TODO

Version 2.14 December 21, 2018

A Partial List of Missing Features

Contributions are welcome. There are plenty of opportunities for visible, important contributions to this module. Here is a partial list of the known problems and missing features:

a. SMB3 (and SMB3.1.1) missing optional features:
   - multichannel (partially integrated), integration of multichannel with RDMA
   - directory leases (improved metadata caching). Currently only implemented for root dir
   - T10 copy offload ie "ODX" (copy chunk, and "Duplicate Extents" ioctl currently the only two server side copy mechanisms supported)

b. improved sparse file support (fiemap and SEEK_HOLE are implemented but additional features would be supportable by the protocol such as FALLOC_FL_COLLAPSE_RANGE and FALLOC_FL_INSERT_RANGE)

c. Directory entry caching relies on a 1 second timer, rather than using Directory Leases, currently only the root file handle is cached longer by leveraging Directory Leases

d. quota support (needs minor kernel change since quota calls otherwise won't make it to network filesystems or deviceless filesystems).

e. Additional use cases can be optimized to use "compounding" (e.g. open/query/close and open/setinfo/close) to reduce the number of roundtrips to the server and improve performance. Various cases (stat, statfs, create, unlink, mkdir, xattrs) already have been improved by using compounding but more can be done. In addition we could significantly reduce redundant opens by using deferred close (with handle caching leases) and better using reference counters on file handles.

f. Finish inotify support so kde and gnome file list windows will autorefresh (partially complete by Asser). Needs minor kernel vfs change to support removing D_NOTIFY on a file.

g. Add GUI tool to configure /proc/fs/cifs settings and for display of the CIFS statistics (started)

h. implement support for security and trusted categories of xattrs (requires minor protocol extension) to enable better support for SELINUX

i. Add support for tree connect contexts (see MS-SMB2) a new SMB3.1.1 protocol feature (may be especially useful for virtualization).

j. Create UID mapping facility so server UIDs can be mapped on a per mount or a per server basis to client UIDs or nobody if no mapping exists. Also better integration with winbind for resolving SID owners

k. Add tools to take advantage of more smb3 specific ioctls and features (passthrough ioctl/fsctl is now implemented in cifs.ko to allow sending various SMB3 fsctls and query info and set info calls directly from user space) Add tools to make setting various non-POSIX metadata attributes easier from tools (e.g. extending what was done in smb-info tool).

l. encrypted file support (currently the attribute showing the file is encrypted on the server is reported, but changing the attribute is not supported).

m. improved stats gathering tools (perhaps integration with nfsometer?) to extend and make easier to use what is currently in /proc/fs/cifs/Stats

n. Add support for claims based ACLs ("DAC")

o. mount helper GUI (to simplify the various configuration options on mount)

p. Expand support for witness protocol to allow for notification of share move, and server network adapter changes. Currently only notifications by the witness protocol for server move is supported by the Linux client.

q. Allow mount.cifs to be more verbose in reporting errors with dialect or unsupported feature errors. This would now be easier due to the implementation of the new mount API.

r. updating cifs documentation, and user guide.

s. Addressing bugs found by running a broader set of xfstests in standard file system xfstest suite.

t. split cifs and smb3 support into separate modules so legacy (and less secure) CIFS dialect can be disabled in environments that don't need it and simplify the code.

v. Additional testing of POSIX Extensions for SMB3.1.1

w. Add support for additional strong encryption types, and additional spnego authentication mechanisms (see MS-SMB2). GCM-256 is now partially implemented.

x. Finish support for SMB3.1.1 compression

Known Bugs

See https://bugzilla.samba.org - search on product "CifsVFS" for current bug list. Also check http://bugzilla.kernel.org (Product = File System, Component = CIFS)

1. existing symbolic links (Windows reparse points) are recognized but can not be created remotely. They are implemented for Samba and those that support the CIFS Unix extensions, although earlier versions of Samba overly restrict the pathnames.
2. follow_link and readdir code does not follow dfs junctions but recognizes them

Misc testing to do

1. check out max path names and max path name components against various server types. Try nested symlinks (8 deep). Return max path name in stat -f information
2. Improve xfstest's cifs/smb3 enablement and adapt xfstests where needed to test cifs/smb3 better
3. Additional performance testing and optimization using iozone and similar - there are some easy changes that can be done to parallelize sequential writes, and when signing is disabled to request larger read sizes (larger than negotiated size) and send larger write sizes to modern servers.
4. More exhaustively test against less common servers
5. Continue to extend the smb3 "buildbot" which does automated xfstesting against Windows, Samba and Azure currently - to add additional tests and to allow the buildbot to execute the tests faster. The URL for the buildbot is: http://smb3-test-rhel-75.southcentralus.cloudapp.azure.com
6. Address various coverity warnings (most are not bugs per-se, but the more warnings are addressed, the easier it is to spot real problems that static analyzers will point out in the future).