Improving access to Remote Files from Linux: review of recent progress in the SMB3.1.1 client

Presented by Steve French
Principal Software Engineer
Microsoft Azure Storage
This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds.

Other company, product, and service names may be trademarks or service marks of others.
Who am I?

- Steve French smfrench@gmail.com
- Author and maintainer of Linux cifs vfs (for accessing Samba, Azure, Windows and various SMB3/CIFS based NAS appliances)
- Co-maintainer of the kernel server (ksmbd)
- Member of the Samba team (co-creator of the “net” utility)
- Coauthor of SNIA CIFS Technical Reference, former SNIA CIFS Working Group chair
- Principal Software Engineer, Azure Storage: Microsoft
Outline

- Overview of Linux FS activity
- Recent ksmbd (server) improvements
- Recent client improvements
- Coming soon … what to look forward to
- Testing improvements
Linux Kernel: A year ago and now ...

- Now: 6.9-rc4 … then: 6.3-rc7
  But it has kept the same unusual name “Hurr durr I’m a ninja sloth”
- The kernel project is HUGE
- 87,735 changesets over that period
- 43,168 files changed
- 3,928,505 insertions (increased)
- 1,539,370 deletions
LSF/MM/eBPF summit

- Picture from May 2023 (looking forward to next, in Salt Lake City soon)
- Many excellent Linux storage developers working together
Some Linux FS topics of interest discussed recently

- Testing … testing … and more automated testing … (e.g. kdevops)
- Folios, netfs, iov_iter, variable size pages, and the redesign of page cache and offline (fscache), io_uring (async i/o improvements)
- Improvements to statx and fsinfo and to inotify/fanotify
- Idmapped mounts, fine grained timestamps
- Leveraging eBPF (not just dynamic tracing)
- Extending in kernel encryption: TLS handshake (for NFS) and QUIC (SMB3.1.1 and other)
- Shift to cloud
- Better support for faster storage (NVME) and net (RDMA/smbdirect)
Linux Filesystems Activity over past year (since 6.3-rc7)

- 8966 filesystems changesets (6.1% of total kernel changesets, carefully watched part of kernel, and FS activity is up 90%!)
- Linux kernel fs are 1.2 million lines of code (measured this week)
  - Some big advancements including
    - Bcachefs added, “ntfs classic” removed (in favor of NTFS3)
    - Ksmbd no longer “experimental”
Most Active Linux Filesystems over the past year

- VFS (mapping layer) 567 changesets (activity up)
- The top filesystems and VFS dominate the activity
- Most active are bcachefs 3425 (!), BTRFS 1017 (down), XFS 609 (up a lot), ext4 402 (down)
- Most active non-local fs is SMB3.1.1 (cifs.ko) 333 (flat)
- Then NFSD (server) 276 (up) and NFS (client) 180 (down)
  - cifs.ko also had many more lines changed. It has been a VERY active year for cifs.ko
- Other: F2FS (272, down) GFS2 (179), ksmbd (153, flat), erofs (124), ceph (124), ntfs3 (116), ceph (105)
SMB3.1.1 Activity was strong this year

- cifs.ko activity was strong, 333 changesets
  - cifs is 63KLOC kernel code (not counting user space utilities)

- ksmbd activity flat
  - 26KLOC kernel code, 453 changesets since its introduction

- Samba server (userspace) is over 3.5 million lines of code (orders of magnitude bigger than the kernel smbd server or any of the NFS servers) and is even more active
Repeating our Goals for SMB3.1.1 on Linux

• Be the fastest, most secure general-purpose way to access file data, whether in the cloud or on premises or virtualized
  • Improve directory lease support
  • Keep improving compounding, multichannel
• Support more Linux/POSIX features – so apps don’t know they run on SMB3 mounts (vs. local)
  • SMB3.1.1 POSIX extensions, new FSCTLs
  • Use xfstests to locate new features to emulate
• As Linux evolves, quickly add features to Linux kernel client and Samba and ksmbd
  • More test automation and keep adding more tests
Linux File API still growing (5 recently)

e.g.: “folios” and io_uring continue to improve as well, and lots of changes to internal APIs (netfs, fscache ...)

