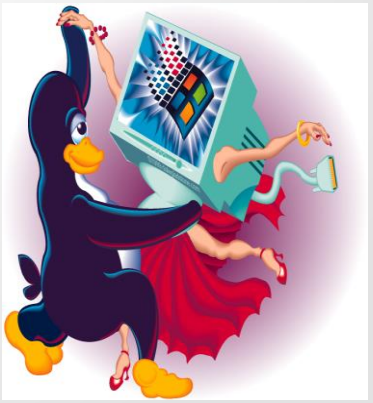
To the Cloud and Beyond

Accessing Files Remotely from Linux via SMB3.1.1

Presented by: Steve French
Principal Software Engineer
Azure Storage - Microsoft



- 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, Macs and various SMB3/CIFS based NAS appliances)
  - Co-maintainer of the new kernel server (ksmbd)
- 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

- Summary of recent Linux VFS and FS Activity
- New Linux Kernel Server
- Recent Linux Client Improvements
- Expected Linux Client Improvements in near future
- Recent cifs-utils improvements
- Testing

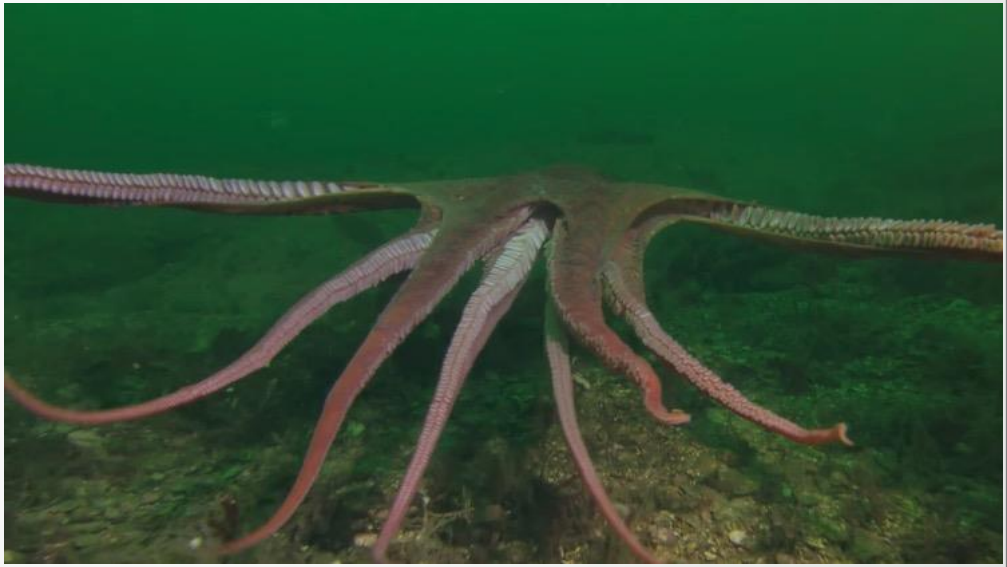
A year ago … and now … kernel (including SMB3 client cifs.ko) improving

Now: Linux 5.16

“Gobble Gobble”

Then: Linux 5.10

“Kleptomaniac Octopus”



A sample of some of topics driving FS development activity

- Folios (changing memory management) and netfs (improving readahead) driven by Matthew Wilcox
- Dave Howell's patches to fscache
- Improved containers support, improved idmapping
- Optional use of QUIC transport for various network filesystems
- Stronger, faster encryption
- Better support for faster storage (NVME, RDMA)
- More improvements around io_uring (async i/o)
- Shift to Cloud (longer latencies, object & file coexisting)

Most Active Linux Filesystems for year

- 5936 kernel filesystem changesets last year (since Linux 5.10) (flat)
- FS activity: 7% of overall kernel changes, up as % of activity
- Kernel is huge (> 21.8 million lines of code, measured last week)
- There are many Linux file systems (>60), but a few (and the VFS layer itself) drive ¾ of activity (e.g. btrfs, xfs, cifs etc)
- File systems represent almost 5% of kernel source code (1 million LOC) but are among the most carefully watched areas
- cifs.ko (cifs/smb3 client) activity is strong, #4 most active fs with 356 changesets over the year!
- 59.5KLOC, up >6% (not counting user space cifs-utils which are now 12% larger at 13KLOC, and samba tools which are larger still)
- At current pace ksmbd will also be one of most fs active components

Linux FS Change Detail since 5.10

- BTRFS 994 changesets (down slightly)
- VFS (overall fs mapping layer and common functions) 1317 (down slightly)
- XFS 544 (flat)
- CIFS/SMB2/SMB3 client 356 (up slightly since last year, up a lot since 4.18)
- NFS client 299 (flat)
- Others: F2FS 240 (down), EXT4 211 (down), GFS2 165(flat), Ceph 138 (up), AFS 52 (down), OCFS2 47 (down), 9p 33 (up) ...
- NFS server 299 (flat). Linux NFS server **MUCH** smaller than Samba
- Samba server is largest, most functionally rich open source fileserver: **Samba is 3.4 million lines of code.** ksmbd activity is also strong and it was merged into mainline in 5.15. – a very exciting time!

Linux filesystems are not easy – API keeps growing, improving
Responsible for more than 200 of 850 syscalls. Added multiple in past year

Syscall name	Kernel Version introduced
epoll_pwait2	5.11
mount_setattr	5.12
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 in the 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

What about SMB3 server (on Linux)?

