
Subject: [PATCH 4/6] c/r: introduce pids objects
Posted by [Oren Laadan](#) on Mon, 07 Feb 2011 17:18:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Add the interface to handle pids as proper c/r shared objects. This is the first step in converting the c/r code to handle pid as pids objects that are namespace aware. Also remove the unused field @root_pid from struct ckpt_ctx.

Signed-off-by: Oren Laadan <orenl@cs.columbia.edu>

```
include/linux/checkpoint_hdr.h | 6 +++++-
include/linux/checkpoint_types.h | 1 -
kernel/checkpoint/checkpoint.c | 2 --
kernel/checkpoint/restart.c | 1 -
kernel/pid.c | 28 ++++++++++++++++++++++++++++++++++++++
5 files changed, 33 insertions(+), 5 deletions(-)
```

diff --git a/include/linux/checkpoint_hdr.h b/include/linux/checkpoint_hdr.h

index b12d586..922eff0 100644

--- a/include/linux/checkpoint_hdr.h

+++ b/include/linux/checkpoint_hdr.h

```
@@ -245,6 +245,8 @@ struct ckpt_hdr_objref {
enum obj_type {
    CKPT_OBJ_IGNORE = 0,
#define CKPT_OBJ_IGNORE CKPT_OBJ_IGNORE
+ CKPT_OBJ_PID,
+#define CKPT_OBJ_PID CKPT_OBJ_PID
    CKPT_OBJ_INODE,
#define CKPT_OBJ_INODE CKPT_OBJ_INODE
    CKPT_OBJ_FILE_TABLE,
@@ -373,7 +375,9 @@ struct ckpt_pids {
} __attribute__((aligned(8)));
```

```
/* pids */
```

```
+#define CKPT_PID_NULL -1
```

```
/* (negative but not valid error) */
```

```
+#define CKPT_PID_NULL (-4096) /* null pid pointer */
```

```
+#define CKPT_PID_ROOT (-4097) /* pid same as root task */
```

```
/* task data */
```

```
struct ckpt_hdr_task {
```

diff --git a/include/linux/checkpoint_types.h b/include/linux/checkpoint_types.h

index cf74d3e..87a569a 100644

--- a/include/linux/checkpoint_types.h

+++ b/include/linux/checkpoint_types.h

```
@@ -35,7 +35,6 @@ struct ckpt_ctx {
    ktime_t ktime_begin; /* operation start time */
```

```

int root_init; /* [container] root init ? */
- pid_t root_pid; /* [container] root pid */
  struct task_struct *root_task; /* [container] root task */
  struct nsproxy *root_nsproxy; /* [container] root nsproxy */
  struct task_struct *root_freezer; /* [container] root task */
diff --git a/kernel/checkpoint/checkpoint.c b/kernel/checkpoint/checkpoint.c
index 01653d7..a850423 100644
--- a/kernel/checkpoint/checkpoint.c
+++ b/kernel/checkpoint/checkpoint.c
@@ -628,8 +628,6 @@ static int init_checkpoint_ctx(struct ckpt_ctx *ctx, pid_t pid)
  * occurs then ckpt_ctx_free() is eventually called.
  */

- ctx->root_pid = pid;
-
  /* root task */
  rcu_read_lock();
  task = find_task_by_vpid(pid);
diff --git a/kernel/checkpoint/restart.c b/kernel/checkpoint/restart.c
index 66b6625..9aaab4f 100644
--- a/kernel/checkpoint/restart.c
+++ b/kernel/checkpoint/restart.c
@@ -1178,7 +1178,6 @@ static struct task_struct *choose_root_task(struct ckpt_ctx *ctx, pid_t
pid)
  struct task_struct *task;

  if (ctx->uflags & RESTART_TASKSELF) {
- ctx->root_pid = pid;
  ctx->root_task = current;
  get_task_struct(current);
  return current;
diff --git a/kernel/pid.c b/kernel/pid.c
index f08f40a..0269991 100644
--- a/kernel/pid.c
+++ b/kernel/pid.c
@@ -36,6 +36,7 @@
#include <linux/pid_namespace.h>
#include <linux/init_task.h>
#include <linux/syscalls.h>
+#include <linux/checkpoint.h>

#define pid_hashfn(nr, ns) \
  hash_long((unsigned long)nr + (unsigned long)ns, pidhash_shift)
@@ -571,6 +572,30 @@ struct pid *find_ge_pid(int nr, struct pid_namespace *ns)
  return pid;
}

```

```

+ #ifdef CONFIG_CHECKPOINT
+
+ /* pid-related checkpoint objects */
+
+ static int obj_pid_grab(void *ptr)
+ {
+     get_pid((struct pid *) ptr);
+     return 0;
+ }
+
+ static void obj_pid_drop(void *ptr, int lastref)
+ {
+     put_pid((struct pid *) ptr);
+ }
+
+ static const struct ckpt_obj_ops ckpt_obj_pid_ops = {
+     .obj_name = "PID",
+     .obj_type = CKPT_OBJ_PID,
+     .ref_drop = obj_pid_drop,
+     .ref_grab = obj_pid_grab,
+ };
+
+ #endif /* CONFIG_CHECKPOINT */
+
+ /*
+  * The pid hash table is scaled according to the amount of memory in the
+  * machine. From a minimum of 16 slots up to 4096 slots at one gigabyte or
+  @@ -605,4 +630,7 @@ void __init pidmap_init(void)
+
+     init_pid_ns.pid_cachep = KMEM_CACHE(pid,
+     SLAB_HWCACHE_ALIGN | SLAB_PANIC);
+ #ifdef CONFIG_CHECKPOINT
+ register_checkpoint_obj(&ckpt_obj_pid_ops);
+ #endif
+ }
+
+ --
+ 1.7.1

```

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