

SNIA DEVELOPER CONFERENCE



BY Developers FOR Developers

September 16-18, 2024
Santa Clara, CA

Elevating Linux File Access

Recent Enhancements to 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 ?

- 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 6.11-rc7 “Baby Opossum Posse”
- Then 6.5 “Hurr durr I’m a ninja sloth”
- 87,300 changesets (non-merge commits) over this period
- 41,673 files changed
- 4 million insertions, 1.4 million deletions



Linux Storage, Filesystem, MM, BPF summit

- This year was in May (at Salt Lake City)

- Great group of talented Linux Linux developers working on storage and FS



Some Linux FS topics of interest discussed recently

- Testing ... testing ... and more automated testing ... (e.g. kdevops)
- Large Folios, netfs mapping layer, variable size pages, and fscache and page caching redesign
- Improvements to statx (for example for subvol/snapshot), improvements to swap
- Idmapped mounts, fine grained timestamps
- How to move Linux kernel drivers and fs to Rust eventually
- Mount API extensions, and finish up of conversion to new mount API
- Move to large block size (LBS)
- Leveraging eBPF (not just dynamic tracing)
- Extending in kernel encryption: TLS handshake (for NFS) and QUIC (SMB3.1.1 and other)
- Shift to cloud and Better support for faster storage (NVME) and net (RDMA/smbdirect)

Linux Filesystems Activity over past year (since 6.5 kernel)

- 9660 filesystem related changesets. 11% of total kernel changesets over this period even though only 4.4% of the lines of code! Huge increase from previous year where it was 6.1% of total changesets.
 - Lots of developer attention on filesystems ...
 - Linux kernel fs are 1.16 million lines of code total (measured last week)
- Lots of progress!
 - Ksmbd no longer “experimental” (as of 6.6 kernel)
 - Bcachefs added (6.7 kernel), ntfs (classic) removed (6.9 kernel)
 - Reiserfs marked ‘obsolete’ with removal targeted next year, will JFS be next?

New fs: Bcachefs (since 6.7 kernel)

- Bcachefs was merged into the Linux kernel with version 6.7. It combines the performance of filesystems like ext4 with advanced features similar to ZFS and Btrfs, such as copy-on-write (CoW), integrated volume management, and improved reliability. Unlike ZFS, Bcachefs is fully GPL-compatible, which avoids licensing issues and positions it as a serious contender in the next-generation filesystem space.
 - Key features they emphasize include: copy on write, full Metadata and data check-summing, replication, advanced caching, data placement, snapshots, scalable, erasure coding

Most Active Linux Filesystems over the past year

- VFS: the mapping layer between syscalls and the various FS
 - 574 changesets, 672 if you include netfs mapping layer)
 - activity up from last year
- Local fs activity last year: Bcachefs (3951 changesets, activity up), BTRFS (958 flat), XFS (793, up), ext4 (263, down), F2FS (247, down), NTFS3 (125) erofs (109)
- Most active non-local fs last year was again SMB3.1.1 client (cifs.ko), 333 changesets, it also had many more lines changed
- Network fs: nfsd (277, flat), nfs client (203), Gfs2 (187), ksmbd (138, flat) ceph (121), AFS (93)

SMB3.1.1 Activity

- Similar to the previous year, 333 changesets
- Due to code cleanup, now a few percent smaller at 61.5KLOC (not counting the various userspace tools)
- Ksmbd server activity flat
 - 27.2KLOC (not counting various userspace helpers), and 514 kernel changesets since its introduction
- Samba server (userspace alternative) just released version 4.21 recently, on September 2nd
 - Now more than 4 million (!) lines of code, orders of magnitude bigger than the kernel ksmbd server or any of the NFS servers, and is much more active, also includes key security services and tools

Repeating our Goals for SMB3.1.1 on Linux

- Be the fastest, most secure general way to access file data, whether in cloud or on-prem or VM
 - 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 improving, lots of changes to internal APIs (netfs, fscache...)

Syscall name	Kernel Version introduced
fchmodat2	6.6-rc1
listmount (query children of mount)	6.8-rc1
statmount (query attributes of mnt)	6.8-rc1
And new fs ioctls:	
FS_IOC_GETFSUUID	6.9-rc1
FS_IOC_GETFSSYSFSPATH	6.9-rc1

One of the strengths of SMB3.1.1 is broad interop testing

- In-person plugfests are back!
- SMB3.1.1 test lab continues to be colocated with SDC again this year
- Many exciting things being tested
- See <https://www.sniadeveloper.org/special-events/smb3-interopability-lab>
- If curious what AI told me when I looked up the SMB3 test event details, see the next slide ...



