Subject: Re: [RFC][PATCH] EXT3: problem with page fault inside a transaction
Posted by Dmitriy Monakhov on Thu, 12 Oct 2006 07:53:56 GMT
View Forum Message <> Reply to Message

Andrew Morton <akpm@osdl.org> writes:

> On Thu, 12 Oct 2006 09:57:26 +0400
> Dmitriy Monakhov <dmonakhov@openvz.org> wrote:
>
>> While reading Andrew's generic_file_buffered_write patches i've remembered
>> one more EXT3 issue.journal_start() in prepare_write() causes different ranking
>> violations if copy_from_user() triggers a page fault. It could cause
>> GFP_FS allocation, re-entering into ext3 code possibly with a different
>> superblock and journal, ranking violation of journalling serialization
>> and mmap_sem and page lock and all other kinds of funny consequences.
>
> With the stuff Nick and I are looking at, we won't take pagefaults inside
> prepare_write()/commit_write() any more.
I'sorry may be i've missed something, but how cant you prevent this?

Let's look at generic_file_buffered_write:
force page fault
fault_in_pages_readable();

find and lock page
 __grab_cache_page()

allocate blocks.This may result in low memory condition
try_to_free_pages->shrink_caches() and etc.
a_ops->prepare_write()

can anyone guarantee that page fault hasn't happened by now ?
user space buffer swapped out, or became invalid.
filemap_copy_from_user()

>
>> Our customers complain about this issue.
>
> Really? How often?
I have't concrete statistic
>
> What on earth are they doing to trigger this? writev() without the 2.6.18
> writev() bugfix?