
Subject: [PATCH 11/11] fuse: optimize __fuse_direct_io()
Posted by [Maxim Patlasov](#) on Wed, 19 Sep 2012 16:33:31 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 | 27 ++++++-----
1 files changed, 23 insertions(+), 4 deletions(-)

diff --git a/fs/fuse/file.c b/fs/fuse/file.c
index 5b0fa5d..5ecab19 100644

--- a/fs/fuse/file.c

+++ b/fs/fuse/file.c

```
@@ -1090,14 +1090,14 @@ static int fuse_get_user_pages(struct fuse_req *req,  
    return 0;  
}
```

```
- while (nbytes < *nbytesp && req->num_pages < FUSE_MAX_PAGES_PER_REQ) {  
+ while (nbytes < *nbytesp && req->num_pages < req->max_pages) {  
    int npages;  
    unsigned long user_addr = fuse_get_ua(*iov_pp, *iov_offset_p);  
    unsigned offset = user_addr & ~PAGE_MASK;  
    size_t frag_size = fuse_get_fr_sz(*iov_pp, *iov_offset_p,  
        *nbytesp - nbytes);
```

```
- int n = FUSE_MAX_PAGES_PER_REQ - req->num_pages;  
+ int 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;  
@@ -1138,6 +1138,24 @@ static int fuse_get_user_pages(struct fuse_req *req,  
    return 0;  
}
```

```
+static inline int fuse_iov_npAGES(const struct iovec *iov,  
+    unsigned long nr_segs, size_t iov_offset)  
+{  
+ int npages = 0;  
+  
+ while (nr_segs-- > 0 && npages < FUSE_MAX_PAGES_PER_REQ) {  
+ unsigned long user_addr = fuse_get_ua(iov, iov_offset);  
+ unsigned offset = user_addr & ~PAGE_MASK;  
+ size_t frag_size = iov->iov_len - iov_offset;
```

```

+
+ npages += (frag_size + offset + PAGE_SIZE - 1) >> PAGE_SHIFT;
+ iov++;
+ iov_offset = 0;
+ }
+
+ 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)
@@ -1150,7 +1168,7 @@ static ssize_t __fuse_direct_io(struct file *file, const struct iovec *iov,
    struct fuse_req *req;
    size_t iov_offset = 0;

- req = fuse_get_req(fc, FUSE_MAX_PAGES_PER_REQ);
+ req = fuse_get_req(fc, fuse_iov_npages(iov, nr_segs, iov_offset));
    if (IS_ERR(req))
        return PTR_ERR(req);

@@ -1186,7 +1204,8 @@ 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_iov_npages(iov, nr_segs, iov_offset));
        if (IS_ERR(req))
            break;
    }

```
