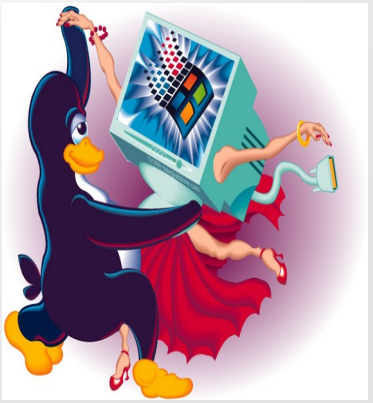
SMB3 Improvements to Linux: Summary of client status

Steve French
Principal Software Engineer
Azure Storage - Microsoft



Legal Statement

- 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, Windows and various SMB3/CIFS based NAS appliances)
- Also wrote initial SMB2 kernel client prototype
- Member of the Samba team, coauthor of SNIA CIFS Technical Reference,former SNIA CIFS Working Group chair
- Principal Software Engineer, Azure Storage: Microsoft

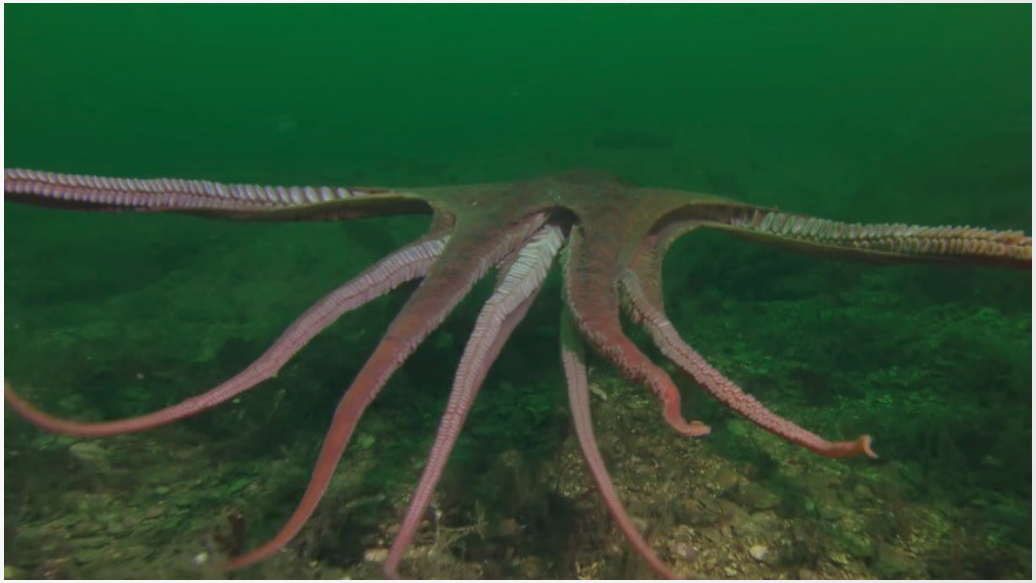
Outline

- General Linux File System Status – Linux FS and VFS Activity
- What are the goals?
- What's New – Key Feature Status
  - Kernel client
  - User space tools (cifs-utils)
- Looking forward, features under development
- Common Configuration Suggestions
- Testing Improvements

A year ago … and now … kernel (including SMB3 client cifs.ko) improving

- Now Linux 5.12 “Frozen Wasteland”

Then Linux 5.7-rc7: “Kleptomaniac Octopus”

