Subject: [PATCH] AB-BA deadlock in drop_caches sysctl
Posted by den on Mon, 03 Dec 2007 13:49:43 GMT
View Forum Message <> Reply to Message

There is a AB-BA deadlock regarding drop_caches sysctl. Here are the code
paths:

```
drop_pagecache
  spin_lock(&inode_lock);
  invalidate_mapping_pages
    try_to_release_page
      ext3_releasepage
        journal_try_to_free_buffers
          __journal_try_to_free_buffer
     spin_lock(&journal->j_list_lock);

journal_commit_transaction
  spin_lock(&journal->j_list_lock);
  __journal_remove_checkpoint
    __journal_refile_buffer
      __journal_unfile_buffer
        __journal_temp_unlink_buffer
   __set_page_dirty_nobuffers
     __mark_inode_dirt
     spin_lock(&inode_lock);
```

The patch tries to address the issue - it drops inode_lock before digging into
invalidate_inode_pages. This seems sane as inode hold should not gone from the
list and should not change its place.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
--- ./fs/drop_caches.c.marker 2006-09-20 07:42:06.000000000 +0400
+++ ./fs/drop_caches.c 2007-12-03 15:43:44.000000000 +0300
@@ -14,15 +14,27 @@ int sysctl_drop_caches;

 static void drop_pagecache_sb(struct super_block *sb)
 {
- struct inode *inode;
+ struct inode *inode, *old;

+ old = NULL;
  spin_lock(&inode_lock);
  list_for_each_entry(inode, &sb->s_inodes, i_sb_list) {
   if (inode->i_state & (I_FREEING|I_WILL_FREE))
    continue;
+  __iget(inode);
+  spin_unlock(&inode_lock);
```

```
+
   invalidate_inode_pages(inode->i_mapping);
+  if (old != NULL)
+   iput(old);
+  old = inode;
+
+  spin_lock(&inode_lock);
  }
  spin_unlock(&inode_lock);
+
+ if (old != NULL)
+  iput(old);
 }

 void drop_pagecache(void)
```