

---

Subject: [PATCH] remove BUG() in possible but rare condition  
Posted by [Glauber Costa](#) on Wed, 11 Apr 2012 18:10:24 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

While stressing the kernel with failing allocations today,  
I hit the following chain of events:

```
alloc_page_buffers():
```

```
    bh = alloc_buffer_head(GFP_NOFS);  
    if (!bh)  
        goto no_grow; <= path taken
```

```
grow_dev_page():  
    bh = alloc_page_buffers(page, size, 0);  
    if (!bh)  
        goto failed; <= taken, consequence of the above
```

and then the failed path BUG()s the kernel.

The failure is inserted a little bit artificially, but even then,  
I see no reason why it should be deemed impossible in a real box.

Even though this is not a condition that we expect to see  
around every time, failed allocations are expected to be handled,  
and BUG() sounds just too much. As a matter of fact, grow\_dev\_page()  
can return NULL just fine in other circumstances, so I propose we just  
remove it, then.

Signed-off-by: Glauber Costa <glommer@parallels.com>  
CC: Linus Torvalds <torvalds@linux-foundation.org>  
CC: Andrew Morton <akpm@linux-foundation.org>

---

```
fs/buffer.c | 1 -  
1 files changed, 0 insertions(+), 1 deletions(-)
```

```
diff --git a/fs/buffer.c b/fs/buffer.c  
index 36d6665..351e18e 100644
```

```
--- a/fs/buffer.c
```

```
+++ b/fs/buffer.c
```

```
@ @ -985,7 +985,6 @ @ grow_dev_page(struct block_device *bdev, sector_t block,  
    return page;
```

```
failed:  
- BUG();  
  unlock_page(page);  
  page_cache_release(page);  
  return NULL;
```



--

1.7.7.6

---