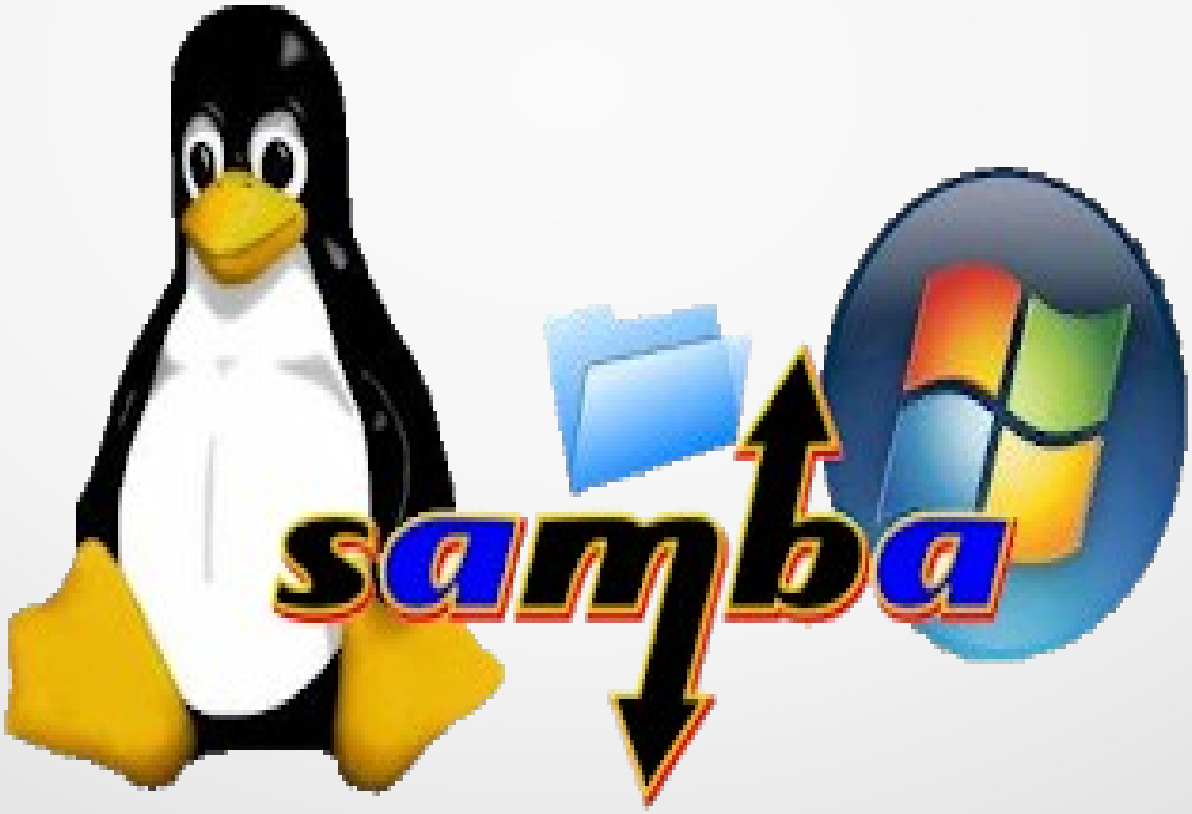


Similar topics continuing to drive some of the FS development activity

- Matthew Wilcox patches for Xarray and netfs
- Dave Howell's patches to fscache
- Improved containers support
- Optional use of QUIC transport for various network filesystems
- Allow querying additional information about mounted fs
- Stronger, faster encryption
- Better support for faster storage (NVME, RDMA)
- More improvements around async i/o and io_uring
- Shift to Cloud (longer latencies, object & file coexisting)

What about the server (on Linux)?

- Samba server is great (and huge, and full function)
  - See multiple different talks at this conference
- But now we also have a kernel server, ksmbd!
  - See Namjae's talk at this conference



No in person Samba Team meeting and testing in Redmond or at SNIA in the Fall (as in years past) but virtual events like this have been helpful ...Looking forward to in person ...

Most Active Linux Filesystems this year

- 5707 kernel filesystem changesets last year (since Linux 5.7-rc7) (up)
  - FS activity: 6.5% of overall kernel changes, down slightly as % of activity
  - Kernel is huge (> 21.6 million lines of code, measured yesterday)
- There are many Linux file systems (>60), but six (and the VFS layer itself) drive ¾ of activity (btrfs, xfs, ext4 and cifs are the most active)
  - File systems represent almost 5% of kernel source code (986KLOC) but are among the most carefully watched areas
- cifs.ko (cifs/smb3 client) activity is strong
  - #4 most active of all fs with 312 changesets!
  - 58.5KLOC, up >6% (not counting user space cifs-utils which are now 12% larger at 13KLOC, and samba tools which are larger)
- At current pace kernel server will also be one of most active components

Linux File System Change Detail for past year (5.7-rc7 to now)

- BTRFS 1065 changesets (up a lot)
- VFS (overall fs mapping layer and common functions) 1429 (up significantly)
- XFS 623 (flat)
- CIFS/SMB2/SMB3 client 312 (since 4.18 kernel activity has gone way up)
- NFS client 202 (down a lot)
- Others: F2FS 302 (up), EXT4 314 (up), Ceph 108 (way down), GFS2 178 (up), AFS 109, OCFS2 51 ...
- NFS server 310 (up). Linux NFS server **MUCH** smaller than CIFS or Samba
- NB: Samba is as active as all Linux file systems put together - broader in scope (by a lot) and also is user space not kernel. **98x larger than the NFS server in Linux! Samba now 3.4 million lines of code** (measured today). And the new kernel server (ksmbd) is also more active than NFS in activity – a very exciting time!

Linux filesystems are not easy! Responsible for more than 200 of 850 syscalls. +4 since last year

Syscall name	Kernel Version introduced
epoll_pwait2	5.11
mount_setattr	5.12
faccessat2	5.8
close_range	5.9

Goals: FAST/EASY/TRANSPARENT!

- Repeating an older slide about goals of SMB3.1.1:
  - Fastest, most secure general purpose way to access file data, whether cloud or on premises or virtualized
  - Implement all reasonable Linux/POSIX features - so apps don’t know they run on SMB3 mounts (vs. local)
  - As Linux evolves, and needs new features, quickly add to Linux kernel client and Samba and ksmbd

Examples of Great Progress we talked about last year!

- Reminders of some amazing progress …
- A few examples

“modefromsid” mount option

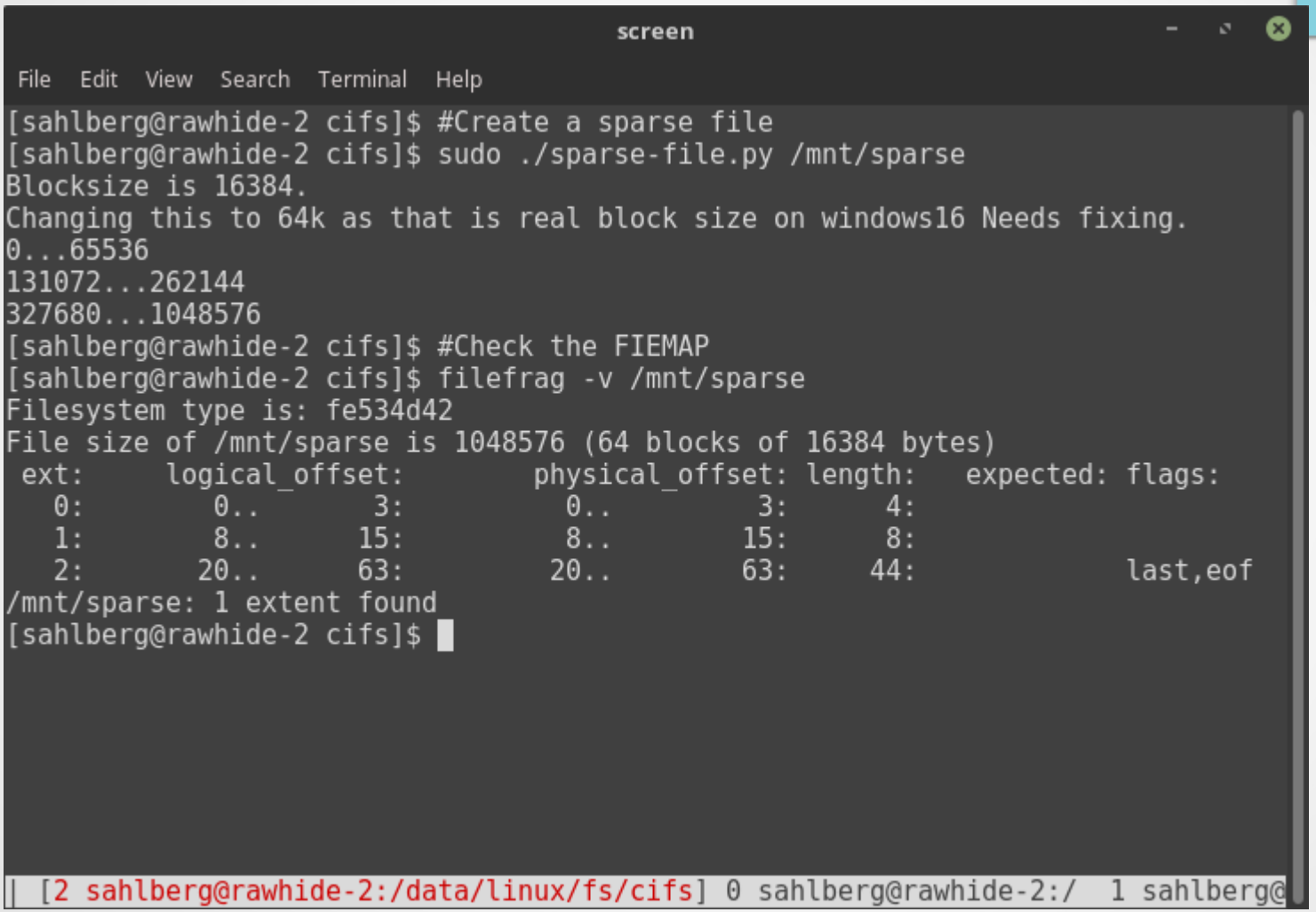
- Useful for “nfs style” security where the client’s permission evaluation matters most
- Stored in ACE with ‘special SID’ unenforced by server
- Creating files with all 4096 mode combinations works