I didn't realize search engine AI was such a big fan 😊 (I asked about details of the upcoming SMB3.1.1 test event)

- Ah, the wonderful world of file access protocols! Let's dive into it, shall we? 😊
- SMB 3.1.1—that's the Server Message Block protocol, for those not yet fluent in acronym-ese—has been getting quite the workout in the Linux ecosystem. It's like the marathon runner of filesystems, clocking in those miles (or should I say bytes?) with gusto. 🏃♂️
- So, what's the buzz about? Well, the Linux SMB3.1.1 client (cifs.ko) has been flexing its muscles lately. It's all about security, performance, and reliability. Picture this: a trusty Linux explorer, backpack loaded with new features, marching confidently toward remote data. 🌐
- Here's the scoop on what's been added recently:
 - Security: Tightened bolts, reinforced gates—because nobody wants unauthorized guests at the data party.
 - Performance: Faster than a caffeinated squirrel on a fiber-optic highway. 🚀
 - Reliability: Like a loyal friend who never cancels plans. You can count on it.
- And guess what? There's more to come! The Linux SMB3.1.1 client is like that overachieving student who raises their hand and says, "Wait, there's extra credit? Sign me up!" Expect even shinier features in the pipeline. ✨
- Now, let's talk Plugfest. It's like a tech playdate where devices gather to test their compatibility. SNIA (Storage Networking Industry Association) sets the stage, and participants—like eager lab partners—work together to define the testing process. Objectives? Accomplished! 🍻
- So, whether you're a filesystem aficionado or just curious about the inner workings of data highways, keep an eye on SMB3.1.1—it's the road less traveled, but oh-so-interesting! 🗺️

Progress on Linux kernel server

(ksmbd update – provided by Namjae Jeon <linkinjeon@kernel.org>)

For more info see:

https://sambaxp.org/fileadmin/user_upload/sambaxp2024-Slides/sxp24-Jeon-ksmbd_status.pdf

ksmbd overview: SMB3.1.1 kernel server

- **Project target**
 - **SMB 2.1 ~ SMB 3.1.1**
 - **Optimize performance for Linux workload**
 - **GPLv2 SMB Server**
 - **Better oplock/lease handling**
 - **Easy to develop new features(e.g. RDMA aka smbdirect)**
- **Very optimal for embedded devices**
 - **Small memory footprint and binary size**
 - **Low CPU usage and better performance**
 - **Easy cross compile without special handling**
- **Used by:**
 - **Embedded opensource projects like openWRT, dd-wrt**
 - **Commercial products like Network camera, Smart whiteboard, etc**
 - **Linux distribution like Ubuntu, Debian, openSUSE**

Status update – move cifs and ksmbd to /fs/smb/

- **Linus suggested moving fs/cifs and ksmbd directory to fs/smb/**
- **And each directory name is also changed to fs/smb/client and server.**
- **Common helper function and macros headers to fs/smb/common/**

fs/cifs -> fs/smb/client

fs/ksmbd -> fs/smb/server

fs/smbfs_common -> fs/smb/common

Status update – No longer Experimental

- No longer experimental module
- Many fixes and much testing for 2 years also lots of fuzzing, and security issues addressed quickly

```
-1,5 +1,5 @@
onfig SMB_SERVER
    tristate "SMB3 server support (EXPERIMENTAL)"
    tristate "SMB3 server support"
    depends on INET
    depends on MULTIUSER
    depends on FILE_LOCKING

ff --git a/fs/smb/server/server.c b/fs/smb/server/server.c
dex 801cd0929209c..5ab2f52f9b355 100644
- a/fs/smb/server/server.c
+ b/fs/smb/server/server.c
-590,8 +590,6 @@ static int __init ksmbd_server_init(void)
    if (ret)
        goto err_crypto_destroy;

    pr_warn_once("The ksmbd server is experimental\n");
```

Status update – directory lease(v2 lease)

- **v2 lease allows the client to cache metadata operations in directory**
- **ksmbd supported directory leases now**
- **Demonstrate operations with smb torture tests.**
- **Currently smb2 leases default disable now**
- **Enable “smb2 leases = yes” in ksmbd.conf**
- **Ready to make it by default enable after fixing a few issues**
- **Plan to enable it after more checking it with cifs.ko**

Status update – Durable handles v1/v2

- **Add support durable handle version 1 and 2 as defined in specification.**
- **ksmbd can handle temporary loss of connection when a file is still opening now**
- **Demonstrated the operation of this feature with smbtorure durable open tests.**
- **Continuous Availability is not implemented yet. (TODO)**

