Subject: Re: [PATCH 4/6] fuse: rework fuse_readpages()
Posted by Miklos Szeredi on Wed, 12 Sep 2012 16:41:40 GMT
View Forum Message <> Reply to Message

Maxim Patlasov <mpatlasov@parallels.com> writes:

> The patch uses 'nr_pages' argument of fuse_readpages() as the heuristics for
> number of page pointers to allocate.
>
> This can be improved further by taking in consideration fc->max_read and gaps
> between page indices, but it's not clear whetheher it's worthy or not.
> ---
> fs/fuse/file.c | 10 ++++++++--
> 1 files changed, 8 insertions(+), 2 deletions(-)
>
> diff --git a/fs/fuse/file.c b/fs/fuse/file.c
> index c811f3d..9a6dcc6 100644
> --- a/fs/fuse/file.c
> +++ b/fs/fuse/file.c
> @@ -641,6 +641,7 @@ struct fuse_fill_data {
>   struct fuse_req *req;
>   struct file *file;
>   struct inode *inode;
> + unsigned nr_pages;
> };
>
> static int fuse_readpages_fill(void *_data, struct page *page)
> @@ -656,8 +657,10 @@ static int fuse_readpages_fill(void *_data, struct page *page)
>   (req->num_pages == FUSE_MAX_PAGES_PER_REQ ||
>   (req->num_pages + 1) * PAGE_CACHE_SIZE > fc->max_read ||
>   req->pages[req->num_pages - 1]->index + 1 != page->index)) {
> + int nr_alloc = min_t(unsigned, data->nr_pages,
> +   FUSE_MAX_PAGES_PER_REQ);
>   fuse_send_readpages(req, data->file);
> - data->req = req = fuse_get_req_multipage(fc, FUSE_MAX_PAGES_PER_REQ);
> + data->req = req = fuse_get_req_multipage(fc, nr_alloc);
>   if (IS_ERR(req)) {
>   unlock_page(page);
>   return PTR_ERR(req);
> @@ -666,6 +669,7 @@ static int fuse_readpages_fill(void *_data, struct page *page)
>   page_cache_get(page);

Okay, this is where things get hairy and where we should do something like:

 if (WARN_ON(req->num_pages >= req->max_pages))
  return -EIO;

>   req->pages[req->num_pages] = page;
>   req->num_pages++;
> + data->nr_pages--;
>   return 0;
> }
>
> @@ -676,6 +680,7 @@ static int fuse_readpages(struct file *file, struct address_space
*mapping,
>   struct fuse_conn *fc = get_fuse_conn(inode);
>   struct fuse_fill_data data;
>   int err;
> + int nr_alloc = min_t(unsigned, nr_pages, FUSE_MAX_PAGES_PER_REQ);
>
>   err = -EIO;
>   if (is_bad_inode(inode))
> @@ -683,7 +688,8 @@ static int fuse_readpages(struct file *file, struct address_space
*mapping,
>
>   data.file = file;
>   data.inode = inode;
> - data.req = fuse_get_req_multipage(fc, FUSE_MAX_PAGES_PER_REQ);
> + data.req = fuse_get_req_multipage(fc, nr_alloc);
> + data.nr_pages = nr_pages;
>   err = PTR_ERR(data.req);
>   if (IS_ERR(data.req))
>   goto out;