Subject: Re: [PATCH] remove BUG() in possible but rare condition
Posted by akpm on Wed, 11 Apr 2012 21:12:44 GMT
View Forum Message <> Reply to Message

On Wed, 11 Apr 2012 17:51:57 -0300
Glauber Costa <glommer@parallels.com> wrote:

> On 04/11/2012 05:26 PM, Andrew Morton wrote:
> >>
> >> >   failed:
> >> > - BUG();
> >> >   unlock_page(page);
> >> >   page_cache_release(page);
> >> >   return NULL;
> > Cute.
> >
> > AFAICT what happened was that in my April 2002 rewrite of this code I
> > put a non-fatal buffer_error() warning in that case to tell us that
> > something bad happened.
> >
> > Years later we removed the temporary buffer_error() and mistakenly
> > replaced that warning with a BUG(). Only it*can* happen.
> >
> > We can remove the BUG() and fix up callers, or we can pass retry=1 into
> > alloc_page_buffers(), so grow_dev_page() "cannot fail". Immortal
> > functions are a silly fiction, so we should remove the BUG() and fix up
> > callers.
> >
> Any particular caller you are concerned with ?

Didn't someone see a buggy caller in btrfs?

I'm thinking that we should retain some sort of assertion (a WARN_ON)
if the try_to_free_buffers() failed. This is a weird case which I
assume handles the situation where a blockdev's blocksize has changed.
The code tries to throw away the old wrongly-sized buffer_heads and to
then add new correctly-sized ones. If that discarding of buffers
fails then the kernel is in rather a mess.

It's quite possible that this code is never executed - we _should_ have
invalidated all the pagecache for that device when changing blocksize.
Or maybe it *is* executed, I dunno. It's one of those things which has
hung around for decades as code in other places has vastly changed.