Status update – SMB 3.1.1 POSIX extensions

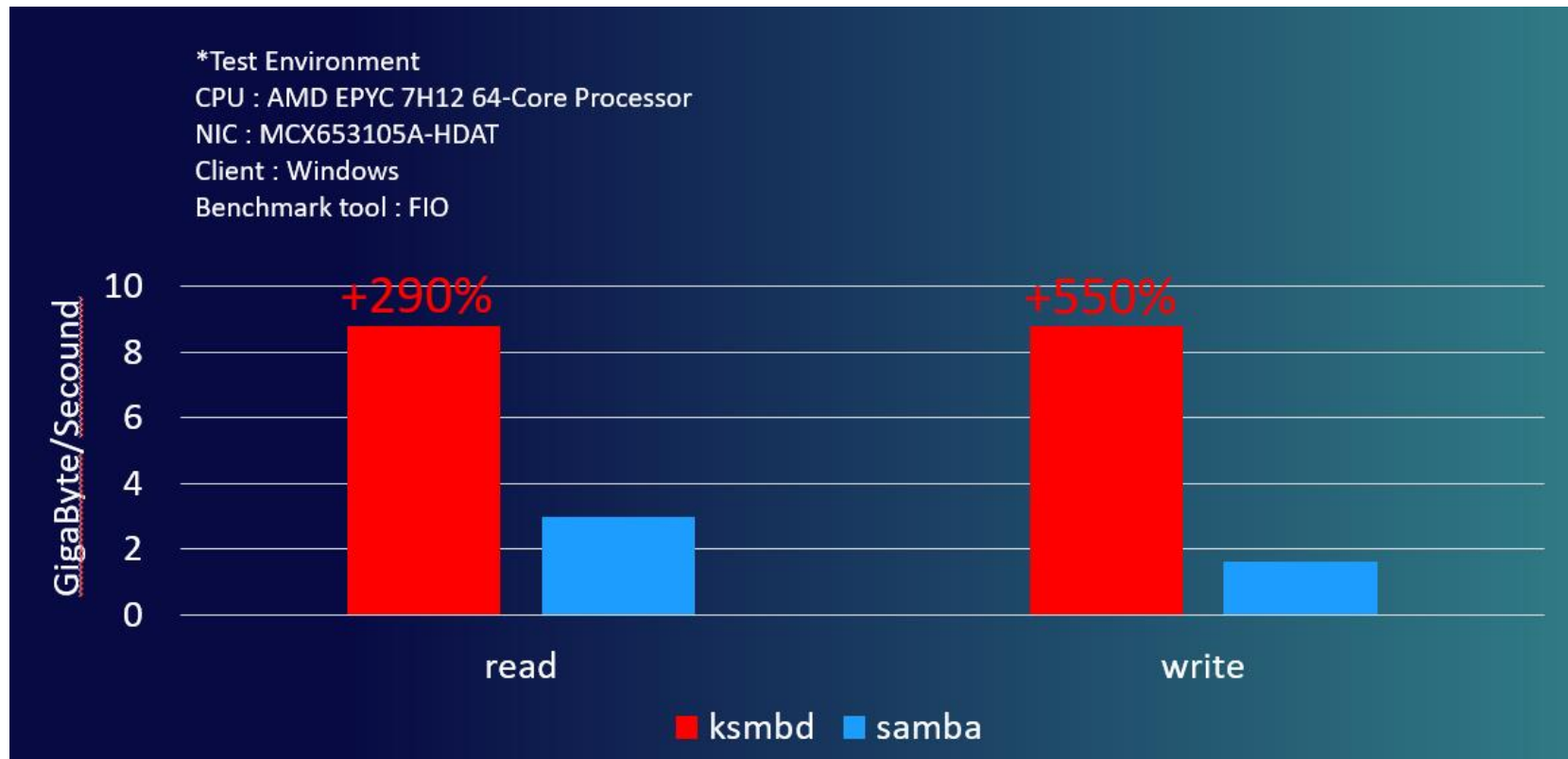
- **Mount from cifs.ko(Linux kernel client) with “posix” option**
- **Fill in SIDs in SMB_FIND_FILE_POSIX_INFO, SMB2_CREATE_POSIX_CONTEXT and SMB_FIND_FILE_POSIX_INFO responses**
- **All posix extensions implemented and supported:**
 - **SMB2_CREATE_POSIX_CONTEXT in smb2 create**
 - **SMB_FIND_FILE_POSIX_INFO in smb2 query directory**
 - **FS_POSIX_INFORMATION in smb2 get info filesystem**
 - **SMB_FIND_FILE_POSIX_INFO in smb2 get info file**

Status update – Fully support SMBDirect

- **Handle large RDMA read/write size(bulk data) supported by SMB Direct multi-descriptors (It supported single descriptor with 512KB size before)**
 - **8MB RDMA read/write size by default**
 - **Control read/write size through ksmbd configuration.(e.g. smb io size = 16MB)**
- **Improve the compatibility with various RDMA types of NICs**
 - **Tested smb-direct working with iWARP(Chelsio, soft-iWARP), Infiniband(Mellanox NICs, Connectx3 ~ x5), ROCE(soft-ROCE).**
- **Auto-detection of RDMA NIC without configuration.**
 - **Server should send RDMA NIC info to client.**
 - **No need to specify RDMA NIC information to smb.conf.**

Status update – Multichannel user report

- Ziwei Xie(high-flyer) give this test reports
- performance difference of 3 times for read and 4 times for write on his setup