- Samba server is great (and huge, and full function)
  - See various talks at sambaxp.org and snia.org
- But now there is a kernel server, ksmbd
  - Can help on some workloads (e.g. RDMA, smbdirect)
  - Also accelerating SMB3.1.1 improvements (and testing)
  - See e.g. Namjae's talk at sambaxp.org

Progress and Status update for Linux Kernel Server (ksmbd)

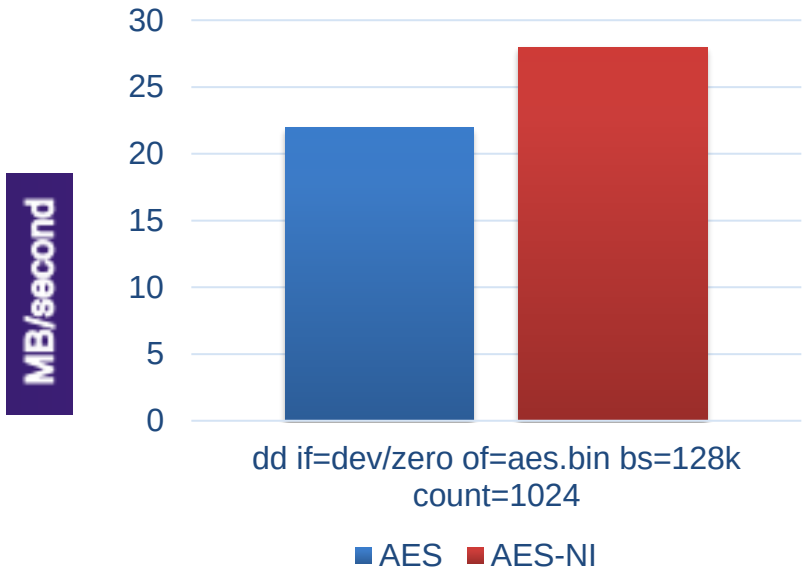
Provided by Namjae Jeon (linkinjeon@kernel.org)

ksmbd merged to mainline in 5.15

- First reviewed for 5 months in Linux-next (when ksmbd v1 patch series went in Linux-next)
- Many high profile developers reviewed, Thank you!
- Ksmbd was merged into linux-5.15 (August 31st)
- To make module and directory name consistent: changed “cifsd” to “ksmbd”
- Common code between client and server is now in “fs/smbfs_common” directory
- Later the cifs source directory will be renamed to smbfs_client to reduce confusion (and to avoid referencing old, deprecated, less secure protocol dialect ‘cifs.’ Modern clients and servers negotiate SMB3 or later, not old cifs)

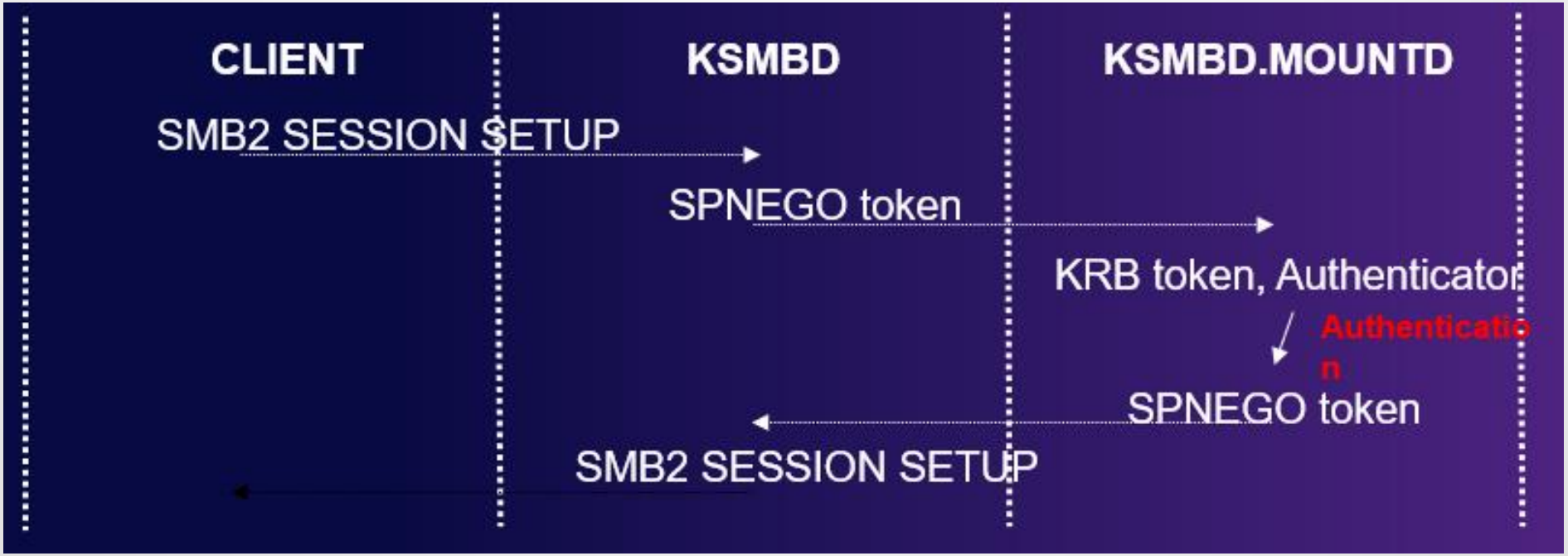
AES-256 encryption support

- ksmbd AES-256 CCM/GCM encryption support now available (strongest encryption)
- Ksmbd accelerated encryption(AES-GCM) performance using AES-NI support in kernel



Kerberos support

- Support authentication with Kerberos
- Ksmbd transmits Kerberos msg to ksmbd.mountd
- Ksmbd.mountd uses libkrb5 library