- -rw--w-rwx 1 root root 14 May 13 00:25 407file
- -rwsrwS--T 1 root root 0 May 13 00:26 4080file
- -rwsrwS--t 1 root root 0 May 13 00:26 4081file
- -rwsrwS-wT 1 root root 14 May 13 00:26 4082file
- -rwsrwS-wt 1 root root 14 May 13 00:26 4083file
- -rwsrwSr-T 1 root root 0 May 13 00:26 4084file
- -rwsrwSr-t 1 root root 0 May 13 00:26 4085file
- -rwsrwSrwT 1 root root 14 May 13 00:26 4086file
- -rwsrwSrwt 1 root root 14 May 13 00:26 4087file
- -rwsrws--T 1 root root 0 May 13 00:26 4088file
- -rwsrws--t 1 root root 0 May 13 00:26 4089file
- -rw--wx--- 1 root root 0 May 13 00:25 408file
- -rwsrws-wT 1 root root 14 May 13 00:26 4090file
- -rwsrws-wt 1 root root 14 May 13 00:26 4091file
- -rwsrwsr-T 1 root root 0 May 13 00:26 4092file
- -rwsrwsr-t 1 root root 0 May 13 00:26 4093file
- -rwsrwsrwT 1 root root 14 May 13 00:26 4094file
- -rwsrwsrwt 1 root root 14 May 13 00:26 4095file
- -rw--wx--x 1 root root 0 May 13 00:25 409file
- ----r-x--- 1 root root 0 May 13 00:25   40file
- -rw--wx-w- 1 root root 14 May 13 00:25 410file
- ...

Multichannel (in 5.5 kernel)

- Thank you Aurelien!
- Expected to be a big performance win …
- Big large I/O performance improvement in 5.8 kernel (up to 5x faster in my testing) and much more stable in 5.13-rc

Sparse File Support (and other network fs can't do this)! Thank your Ronnie!



GCM Fast

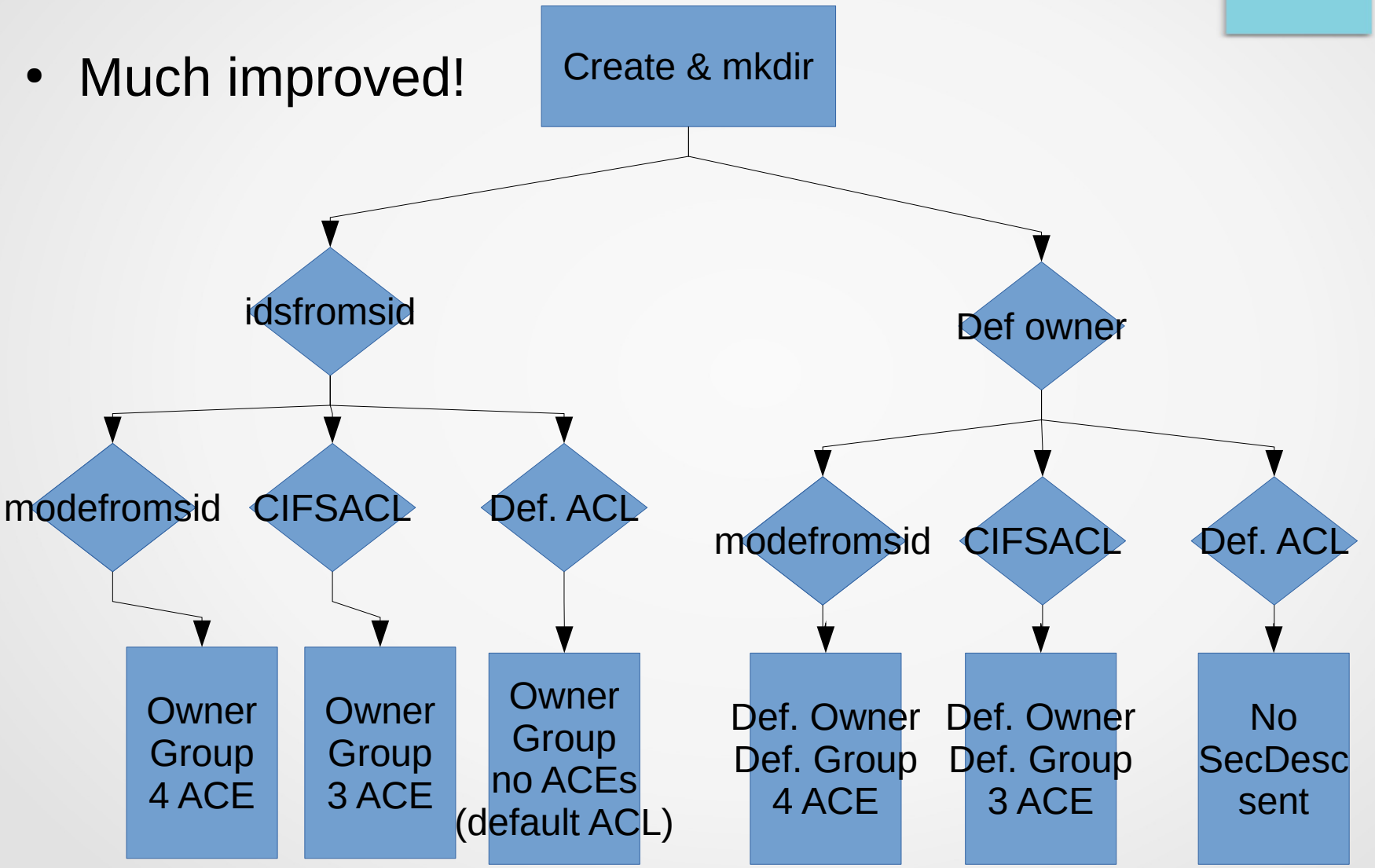
- Can more than double perf to Azure in some configurations in recent testing I tried
- Also speeds up encrypted mounts to Windows and Samba
- In 5.3 kernel