Status update – SMB-Direct user report

- Yufan Chen give test results for SMB-Direct
- Benchmark tool
 - Framtest (<https://support.dvsus.com/hc/en-us/articles/212925466-How-to-use-frametest>)

- **Server**

CPU: intel Silver 4114 x 2

DRAM: 512GB

NVMe SSD: Kioxia CM6 1.9T x 9(mdadm raid0 with XFS)

NIC: MCX516A-CCAT 100GbE

- **Client**

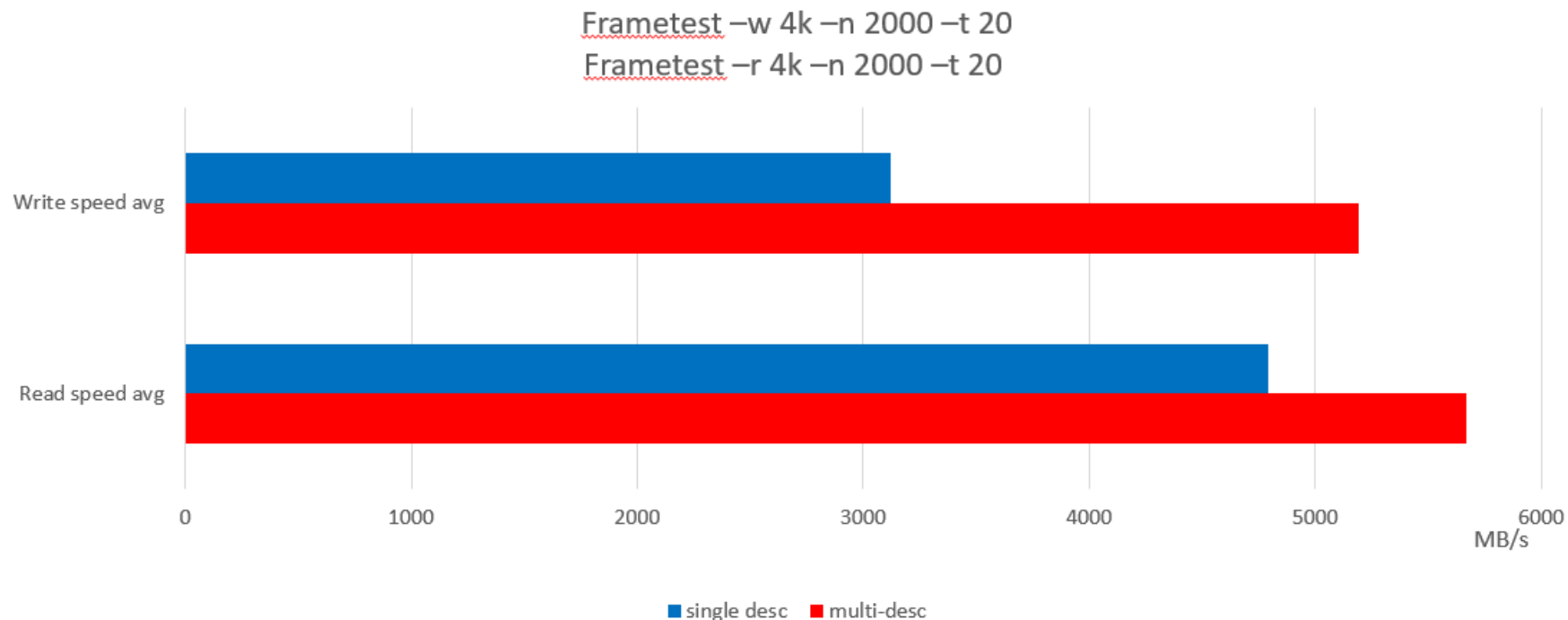
CPU: intel Silver 4215 x 2

DRAM: 64GB

NIC: MCX516A-CCAT

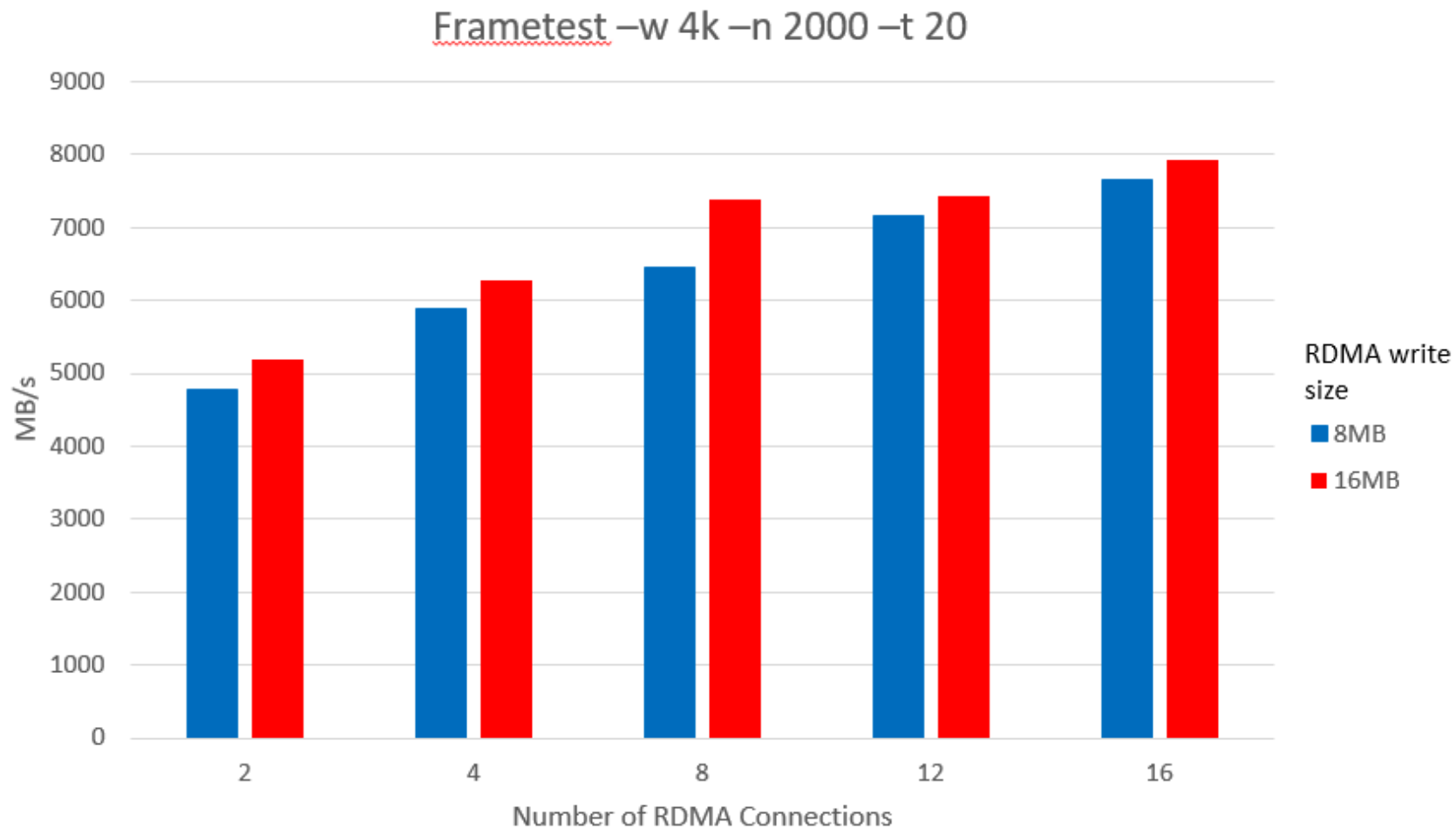
Status update – SMB-Direct user report

- Performance comparison between single and multi-descriptor



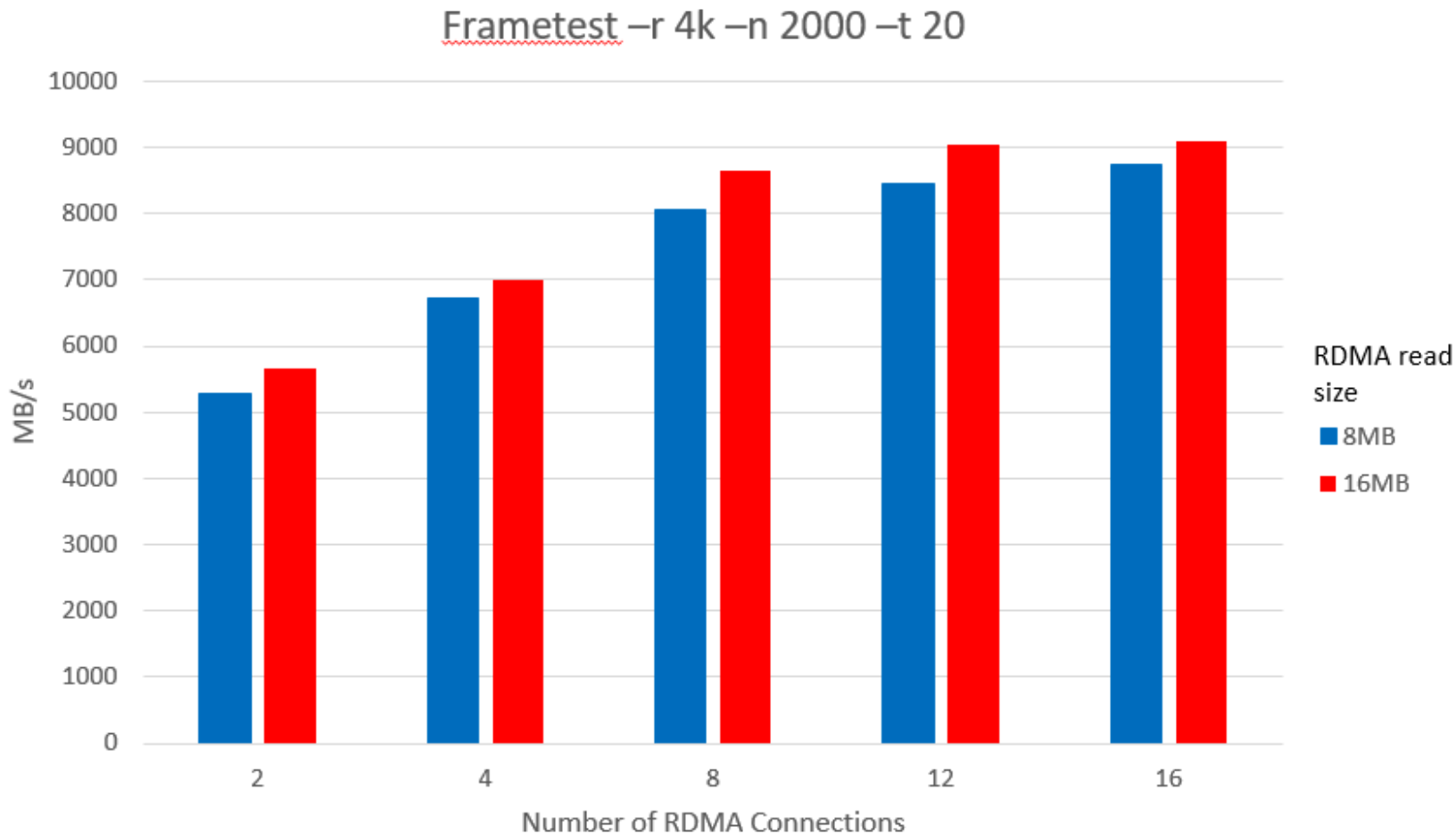
Status update – Fully support SMB-Direct