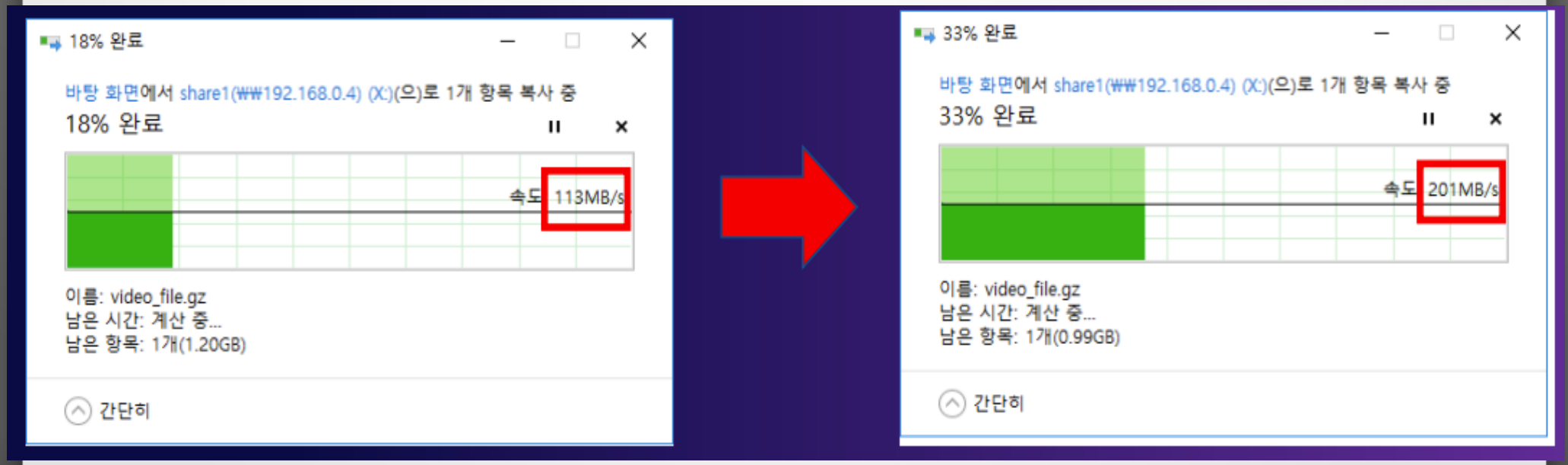


Duplicate extent support

- Ksmbd has added support for FSCTL_DUPLICATE_EXTENT_TO_FILE
- Can be used if share is on a local fs which supports reflink
- Linux client uses duplicate extents for some fallocate related operations like insert range)
- Additional xfstests tests pass
- Ksmbd doesn’t have to deal with VFS mapping (btrfs, etc.) layer like samba.

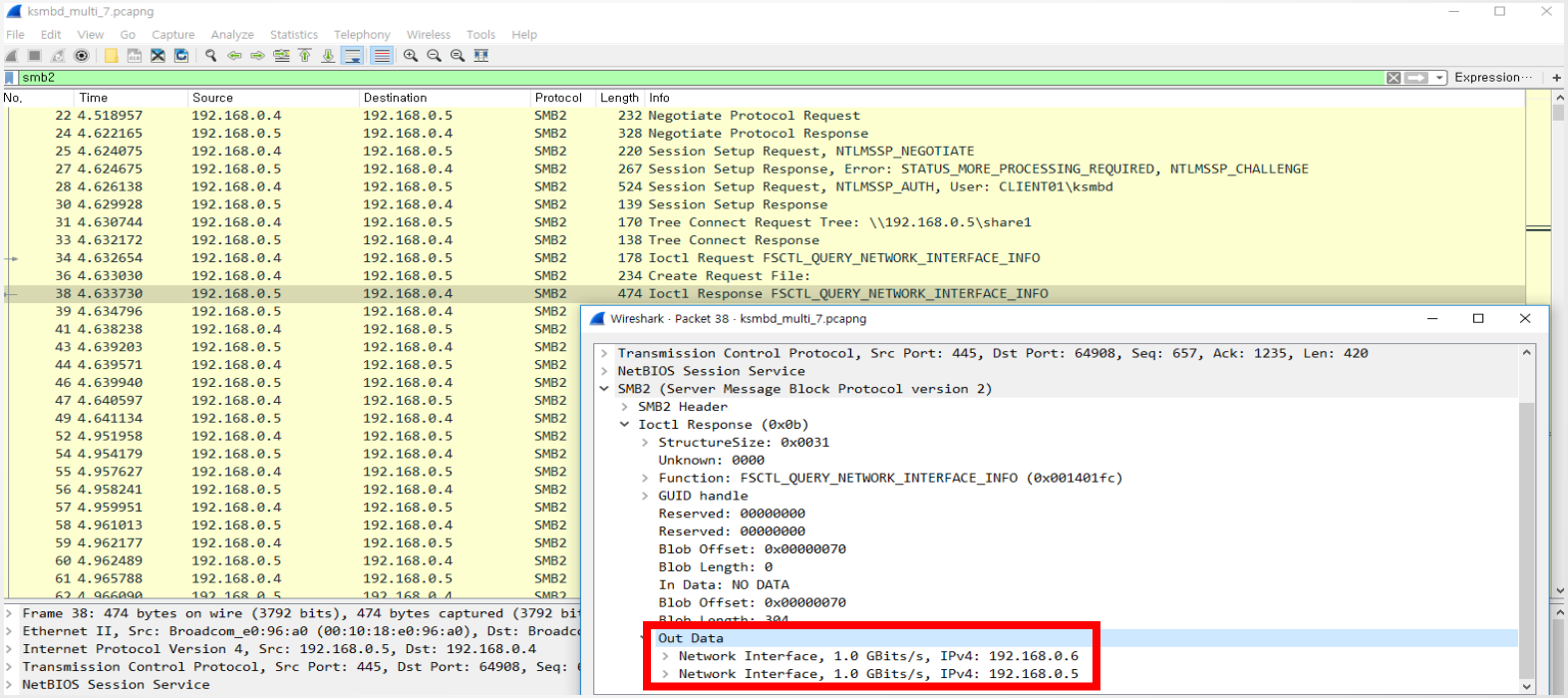
SMB3 multi-channel support

- SMB3 Multichannel feature greatly improves performance on Multi-port NIC or multiple NICs.
- Ksmbd kernel server now supports SMB3 multichannel.
- TODO Replay/retry features on channel failure.



