

Re: If not readdir() then what?

Re: If not readdir() then what?

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-04/msg02997.html>

- *From:* Trond Myklebust <trond.myklebust@xxxxxxxxxx>
 - *Date:* Mon, 09 Apr 2007 10:03:15 -0400
-

On Mon, 2007-04-09 at 09:19 -0400, Theodore Tso wrote:

On Mon, Apr 09, 2007 at 08:31:37AM -0400, Trond Myklebust wrote:

On Mon, 2007-04-09 at 13:09 +0200, Jörn Engel wrote:

That surely doesn't make life any easier for filesystem developers, I agree. From that point of view, all telldir cookies should end their life at closedir time. For "rm -r" it would be sufficient if the nfs client simply didn't seekdir at all. For "ls -lR", this would return duplicate dentries.

Please go read the NFS spec. The only thing an NFS client has in order to read a directory is a READDIR operation that in essence takes a filehandle and a cookie as its arguments. Unless the server is able to return the entire rest of the directory in one RPC reply, the client needs to send a second READDIR operation with a cookie from the previous READDIR operation. The server is expected to return cookies for `_each_` entry in the directory.

That is a protocol limitation, not a client limitation.

<Groan>

And after quickly checking RFC 3010, I see this limitation hasn't been lifted in NFSv4.

If we can come up with an interface that makes sense in the context of NFS, then we should be able to push it into a future minor revision of NFSv4. It is unfortunately looking too late to push it into v4.1, since the final drafts of the RFC are already circulating.

Re: If not readdir() then what?

Re: If not readdir() then what?

Speaking of which, right now ext3 doesn't know whether it's talking to an NFSv2 or NFS v3/v4 server, so it's always passing a 32-bit cookie. If NFSv3/v4 could use an explicit interface to request a 64-bit cookie, instead of just relying on the f_pos field in the file handle, we can reduce the chance of hash collisions when reading an ext3 directory significantly.

We could perhaps teach nfsd to open the file without the O_LARGEFILE attribute in the case of NFSv2?

If there are 2 or 3 directory entries that have a hash collision, would the NFS protocol allow the server to juggle things so that those 2-3 directory entries with the hash collision are sent back in a single readdir RPC reply? Is it acceptable/legal to have multiple entries in the same READDIR reply packet have the same cookie value?

Not really.

However on NFSv3 and v4 there is actually a mechanism for declaring that the existing set of cookies have expired and are no longer valid: you have an 8-byte opaque 'verifier' which is supplied by the server, and which is supposed to be returned by the client on every call to READDIR. If the server wants to change its cookie scheme, then it signals it to the client by changing its verifier, and returning an error whenever the client tries to use the old verifier. Upon receiving that error, the client is supposed to clear out all cached cookies, and read the directory in again from the start.

In practice, though, this sort of behaviour has to be managed carefully by the server. Forcing a client to re-read the entire contents of the directory doesn't really scale too well...

Note also that we would have to fix the client implementation. Nobody has bothered working on the code to handle verifier changes since there are no servers out there in the wild that use it.

Cheers
Trond

—

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: If not readdir() then what?