- RDMA write performance per number of connections



Status update – Fully support SMB-Direct

- RDMA read performance per number of connections



Status update – ksmbd-tools

- **Renamed smb.conf to ksmbd.conf**
- **Atte Heikkilä refactored ksmbd-tools codes**
- **Add manpages for all utils and ksmbd.conf**
- **Added new parameters:**
 - **durable handles**
 - **crossmnt**
 - **smb3 encryption**
 - **smbd io size**
 - **smb2 max credits**

Working on the following

- **ksmbd over QUIC introduced by Xin Long**
- **Reply operations support as next step of durable handle**

Progress on Linux kernel client

(cifs.ko update)

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 (ie 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 SMB2) Default: 32767 (int)
parm: dir_cache_timeout:Number of seconds to cache directory contents for which we have a lease. Default: 30 (int)
parm: slow_rsp_threshold:Amount of time (in seconds) to wait before logging that a response is delayed. Default: 30 (int)
parm: enable_oplocks:Enable or disable oplocks. Default: y/Y/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 ~]# modinfo cifs
```

```
[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 (now can update on active mounts)

- Password rotation (key rotation) becoming common requirement due to security challenges e.g.

smftestdiag102


Access keys ☆ ...

« 🕒 Set rotation reminder 🔄 Refresh 🗨️ Give feedback

Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.

Remember to update the keys with any Azure resources and apps that use this storage account.
[Learn more about managing storage account access keys](#)


Storage account name

smftestdiag102 

key1 🔄 Rotate key

Last rotated: 4/9/2024 (0 days ago)

Key



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)

```
[root@fedora29 ~]# stat /mnt1
stat: cannot stat '/mnt1': Host is down
[root@fedora29 ~]# stat /mnt2
  File: /mnt2
  Size: 0          Blocks: 0          IO Block: 1048576 directory
Device: 32h/50d Inode: 7957636952732887613  Links: 2
Access: (0777/drwxrwxrwx)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2024-04-09 15:50:30.899129800 -0500
Modify: 2024-04-09 15:50:30.899129800 -0500
Change: 2024-04-09 15:50:30.899129800 -0500
```

```
0.150.38.8 Uses: 1 Capability: 0x300057  Session Status: 3 password no longer valid
: RawNTLMSSP SessionId: 0x721c80388000e11 encrypted
User: 0 Primary channel: DISCONNECTED
```