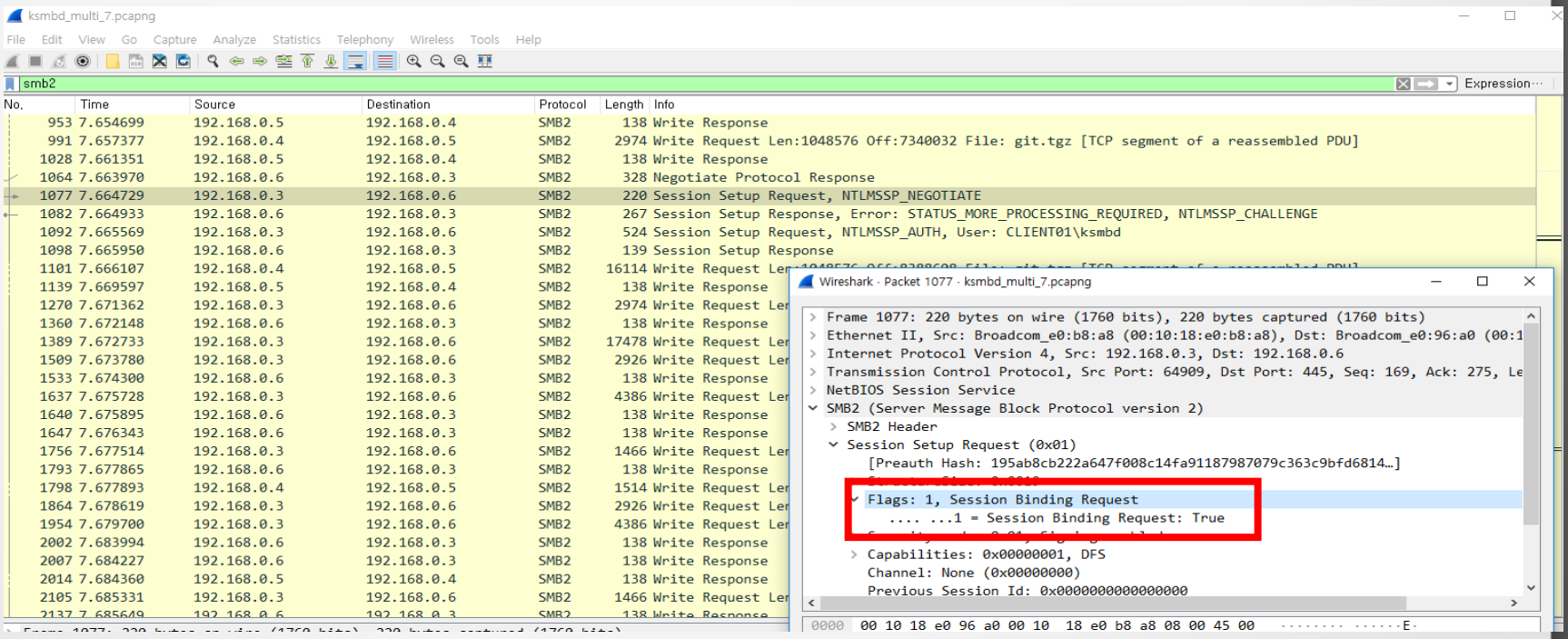
SMB3 multi-channel support

Send NICs information to client through FSCTL_QUERY_NETWORK_INTERFACE_INFO command



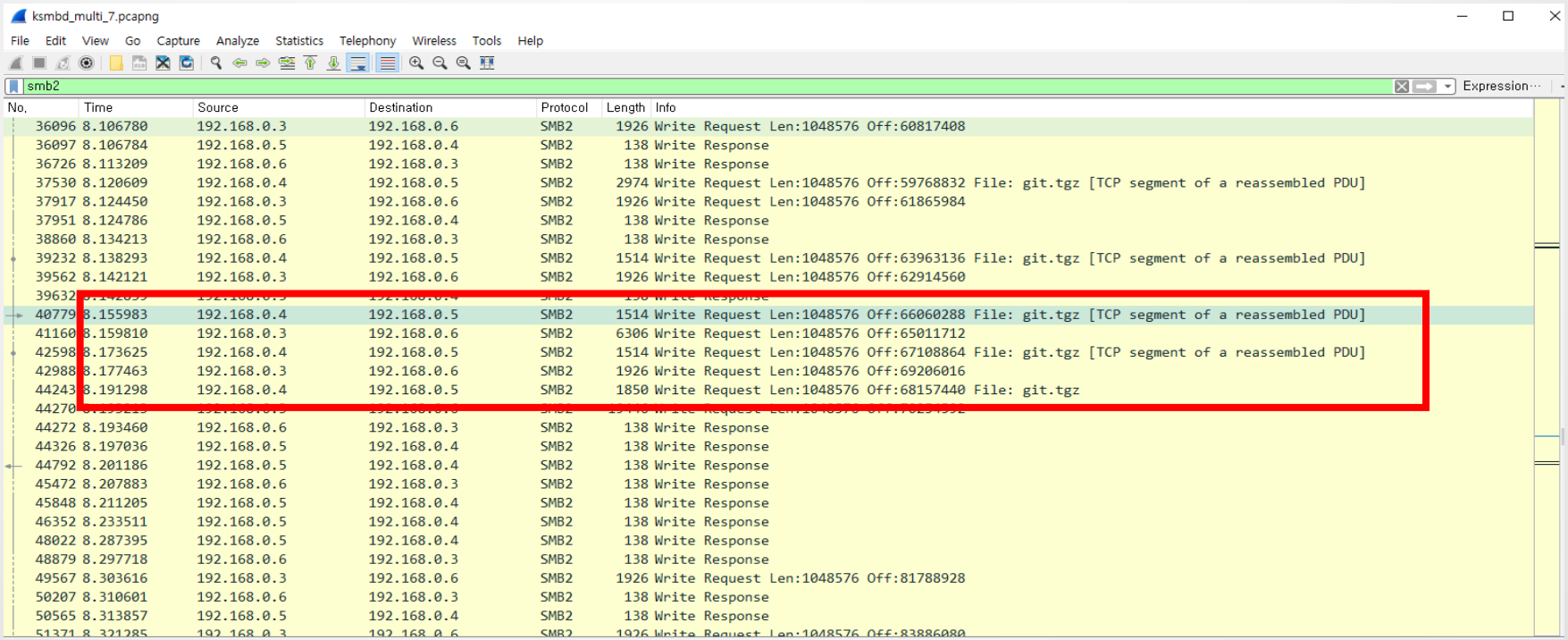
SMB3 multi-channel support

Client sending session binding request to ksmbd.



SMB3 multi-channel support

Client sending interleaved write requests to dual channels(192.168.0.3, 192.168.0.4)

