Subject: Re: [RFC][PATCH 1/5] Virtualization/containers: startup
Posted by Dave Hansen on Fri, 03 Feb 2006 18:34:01 GMT
View Forum Message <> Reply to Message

On Fri, 2006-02-03 at 09:49 -0800, Linus Torvalds wrote:
> One thing I don't particularly like is some of the naming. To me "vps"
> doesn't sound particularly generic or logical. I realize that it probably
> makes perfect sense to you (and I assume it just means "virtual private
> servers"), but especially if you see patches 1-3 to really be independent
> of any "actual" virtualization code that is totally generic, I'd actually
> prefer a less specialized name.

I just did a global s/vps/container/ and it looks pretty reasonable, at least from my point of view.

Couple of minor naming nitpick questions, though. Is vps/container_info really what we want to call it? It seems to me to be the basis for a real "container", without the _info part.

"tsk->owner_container" That makes it sound like a pointer to the "task owner's container". How about "owning_container"? The "container owning this task". Or, maybe just "container"?

Any particular reason for the "u32 id" in the vps_info struct as opposed to one of the more generic types? Do we want to abstract this one in the same way we do pid_t?

The "host" in "host_container_info" doesn't mean much to me. Though, I guess it has some context in the UML space. Would "init_container_info" or "root_container_info" be more descriptive?

Lastly, is this a place for krefs? I don't see a real need for a destructor yet, but the idea is fresh in my mind.

How does the attached patch look?

-- Dave

```
 memhotplug-dave/include/linux/container.h  |   40 ++++++++++++++++++++++++
 memhotplug-dave/include/linux/init_task.h  |    2 +
 memhotplug-dave/include/linux/sched.h      |    4 ++
 memhotplug-dave/init/main.c                |    8 +++++
 memhotplug-dave/kernel/exit.c              |    2 +
 memhotplug-dave/kernel/fork.c              |    2 +
 6 files changed, 58 insertions(+)
```

diff -puN include/linux/container.h~container-base include/linux/container.h
--- memhotplug/include/linux/container.h~container-base 2006-02-03 10:21:36.000000000 -0800

```
+++ memhotplug-dave/include/linux/container.h 2006-02-03 10:21:36.000000000 -0800
@@ -0,0 +1,40 @@
+#ifndef __LINUX_VPS_INFO_H_
+#define __LINUX_VPS_INFO_H_
+
+#include <asm/types.h>
+#include <asm/atomic.h>
+
+struct task_struct;
+
+struct container {
+ u32 id;
+ struct task_struct *init_task;
+ atomic_t refcnt;
+};
+
+extern struct container root_container;
+
+static inline container_t get_container(container_t container)
+{
+ atomic_inc(&container->refcnt);
+ return container;
+}
+
+static inline void put_container(container_t container)
+{
+ atomic_dec(&container->refcnt);
+}
+
+#include <linux/sched.h>
+#include <asm/current.h>
+
+static inline container_t set_current_container(container_t container)
+{
+ container_t ret;
+
+ ret = current->owning_container;
+ current->owning_container = container;
+ return ret;
+}
+
+#endif
diff -puN include/linux/init_task.h~container-base include/linux/init_task.h
--- memhotplug/include/linux/init_task.h~container-base 2006-02-03 10:21:36.000000000 -0800
+++ memhotplug-dave/include/linux/init_task.h 2006-02-03 10:23:49.000000000 -0800
@@ -3,6 +3,7 @@

 #include <linux/file.h>
```

```
 #include <linux/rcupdate.h>
+#include <linux/container.h>

 #define INIT_FDTABLE \
 {       \
@@ -121,6 +122,7 @@ extern struct group_info init_groups;
  .journal_info = NULL,      \
  .cpu_timers = INIT_CPU_TIMERS(tsk.cpu_timers),  \
  .fs_excl = ATOMIC_INIT(0),    \
+ .owning_container = &root_container,    \
 }


diff -puN include/linux/sched.h~container-base include/linux/sched.h
--- memhotplug/include/linux/sched.h~container-base 2006-02-03 10:21:36.000000000 -0800
+++ memhotplug-dave/include/linux/sched.h 2006-02-03 10:21:36.000000000 -0800
@@ -687,6 +687,7 @@ static inline void prefetch_stack(struct

 struct audit_context;  /* See audit.c */
 struct mempolicy;
+struct container;

 struct task_struct {
  volatile long state; /* -1 unrunnable, 0 runnable, >0 stopped */
@@ -844,6 +845,7 @@ struct task_struct {
  struct backing_dev_info *backing_dev_info;

  struct io_context *io_context;
+ struct container *owning_container;

  unsigned long ptrace_message;
  siginfo_t *last_siginfo; /* For ptrace use.  */
@@ -874,6 +876,8 @@ struct task_struct {
  struct rcu_head rcu;
 };

+#define current_container() (current->owning_container)
+
 static inline pid_t process_group(struct task_struct *tsk)
 {
  return tsk->signal->pgrp;
diff -puN init/main.c~container-base init/main.c
--- memhotplug/init/main.c~container-base 2006-02-03 10:21:36.000000000 -0800
+++ memhotplug-dave/init/main.c 2006-02-03 10:21:36.000000000 -0800
@@ -47,6 +47,7 @@
 #include <linux/rmap.h>
 #include <linux/mempolicy.h>
 #include <linux/key.h>
```

```
+#include <linux/container.h>

 #include <asm/io.h>
 #include <asm/bugs.h>
@@ -434,6 +435,13 @@ void __init parse_early_param(void)
  done = 1;
 }

+struct container root_container = {
+ .id  = 0,
+ .init_task = &init_task,
+ .refcnt  = ATOMIC_INIT(1),
+};
+
+EXPORT_SYMBOL(root_container);
 /*
  * Activate the first processor.
  */
diff -puN kernel/exit.c~container-base kernel/exit.c
--- memhotplug/kernel/exit.c~container-base 2006-02-03 10:21:36.000000000 -0800
+++ memhotplug-dave/kernel/exit.c 2006-02-03 10:21:36.000000000 -0800
@@ -31,6 +31,7 @@
 #include <linux/signal.h>
 #include <linux/cn_proc.h>
 #include <linux/mutex.h>
+#include <linux/container.h>

 #include <asm/uaccess.h>
 #include <asm/unistd.h>
@@ -107,6 +108,7 @@ repeat:
  spin_unlock(&p->proc_lock);
  proc_pid_flush(proc_dentry);
  release_thread(p);
+ put_container(p->owning_container);
  put_task_struct(p);

  p = leader;
diff -puN kernel/fork.c~container-base kernel/fork.c
--- memhotplug/kernel/fork.c~container-base 2006-02-03 10:21:36.000000000 -0800
+++ memhotplug-dave/kernel/fork.c 2006-02-03 10:21:36.000000000 -0800
@@ -44,6 +44,7 @@
 #include <linux/rmap.h>
 #include <linux/acct.h>
 #include <linux/cn_proc.h>
+#include <linux/container.h>

 #include <asm/pgtable.h>
 #include <asm/pgalloc.h>
```

```
@@ -1132,6 +1133,7 @@ static task_t *copy_process(unsigned lon
  p->ioprio = current->ioprio;

  SET_LINKS(p);
+ get_container(p->owning_container);
  if (unlikely(p->ptrace & PT_PTRACED))
   __ptrace_link(p, current->parent);
```

_