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)
Access: 2024-04-09 15:50:30.899129800 -0500
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: \\smftestdiag102.file.core.windows.net Send error in SessSetup = -13
[ 1160.094753] CIFS: Status code returned 0xc000006d STATUS_LOGON_FAILURE
[ 1160.094800] CIFS: VFS: \\smftestdiag102.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 ~]# █
```

Summary of Features added to allow password rotation

- 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 the 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
 - <https://lwn.net/Articles/937239/>
 - Also nice to see move to larger block sizes:
<https://lwn.net/Articles/945646/>
- A lot of this is thanks to David Howells!
 - Move VM, pagecache & folios out of individual filesystems to shared code (“netfs”)
 - Filesystem just supplies read & write ops to/from iterators
 - Interleaving reads between local cache, servers
 - Write caching now being done even for file open only for write
 - Many perf improvements to read and write caching (e.g. read caching no longer holds up page updates).
 - fsstress e.g. 5-10% faster since 6.9 kernel over smb3.1.1 mounts

Folios/Netfs/MM caching improvements

- David Howells added large patch set for netfs/cifs.ko in 6.10 kernel which e.g. better handled retries, and improved credit renegotiation
- Support for write-through caching and support for vectored writes
- Currently working on set of even more perf improvements for 6.11-rc
- And after that adding support for content crypto (in devel)

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 (thank you Paulo!)
- 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
 - ioctl “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

6.8 kernel (March 2024) (cifs.ko version: 2.47)

- 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 (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 and perf improvement for avoiding lease breaks on compounded operations
- 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

6.10 kernel (July 2024) (cifs.ko vers: 2.49)

- Various performance improvements related to folios (cached i/o) as move to using netfs more fully
- Reenable swapfile support over SMB3 mounts
- Add support for creating sockets with “sfu” mount option

6.11-rc7 kernel (September 2024) (cifs.ko vers: 2.50)

- Add various dynamic tracepoints to improve debugging
- Many important netfs/folios (cached read and write) and fallocation fixes and improvements
- Important RDMA (smbdirect) fixes

Recent Debugging Improvements


- More dynamic trace points added
- Ability to query the session id and tid for a mount (e.g. “smbinfo getconninfo /mnt”) which can help correlate with /proc/fs/cifs/DebugData
- 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 100 dynamic trace points 16 months ago ...

```
root@smfrench-ThinkPad-P52:/sys/kernel/tracing/events/cifs# ls
cifs_flush_err          smb3_flush_err          smb3_posix_mkdir_enter  smb3_ses_expired
cifs_fsync_err          smb3_fsctl_err          smb3_posix_mkdir_err    smb3_set_credits
enable                  smb3_hardlink_done      smb3_posix_query_info_compound_done  smb3_set_eof
filter                  smb3_hardlink_enter     smb3_posix_query_info_compound_enter  smb3_set_eof_done
smb3_add_credits        smb3_hardlink_err       smb3_posix_query_info_compound_err     smb3_set_eof_enter
smb3_adj_credits        smb3_hdr_credits        smb3_query_dir_done      smb3_set_eof_err
smb3_close_done        smb3_insufficient_credits  smb3_query_dir_enter     smb3_set_info_compound_done
smb3_close_enter       smb3_lease_done         smb3_query_dir_err       smb3_set_info_compound_enter
smb3_close_err         smb3_lease_err          smb3_query_info_compound_done          smb3_set_info_compound_err
smb3_cmd_done          smb3_lease_not_found    smb3_query_info_compound_enter          smb3_set_info_err
smb3_cmd_enter         smb3_lock_err           smb3_query_info_compound_err           smb3_slow_rsp
smb3_cmd_err           smb3_mkdir_done         smb3_query_info_done      smb3_tcon
smb3_connect_done      smb3_mkdir_enter        smb3_query_info_enter     smb3_tdis_done
smb3_connect_err       smb3_mkdir_err          smb3_query_info_err       smb3_tdis_enter
smb3_credit_timeout    smb3_nblk_credits       smb3_read_done            smb3_tdis_err
smb3_delete_done       smb3_notify_done        smb3_read_enter           smb3_too_many_credits
smb3_delete_enter      smb3_notify_enter       smb3_read_err             smb3_wait_credits
smb3_delete_err        smb3_notify_err         smb3_reconnect            smb3_waitff_credits
smb3_enter             smb3_open_done          smb3_reconnect_detected   smb3_write_done
smb3_exit_done         smb3_open_enter         smb3_reconnect_with_invalid_credits    smb3_write_enter
smb3_exit_err          smb3_open_err           smb3_rename_done          smb3_write_err
smb3_falloc_done       smb3_oplock_not_found   smb3_rename_enter         smb3_zero_done
smb3_falloc_enter      smb3_overflow_credits   smb3_rename_err           smb3_zero_enter
smb3_falloc_err        smb3_partial_send_reconnect  smb3_rmdir_done          smb3_zero_err
smb3_flush_done        smb3_pend_credits       smb3_rmdir_enter
smb3_flush_enter       smb3_posix_mkdir_done   smb3_rmdir_err
root@smfrench-ThinkPad-P52:/sys/kernel/tracing/events/cifs# ls | wc
   102    102   1877
```

To 127 now (27 more eBPF trace points added)