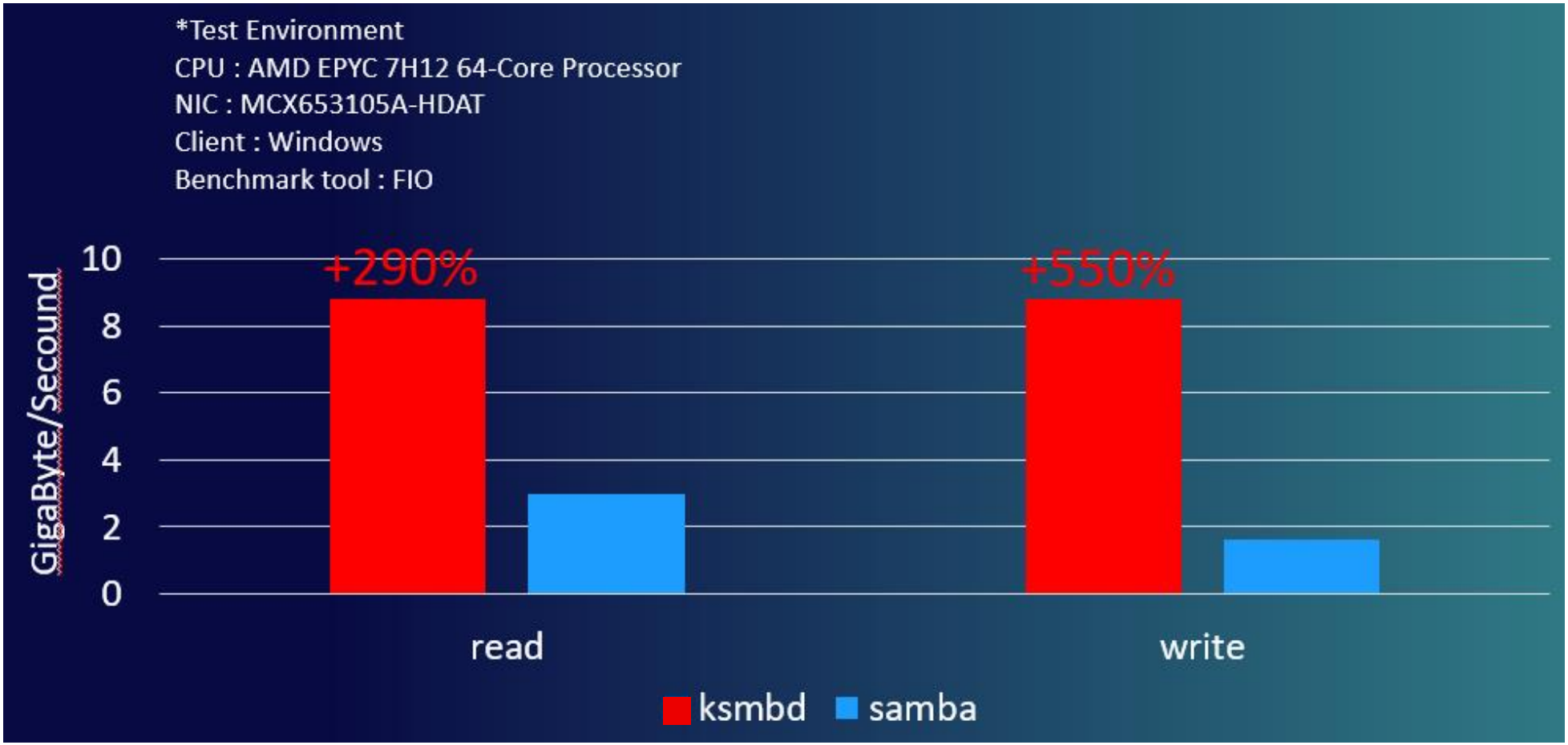


Currently working features

- SMB Direct with windows client
  - Got test HW support from Chelsio (Bob Dugan)
  - Patches (multiple buffer descriptor) in progress for performance improvement
  - Credit management rework
- SMB2 directory leases
- SMB2 change notify
  - considering using fanotify instead of inotify for SMB2_WATCH_TREE
  - Need to change fanotify codes as export symbol to call function by ksmbd.

RSS(Receive Side Scaling) mode support

Ksmbd now supports RSS mode
Ziwei Xie(high-flyer) compared samba and ksmbd in (multichannel+RSS) test environment. Thanks Ziwei!



SMB Direct(RDMA) support from Windows client

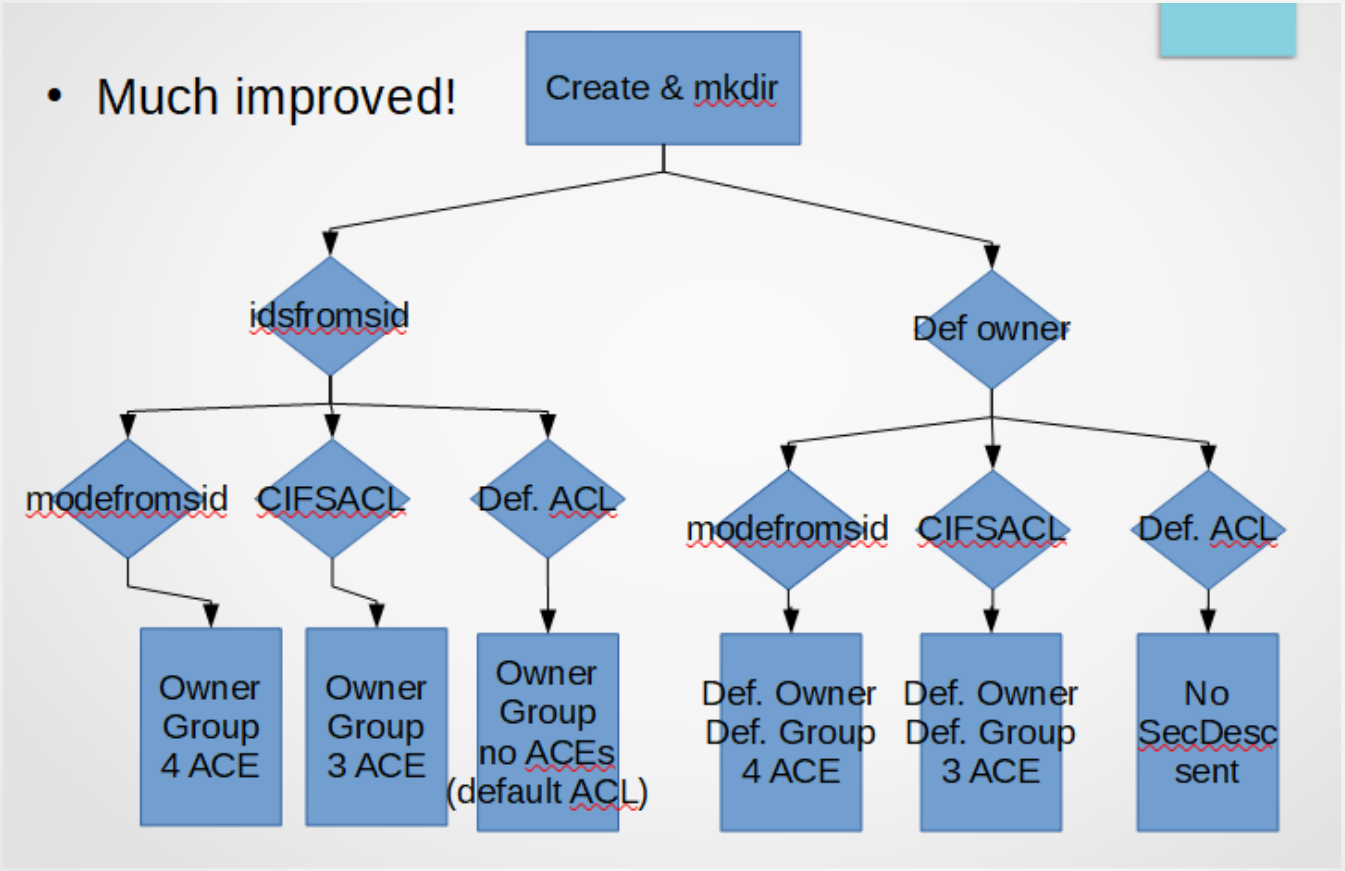
- Ksmbd supported SMB Direct with only linux client(cifs)
- Ksmbd in linux 5.17 kernel will support it with Windows client as well
- TODO: Working Multiple buffer descriptor support for large read/write size. Currently, Only MAX 1MB read/write size is supported

