
+ int chain_length = 0;
+ static int chain_length_limit = 5;
+ static int issued_warning = 0;

  hlist_for_each_entry_rcu(pid, elem,
    &pid_hash[pid_hashfn(nr)], pid_chain) {
+  if (!issued_warning && (chain_length++ > chain_length_limit)) {
+   issued_warning = 1;
+   printk(KERN_WARN "%s() pid hash chain length "
+      "exceeded %d elements\n",
+      __FUNCTION__, chain_length);
+  }
   WARN_ON(!pid->nr); /* to be removed soon */
```

```
    if (pid->nr == nr)
     return pid;
```
–

-- Dave