
Subject: [PATCH 7/12] gfs2: make gfs2_holder.gh_owner_pid be a struct pid *
Posted by [Pavel Emelianov](#) on Tue, 29 Jan 2008 13:57:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

The gl_owner_pid field is used to get the holder task by its pid and check whether the current is a holder, so make it in a proper manner, i.e. via the struct pid * manipulations.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
---
fs/gfs2/glock.c | 18 ++++++++-----
fs/gfs2/glock.h |  4 +++-
fs/gfs2/incore.h |  2 +-
3 files changed, 15 insertions(+), 9 deletions(-)

diff --git a/fs/gfs2/glock.c b/fs/gfs2/glock.c
index 5fe585f..5a2b625 100644
--- a/fs/gfs2/glock.c
+++ b/fs/gfs2/glock.c
@@ -399,7 +399,7 @@ void gfs2_holder_init(struct gfs2_glock *gl, unsigned int state, unsigned
flags,
    INIT_LIST_HEAD(&gh->gh_list);
    gh->gh_gl = gl;
    gh->gh_ip = (unsigned long) __builtin_return_address(0);
- gh->gh_owner_pid = current->pid;
+ gh->gh_owner_pid = get_pid(task_pid(current));
    gh->gh_state = state;
    gh->gh_flags = flags;
    gh->gh_error = 0;
@@ -433,6 +433,6 @@ void gfs2_holder_reinit(unsigned int state, unsigned flags, struct
gfs2_holder *
void gfs2_holder_uninit(struct gfs2_holder *gh)
{
+ put_pid(gh->gh_owner_pid);
    gfs2_glock_put(gh->gh_gl);
    gh->gh_gl = NULL;
    gh->gh_ip = 0;
@@ -1050,6 +1050,6 @@ static int glock_wait_internal(struct gfs2_holder *gh)
}

static inline struct gfs2_holder *
-find_holder_by_owner(struct list_head *head, pid_t pid)
+find_holder_by_owner(struct list_head *head, struct pid *pid)
{
    struct gfs2_holder *gh;
```

```

@@ -1087,7 +1088,7 @@ static void add_to_queue(struct gfs2_holder *gh)
    struct gfs2_glock *gl = gh->gh_gl;
    struct gfs2_holder *existing;

- BUG_ON(!gh->gh_owner_pid);
+ BUG_ON(gh->gh_owner_pid == NULL);
    if (test_and_set_bit(HIF_WAIT, &gh->gh_iflags))
        BUG();

@@ -1097,12 +1098,14 @@ static void add_to_queue(struct gfs2_holder *gh)
    if (existing) {
        print_symbol(KERN_WARNING "original: %s\n",
            existing->gh_ip);
-    printk(KERN_INFO "pid : %d\n", existing->gh_owner_pid);
+    printk(KERN_INFO "pid : %d\n",
+        pid_nr(existing->gh_owner_pid));
        printk(KERN_INFO "lock type : %d lock state : %d\n",
            existing->gh_gl->gl_name.ln_type,
            existing->gh_gl->gl_state);
        print_symbol(KERN_WARNING "new: %s\n", gh->gh_ip);
-    printk(KERN_INFO "pid : %d\n", gh->gh_owner_pid);
+    printk(KERN_INFO "pid : %d\n",
+        pid_nr(gh->gh_owner_pid));
        printk(KERN_INFO "lock type : %d lock state : %d\n",
            gl->gl_name.ln_type, gl->gl_state);
        BUG();
@@ -1803,8 +1806,9 @@ static int dump_holder(struct glock_iter *gi, char *str,

    print_dbg(gi, " %s\n", str);
    if (gh->gh_owner_pid) {
-    print_dbg(gi, " owner = %ld ", (long)gh->gh_owner_pid);
-    gh_owner = find_task_by_pid(gh->gh_owner_pid);
+    print_dbg(gi, " owner = %ld ",
+        (long)pid_nr(gh->gh_owner_pid));
+    gh_owner = pid_task(gh->gh_owner_pid, PIDTYPE_PID);
        if (gh_owner)
            print_dbg(gi, "(%s)\n", gh_owner->comm);
        else
diff --git a/fs/gfs2/glock.h b/fs/gfs2/glock.h
index b16f604..2f9c6d1 100644
--- a/fs/gfs2/glock.h
+++ b/fs/gfs2/glock.h
@@ -36,11 +36,13 @@ static inline int gfs2_glock_is_locked_by_me(struct gfs2_glock *gl)
{
    struct gfs2_holder *gh;
    int locked = 0;
+    struct pid *pid;

```

```

/* Look in glock's list of holders for one with current task as owner */
spin_lock(&gl->gl_spin);
+ pid = task_pid(current);
  list_for_each_entry(gh, &gl->gl_holders, gh_list) {
- if (gh->gh_owner_pid == current->pid) {
+ if (gh->gh_owner_pid == pid) {
    locked = 1;
    break;
  }
diff --git a/fs/gfs2/incore.h b/fs/gfs2/incore.h
index 8ad1c3f..2a8d810 100644
--- a/fs/gfs2/incore.h
+++ b/fs/gfs2/incore.h
@@ -151,7 +151,7 @@ struct gfs2_holder {
  struct list_head gh_list;

  struct gfs2_glock *gh_gl;
- pid_t gh_owner_pid;
+ struct pid *gh_owner_pid;
  unsigned int gh_state;
  unsigned gh_flags;

```

--

1.5.3.4