Ksmbd status summary

- In mainline kernel, and stability and features rapidly improving
  - Marked as experimental, disabled by default
- Initial focus was on functional testing, but soon after merge… serious security issues were identified as ksmbd got additional reviews and testing
  - Patches for these were merged into 5.15 and 5.16
  - Tracking progress at https://wiki.samba.org/index.php/Ksmbd-review
  - Additional reviews would be welcome. Progress on these has been good
- Roles: there are multiple developers helping Namjae (the maintainer). I am managing the git merges, ensuring additional functional testing is done regularly, and reviewing patches as requested by Namjae (my focus is largely on the client)
- Namjae would welcome additional help with code reviews, security auditing, testing and new features
- Very exciting time!

Examples of great recent progress on client (cifs.ko)

Remember the security models: idsfromsid, modefromsid, cifsacl (improved in 5.9 kernel)



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 (Azure, Windows, ksmbd etc. support it)
- Client can require (force) AES-GCM-256 if new module parm “require_gcm_256” set since 5.12 kernel

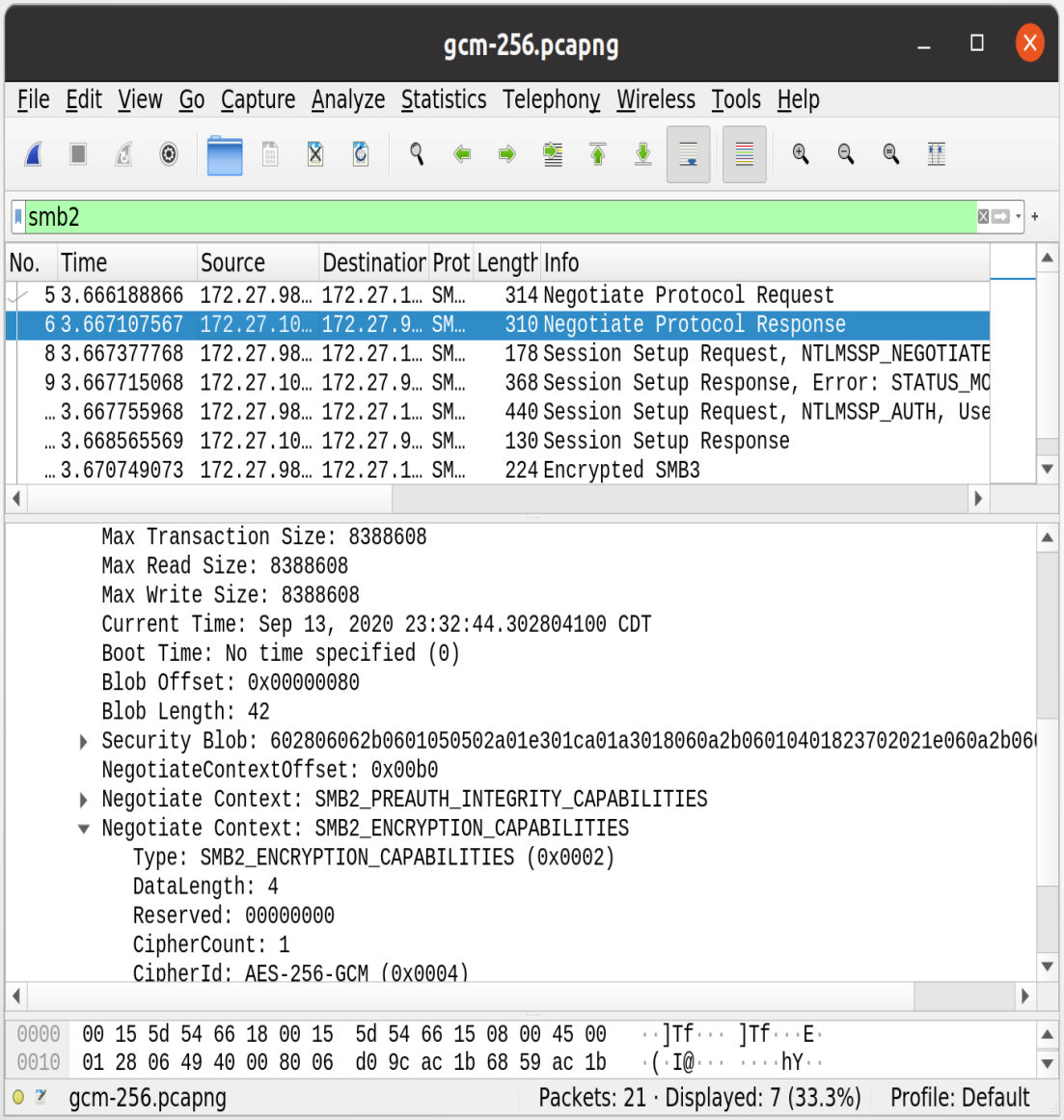
```
root@smfrench-Virtual-Machine:~# mount | grep cifs
//172.25.223.247/test on /mnt type cifs (rw,relatime,vers=3.1.1,cache=strict,username=testuser,uid=0,nof
orceuid,gid=0,noforcegid,addr=172.25.223.247,file_mode=0755,dir_mode=0755,seal,soft,nounix,serverino,map
posix,noperm,rsize=4194304,wsize=4194304,bsize=1048576,echo_interval=60,actimeo=1)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



Multichannel (much improved in 5.13)

- Thank you Aurelien! Opportunity for huge perf gains
- Originally added in 5.5 kernel as experimental
- large I/O performance much improved in 5.8 kernel (up to 5x faster in my testing) now much more stable in 5.13
- Reconnect improvements added in 5.16 and more being worked on for 5.17