Examples of Great Progress more recently!

- This has been a great year for cifs.ko...

Remember the security models: idsfromsid, modefromsid, cifsacl

- Much improved!



What about Security Improvements?

- Four key parts:
  - Authentication: improvements to Kerberos mounts (thanks Shyam) and an enhancement to NTLMSSP security in progress (expected soon)
  - What permissions you have. The 3 security models:
    - The two non default options:. “multiuser, server enforced” (ie cifsacl) vs. “client enforced” (modefromsid,idsfromsid) are greatly improved
  - Who you are: additional options possible now with “idsfromsid”
  - Encryption: with addition of GCM256 now have option of strongest encryption (and GCM encryption is really fast too). And when QUIC is added we will have even more choices for encryption
- And don’t forget managing access control and auditing: much improved ability to query and set this information through our tooling (cifs-utils)

AES-GCM-256 (strongest encryption)

- Negotiates it with server by default now if server requires it
- Client can require (force) AES-GCM-256 as well if new module parm “require_gcm_256” set

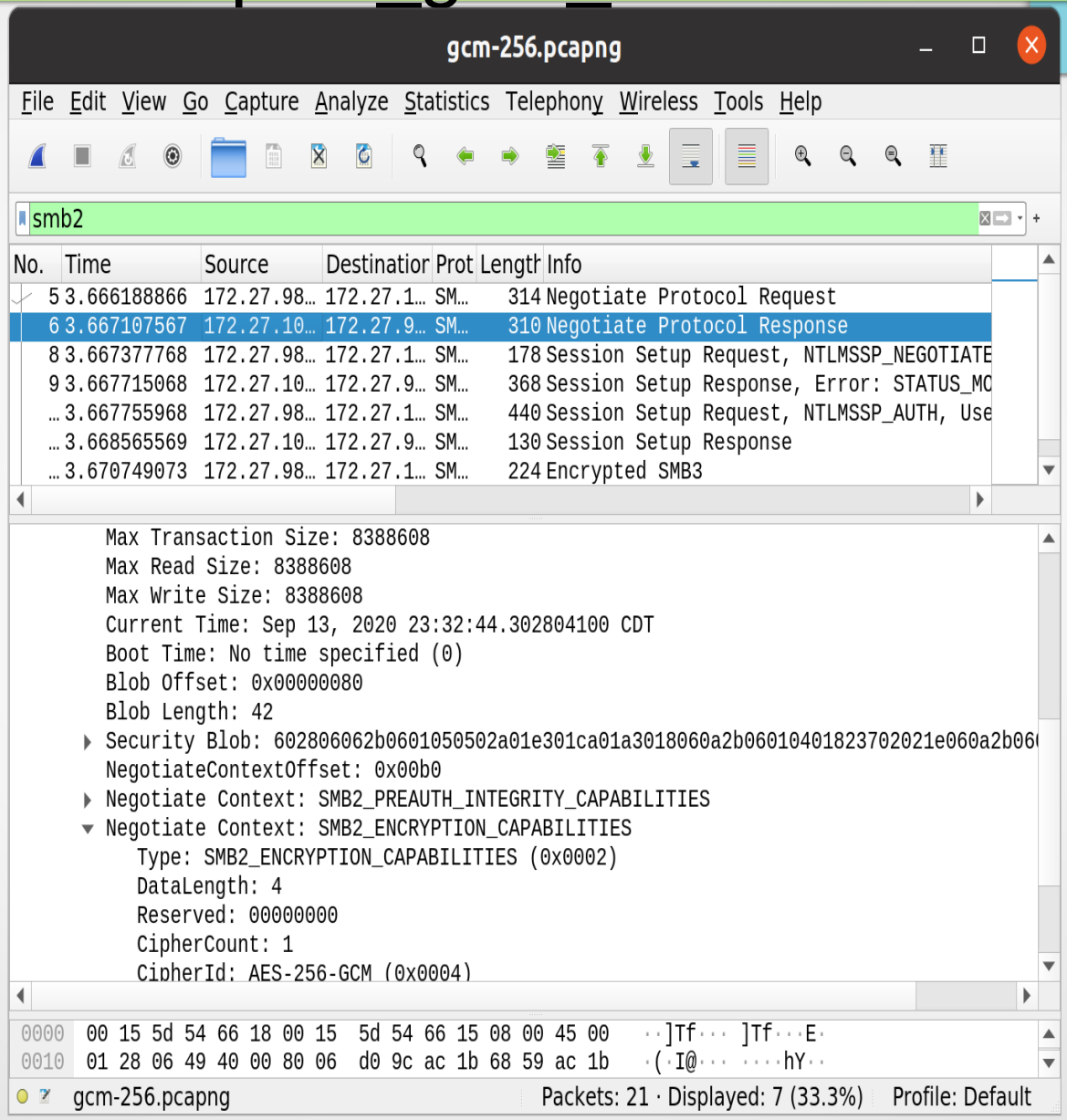
```
root@smfrench-Virtual-Machine:~# mount | grep cifs
//172.25.223.247/test on /mnt type cifs (rw,relatime,vers=3.1.1,cache=strict,username=testuser,uid=0,nof
orceuid,gid=0,noforcegid,addr=172.25.223.247,file_mode=0755,dir_mode=0755,seal,soft,nounix,serverino,map
posix,noperm,rsize=4194304,wsize=4194304,bsize=1048576,echo_interval=60,actimeo=1)
root@smfrench-Virtual-Machine:~# cat /sys/module/cifs/parameters/require_gcm_256
Y
root@smfrench-Virtual-Machine:~# cat /sys/module/cifs/parameters/enable_gcm_256
Y
root@smfrench-Virtual-Machine:~# cat /proc/fs/cifs/DebugData | grep Encrypted -C3

        Shares:
        0) IPC: \\172.25.223.247\IPC$ Mounts: 1 DevInfo: 0x0 Attributes: 0x0
        PathComponentMax: 0 Status: 1 type: 0 Serial Number: 0x0 Encrypted
        Share Capabilities: None        Share Flags: 0x30
        tid: 0x5        Maximal Access: 0x11f01ff

        1) \\172.25.223.247\test Mounts: 1 DevInfo: 0x20020 Attributes: 0x5c4402cf
        PathComponentMax: 255 Status: 1 type: DISK Serial Number: 0x4a6aea0a Encrypted
        Share Capabilities: None Aligned, Partition Aligned, TRIM-support,      Share Flags: 0x0
        tid: 0x1        Optimal sector size: 0x1000     Maximal Access: 0x1f01ff

root@smfrench-Virtual-Machine:~# cat /proc/fs/cifs/DebugData | grep Version
CIFS Version 2.32
```

