

[View Forum Message](#) <> [Reply to Message](#)

Dmitriy Monakhov <dmonakhov@sw.ru> wrote:

- > incorrect direct io error handling (v3)
- > Changes from v2:
  - > - Remove BUG\_ON(!mutex\_is\_locked(..)) for non blkdev.
  - > - vmtruncate() called from generic\_file\_aio\_write().
  - > - depends on patch titled:
  - > [PATH][RFC] mm: Move common segments checks to separate function

drat, I skipped that patch due to rejects, and because Nick is working on things in the same area.

[illegible]

```

> #fsck.ext3 -f /dev/VG/test
> e2fsck 1.39 (29-May-2006)
> Pass 1: Checking inodes, blocks, and sizes
> Inode 14, i_size is 0, should be 56556544. Fix<y>? yes
> Pass 2: Checking directory structure
>
> Signed-off-by: Dmitriy Monakhov <dmonakhov@openvz.org>
> -----
>

```

```

diff --git a/mm/filemap.c b/mm/filemap.c
> index d01abb6..96840e5 100644
> --- a/mm/filemap.c
> +++ b/mm/filemap.c
> @@ -2058,8 +2058,9 @@ generic_file_direct_write(struct kiocb *
> /*
>  * Sync the fs metadata but not the minor inode changes and
>  * of course not the data as we did direct DMA for the IO.
> - * i_mutex is held, which protects generic_osync_inode() from
> - * livelocking. AIO O_DIRECT ops attempt to sync metadata here.
> + * i_mutex may not being held, if so some specific locking
> + * ordering must protect generic_osync_inode() from livelocking.
> + * AIO O_DIRECT ops attempt to sync metadata here.
> */
> if ((written >= 0 || written == -EIOCBQUEUED) &&
>     ((file->f_flags & O_SYNC) || IS_SYNC(inode))) {
> @@ -2365,6 +2366,17 @@ ssize_t generic_file_aio_write(struct ki
>     &iocb->ki_pos);
>     mutex_unlock(&inode->i_mutex);
>
> + if (unlikely(ret < 0 && (file->f_flags & O_DIRECT))) {
> +     ssize_t cnt = generic_segment_checks(nr_segs, iov, VERIFY_READ);
> +     loff_t isize = i_size_read(inode);
> +     /*
> +      * generic_file_direct_write() may have instantiated a few
> +      * blocks outside i_size. Trim these off again.
> +      */
> +     if (cnt > 0 && (pos + cnt > isize))
> +         vmtruncate(inode, isize);
> + }
> +

```

vmtruncate() really wants i\_mutex to be held. Can't we do that here?

---