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!

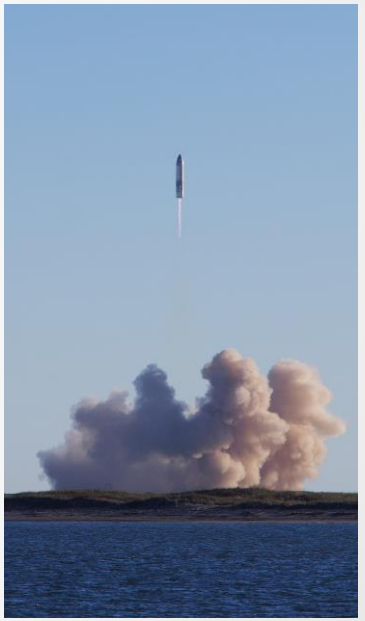
And another one … (Thank you Rohith!)

- Support added for handle leases (deferred close) in 5.13 kernel. 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

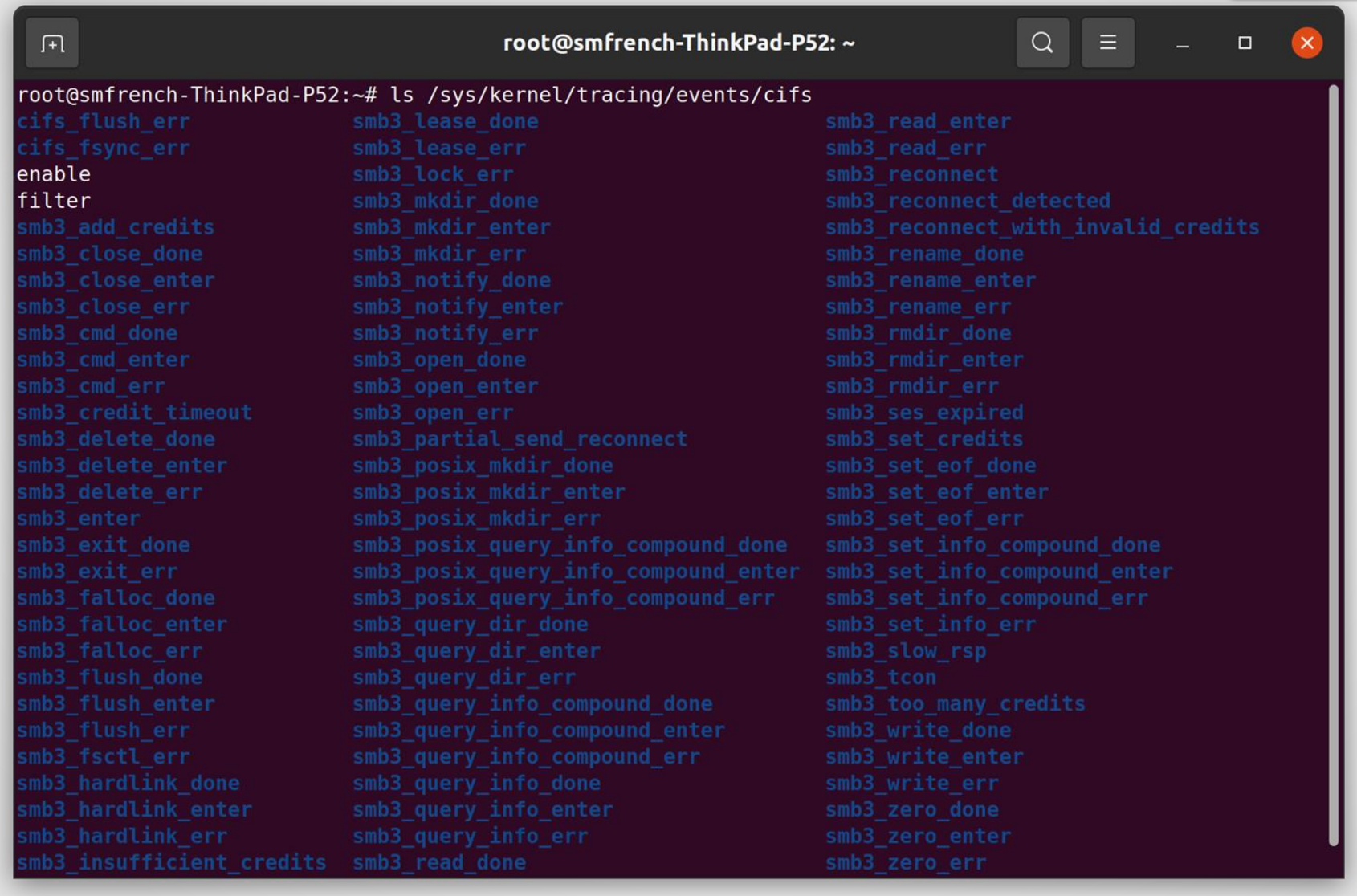
- Before:2.0 GiB copied, 0.583143 s, 3.7 GB/s
- Current: 2.0 GiB copied, 0.159237 s, 13.5 GB/s
- Read the same 4GB file twice (2nd time is 3x faster)
- dd if=4G of=/dev/null bs=1M count=4K ;
- dd if=4G of=/dev/null bs=1M count=4K
- Before: 4.0 GiB copied, 1.36794 s, 3.1 GB/s
- Current: 4.0 GiB copied, 0.441635 s, 9.7 GB/s



And another one added in the 5.12 kernel…

- Metadata caching performance can now be controlled more granularly
  - acregmax for caching file metadata (defaults to 1 sec)
  - acdirmax for caching directory metadata (defaults to 1 second, can often be set much higher)
    - Setting this higher can allow caching components of long path names allowing faster lookup of pathnames and opening of files, especially in deep directory trees
  - actimeo to set both

Better debugging: now 87 smb3 dynamic tracepoints



And another new feature … “shutdown” (added to 5.13 kernel)

- 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

Detailed feature list by release

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 kernel (June 27th 2021) 66 changesets. 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”)*

5.14 kernel (August 29th) 71 changesets cifs.ko version 2.33

- Fallocate improvements (can now alloc smaller ranges up to 1MB). Thank you Ronnie!
- DFS reconnect improvements, and reconnect retry improvements. Thank you Paulo!
- Experimental support added for negotiating signing algorithm
- 5.15 kernel (Oct 31, 2021) 26 changesets
  - Important deferred close (handle lease) bug fixes
  - Support for weaker authentication (NTLMv1 and LANMAN) removed