Trace of Linux AES-GCM-256 mount to Windows with “require_gcm_256” set



What about Performance Improvements?

- It rocks! Let's take a simple example and copy 10GB from Azure server down to Linux client VM

  "dd if=/mnt/10GB of=/dev/null bs=1M count=10K"

  - Old defaults (3.0) 143MB/sec
  - With 3.1.1 201MB/sec (41% faster)
  - And go to 2 channels & set new parm "rasize" to 4MB
    - 453MB/sec
      - More than 3x faster!!
- Lots of great perf improvements!

And another one … (Thank you Rohith!)

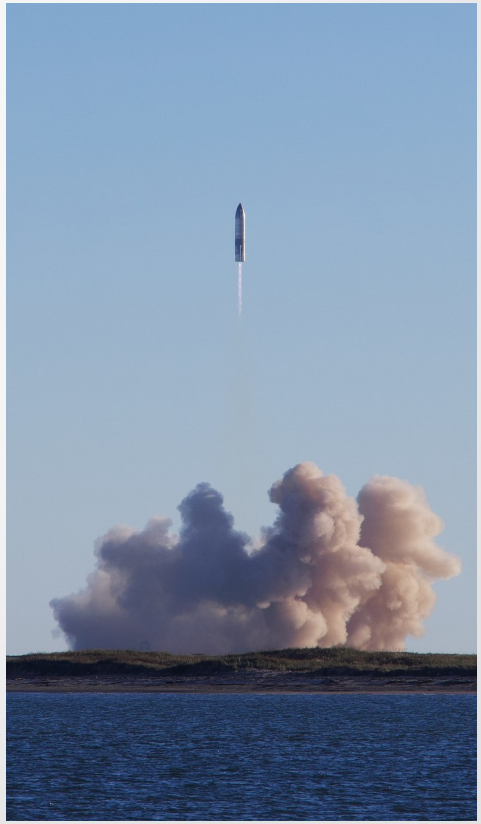
- Support added for handle leases (deferred close). Here are two simple example of the huge caching perf gains even copying to Samba localhost
  - Create a 2GB file and read it back (read is 4x faster)

    dd if=/dev/urandom of=2G bs=1M count=2K ;

    dd if=2G bs=1M count=2K of=/dev/null
    - Before:2.0 GiB copied, 0.583143 s, 3.7 GB/s
    - Current: 2.0 GiB copied, 0.159237 s, 13.5 GB/s
  - Read the same 4GB file twice (2nd time is 3x faster)

    dd if=4G of=/dev/null bs=1M count=4K ;

    dd if=4G of=/dev/null bs=1M count=4K
    - Before: 4.0 GiB copied, 1.36794 s, 3.1 GB/s
    - Current: 4.0 GiB copied, 0.441635 s, 9.7 GB/s

