Subject: [PATCH 11/12] fuse: optimize fuse_get_user_pages()
Posted by Maxim Patlasov on Fri, 26 Oct 2012 15:50:29 GMT
View Forum Message <> Reply to Message

Let fuse_get_user_pages() pack as many iov-s to a single fuse_req as
possible. This is very beneficial in case of iov[] consisting of many
iov-s of relatively small sizes (e.g. PAGE_SIZE).

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

```
---
 fs/fuse/file.c |   79 +++++++++++++++++++++++++++++++++++---------------------
 1 files changed, 51 insertions(+), 28 deletions(-)

diff --git a/fs/fuse/file.c b/fs/fuse/file.c
index 9934321..fad8c7b 100644
--- a/fs/fuse/file.c
+++ b/fs/fuse/file.c
@@ -1047,29 +1047,37 @@ static void fuse_release_user_pages(struct fuse_req *req, int write)
  }
 }

-static inline void fuse_page_descs_length_init(struct fuse_req *req)
+static inline void fuse_page_descs_length_init(struct fuse_req *req,
+  unsigned index, unsigned nr_pages)
 {
  int i;

- for (i = 0; i < req->num_pages; i++)
+ for (i = index; i < index + nr_pages; i++)
   req->page_descs[i].length = PAGE_SIZE -
    req->page_descs[i].offset;
 }

+static inline unsigned long fuse_get_user_addr(const struct iov_iter *ii)
+{
+ return (unsigned long)ii->iov->iov_base + ii->iov_offset;
+}
+
+static inline size_t fuse_get_frag_size(const struct iov_iter *ii,
+     size_t max_size)
+{
+ return min(iov_iter_single_seg_count(ii), max_size);
+}
+
 static int fuse_get_user_pages(struct fuse_req *req, struct iov_iter *ii,
          size_t *nbytesp, int write)
 {
- size_t nbytes = *nbytesp;
```

```
- size_t frag_size = min(iov_iter_single_seg_count(ii), nbytes);
- unsigned long user_addr;
- unsigned offset;
- int npages;
-
- user_addr = (unsigned long)ii->iov->iov_base + ii->iov_offset;
- offset = user_addr & ~PAGE_MASK;
+ size_t nbytes = 0;  /* # bytes already packed in req */

  /* Special case for kernel I/O: can copy directly into the buffer */
  if (segment_eq(get_fs(), KERNEL_DS)) {
+  unsigned long user_addr = fuse_get_user_addr(ii);
+  size_t frag_size = fuse_get_frag_size(ii, *nbytesp);
+
   if (write)
    req->in.args[1].value = (void *) user_addr;
   else
@@ -1080,30 +1088,45 @@ static int fuse_get_user_pages(struct fuse_req *req, struct iov_iter
*ii,
   return 0;
  }

- nbytes = min_t(size_t, frag_size, FUSE_MAX_PAGES_PER_REQ << PAGE_SHIFT);
- npages = (nbytes + offset + PAGE_SIZE - 1) >> PAGE_SHIFT;
- npages = clamp(npages, 1, FUSE_MAX_PAGES_PER_REQ);
- npages = get_user_pages_fast(user_addr, npages, !write, req->pages);
- if (npages < 0)
-  return npages;
+ while (nbytes < *nbytesp && req->num_pages < FUSE_MAX_PAGES_PER_REQ) {
+  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;

- req->num_pages = npages;
- req->page_descs[0].offset = offset;
- fuse_page_descs_length_init(req);
+  unsigned n = FUSE_MAX_PAGES_PER_REQ - req->num_pages;
+  frag_size = min_t(size_t, frag_size, n << PAGE_SHIFT);
+
+  npages = (frag_size + offset + PAGE_SIZE - 1) >> PAGE_SHIFT;
+  npages = clamp(npages, 1U, n);
+
+  ret = get_user_pages_fast(user_addr, npages, !write,
+        &req->pages[req->num_pages]);
+  if (ret < 0)
+   return ret;
```

```
+
+  npages = ret;
+  frag_size = min_t(size_t, frag_size,
+      (npages << PAGE_SHIFT) - offset);
+  iov_iter_advance(ii, frag_size);
+
+  req->page_descs[req->num_pages].offset = offset;
+  fuse_page_descs_length_init(req, req->num_pages, npages);
+
+  req->num_pages += npages;
+  req->page_descs[req->num_pages - 1].length -=
+    (npages << PAGE_SHIFT) - offset - frag_size;
+
+  nbytes += frag_size;
+ }

   if (write)
    req->in.argpages = 1;
   else
    req->out.argpages = 1;

- nbytes = (req->num_pages << PAGE_SHIFT) - req->page_descs[0].offset;
-
- if (frag_size < nbytes)
-   req->page_descs[req->num_pages - 1].length -=
-    nbytes - frag_size;
-
- *nbytesp = min(frag_size, nbytes);
- iov_iter_advance(ii, *nbytesp);
+ *nbytesp = nbytes;

   return 0;
 }
@@ -1948,7 +1971,7 @@ long fuse_do_ioctl(struct file *file, unsigned int cmd, unsigned long
arg,
   }
   memcpy(req->pages, pages, sizeof(req->pages[0]) * num_pages);
   req->num_pages = num_pages;
- fuse_page_descs_length_init(req);
+ fuse_page_descs_length_init(req, 0, req->num_pages);

   /* okay, let's send it to the client */
   req->in.h.opcode = FUSE_IOCTL;
```