Subject: [PATCH] Check for error returned by kthread_create on creating journal thread
Posted by xemul on Mon, 16 Apr 2007 07:40:28 GMT
View Forum Message <> Reply to Message

If the thread failed to create the subsequent wait_event
will hang forever.

This is likely to happen if kernel hits max_threads limit.

Will be critical for virtualization systems that limit the
number of tasks and kernel memory usage within the container.

```
--- ./fs/jbd/journal.c.jbdthreads 2007-04-16 11:17:36.000000000 +0400
+++ ./fs/jbd/journal.c 2007-04-16 11:30:09.000000000 +0400
@@ -211,10 +211,16 @@ end_loop:
  return 0;
 }

-static void journal_start_thread(journal_t *journal)
+static int journal_start_thread(journal_t *journal)
 {
- kthread_run(kjournald, journal, "kjournald");
+ struct task_struct *t;
+
+ t = kthread_run(kjournald, journal, "kjournald");
+ if (IS_ERR(t))
+  return PTR_ERR(t);
+
  wait_event(journal->j_wait_done_commit, journal->j_task != 0);
+ return 0;
 }

 static void journal_kill_thread(journal_t *journal)
@@ -840,8 +846,7 @@ static int journal_reset(journal_t *jour

  /* Add the dynamic fields and write it to disk. */
  journal_update_superblock(journal, 1);
- journal_start_thread(journal);
- return 0;
+ return journal_start_thread(journal);
 }

 /**
--- ./fs/jbd2/journal.c.jbdthreads 2007-04-16 11:17:36.000000000 +0400
+++ ./fs/jbd2/journal.c 2007-04-16 11:27:45.000000000 +0400
@@ -211,10 +211,16 @@ end_loop:
  return 0;
```

```
 }

-static void jbd2_journal_start_thread(journal_t *journal)
+static int jbd2_journal_start_thread(journal_t *journal)
 {
- kthread_run(kjournald2, journal, "kjournald2");
+ struct task_struct *t;
+
+ t = kthread_run(kjournald2, journal, "kjournald2");
+ if (IS_ERR(t))
+  return PTR_ERR(t);
+
  wait_event(journal->j_wait_done_commit, journal->j_task != 0);
+ return 0;
 }

 static void journal_kill_thread(journal_t *journal)
@@ -840,8 +846,7 @@ static int journal_reset(journal_t *jour

  /* Add the dynamic fields and write it to disk. */
  jbd2_journal_update_superblock(journal, 1);
- jbd2_journal_start_thread(journal);
- return 0;
+ return jbd2_journal_start_thread(journal);
 }

 /**
```