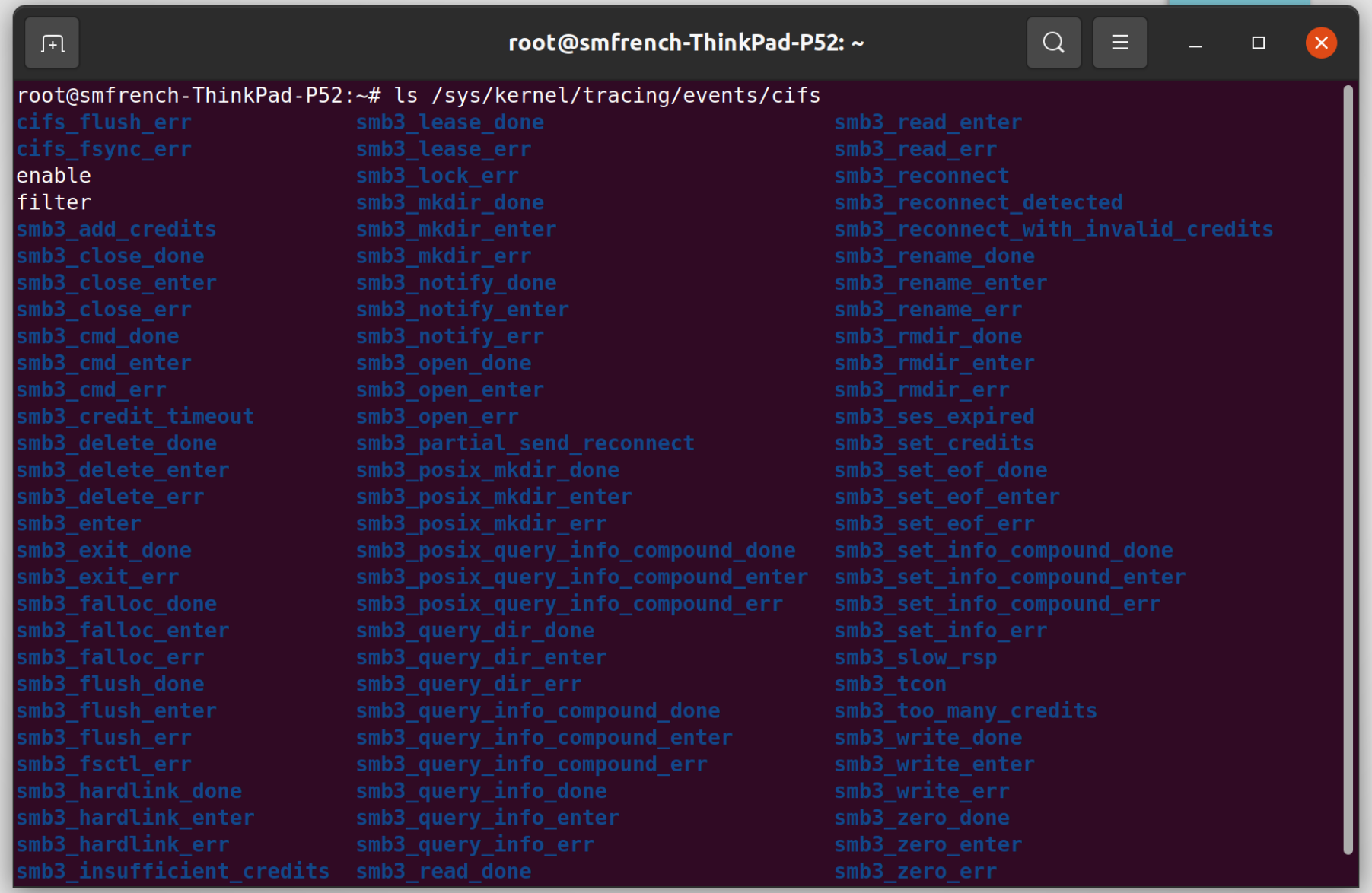


And more info on why it is faster...

- Easy to see why open/write/close/open/read/close is so much faster
- Before: Total vfs operations: 562

  SMBs: 1064
  Bytes read: 2147483648 Bytes written: 2147483648
  Creates: 12 total 0 failed
  Closes: 12 total 0 failed
  Flushes: 1 total 0 failed
  Reads: 513 total 0 failed
  Writes: 513 total 0 failed
- After (with handle lease patches): Total vfs operations: 42
  SMBs: 548
  Bytes read: 0 Bytes written: 2147483648
  Creates: 11 total 0 failed
  Closes: 11 total 0 failed
  Flushes: 1 total 0 failed
  Reads: 0 total 0 failed
  Writes: 512 total 0 failed

Better debugging: now 85 smb3 dynamic tracepoints (10% more than last year)



And another new feature ... “shutdown”

- Shutdown call (see https://man7.org/linux/man-pages/man2/ioctl_xfs_goingdown.2.html for more details or tools like “godown”)
- root@smfrench-ThinkPad-P52:~# mount | grep cifs

  //localhost/test on /mnt1 type cifs

  root@smfrench-ThinkPad-P52:~# touch /mnt1/file

  root@smfrench-ThinkPad-P52:~# ~/xfstests-dev/src/godown /mnt1/

  root@smfrench-ThinkPad-P52:~# touch /mnt1/file

  touch: cannot touch '/mnt1/file': Input/output error

  root@smfrench-ThinkPad-P52:~# mount -t cifs //localhost/test /mnt1 -o remount

  root@smfrench-ThinkPad-P52:~# touch /mnt1/file

And another new feature ... “fcollapse” (thanks Ronnie!)

- COLLAPSE_RANGE and INSERT_RANGE are important for some sparse file workloads (especially fcollapse). Here is an example of removing (fcollapse) 10MB from the beginning of a file
- root@smfrench-ThinkPad-P52:~# dd if=/dev/random of=/mnt1/300M bs=1M count=300

  root@smfrench-ThinkPad-P52:~# du -h /mnt1/300M ; xfs_io -c "fcollapse 0 10000000" /mnt1/300M

