Subject: Re: [PATCH 5/6] fuse: rework fuse_perform_write()
Posted by Miklos Szeredi on Wed, 12 Sep 2012 16:43:54 GMT
View Forum Message <> Reply to Message

Maxim Patlasov <mpatlasov@parallels.com> writes:

> The patch allocates as many page pointers in fuse_req as needed to cover
> interval [pos .. pos+len-1]. FUSE_WR_PAGES macro is introduced to hide this
> cumbersome arithmetics.

Please don't ever use a macro for something which you can use an inline
function.

> ---
> fs/fuse/file.c | 15 ++++++++++----
> 1 files changed, 11 insertions(+), 4 deletions(-)
>
> diff --git a/fs/fuse/file.c b/fs/fuse/file.c
> index 9a6dcc6..84cc83c 100644
> --- a/fs/fuse/file.c
> +++ b/fs/fuse/file.c
> @@ -815,7 +815,8 @@ static size_t fuse_send_write_pages(struct fuse_req *req, struct file
*file,
>
> static ssize_t fuse_fill_write_pages(struct fuse_req *req,
>   struct address_space *mapping,
> -   struct iov_iter *ii, loff_t pos)
> +   struct iov_iter *ii, loff_t pos,
> +   unsigned nr_pages)
> {
>   struct fuse_conn *fc = get_fuse_conn(mapping->host);
>   unsigned offset = pos & (PAGE_CACHE_SIZE - 1);
> @@ -875,11 +876,16 @@ static ssize_t fuse_fill_write_pages(struct fuse_req *req,
>   if (!fc->big_writes)
>   break;
>   } while (iov_iter_count(ii) && count < fc->max_write &&
> -   req->num_pages < FUSE_MAX_PAGES_PER_REQ && offset == 0);
> +   req->num_pages < nr_pages && offset == 0);
>
>   return count > 0 ? count : err;
> }
>
> +#define FUSE_WR_PAGES(pos, len) min_t(unsigned,   \
> +   ((pos + len - 1) >> PAGE_CACHE_SHIFT) - \
> +   (pos >> PAGE_CACHE_SHIFT) + 1,   \
> +   FUSE_MAX_PAGES_PER_REQ)
> +
> static ssize_t fuse_perform_write(struct file *file,

```
>        struct address_space *mapping,
>        struct iov_iter *ii, loff_t pos)
> @@ -895,14 +901,15 @@ static ssize_t fuse_perform_write(struct file *file,
>   do {
>    struct fuse_req *req;
>    ssize_t count;
> +  unsigned nr_pages = FUSE_WR_PAGES(pos, iov_iter_count(ii));
>
> -  req = fuse_get_req_multipage(fc, FUSE_MAX_PAGES_PER_REQ);
> +  req = fuse_get_req_multipage(fc, nr_pages);
>    if (IS_ERR(req)) {
>     err = PTR_ERR(req);
>     break;
>    }
>
> -  count = fuse_fill_write_pages(req, mapping, ii, pos);
> +  count = fuse_fill_write_pages(req, mapping, ii, pos, nr_pages);
>    if (count <= 0) {
>     err = count;
>    } else {
```