

Hi,

Existing fuse implementation processes scatter-gather direct IO in suboptimal way: fuse_direct_IO passes iovec[] to fuse_loop_dio and the latter calls fuse_direct_read/write for each iovec from iovec[] array. Thus we have as many submitted fuse-requests as the number of elements in iovec[] array. This is pure waste of resources and affects performance negatively especially for the case of many small chunks (e.g. page-size) packed in one iovec[] array.

The patch-set amends situation in a natural way: let's simply pack as many iovec[] segments to every fuse-request as possible.

To estimate performance improvement I used slightly modified fusexmp over tmpfs (clearing O_DIRECT bit from fi->flags in xmp_open). The test opened a file with O_DIRECT, then called readv/writev in a loop. An iovec[] for readv/writev consisted of 32 segments of 4K each. The throughput on some commodity (rather feeble) server was (in MB/sec):

	original	/	patched
writev:	~107	/	~480
readv:	~114	/	~569

We're exploring possibility to use fuse for our own distributed storage implementation and big iovec[] arrays of many page-size chunks is typical use-case for device virtualization thread performing i/o on behalf of virtual-machine it serves.

Changed in v2:

- inline array of page pointers req->pages[] is replaced with dynamically allocated one; the number of elements is calculated a bit more intelligently than being equal to FUSE_MAX_PAGES_PER_REQ; this is done for the sake of memory economy.
- a dynamically allocated array of so-called 'page descriptors' - an offset in page plus the length of fragment - is added to fuse_req; this is done to simplify processing fuse requests covering several iov-s.

Changed in v3:

- used iov_iter in fuse_get_user_pages() and __fuse_direct_io()
- zeroed req->pages[] array on allocation
- a bunch of minor cleanup changes:
 - used unsigned for npages
 - freed req->pages[] and req->page_descs[] together
 - renamed fuse_get_ua() to fuse_get_user_addr()
 - renamed fuse_get_fr_sz() to fuse_get_user_size()

- simplified loop in fuse_page_descs_length_init()
- rebased on v3.7-rc2

Thanks,
Maxim

Maxim Patlasov (12):

- fuse: general infrastructure for pages[] of variable size
- fuse: categorize fuse_get_req()
- fuse: rework fuse_retrieve()
- fuse: rework fuse_readpages()
- fuse: rework fuse_perform_write()
- fuse: rework fuse_do_ioctl()
- fuse: add per-page descriptor <offset, length> to fuse_req
- fuse: use req->page_descs[] for arpages cases
- mm: minor cleanup of iov_iter_single_seg_count()
- fuse: pass iov[] to fuse_get_user_pages()
- fuse: optimize fuse_get_user_pages()
- fuse: optimize __fuse_direct_io()

```
fs/fuse/cuse.c | 3 -
fs/fuse/dev.c | 98 ++++++
fs/fuse/dir.c | 39 +++++
fs/fuse/file.c | 242 ++++++
fs/fuse/fuse_i.h | 47 ++++++
fs/fuse/inode.c | 6 +
include/linux/fs.h | 2
mm/filemap.c | 2
8 files changed, 292 insertions(+), 147 deletions(-)
```

--

Signature