  301M /mnt1/300M

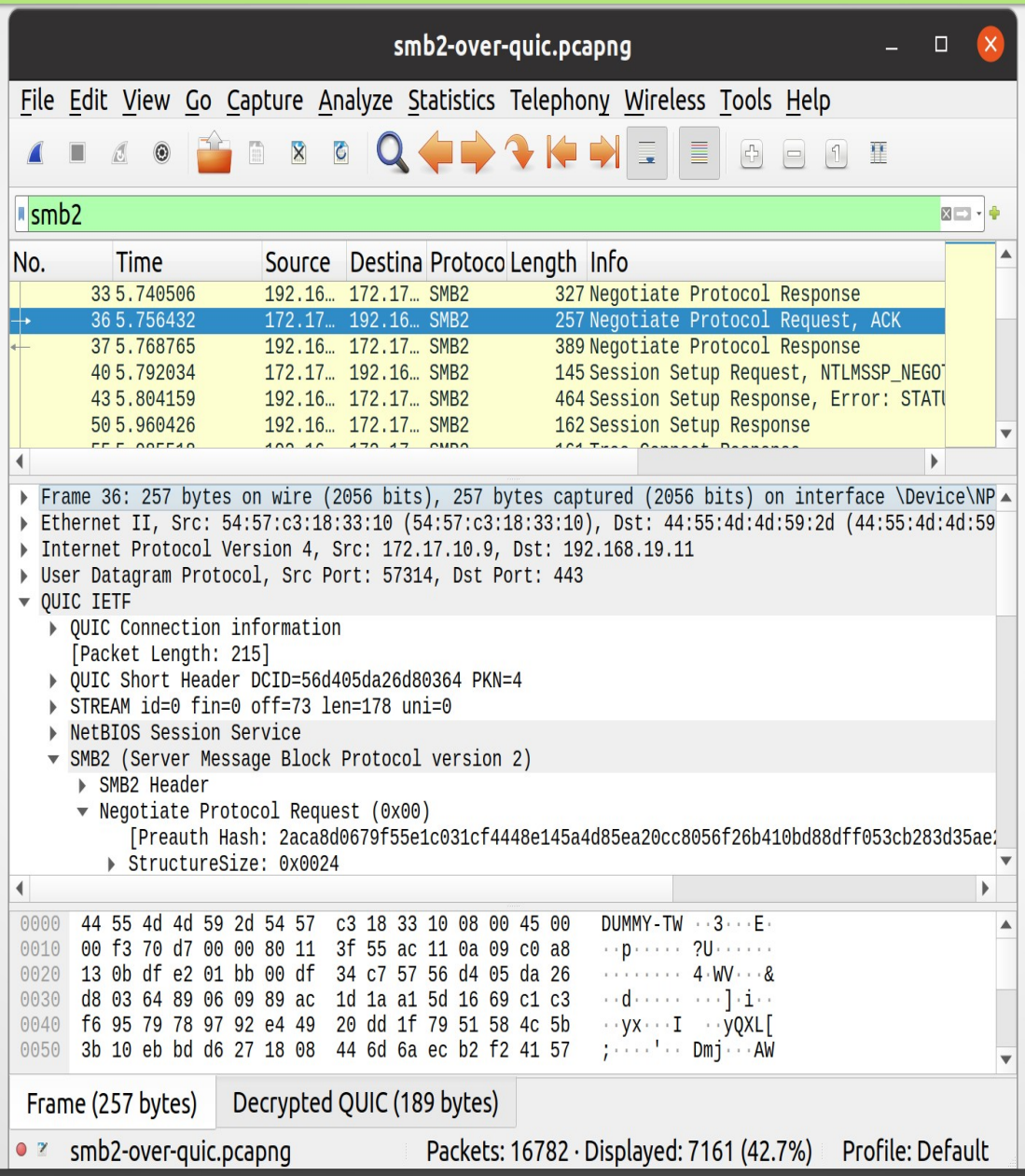
  root@smfrench-ThinkPad-P52:~# du -h /mnt1/300M

  291M /mnt1/300M
- On earlier kernels it would fail:

  xfs_io -c "fcollapse 0 10000000" /mnt1/300M

  fallocate: Operation not supported

Wireshark now has support for SMB3.1.1 over QUIC



Detailed feature list by release

5.3 (55 changesets) Sept 15th, 2019 (cifs internal module number 2.22)

- Improve performance of open (cut network requests from 3 to 2), improves perf about 10%
- Improve encrypted read and write perf with the addition of GCM crypto (e.g. can more than double encrypted write performance and large reads MUCH faster as well)
- copy_file_range (fast server side copy) now supports cross share copy offload
- smbdirect (SMB3 over RDMA) no longer ‘experimental’ (thanks Long Li!)
- Send netname context on negotiate protocol (could help load balancers eg.)
- Can query symlinks stored as reparse points

5.4 (76 changesets). Nov. 24th, 2019
Cifs version 2.23

- Boot from cifs (root file system on cifs). Networking dependencies went in 5.5. Thank you Paulo from SuSE!
- mount parm “modefromsid” to allow setting mode bits in special ACE
- Allow decryption for large reads to be offloaded: new mount parm

  “esize=<min-offload-size>” to improve encrypted read performance via parallel decryption
- Allow disabling requesting leases for a mount (“nolease” mount parm)
- Add passthrough ioctl for SMB3 SetInfo. Thank you Ronnie from Redhat!
- Add new mount options for forced caching (“cache=ro” and “cache=singleclient”) and improved signing perf (“signloosely”)
- Display max requests in flight.
- Can get keys for Wireshark encryption more easily via smbinfo <filename>

5.5 (61 changesets). January 26th, 2020 Cifs version 2.24

- Add support for flock
- SMB3 Multichannel support (Thank You Aurelien)
- Performance optimization query attributes on close (also is more correct for cases where timestamp update delayed to close time)
- Improvements to Boot from cifs (root file system on cifs) – network dependencies merged
- Readdir performance optimization (reparse points)

5.6 kernel March 2020 – cifs.ko version 2.25

- “modefromsid” mount option much improved to set better ACL at file create time
- Add support for fallocate mode 0 for non-sparse files
- Allow setting owner info, DOS attributes and creation time from user space backup/restore tools (Thank you Boris Protopopov)
- Readdir performance optimization (add compouding support for readdir, cuts roundtrips for typical ls from about 9 to 7) (Thank you Ronnie)
- Readdir improvements for modefromsid and cifsacl (so mode bits don’t get overwritten by default mode in readdir)
- Add new ioctl for change notify (for user space tools to wait on directory change notifications)

5.7 kernel. 5/31/2020. 49 changesets, cifs.ko version 2.26

- Big perf improvement for signed connections (when multiple requests sent at same time)
- RDMA (smbdirect) improvements
- Swap over SMB3
- Support for POSIX readdir

5.8 kernel. 8/2/2020. 61 changesets cifs.ko version 2.28

- Big perf improvement for large I/O with multichannel (often > 4x faster) and for read with large pages
- Support for “idsfromsid” (allowing alternate way of handling chown - mapping of POSIX uid/gid, owner information, into ‘special SID’)
- Support for POSIX queryinfo (All key parts of SMB3.1.1 POSIX extensions support complete)
- “nodelete” mount parm added (there were cases where mounting read only couldn’t handle some uses cases)

