

Re: [PATCH] NFS server does not update mtime on setattr request

Re: [PATCH] NFS server does not update mtime on setattr request

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-06/msg01910.html>

- *From:* Peter Staubach <staubach@xxxxxxxxxx>
 - *Date:* Wed, 07 Jun 2006 10:44:50 -0400
-

Trond Myklebust wrote:

On Tue, 2006-06-06 at 14:05 -0400, Peter Staubach wrote:

On the NFS client side, there was an optimization added which attempted to avoid an over the wire call if the size of the file was not going to change. This would be great, except for the side effect of the mtime on the file needing to change anyway. The solution is just to issue the over the wire call anyway, which, as a side effect, updates the mtime and ctime fields.

Vetoed!

The current code gets it quite right: if someone calls `open(O_TRUNC)`, then `may_open()` calls `do_truncate()` with the `ATTR_MTIME|ATTR_CTIME` flags set. That will cause the client to do the right thing `_regardless_` of the size optimisation.

You are right. My testing originally showed something different, but testing again shows the correct semantics.

I think that the conservative thing to do though, since an over the wire call is being made anyway, is to remove the optimization and retain the size change. The server is already going to have such a check in it anyway and issuing the `SETATTR` with the size change in it may reduce some races.

Re: [PATCH] NFS server does not update mtime on setattr request

Re: [PATCH] NFS server does not update mtime on setattr request

On the NFS server side, there was a change to the routine, `inode_setattr()`, which now relies upon the caller to set the `ATTR_MTIME` and `ATTR_CTIME` flags in `ia_valid` in addition to the `ATTR_SIZE`. Previously, this routine would force these bits on if the size of the file was not changing. Now, this routine relies upon the caller to specify all of the fields which need to be updated.

Also wrong.

This change causes the server to do entirely the wrong thing for `truncate()/ftruncate()` calls: in the SuSv3 spec, a call that fails to change the file length is supposed to leave the file entirely unchanged: that includes `mtime/ctime` as well as `suid/sgid` bits.

I saw that wording too and assumed what I think that you assumed. I assumed that that meant that if the new size is equal to the old size, then nothing should be changed. However, that does not seem to be how those words are to be interpreted. They are to be interpreted as "if the new length of the file can be successfully set, then the `mtime/ctime` should be changed". It does not matter if the new length was the same as the old length or not. Linux implements this semantic, as you pointed out above regarding the client side changes with the passing of `ATTR_MTIME|ATTR_CTIME` to `do_truncate()`. SunOS also implements this semantic.

Therefore, I believe that this patch should stand because it modifies the NFS server to do the right thing and indeed, to match the current semantics of `do_truncate()`. The older version of `do_truncate()` would have set `ATTR_MTIME` and `ATTR_CTIME` itself if the file size was not actually changing. Clients such as SunOS and older versions of Linux will require this change in order to properly interoperate with a new enough Linux NFS server.

Neil, can we get these changes integrated, please?

Thanx...

ps

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to `majordomo@xxxxxxxxxxxxxxxxx`

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>

Re: [PATCH] NFS server does not update mtime on setattr request