
Subject: [PATCH 4/4] fuse: re-work fuse_direct_IO()
Posted by [Maxim Patlasov](#) on Fri, 20 Jul 2012 11:50:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

The patch eliminates `fuse_loop_dio()` by passing `iovec[]` transparently from `fuse_direct_IO()` to `__fuse_direct_io()`. The latter is responsible now for processing elements of `iovec[]`. This allows `__fuse_direct_io()` to pack many `iovec`-s to each `fuse_req`, effectively minimizing number of `fuse_req`-s required.

Signed-off-by: Maxim Patlasov <mpatlasov@parallels.com>

`fs/fuse/file.c | 64 ++++++-----`

1 files changed, 21 insertions(+), 43 deletions(-)

```
diff --git a/fs/fuse/file.c b/fs/fuse/file.c
index 6c8e24f..0355128 100644
--- a/fs/fuse/file.c
+++ b/fs/fuse/file.c
@@ -1158,8 +1158,8 @@ ssize_t fuse_direct_io(struct file *file, const char __user *buf,
}
EXPORT_SYMBOL_GPL(fuse_direct_io);

-static ssize_t fuse_direct_read(struct file *file, char __user *buf,
-    size_t count, loff_t *ppos)
+static ssize_t __fuse_direct_read(struct file *file, const struct iovec *iov,
+    unsigned long nr_segs, loff_t *ppos)
{
    ssize_t res;
    struct inode *inode = file->f_path.dentry->d_inode;
@@ -1167,22 +1167,31 @@ static ssize_t fuse_direct_read(struct file *file, char __user *buf,
    if (is_bad_inode(inode))
        return -EIO;

-    res = fuse_direct_io(file, buf, count, ppos, 0);
+    res = __fuse_direct_io(file, iov, nr_segs, iov_length(iov, nr_segs),
+        ppos, 0);

    fuse_invalidate_attr(inode);

    return res;
}

-static ssize_t __fuse_direct_write(struct file *file, const char __user *buf,
-    size_t count, loff_t *ppos)
+static ssize_t fuse_direct_read(struct file *file, char __user *buf,
+    size_t count, loff_t *ppos)
+{

```

```

+ struct iovec iov = { .iov_base = (void *)buf, .iov_len = count };
+ return __fuse_direct_read(file, &iov, 1, ppos);
+}
+
+static ssize_t __fuse_direct_write(struct file *file, const struct iovec *iov,
+      unsigned long nr_segs, loff_t *ppos)
{
    struct inode *inode = file->f_path.dentry->d_inode;
+ size_t count = iov_length(iov, nr_segs);
    ssize_t res;

    res = generic_write_checks(file, ppos, &count, 0);
    if (!res) {
-     res = fuse_direct_io(file, buf, count, ppos, 1);
+     res = __fuse_direct_io(file, iov, nr_segs, count, ppos, 1);
        if (res > 0)
            fuse_write_update_size(inode, *ppos);
    }
@@ -1195,6 +1204,7 @@ static ssize_t __fuse_direct_write(struct file *file, const char __user
*buf,
 static ssize_t fuse_direct_write(struct file *file, const char __user *buf,
     size_t count, loff_t *ppos)
{
+ struct iovec iov = { .iov_base = (void *)buf, .iov_len = count };
    struct inode *inode = file->f_path.dentry->d_inode;
    ssize_t res;

@@ -1203,7 +1213,7 @@ static ssize_t fuse_direct_write(struct file *file, const char __user *buf,
/* Don't allow parallel writes to the same file */
mutex_lock(&inode->i_mutex);
- res = __fuse_direct_write(file, buf, count, ppos);
+ res = __fuse_direct_write(file, &iov, 1, ppos);
    mutex_unlock(&inode->i_mutex);

    return res;
@@ -2161,41 +2171,6 @@ int fuse_notify_poll_wakeup(struct fuse_conn *fc,
    return 0;
}

-static ssize_t fuse_loop_dio(struct file *filp, const struct iovec *iov,
-      unsigned long nr_segs, loff_t *ppos, int rw)
-{
- const struct iovec *vector = iov;
- ssize_t ret = 0;
-
- while (nr_segs > 0) {
- void __user *base;

```

```

- size_t len;
- ssize_t nr;
-
- base = vector->iov_base;
- len = vector->iov_len;
- vector++;
- nr_segs--;
-
- if (rw == WRITE)
- nr = __fuse_direct_write(filp, base, len, ppos);
- else
- nr = fuse_direct_read(filp, base, len, ppos);
-
- if (nr < 0) {
- if (!ret)
- ret = nr;
- break;
- }
- ret += nr;
- if (nr != len)
- break;
- }
-
- return ret;
-}
-
-
static ssize_t
fuse_direct_IO(int rw, struct kiocb *iocb, const struct iovec *iov,
loff_t offset, unsigned long nr_segs)
@@ -2207,7 +2182,10 @@ fuse_direct_IO(int rw, struct kiocb *iocb, const struct iovec *iov,
file = iocb->ki_filp;
pos = offset;

- ret = fuse_loop_dio(file, iov, nr_segs, &pos, rw);
+ if (rw == WRITE)
+ ret = __fuse_direct_write(file, iov, nr_segs, &pos);
+ else
+ ret = __fuse_direct_read(file, iov, nr_segs, &pos);

return ret;
}

```
