Subject: Re: [PATCH 0/16] Pid namespaces
Posted by Sukadev Bhattiprolu on Mon, 09 Jul 2007 21:42:44 GMT
View Forum Message <> Reply to Message

Pavel Emelianov [xemul@openvz.org] wrote:

| This is "submition for inclusion" of hierarchical, not kconfig
| configurable, zero overheaded ;) pid namespaces.
|
| The overall idea is the following:
|
| The namespace are organized as a tree - once a task is cloned
| with CLONE_NEWPIDS (yes, I've also switched to it :) the new
| namespace becomes the parent's child and tasks living in the
| parent namespace see the tasks from the new one. The numerical
| ids are used on the kernel-user boundary, i.e. when we export
| pid to user we show the id, that should be used to address the
| task in question from the namespace we're exporting this id to.
|
| The main difference from Suka's patches are the following:
|
| 0. Suka's patches change the kernel/pid.c code too heavy.
| This set keeps the kernel code look like it was without
| the patches. However, this is a minor issue. The major is:
|
| 1. Suka's approach is to remove the notion of the task's
| numerical pid from the kernel at all. The numbers are
| used on the kernel-user boundary or within the kernel but
| with the namespace this nr belongs to. This results in
| massive changes of struct's members fro int pid to struct
| pid *pid, task->pid becomes the virtual id and so on and
| so forth.

Your basic design is similar to what our patchset has been for
a while, with a few changes.

My patchset does not remove the task->pid. It still uses it
with the caveat that with multiple namespaces it is not unique.
getpid() implementation does not changes for instance.

Basically our patchset has init_pid_ns as the last element in the
pid->numbers[] array while yours is having it as the first. How
big a difference it makes, I am not sure.

|
| My approach is to keep the good old logic in the kernel.
| The task->pid is a global and unique pid, find_pid() finds
| the pid by its global id and so on. The virtual ids appear

| on the user-kernel boundary only. Thus drivers and other
| kernel code may still be unaware of pids unless they do not
| communicate with the userspace and get/put numerical pids.

Even in my patchset, drivers or other kernel code have no need
to know anything about namespaces.

Actually you seem to introduce a new function find_vpid() that
is used in a driver. So a driver-writer needs to know whether
to call find_pid() or find_vpid().

|
| And some more minor differences:
|
| 2. Suka's patches have the limit of pid namespace nesting.
| My patches do not.

Yes - its a compile-time constant (MAX_NESTED_PID_NS) that I
introduced just in the last version to simplify allocation.
Ecspecially after you argued against arbitrary depth before :-)

The basic design of your new 'struct pid' data structure is very
similar to what we have had for the last couple of rounds and we
could just as easily remove MAX_NESTED_PID_NS.

|
| 3. Suka assumes that pid namespace can live without proc mount
| and tries to make the code work with pid_ns->proc_mnt change
| from NULL to not-NULL from times to times.
| My code calls the kern_mount() at the namespace creation and
| thus the pid_namespace always works with proc.

Yes, we have been debating about the better approach for this yet.
We have been considering doing the kern_mount, as we do in
init_pid_ns at present.

|
| There are some small issues that I can describe if someone is
| interested.
|
| The tests like nptl perf, unixbench spawn, getpid and others
| didn't reveal any performance degradation in init_namespace
| with the RHEL5 kernel .config file. I admit, that different
| .config-s may show that patches hurt the performance, but the
| intention was *not* to make the kernel work worse with popular
| distributions.
|
| This set has some ways to move forward, but this is some kind

| of a core, that do not change the init_pid_namespace behavior
| (checked with LTP tests) and may require some hacking to do
| with the namespaces only.
|
| Patches apply to 2.6.22-rc6-mm1.

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