5.9 kernel. 10/11/2020. 30 changesets cifs.ko version 2.28

- Fixes, for example:
  - Ownership now properly saved for idsfromsid on mdkir
  - DFS fixes

5.10 kernel. 12/13/2020. 43 changesets cifs.ko version 2.29

- idsfromsid mount option now works to Azure
  - Needed for “client enforced” security workloads (where default mode bits or alternatively cifsacl can’t be used)
- Special files (fifo, char, block, symlink etc. are saved as reparse points by WSL) created by Linux apps on Windows are now recognized
- Fixes for SMB3.1.1 POSIX Extensions return owner information properly

5.11 kernel. 2/14/2021. 80 changesets cifs.ko version 2.30

- Add support for new Linux mount API which allows
  - Better error handling, messages on mount failures
  - Better support for changing an active mount (remount)
- Can get/set auditing information (SACL)
- Support for server notification of changes (add support for the “Witness Protocol”) such as server moving, address changes

5.12 kernel. 4/25/2021. 51 changesets cifs.ko version 2.31

- New mount options to improve performance
  - “actimeo” metadata caching timeout can now be configured differently for files (“acregmax”) or directories (“acdirmax”)
- “vers=3” mount option now will also include SMB3.1.1 (not just SMB3.0 dialect). To mount with only SMB3 (and not request SMB3.1.1) can still use “vers=3.0” but “vers=3” means “version 3 or later, including 3.1.1)
- Fixes for saving mode bits (“cifsacl” and “modefromsid”)
- Important fix for reconnect when server’s ip address changed
- Support added for idmapped mounts (user namespace mappings), added for cifs.ko and more generally in the Linux VFS as well

5.13-rc kernel (expected July 2021) 54 changesets so far. cifs.ko version 2.32

- Huge performance boost for readahead in some configurations by setting new mount parameter (“rasize=”) larger than rsize
- Add support for fcollapse and finsert (collapse and insert range calls)
- Add support for deferred close (handle leases), greatly improving performance of some workloads
- improvements to directory caching of the root directory
- Strongest type of encryption (GCM256) is now sent by default in the list of allowed encryption algorithms (GCM128 preferred, then GCM256 then CCM128) and does not have to be enabled manually in module load time parameters
- Debugging of encrypted mounts improved (e.g. for multiuser mounts and also for GCM256)
- Add support for shutddown ioctl (useful to halt new activity to better allow emergency unmounts, and also required for some common testcases)
- Mount error handling improvements (see *“/proc/fs/cifs/mount_params”)*

Cifs-utils improvements (thank you Pavel for coordinating this ...)

- 6.12 version released on December 31st
  - mount.cifs improvements
  - Add SACL (auditing) support to get/setcifsacl, and support for changing ownership
- 6.13 released April 12th : added container namespace support
- Next release:
  - smbinfo can now display alternate data streams
  - setcifsacl can now set (optionally reorder) ACEs in the preferred order
- Smbinfo is a great tool and now rewritten in python to be even easier to extend
- Tools like smb2-quota show how easy it is to use SMB3 query_info/set_info fsctls (the SMB3 pass through fsctl feature now possible in cifs.ko) to access SMB3 server features that aren't available in POSIX/Linux API

What about the future? What should we expect?

- Multichannel reconnect improvements
- Support for SMB3.1.1 over QUIC (probably using user space upcalls first)
- Additional sparse file improvements (e.g. fallocate of file ranges)
- Support for compression of SMB3.1.1 network traffic
- POSIX emulation improvements such as better “silly-rename” workarounds for rename of an open file, and support for “\” in file names and better special file support
- More performance improvements, e.g. more general use of directory leases (beyond the root dir)
- Improved packet signing performance
- More multichannel features (dynamic channel usage, RDMA with multichannel support, witness protocol multichannel notifications)
- More idmapping choices (e.g. for when RFC2307 not available)
- More use of compounding for ACL related operations
- Improvements to the POSIX extensions
- Support for additional authentication options (e.g. PKU2U)
- Add support for more misc Linux features: tmpfile support, “freeze” ioctl, richacl xattr support, improved SELinux emulation

Some general configuration advice

- Lots of mount options (and “/proc/fs/cifs” and “/sys/module/cifs” parameters) but focus should be on a very small subset of these options:
- Commonly used:
  - username,password (or use credentials=)
  - mfsymlinks, seal (encrypt)
- Security model (three common choices, first two often with “noperm”):
  - “uid=,gid=,dir_mode=,file_mode=” or “cifsacl,multiuser” or “idsfromsid,modefromsid”
  - “sec=krb5” is also commonly chosen
- Often recommended, especially on very recent kernels are some of the following 5:
  - nostrictsync,rasize=,acdirmax=,acregmax=,multichannel
- And if server and client have rdma cards: “rdma”
- Sometimes used: “snapshot=” … “persistenthandles” … “nobrl”

Testing … testing … testing

- The “buildbot” - automated regression testing keeps getting better. Thank you Paulo, Ronnie and Aurelien. See:

  http://smb3-test-rhel-75.southcentralus.cloudapp.azure.com
- See xfstesting page in cifs wiki https://wiki.samba.org/index.php/Xfstesting-cifs
- Tests are getting broader. Git regression tests recently added e.g.
- Easy to setup, exclude file for slow tests or failing ones
- Huge improvement in XFSTEST – more than 160 groups of tests run over SMB3 (more than run over NFS)! And more being added every release

Thanks to the buildbot – Best Releases Ever for SMB3!

- Prevents regressions
- Continues to improve quality

Thank you for your time

- Future is very bright!

+ SMB3

Additional Resources to Explore for SMB3 and Linux

- https://msdn.microsoft.com/en-us/library/gg685446.aspx
  - In particular MS-SMB2.pdf at https://msdn.microsoft.com/en-us/library/cc246482.aspx
- https://wiki.samba.org/index.php/Xfstesting-cifs
- Linux CIFS client https://wiki.samba.org/index.php/LinuxCIFS
- Samba-technical mailing list and IRC channel
- And various presentations at http://www.sambaxp.org and Microsoft channel 9 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/cifs
  - 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://github.com/smfrench/smb3-kernel/tree/cifsd-for-next