<table>
<thead>
<tr>
<th>Syscall name</th>
<th>Kernel Version introduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>fchmodat2</td>
<td>6.6-rc1</td>
</tr>
<tr>
<td>listmount (query children of mount)</td>
<td>6.8-rc1</td>
</tr>
<tr>
<td>statmount (query attributes of mnt)</td>
<td>6.8-rc1</td>
</tr>
<tr>
<td>And new fs ioctls:</td>
<td></td>
</tr>
<tr>
<td>FS_IOC_GETUUID</td>
<td>6.9-rc1</td>
</tr>
<tr>
<td>FS_IOC_GETFSSYSFSPATH</td>
<td>6.9-rc1</td>
</tr>
</tbody>
</table>
One of the strengths of SMB3.1.1 is broad interop testing

• In-person plugfests are back!
• SMB3.1.1 plugfest restarted, colocated with SDC last two years
  − And will be again this year!
• Many exciting things being tested
Progress and Status update for Linux Kernel Server (ksmbd)

See Namjae Jeon (linkinjeon@kernel.org) talk, also at SambaXP
Improving the network stack: progress on QUIC and SMB3.1.1 for Linux

Very exciting progress in the experimental Linux kernel QUIC driver
See my earlier SambaXP talk
Recent improvements in the kernel client

(cifs.ko)
DFS ie “The Global Namespace” - improvements

- DFS Interlink support (DFS link that points to another namespace)
- Sharing of DFS connections between mounts
- Reparse mount points - as cross fs boundaries on server - create submounts with noserverino
- Reconnection improved for DFS use cases
- Now handles cases where nested links have some bad targets (and can handle some scenarios which other clients can't)
- Thank you Paulo!
POSIX Extensions

- SMB3.1.1 POSIX/Linux extensions supported by ksmbd and Samba
- Details presented in earlier conferences
- Improved with recent enhancements for fifo, block, char, symlinks
Multichannel improvements

• Multiple Reconnect and Perf improvements including improved channel allocation for SMB3.1.1 requests (thank you Shyam Prasad)
• Soon will be enabled by default (when server supports multiple interfaces or RSS)
Directory and file caching improvements

- New module parm “dir_cache_timeout”
- Mount parms “max_cached_dirs=” and “handletimeout=“

```
parm: cifs_max_pending:Simultaneous requests to server for CIFS/SMB1 dialect (N/A for SMB3) Default: 52767
parm: dir_cache_timeout:Number of seconds to cache directory contents for which we have a lease. Default: 30
parm: slow_rsp_threshold:Amount of time (in seconds) to wait before logging that a response is delayed. Default: 30
parm: enable_oplocks:Enable or disable oplocks. Default: y/N/1 (bool)
parm: enable_gcm_256:Enable requesting strongest (256 bit) GCM encryption. Default: n/N/0 (bool)
parm: require_gcm_256:Require strongest (256 bit) GCM encryption. Default: n/N/0 (bool)
parm: enable_negotiate_signing:Enable negotiating packet signing algorithm with server. Default: n/N/0 (bool)
parm: disable_legacy_dialects:To improve security it may be helpful to restrict the ability to override the (CIFS/SMB1 and SMB2) since vers=1.0 (CIFS/SMB1) and vers=2.0 are weaker and less secure. Default: n/N/0 (bool)
```

```
[root@fedora29 ~]# ls /proc/fs/cifs
cifsFYI DebugData dfscache LinuxExtensionsEnabled LookupCacheEnabled mount_params open_files SecurityFlags Stats traceSMB
[root@fedora29 ~]# cat /proc/fs/cifs/lookupcacheEnabled
1
[root@fedora29 ~]# ls /sys/module/cifs/parameters/
CIFSMaxBufSize cifs_min_rcv dir_cache_timeout enable_gcm_256 enable_oplocks slow_rsp_threshold
CIFS_max_pending cifs_min_small disable_legacy_dialects enable_negotiate_signing require_gcm_256
[root@fedora29 ~]# 
```
Improvements to special file support

- Much broader support for different types of special files (exported different ways by servers). Thank you Paulo!
- New mount parm “reparse=” allows you to choose whether reparse points (that encode special files like FIFOs, symlinks, block and char devices) should default to “wsl” format or the older Windows “nfs” server’s format.
Password rotation (key rotation) becoming common requirement due to security challenges e.g.
password rotation (now can update on active mounts)

- If you had two mounts to the same server, one ("/mnt1") with key 1 and one with key 2 ("/mnt2"), but then changed the first password ("rotate key") but not the second then the first mount would be inaccessible (and before this change require "umount /mnt1" then "mount" again with new password) and DebugData would show DISCONNECTED (and password no longer valid)
password rotation (now can update on active mounts)

- Remount with new password (now in 6.9 kernel but also already backported to stable 6.8, 6.6 etc ...)

