
Subject: Re: [patch -mm 08/17] nsproxy: add hashtable
Posted by [Herbert Poetzl](#) on Tue, 05 Dec 2006 23:36:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, Dec 05, 2006 at 11:28:00AM +0100, clg@fr.ibm.com wrote:

> From: Cedric Le Goater <clg@fr.ibm.com>

>

> This patch adds a hashtable of nsproxy using the nsproxy as a key.

> init_nsproxy is hashed at init with key 0. This is considered to be

> the 'host' nsproxy.

hmm? how is that going to work?

I mean, a simple clone() call (on the host) will
change the nsproxy and spawn new ones, but that
doesn't mean that the process has left the 'host'
it just means that some namespace is not shared
with the 'other' processes ...

best,
Herbert

> Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

> ---

> include/linux/nsproxy.h | 3 +++

> kernel/nsproxy.c | 43 ++++++++++++++++++++++++++++++++++++++

> 2 files changed, 46 insertions(+)

>

> Index: 2.6.19-rc6-mm2/include/linux/nsproxy.h

> =====

> --- 2.6.19-rc6-mm2.orig/include/linux/nsproxy.h

> +++ 2.6.19-rc6-mm2/include/linux/nsproxy.h

> @@ -2,6 +2,7 @@

> #define _LINUX_NSProxy_H

>

> #include <linux/spinlock.h>

> +#include <linux/list.h>

>

> struct task_struct;

>

> @@ -34,6 +35,8 @@ struct nsproxy {

> struct pid_namespace *pid_ns;

> struct net_namespace *net_ns;

> struct user_namespace *user_ns;

> +

> + struct hlist_node ns_hash_node;

> };

> extern struct nsproxy init_nsproxy;

```

>
> Index: 2.6.19-rc6-mm2/kernel/nsproxy.c
> =====
> --- 2.6.19-rc6-mm2.orig/kernel/nsproxy.c
> +++ 2.6.19-rc6-mm2/kernel/nsproxy.c
> @@ -22,6 +22,15 @@
> #include <linux/pid_namespace.h>
> #include <linux/net_namespace.h>
>
> + #define NS_HASH_BITS 3 /* this might need some configuration */
> + #define NS_HASH_SIZE (1 << NS_HASH_BITS)
> + #define NS_HASH_MASK (NS_HASH_SIZE - 1)
> + #define ns_hashfn(id) (((id >> NS_HASH_BITS) + id) & NS_HASH_MASK)
> + #define ns_hash_head(id) &ns_hash[ns_hashfn(id)]
> +
> + static struct hlist_head ns_hash[NS_HASH_SIZE];
> + static DEFINE_SPINLOCK(ns_hash_lock);
> +
> struct nsproxy init_nsproxy = INIT_NS_PROXY(init_nsproxy);
>
> static inline void get_nsproxy(struct nsproxy *ns)
> @@ -190,3 +199,37 @@ void put_nsproxy(struct nsproxy *ns)
> free_nsproxy(ns);
> }
> }
> +
> + /*
> + * This routine must be called with the ns_hash spin_locked
> + */
> + static inline struct nsproxy *ns_hash_find(int id)
> + {
> + struct hlist_node *elem;
> + struct nsproxy *ns;
> +
> + hlist_for_each_entry(ns, elem, ns_hash_head(id), ns_hash_node) {
> + if (ns->id == id) {
> + get_nsproxy(ns);
> + return ns;
> + }
> + }
> +
> + return NULL;
> + }
> +
> + static int __init nshash_init(void)
> + {
> + int i;
> +

```

```
> + for (i = 0; i < NS_HASH_SIZE; ++i)
> + INIT_HLIST_HEAD(&ns_hash[i]);
> +
> + spin_lock_irq(&ns_hash_lock);
> + hlist_add_head(&init_nsproxy.ns_hash_node, ns_hash_head(0));
> + spin_unlock_irq(&ns_hash_lock);
> +
> + return 0;
> +}
> +
> +module_init(nshash_init);
>
> --
>


---


> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers
```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
