

---

Subject: Re: [PATCH 1/4] fuse: add basic support of iovec[] to fuse\_req

Posted by [Miklos Szeredi](#) on Wed, 08 Aug 2012 16:02:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Maxim Patlasov <[mpatlasov@parallels.com](mailto:mpatlasov@parallels.com)> writes:

```
> The patch allows fuse_req to refer to array of iovec-s describing
> layout of user-data over req->pages. fuse_copy_pages() is re-worked to
> support both cased: former layout where pages[] corresponded to <buf, len>
> and newer one where pages[] corresponds to iovec[].
>
> Signed-off-by: Maxim Patlasov <mpatlasov@parallels.com>
> ---
> fs/fuse/dev.c | 52 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> fs/fuse/fuse_i.h | 12 ++++++
> 2 files changed, 59 insertions(+), 5 deletions(-)
>
> diff --git a/fs/fuse/dev.c b/fs/fuse/dev.c
> index 7df2b5e..cdae525 100644
> --- a/fs/fuse/dev.c
> +++ b/fs/fuse/dev.c
> @@ -850,9 +850,9 @@ static int fuse_copy_page(struct fuse_copy_state *cs, struct page
> **pagep,
> return 0;
> }
>
> -/* Copy pages in the request to/from userspace buffer */
> -static int fuse_copy_pages(struct fuse_copy_state *cs, unsigned nbytes,
> -    int zeroing)
> +/* Start from addr(pages[0]) + page_offset. No holes in the middle. */
> +static int fuse_copy_pages_for_buf(struct fuse_copy_state *cs, unsigned nbytes,
> +    int zeroing)
> {
>     unsigned i;
>     struct fuse_req *req = cs->req;
> @@ -874,6 +874,52 @@ static int fuse_copy_pages(struct fuse_copy_state *cs, unsigned
> nbytes,
> return 0;
> }
>
> +/* Take iov_offset as offset in iovec[0]. Iterate based on iovec[].iov_len */
> +static int fuse_copy_pages_for_iovec(struct fuse_copy_state *cs,
> +    unsigned nbytes, int zeroing)
> +{
>     unsigned i;
>     struct fuse_req *req = cs->req;
>     const struct iovec *iov = req->iovec;
>     unsigned iov_offset = req->iov_offset;
```

```
> +  
> + for (i = 0; i < req->num_pages && (nbytes || zeroing); i++) {  
> + int err;  
> + unsigned long user_addr = (unsigned long)iov->iov_base +  
> +     iov_offset;  
> + unsigned offset = user_addr & ~PAGE_MASK;  
> + unsigned count = min_t(size_t, PAGE_SIZE - offset,  
> +     iov->iov_len - iov_offset);
```

It would be much cleaner if we didn't have to deal with the original iovec here, but only offset and length relative to the page.

I understand that that would mean allocating an array for these. In the other thread I mentioned the possibility of allocating the page array. Instead of a page array, we could have an array of (pageptr, offset, len) which would simplify the whole thing.

Thanks,  
Miklos

---