   ("mount -t cifs //server/share /mnt1 -o remount,password=newpassword")

```
[root@fedora29 ~]# ./remount1
[root@fedora29 ~]# stat /mnt1
  File: /mnt1
  Size: 0   Blocks: 0   IO Block: 1048576 directory
  Device: 31h/49d   Inode: 7957636952732887613   Links: 2
  Access: (0777/drwxrwxrwx)   Uid: ( 0/ root)   Gid: ( 0/ root)
  Modify: 2024-04-09 15:50:30.899129800 -0500
  Change: 2024-04-09 15:50:30.899129800 -0500
  Birth: -
[root@fedora29 ~]#
```
password rotation (now can update on active mounts)

- But remount of working mount (“/mnt2” in this example) with new (changed) password is not permitted since session not disconnected. Only allowed if server returned EACCESS or EKEYEXPIRED etc. and session is down, so we added a way to handle this with alt password (new mount parm “password2=”)
- “mount -t cifs //server/share /mnt2 -o remount,password2=newpassword"

[1158.050089] CIFS: VFS: \smfptestdiag102.file.core.windows.net Send error in SessSetup = -13
[1160.094753] CIFS: Status code returned 0xc000006d STATUS_LOGON_FAILURE
[1160.094800] CIFS: VFS: \smfptestdiag102.file.core.windows.net Send error in SessSetup = -13
[root@fedora29 ~]# ./remount2
mount error(22): Invalid argument
Refer to the mount.cifs(8) manual page (e.g. man mount.cifs) and kernel log messages (dmesg)
[root@fedora29 ~]# dmesg
[1343.658694] can not change password of active session during remount
[1343.658704] CIFS: VFS: can not change password of active session during remount
[root@fedora29 ~]# 1
• Until these recent changes:
  • Changing a password on a mount required “unmount” then “mount” again for Linux.
    • Which isn’t always practical if the app has open files on the mount or is hard to shutdown
  • Added ability to change the password on remount (which doesn’t require apps to exit or files to close). Now can do:
    • mount -t cifs //srv/share /target -o remount, username=<user>,password=<newpassword>
    • Previously this would have returned “Invalid argument”

• Also added new mount option (also works on remount) “password2=“ so can have two passwords saved in session structure (and if reconnect fails with first one with access denied or key expired then is switched with “password2”)

• Note that the “cifscreds” approach (used for cifs multiuser mounts in the non-krb5 case) which leverages the kernel keyring to save passwords instead of the session structure could not be used because the process doing the reconnect is not a child process of the mount process (or the process which launched cifscreds). The remount change and the “password2=“ approach use the normal session structure (in kernel memory but not in the kernel keyring) to store the password2 (where the “password” field is stored)

• The ability to remount with updated password is in Linux kernel now (6.9-rc1) and has been backported to 6.8 and 6.6 kernels (among others) already, and is expected to be broadly updated in distros over the coming months. The new mount option (password2) is in Linux starting with 6.9-rc4 but will be backported soon to some older kernels, and will request that it be picked up by other distros as well over the coming months.
Folios/Netfs/MM caching improvements

- Kernel multipage folios (supported by local fs and AFS so far, support pending for cifs.ko)
  - https://lwn.net/Articles/937239/
  - Also nice to see move to larger block sizes: https://lwn.net/Articles/945646/

- Merged so far (Thanks to David Howells!):
  - Move VM, pagecache & folios out of individual filesystems
  - Share code between 9p, afs, cifs, ceph
  - Filesystem just supplies read & write ops to/from iterators
  - Interleaving reads between local cache, servers
Currently being worked on:

- David Howells has posted a large patch set for eval for 6.10 kernel
  - See the vfs.git vfs.netfs branch on kernel.org e.g. (in linux-next)
- Better handling retries, and handling credit renegotiation
- Support for write-through caching
- Support for vectored writes
- Support for content crypto (in devel)
6.3 kernel (April 23rd, 2023) (cifs module 2.42) very active release!

