Subject: [PATCH 14/33] task-containersv11-basic-task-container-framework-containers-fix-refcount-bug
Posted by Paul Menage on Mon, 17 Sep 2007 21:03:21 GMT
View Forum Message <> Reply to Message

Fix a reference counting bug in cgroupfs

As part of the extraction of cpusetfs to cgroupfs, a call to
cpuset_get_dentry() was lost (justified by the fact that the dentry in
question was now being passed down by the caller). Since
cpuset_get_dentry() called lookup_one_len(), this resulted in a reference
count being missed from the directory dentry.

This patch removes cgroup_get_dentry() and replaces it with direct calls
to lookup_one_len(); the initialization of cgroupfs dentry ops is done
now in cgroup_create_file() at dentry creation time.

Signed-off-by: Paul Menage <menage@google.com>

---
 kernel/cgroup.c | 24 +++++++----------------
 1 file changed, 7 insertions(+), 17 deletions(-)

Index: linux-2.6.23-rc4-mm1-new/kernel/cgroup.c
===
--- linux-2.6.23-rc4-mm1-new.orig/kernel/cgroup.c
+++ linux-2.6.23-rc4-mm1-new/kernel/cgroup.c
@@ -604,19 +604,6 @@ static void cgroup_diput(struct dentry *
  iput(inode);
 }

-static struct dentry *cgroup_get_dentry(struct dentry *parent,
-        const char *name)
-{
- struct dentry *d = lookup_one_len(name, parent, strlen(name));
- static struct dentry_operations cgroup_dops = {
- .d_iput = cgroup_diput,
- };
-
- if (!IS_ERR(d))
- d->d_op = &cgroup_dops;
- return d;
-}
-
 static void remove_dir(struct dentry *d)
 {
  struct dentry *parent = dget(d->d_parent);
@@ -1507,6 +1494,10 @@ static struct inode_operations cgroup_di

```
 static int cgroup_create_file(struct dentry *dentry, int mode,
     struct super_block *sb)
 {
+ static struct dentry_operations cgroup_dops = {
+  .d_iput = cgroup_diput,
+ };
+
  struct inode *inode;

  if (!dentry)
@@ -1532,7 +1523,7 @@ static int cgroup_create_file(struct den
   inode->i_size = 0;
   inode->i_fop = &cgroup_file_operations;
  }
-
+ dentry->d_op = &cgroup_dops;
  d_instantiate(dentry, inode);
  dget(dentry); /* Extra count - pin the dentry in core */
  return 0;
@@ -1553,13 +1544,12 @@ static int cgroup_create_dir(struct cgro
  int error = 0;

  parent = cont->parent->dentry;
- if (IS_ERR(dentry))
-  return PTR_ERR(dentry);
  error = cgroup_create_file(dentry, S_IFDIR | mode, cont->root->sb);
  if (!error) {
   dentry->d_fsdata = cont;
   inc_nlink(parent->d_inode);
   cont->dentry = dentry;
+  dget(dentry);
  }
  dput(dentry);

@@ -1581,7 +1571,7 @@ int cgroup_add_file(struct cgroup *cont,
  }
  strcat(name, cft->name);
  BUG_ON(!mutex_is_locked(&dir->d_inode->i_mutex));
- dentry = cgroup_get_dentry(dir, name);
+ dentry = lookup_one_len(name, dir, strlen(name));
  if (!IS_ERR(dentry)) {
   error = cgroup_create_file(dentry, 0644 | S_IFREG,
       cont->root->sb);

--
```

_____
Containers mailing list
Containers@lists.linux-foundation.org