Subject: Re: [PATCH 0/6] fuse: allocate req->pages[] dynamically
Posted by Maxim Patlasov on Fri, 14 Sep 2012 13:18:17 GMT
View Forum Message <> Reply to Message

Hi Miklos,

> "Maxim V. Patlasov" <mpatlasov@parallels.com> writes:
>
>> Hi Miklos,
>>
>> So far as no objections appeared, I'll go ahead and replace fuse req->page with
>> req->pagevec. It will point to an array of structs:
>>
>> struct page_vec {
>>  struct page  *pv_page;
>>  unsigned int  pv_len;
>>  unsigned int  pv_offset;
>> };
>>
>> instead of 'struct page *' as it used to be. It seems to be what you suggested
>> in one of your comments. Are you OK about it?
> Yes, that's exactly what I was thinking.

I've encountered a problem while trying to follow this approach.
fuse_get_user_pages() passes 'req->pages' to get_user_pages_fast().
get_user_pages_fast() and friends are not ready to get a pointer to
array of page_vec-s from fuse. I can see five ways to solve the problem:

1. Re-work get_user_pages_fast() and friends adding ability to fill
page_vec array. Too much work. Very ugly. I strongly dislike this way.

2. Allocate a temporary array of page pointers in fuse_get_user_pages()
to use as argument to get_user_pages_fast(). Ugly and may have
performance impact. I dislike this way too.

3. Call get_user_pages_fast() for each page (i.e. pass npages == 1 to
it). Easy to implement but may have performance impact. I'd refrain from it.

4. Keep req->pages 'as is', but add req->page_descs pointing to an array
of <offset, len> structures. Looks clumsy, straightforward, but quite
doable.

5. Use a hack in fuse_get_user_pages(): temporarily cast req->pagevecs
to 'struct page **pages', pass it get_user_pages_fast(), then transform
the content of req->pagevecs[] to have page pointers stored in proper
places (like 'for (i=...) pagevecs[i].pv_page = pages[i];').

What do you think?

Btw, thanks a lot for careful review of patch-set. I agree with your comments. Next version will have those findings fixed.

Thanks,
Maxim