- Kernel idmapping improvements
- Improvements to use folios (better mm integration and cached writes)
- RDMA (smbdirect) improvements (thanks Metze and David)
- Many multichannel improvements (including using least loaded channel for sending I/O, and improvements for reconnect). Thanks Shyam!
- Various DFS fixes (thanks Paulo)
- Lower default deferred close timeout (from 5 to 1 second)
6.4 kernel (June 2023) (cifs.ko version: 2.43)

- Important deferred close (lease break corner case) fixes
- Reconnect and DFS fixes
- Important crediting improvements, and debugging improvements
- Perf improvement for large reads (rasize now defaults to 2x max rsize)
- At Linus’s suggestion source directories renamed
  - More intuitive
  - fs/cifs → fs/smb/client
  - fs/smbfs_common → fs/smb/common
  - fs/ksmbd → fs/smb/server
6.5 kernel (August 2023) (cifs.ko version: 2.44)

- Deferred close perf improvement (avoid unneeded lease break acks)
- Crediting (flow control) improvements to avoid low credit perf issue
- Reconnect and DFS fixes
- Fix null auth (sec=none) regression
- Allow dumping decryption keys (eg for reading network traces) via directory name, not just file.
- Directory caching improvement (‘laundromat’ thread added to clean up)
- Display client GUID and network namespace in /proc/fs/cifs/DebugData (can help with debugging containers e.g.)
6.6 kernel (October 2023) (cifs.ko version: 2.45)

- DFS (global namespace) fixes
- Improvements handling reparse points.
- Perf improvement for querying reparse point symlinks
- Reconnect improvement (write retry with channel sequence number)
- Add new mount parm "max\_cached\_dirs" to control how many directories are cached when server supports directory leases.
- Add new module parm /sys/module/cifs/parameters/dir\_cache\_timeout to control length of time a directory is cached with directory leases
6.7 kernel (Jan 2024) (cifs.ko version: 2.46)

- Reconnect and multichannel improvements
- Debugging improvements:
  - including client version in NTLM auth
  - loctl “CIFS_IOC_GET_TCON_INFO” to get sessid & tcon id of mnt
- Fallocate improvements for insert and zero range
- Fixes for metadata on server side symlinks (when reparse points)
- Very interesting OOB fixes found by Dr. Robert Morris with fuzzing
- netfs/folios (cached read/write) optimizations
- Compounding improvements
- Special file handling improvements (fifos, char/block, symlinks, and perf improvements for reparse points)
- Stats now show timestamp of when begun
- New mount option “retrans” (how often to retry on EAGAIN errors) and retry improvements
- Multichannel improvements
6.9 kernel (expected May 2024) (cifs.ko vers: 2.48)

- Allow updating password on remount (if session down due to password change on server, e.g. password rotation is getting common now)
- Allow alt password ("password2=") on mount/remount to better handle key rotation e.g. plan for changed password rolled out over server(s)
- Metadata caching improvements
- Add retry for some types of failed close operations
- Delete of open file improvements (do not defer close)
- Reparse mount option and add support for WSL reparse points
- Prep support for SMB3.1.1 compression over the wire (expect in 6.10)
Recent Debugging Improvements

- More dynamic trace points added
- Ability to query the session id and tid for a mount
- Start time for stats visible (and reset when echo > /proc/fs/cifs/Stats)
- DebugData improvements:
  - Network namespace now shown for session (e.g. can help debug container reconnects)
  - Server capabilities now displayed
  - ClientGUID displayed
  - Unknown link (network) speed now shown properly
From 102 dynamic trace points last year at SambaXP ...
To 118 now (16 more eBPF trace points added)
Planned and in-progress Improvements (To-dos)

(cifs.ko)
What improvements to expect in next few releases?