  - (And experimental kernel server, ksmbd, merged)

5.16 kernel (Jan 9, 2022) 46 changesets cifs.ko version 2.34

- Performance improvements for stat, setfilesize and set_file_info (additional uses of compounding)
- Multichannel improvements (thanks Shyam!)
- Reconnect improvements
- Fscache fixes
- New mount parm “tcpnodelay”

What about the future? What should we expect?

- Significant multichannel improvements in 5.17 kernel
- Integration w/new page cache readahead mechanism (“netfs”) and offline caching features (fscache) from Dave Howells
- Support for SMB3.1.1 over QUIC (probably using user space upcalls first to well tested module like msquic)
- Additional sparse file improvements (including more fallocate improvements)
- 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. peer to peer kerberos)
- Add support for more misc Linux features: tmpfile support, “freeze” ioctl, richacl xattr support, improved SELinux emulation

Client tooling (cifs-utils) improvements

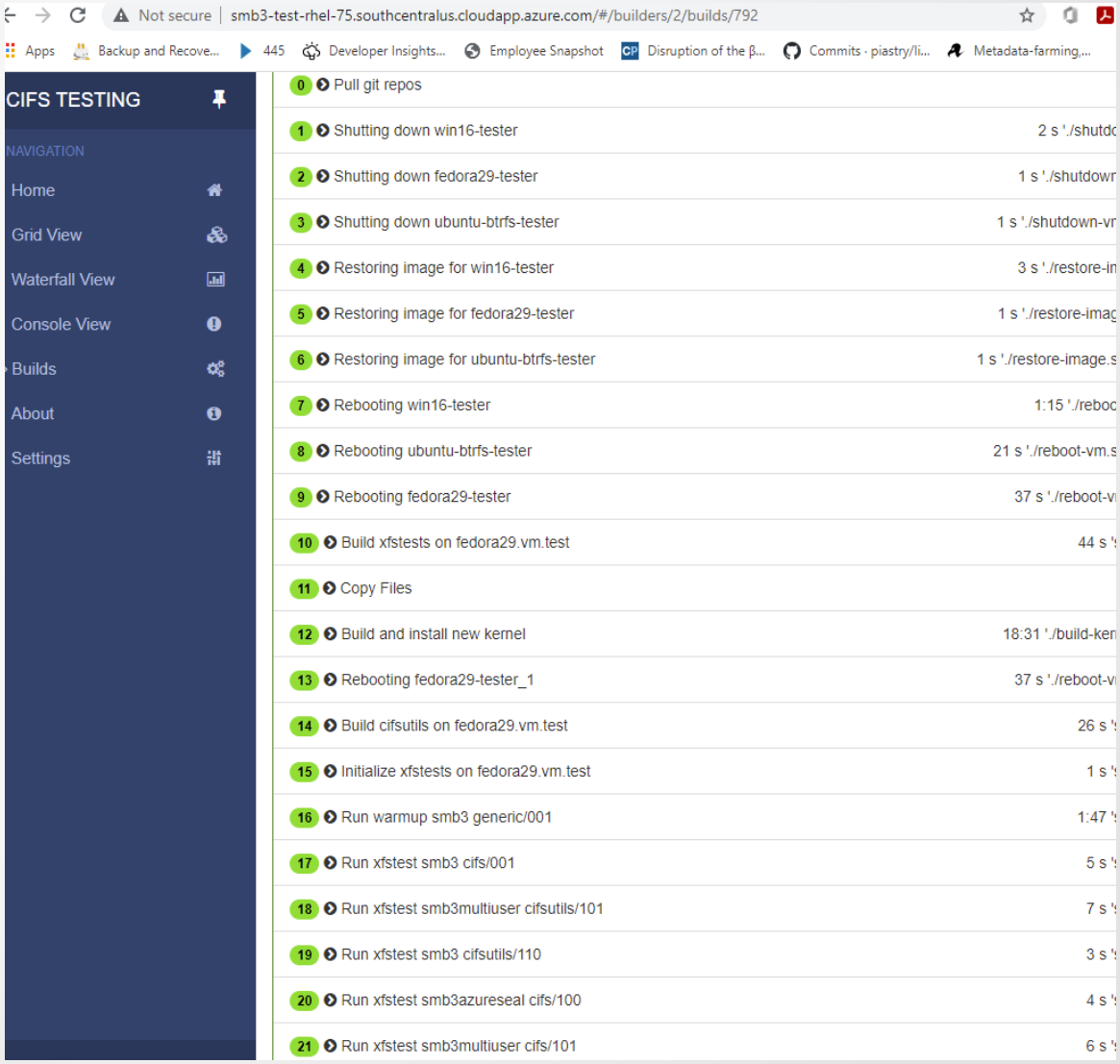
- cifs-utils 6.14 released in Sept
  - Add commands to view Alternate Data Streams
  - setcifsacl improvements
  - Improved debugging (keydump)
- cifs-utils version 6.13 released in April
  - Improvements to smbinfo to make snapshot mounts easier (mounting previous versions of a share)
  - Add ability to display alternate data streams (“smbinfo filestreaminfo”)
  - Improved support for containers
  - Improved debugging (“smbinfo keys”) of encrypted mounts
  - Getcifsacl/setcifsacl can now dump SACLs not just DACLs

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”

Thanks to the buildbot – Best Releases Ever for SMB3!

- Prevents regressions
- Continues to improve quality
- Added 40+ tests to main test group over past year!
- And more in other xfstest groups



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://git.samba.org/?p=ksmbd.git;a=shortlog;h=refs/heads/cifsd-for-next