

[Patch 6/10]: ext3 online resize: remove on-stack bogus inode

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-09/9042.html>

From: Stephen Tweedie (sct_at_redhat.com)

Date: 09/30/04

Date: Thu, 30 Sep 2004 14:23:45 +0100

To: linux-kernel@vger.kernel.org, Andrew Morton <akpm@osdl.org>, Andreas Dilger <[Remove the "bogus" on-stack inode. It's only used as a marker for indirectly passing the sb to various functions, so also add `ext3_journal_start_sb\(\)` which starts a transaction for a given `super_block` rather than for an inode.](mailto:adilger@clusterf</p></div><div data-bbox=)

Signed-off-by: Stephen Tweedie <sct@redhat.com>

```
--- linux-2.6.9-rc2-mm4/fs/ext3/resize.c.=K0005=.orig
+++ linux-2.6.9-rc2-mm4/fs/ext3/resize.c
@@ -159,7 +159,7 @@ static void mark_bitmap_end(int start_bi
 *
 * We only pass inode because of the ext3 journal wrappers.
 */
-static int setup_new_group_blocks(struct super_block *sb, struct inode *inode,
+static int setup_new_group_blocks(struct super_block *sb,
                                struct ext3_new_group_data *input)
{
    struct ext3_sb_info *sbi = EXT3_SB(sb);
@@ -175,8 +175,8 @@ static int setup_new_group_blocks(struct
    int i;
    int err = 0, err2;

- handle = ext3_journal_start(inode, reserved_gdb + gdblocks +
- 2 + sbi->s_itb_per_group);
+ handle = ext3_journal_start_sb(sb, reserved_gdb + gdblocks +
+ 2 + sbi->s_itb_per_group);
    if (IS_ERR(handle))
        return PTR_ERR(handle);

@@ -616,7 +616,7 @@ exit_free:
 *
 * We only pass inode because of the ext3 journal wrappers.
 */
-static void update_backups(struct super_block *sb, struct inode *inode,
+static void update_backups(struct super_block *sb,
```

Linux-Kernel: [Patch 6/10]: ext3 online resize: remove on-stack bogus inode

```

        int blk_off, char *data, int size)
{
    struct ext3_sb_info *sbi = EXT3_SB(sb);
@@ -630,7 +630,7 @@ static void update_backups(struct super_
    handle_t *handle;
    int err = 0, err2;

- handle = ext3_journal_start(inode, EXT3_MAX_TRANS_DATA);
+ handle = ext3_journal_start_sb(sb, EXT3_MAX_TRANS_DATA);
    if (IS_ERR(handle)) {
        group = 1;
        err = PTR_ERR(handle);
@@ -706,7 +706,6 @@ int ext3_group_add(struct super_block *s
    struct buffer_head *primary = NULL;
    struct ext3_group_desc *gdp;
    struct inode *inode = NULL;
- struct inode bogus;
    handle_t *handle;
    int gdb_off, gdb_num;
    int err, err2;
@@ -735,16 +734,12 @@ int ext3_group_add(struct super_block *s
        iput(inode);
        return -ENOENT;
    }
- } else {
- /* Used only for ext3 journal wrapper functions to get sb */
- inode = &bogus;
- bogus.i_sb = sb;
    }

    if ((err = verify_group_input(sb, input)))
        goto exit_put;

- if ((err = setup_new_group_blocks(sb, inode, input)))
+ if ((err = setup_new_group_blocks(sb, input)))
        goto exit_put;

    /*
@@ -754,8 +749,9 @@ int ext3_group_add(struct super_block *s
    * are adding a group with superblock/GDT backups we will also
    * modify each of the reserved GDT dindirect blocks.
    */
- handle = ext3_journal_start(inode, ext3_bg_has_super(sb, input->group) ?
- 3 + reserved_gdb : 4);
+ handle = ext3_journal_start_sb(sb,
+ ext3_bg_has_super(sb, input->group) ?
+ 3 + reserved_gdb : 4);
    if (IS_ERR(handle)) {
        err = PTR_ERR(handle);
        goto exit_put;
@@ -822,14 +818,13 @@ exit_journal:

```

Linux-Kernel: [Patch 6/10]: ext3 online resize: remove on-stack bogus inode

```

    if ((err2 = ext3_journal_stop(handle)) && !err)
        err = err2;
    if (!err) {
- update_backups(sb, inode, sbi->s_sbh->b_blocknr, (char *)es,
+ update_backups(sb, sbi->s_sbh->b_blocknr, (char *)es,
                    sizeof(struct ext3_super_block));
- update_backups(sb, inode, primary->b_blocknr, primary->b_data,
+ update_backups(sb, primary->b_blocknr, primary->b_data,
                    primary->b_size);
    }
    exit_put:
- if (inode != &bogus)
- iput(inode);
+ iput(inode);
    return err;
} /* ext3_group_add */

@@ -949,7 +944,7 @@ int ext3_group_extend(struct super_block
    if (test_opt(sb, DEBUG))
        printk(KERN_DEBUG "EXT3-fs: extended group to %u blocks\n",
                le32_to_cpu(es->s_blocks_count));
- update_backups(sb, inode, EXT3_SB(sb)->s_sbh->b_blocknr, (char *)es,
+ update_backups(sb, EXT3_SB(sb)->s_sbh->b_blocknr, (char *)es,
                    sizeof(struct ext3_super_block));
    exit_put:
        iput(inode);
--- linux-2.6.9-rc2-mm4/fs/ext3/super.c=K0005=.orig
+++ linux-2.6.9-rc2-mm4/fs/ext3/super.c
@@ -59,19 +59,19 @@ static int ext3_sync_fs(struct super_block
 * that sync() will call the filesystem's write_super callback if
 * appropriate.
 */
-handle_t *ext3_journal_start(struct inode *inode, int nblocks)
+handle_t *ext3_journal_start_sb(struct super_block *sb, int nblocks)
{
    journal_t *journal;

- if (inode->i_sb->s_flags & MS_RDONLY)
+ if (sb->s_flags & MS_RDONLY)
    return ERR_PTR(-EROFS);

    /* Special case here: if the journal has aborted behind our
     * backs (eg. EIO in the commit thread), then we still need to
     * take the FS itself readonly cleanly. */
- journal = EXT3_JOURNAL(inode);
+ journal = EXT3_SB(sb)->s_journal;
    if (is_journal_aborted(journal)) {
- ext3_abort(inode->i_sb, __FUNCTION__,
+ ext3_abort(sb, __FUNCTION__,
                "Detected aborted journal");
    return ERR_PTR(-EROFS);

```

Linux-Kernel: [Patch 6/10]: ext3 online resize: remove on-stack bogus inode

```
    }
--- linux-2.6.9-rc2-mm4/include/linux/ext3_jbd.h.=K0005=.orig
+++ linux-2.6.9-rc2-mm4/include/linux/ext3_jbd.h
@@ -188,9 +188,14 @@ __ext3_journal_dirty_metadata(const char
#define ext3_journal_dirty_metadata(handle, bh) \
    __ext3_journal_dirty_metadata(__FUNCTION__, (handle), (bh))

-handle_t *ext3_journal_start(struct inode *inode, int nblocks);
+handle_t *ext3_journal_start_sb(struct super_block *sb, int nblocks);
int __ext3_journal_stop(const char *where, handle_t *handle);

+static inline handle_t *ext3_journal_start(struct inode *inode, int nblocks)
+{
+ return ext3_journal_start_sb(inode->i_sb, nblocks);
+}
+
#define ext3_journal_stop(handle) \
    __ext3_journal_stop(__FUNCTION__, (handle))

-
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>