- Prototype of SMB3.1.1 over QUIC (new encrypted network transport)
- Reenabling support for swapfile over SMB3.1.1 mounts (was delayed due to MM/folios/netfs changes)
- Support for creating with O_TMPFILE
- More testing of SMB3.1.1 mounts to Samba with the POSIX extensions
- Support for new auth mechanisms (e.g. local KDC, IAKERB)
- Support for SMB3.1.1 compression (some of Enzo’s series has already been merged into mainline). See slides 48-55 of SNIA-SDC23-French-Advancing-Accessing-Remote-Files-SMB3_2.pdf
- Better use of parent lease key to avoid unneeded lease breaks
Anyone interested in the fastest transports ...?

- Smbdirect (RDMA abstraction layer) is very powerful
- This could be available as a common code layer used by Samba, ksmbsd server and cifs.ko client and other applications
- Tom Talpey and Metze had some great ideas (some described in my 2023 SDC presentation) on how to improve smbdirect (RDMA) for SMB3.1.1 mounts
- Also important is improving test automation for RDMA (smbdirect)
  - Ideas welcome
What about new Linux syscalls and Linux extensions

- Can consider new FSCTLs and infolevels to deal with the new Linux features, syscalls, fallocate flags etc.
- Xfstests help discover new Linux features that are missing
Work on SMB3.1.1 compression over network

- Can now negotiate compression algorithms with the server
- Will be big help in long latency networks
- Enzo is working on additional patches. See my SDC2023 presentation pages 48-52 for details
Optimizing common cases: improve “ls” (readdir)

- Current behavior wastes a roundtrip for small directories
  - Extra Find Request (frames 6 & 7) could be compounded
Optimizing common cases: improve “ls -l” when symlinks

Current behavior with “mfsymlinks” (note repeated read e.g.)
Optimizing common cases: improve “ls” when symlinks

Current behavior with “mfsymlinks”
Optimizing common cases: improve creating symlinks

Current behavior with "mfsymlinks"

```
mfrench@smfrench-ThinkPad-P52:~$ cd /mnt2/dir1
mfrench@smfrench-ThinkPad-P52:/mnt2/dir1$ ln -s file symlinktofile
```
Recent improvements in the user space tools

(cifs-utils)
Cifs-utils improvements

- Various important man page updates (e.g. new mount parms)
- Various minor fixes
- New feature to use CLDAP to find closest site (thanks to David Voit)
  - Being reviewed now
- Additional debugging tools are being considered, as well as additional tools for change notify and making updating passwords easier
Testing Improvements

Test ... test ... test ...
Additional tests are encouraged (generic or smb specific)

- Xfstests are the standard Linux filesystem functional tests
- Over last 9 months added 46 to the main “cifs-testing” regression testing group (288 tests run on every checkin from this group)
- Various server specific groups have added even more
  - Azure SMB3.1.1 multichannel: added 20 more tests, now 153 tests
    - e.g. http://smb311-linux-testing.southcentralus.cloudapp.azure.com/#/builders/5/builds/91
  - Ksmbd (Linux kernel server target) added 77 more tests, now includes 221 tests
    - e.g. http://smb311-linux-testing.southcentralus.cloudapp.azure.com/#/builders/10/builds/52
- Detailed wiki pages on wiki.samba.org go through how to setup xfstests with cifs.ko, and what features need to be added to enable more tests (tests that currently skip or fail so aren’t run in the ‘buildbot’)
Thank you for your time

- Future is very bright!
Additional Resources to Explore for SMB3 and Linux


• In particular MS-SMB2.pdf at https://msdn.microsoft.com/en-us/library/cc246482.aspx

https://wiki.samba.org/index.php/Xfstesting-cifs and test results

• http://smb311-linux-testing.southcentralus.cloudapp.azure.com/#/

Linux CIFS client https://wiki.samba.org/index.php/LinuxCIFS

Samba-technical mailing list

And various presentations at http://www.sambaxp.org and Microsoft Learn (learn.microsoft.com) and of course SNIA … http://www.snia.org/events/storage-developer

And the code:

• https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/fs/smb

• For pending changes, soon to go into upstream kernel see:
  − https://git.samba.org/?p=sfrench/cifs-2.6.git;a=shortlog;h=refs/heads/for-next

• Kernel server code: https://git.samba.org/ksmbd.git/?p=ksmbd.git (ksmbd-for-next branch)