

## [Patch 8/10]: ext3 online resize: remove s\_debts

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-09/9039.html>

---

**From:** Stephen Tweedie ([sct\\_at\\_redhat.com](mailto:sct_at_redhat.com))

**Date:** 09/30/04

Date: Thu, 30 Sep 2004 14:23:55 +0100

To: linux-kernel@vger.kernel.org, Andrew Morton <[akpm@osdl.org](mailto:akpm@osdl.org)>, Andreas Dilger <[s\\_debts is currently not used by ext3 \(it is created, destroyed and checked but never set\). Remove it for now.](mailto:adilger@clusterf</a></p></div><div data-bbox=)

Resurrecting this will require adding it back in changed form. In existing form it's already unsafe wrt. byte-tearing as it performs unlocked byte increment/decrement on words which may be being accessed simultaneously on other CPUs. It is also the only in-memory dynamic table which needs to be extended by online-resize, so locking it will require care.

Signed-off-by: Stephen Tweedie <[sct@redhat.com](mailto:sct@redhat.com)>

```
--- linux-2.6.9-rc2-mm4/fs/ext3/ialloc.c.=K0007=.orig
+++ linux-2.6.9-rc2-mm4/fs/ext3/ialloc.c
@@ -320,8 +320,6 @@ static int find_group_orlov(struct super
     desc = ext3_get_group_desc (sb, group, &bh);
     if (!desc || !desc->bg_free_inodes_count)
         continue;
- if (sbi->s_debts[group] >= max_debt)
- continue;
     if (le16_to_cpu(desc->bg_used_dirs_count) >= max_dirs)
         continue;
     if (le16_to_cpu(desc->bg_free_inodes_count) < min_inodes)
--- linux-2.6.9-rc2-mm4/fs/ext3/super.c.=K0007=.orig
+++ linux-2.6.9-rc2-mm4/fs/ext3/super.c
@@ -400,7 +400,6 @@ void ext3_put_super (struct super_block
     for (i = 0; i < sbi->s_gdb_count; i++)
         brelse(sbi->s_group_desc[i]);
     kfree(sbi->s_group_desc);
- kfree(sbi->s_debts);
     brelse(sbi->s_sbh);
#ifdef CONFIG_QUOTA
     for (i = 0; i < MAXQUOTAS; i++) {
@@ -1460,13 +1459,6 @@ static int ext3_fill_super (struct super
         printk (KERN_ERR "EXT3-fs: not enough memory\n");
         goto failed_mount;
     }
}
```

Linux-Kernel: [Patch 8/10]: ext3 online resize: remove s\_debts

```
- sbi->s_debts = kmalloc(sbi->s_groups_count * sizeof(u8),
- GFP_KERNEL);
- if (!sbi->s_debts) {
- printk("EXT3-fs: not enough memory to allocate s_bgi\n");
- goto failed_mount2;
- }
- memset(sbi->s_debts, 0, sbi->s_groups_count * sizeof(u8));

    percpu_counter_init(&sbi->s_freeblocks_counter);
    percpu_counter_init(&sbi->s_freeinodes_counter);
@@ -1618,7 +1610,6 @@ static int ext3_fill_super (struct super
failed_mount3:
    journal_destroy(sbi->s_journal);
failed_mount2:
- kfree(sbi->s_debts);
    for (i = 0; i < db_count; i++)
        brelse(sbi->s_group_desc[i]);
    kfree(sbi->s_group_desc);
--- linux-2.6.9-rc2-mm4/include/linux/ext3_fs_sb.h.=K0007=.orig
+++ linux-2.6.9-rc2-mm4/include/linux/ext3_fs_sb.h
@@ -54,7 +54,6 @@ struct ext3_sb_info {
    u32 s_next_generation;
    u32 s_hash_seed[4];
    int s_def_hash_version;
- u8 *s_debts;
    struct percpu_counter s_freeblocks_counter;
    struct percpu_counter s_freeinodes_counter;
    struct percpu_counter s_dirs_counter;
-
```

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/>