

---

Subject: [PATCH 12/12] fuse: optimize \_\_fuse\_direct\_io()  
Posted by [Maxim Patlasov](#) on Fri, 26 Oct 2012 15:50:36 GMT  
[View Forum Message](#) <[Reply to Message](#)

---

\_\_fuse\_direct\_io() allocates fuse-requests by calling fuse\_get\_req(fc, n). The patch calculates 'n' based on iov[] array. This is useful because allocating FUSE\_MAX\_PAGES\_PER\_REQ page pointers and descriptors for each fuse request would be waste of memory in case of iov-s of smaller size.

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

---

fs/fuse/file.c | 25 ++++++-----  
1 files changed, 21 insertions(+), 4 deletions(-)

```
diff --git a/fs/fuse/file.c b/fs/fuse/file.c
index fad8c7b..1a9ae5a 100644
--- a/fs/fuse/file.c
+++ b/fs/fuse/file.c
@@ -1088,14 +1088,14 @@ static int fuse_get_user_pages(struct fuse_req *req, struct iov_iter
*ii,
    return 0;
}

- while ( nbytes < *nbytesp && req->num_pages < FUSE_MAX_PAGES_PER_REQ) {
+ while ( nbytes < *nbytesp && req->num_pages < req->max_pages) {
    unsigned npages;
    unsigned long user_addr = fuse_get_user_addr(ii);
    unsigned offset = user_addr & ~PAGE_MASK;
    size_t frag_size = fuse_get_frag_size(ii, *nbytesp - nbytes);
    int ret;

-    unsigned n = FUSE_MAX_PAGES_PER_REQ - req->num_pages;
+    unsigned n = req->max_pages - req->num_pages;
    frag_size = min_t(size_t, frag_size, n << PAGE_SHIFT);

    npages = (frag_size + offset + PAGE_SIZE - 1) >> PAGE_SHIFT;
@@ -1131,6 +1131,23 @@ static int fuse_get_user_pages(struct fuse_req *req, struct iov_iter *ii,
    return 0;
}

+static inline int fuse_iter_npaging(const struct iov_iter *ii_p)
+{
+    struct iov_iter ii = *ii_p;
+    int npages = 0;
+
+    while (iov_iter_count(&ii) && npages < FUSE_MAX_PAGES_PER_REQ) {
+        unsigned long user_addr = fuse_get_user_addr(&ii);
+        unsigned offset = user_addr & ~PAGE_MASK;
```

```

+ size_t frag_size = iov_iter_single_seg_count(&ii);
+
+ npages += (frag_size + offset + PAGE_SIZE - 1) >> PAGE_SHIFT;
+ iov_iter_advance(&ii, frag_size);
+ }
+
+ return min(npages, FUSE_MAX_PAGES_PER_REQ);
+}
+
static ssize_t __fuse_direct_io(struct file *file, const struct iovec *iov,
    unsigned long nr_segs, size_t count,
    loff_t *ppos, int write)
@@ -1145,7 +1162,7 @@ static ssize_t __fuse_direct_io(struct file *file, const struct iovec *iov,
    iov_iter_init(&ii, iov, nr_segs, count, 0);

- req = fuse_get_req(fc, FUSE_MAX_PAGES_PER_REQ);
+ req = fuse_get_req(fc, fuse_iter_npates(&ii));
if (IS_ERR(req))
    return PTR_ERR(req);

@@ -1180,7 +1197,7 @@ static ssize_t __fuse_direct_io(struct file *file, const struct iovec *iov,
    break;
if (count) {
    fuse_put_request(fc, req);
- req = fuse_get_req(fc, FUSE_MAX_PAGES_PER_REQ);
+ req = fuse_get_req(fc, fuse_iter_npates(&ii));
if (IS_ERR(req))
    break;
}

```

---