Subject: Re: Re: [PATCH 22/28] [MULTI 1/6] Changes in data structures for multilevel model
Posted by Pavel Emelianov on Tue, 19 Jun 2007 07:49:05 GMT
View Forum Message <> Reply to Message

sukadev@us.ibm.com wrote:
> Pavel Emelianov [xemul@openvz.org] wrote:
> | This patch opens the multilevel model patches.
> |
> | The multilevel model idea is basically the same as for the flat one,
> | but in this case task may have many virtual pids - one id for each
> | sub-namespace this task is visible in. The struct pid carries the
> | list of pid_number-s and two hash tables are used to find this number
> | by numerical id and by struct pid.
> |
> |
> |
> | The struct pid doesn't need the numerical ids any longer. Instead it
> | has a single linked list of struct pid_number-s which are hashed
> | for quick search and have the numerical id.
> |
> | Signed-off-by: Pavel Emelianov <xemul@openvz.org>
> |
> | ---
> |
> | pid.h | 31 +++++++++++++++++++++++++++++++
> | 1 files changed, 31 insertions(+)
> |
> | --- ./include/linux/pid.h.multdatast 2007-06-15 15:23:00.000000000 +0400
> | +++ ./include/linux/pid.h 2007-06-15 15:32:15.000000000 +0400
> | @@ -50,9 +50,33 @@ enum pid_type
> | * id must be used.
> | */
> |
> | +/*
> | + * multilevel pid namespaces
> | + * each task may belong to any number of namespaces and thus struct pid do
> | + * not carry the number any longer. instead if this struct pid has a list of
> | + * pid_number-s each belonging to one namespace. two hashes are used to find
> | + * the number - by the numerical id and by the struct pid this nr belongs to.
> | + * this allows for creating namespaces of infinite nesting, but has slight
> | + * performance problems.
> | + */
> | +
> | +struct pid_number
> | +{
> | + int nr;
> | + struct pid_namespace *ns;

> | + struct pid *pid;
> | +
> | + struct hlist_node pid_chain;
> | + struct hlist_node nr_chain;
> | + struct pid_number *next;
>
> As you probably noticed, we had a similar linked list until recently.
> But since we use only clone() to create a new pid namespace, we figured
> we could use an array of 'struct pid_number' elements. That may perform
> slightly better since all 'pid_number elements' are co-located.

Yes, I know it. This ability is a good reason to clone the namespace
via clone()...

> We obviously need a list like this if we unshare (rather than clone())
> pid namespace.
>
> I have a few questions - not that I see any problems yet - just for my
> understanding (they may be addressed in other patches, but am still
> reviewing them).
>
> - Can one process unshare() its namespace, create a few children,
> and unshare its namespace again ?

Yes, it can.

> - If so, will that same process be the reaper for multiple pid
> namespaces ?

It will. The process that created the namespace will become its reaper.
But I think this is wrong... Reaper should be such for the only namespace.

> - Will we terminate all those namespaces if the reaper is terminated ?

As you will see I do not terminate the namespace on reaper's death.

But it looks like you have caught a BUG in my patches - when a task
is a reaper for multiple namespaces and when he exits the namespaces
will point to the exited task as a reaper :(I will fix it.

>
> | +};
> | +
> | struct pid
> | {
> | atomic_t count;
> | +#ifdef CONFIG_PID_NS_MULTILEVEL
> | + struct pid_number *pid_nrs;

```
> | +#else
> |   /* Try to keep pid_chain in the same cacheline as nr for find_pid */
> |   int nr;
> |   struct hlist_node pid_chain;
> | @@ -65,11 +89,18 @@ struct pid
> |   struct pid_namespace *ns;
> |   struct hlist_node vpid_chain;
> |  #endif
> | +#endif
> |   /* lists of tasks that use this pid */
> |   struct hlist_head tasks[PIDTYPE_MAX];
> |   struct rcu_head rcu;
> |  };
> |
> | +#ifdef CONFIG_PID_NS_MULTILEVEL
> | +/* small helper to iterate over the pid's numbers */
> | +#define for_each_pid_nr(nr, pid) \
> | +  for (nr = pid->pid_nrs; nr != NULL; nr = nr->next)
> | +#endif
> | +
> |  extern struct pid init_struct_pid;
> |
> |  struct pid_link
> _____________________________________________________
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers
>
_____________________________________________________
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers
```