```
root@smfrench-ThinkPad-P52:/sys/kernel/tracing/events/cifs# ls
cifs_flush_err          smb3_get_reparse_compound_done  smb3_posix_mkdir_err          smb3_set_eof_done
cifs_fsync_err          smb3_get_reparse_compound_enter  smb3_posix_query_info_compound_done  smb3_set_eof_enter
enable                  smb3_get_reparse_compound_err    smb3_posix_query_info_compound_enter  smb3_set_eof_err
filter                  smb3_hardlink_done               smb3_posix_query_info_compound_err    smb3_set_info_compound_done
smb3_add_credits        smb3_hardlink_enter              smb3_qfs_done                    smb3_set_info_compound_enter
smb3_adj_credits        smb3_hardlink_err                smb3_query_dir_done              smb3_set_info_compound_err
smb3_clone_done         smb3_hdr_credits                 smb3_query_dir_enter              smb3_set_info_err
smb3_clone_enter        smb3_insufficient_credits         smb3_query_dir_err                smb3_set_reparse_compound_done
smb3_clone_err          smb3_ioctl                        smb3_query_info_compound_done      smb3_set_reparse_compound_enter
smb3_close_done         smb3_key_expired                 smb3_query_info_compound_enter      smb3_set_reparse_compound_err
smb3_close_enter        smb3_lease_done                  smb3_query_info_compound_err        smb3_shutdown_done
smb3_close_err          smb3_lease_err                   smb3_query_info_done              smb3_shutdown_enter
smb3_cmd_done           smb3_lease_not_found             smb3_query_info_enter              smb3_shutdown_err
smb3_cmd_enter          smb3_lock_err                    smb3_query_info_err                smb3_slow_rsp
smb3_cmd_err            smb3_mkdir_done                  smb3_query_wsl_ea_compound_done      smb3_smbd_connect_done
smb3_connect_done       smb3_mkdir_enter                 smb3_query_wsl_ea_compound_err       smb3_smbd_connect_err
smb3_connect_err        smb3_mkdir_err                   smb3_read_done                     smb3_tcon
smb3_copychunk_done     smb3_mknod_done                  smb3_read_enter                     smb3_tcon_ref
smb3_copychunk_enter    smb3_mknod_enter                 smb3_read_err                       smb3_tdis_done
smb3_credit_timeout     smb3_mknod_err                   smb3_reconnect                      smb3_tdis_enter
smb3_delete_done        smb3_nblk_credits                smb3_reconnect_detected              smb3_tdis_err
smb3_delete_enter       smb3_notify_done                 smb3_reconnect_with_invalid_credits  smb3_too_many_credits
smb3_delete_err         smb3_notify_enter                smb3_rename_done                    smb3_wait_credits
smb3_enter              smb3_notify_err                  smb3_rename_enter                   smb3_waitff_credits
smb3_exit_done          smb3_open_done                   smb3_rename_err                     smb3_write_done
smb3_exit_err           smb3_open_enter                  smb3_rmdir_done                     smb3_write_enter
smb3_falloc_done        smb3_open_err                    smb3_rmdir_enter                    smb3_write_err
smb3_falloc_enter       smb3_oplock_not_found            smb3_rmdir_err                      smb3_zero_done
smb3_falloc_err         smb3_overflow_credits            smb3_rw_credits                     smb3_zero_enter
smb3_flush_done         smb3_partial_send_reconnect      smb3_ses_expired                    smb3_zero_err
smb3_flush_enter        smb3_pend_credits                smb3_ses_not_found
smb3_flush_err          smb3_posix_mkdir_done            smb3_set_credits
smb3_fsctl_err          smb3_posix_mkdir_enter           smb3_set_eof
root@smfrench-ThinkPad-P52:/sys/kernel/tracing/events/cifs# ls | wc
129      129      2456
```

Feature work in progress for the Linux client

Section Subtitle

Coming soon (being tested at collocated SMB3.1.1 test event)

- SMB3.1.1 Compression support (thank you Enzo!)
- Support for O_TMPFILE (creating temp files – thank you Jun Ma))
- Improved MacOS SMB3.1.1 interop
- Improve content crypto (great idea David Howells has been investigating)
- Support for additional auth mechanisms (to make it easier to move away from NTLMv2 when not domain joined and able to use sec=krb5)

Userspace tools & cifs-utils

Section Subtitle

- Recent update: cifs-utils version 7.1
- Example of some new features:
 - CLDAP ping to find closest site
 - Updated documentation (descriptions of various mount options e.g. were missing)
 - Improved debugging “smbinfo gettconinfo”
- Various new tools being considered (and new ideas welcome) for mount configuration, improved debugging and more

Testing ... testing ... and more testing ...

Section Subtitle

Hundreds of standard tests run to wide variety of servers. See <http://smb311-linux-testing.southcentralus.cloudapp.azure.com/>

The screenshot shows the CIFS TESTING dashboard for build azure-multichannel/227. The build is finished 3 days ago. The build steps are as follows:

Step	Description	Duration	Status
0	worker_preparation	0 s	worker cifs-testing ready
1	Pull git repos	1 s	'/update-git.sh'
2	Shutting down win16-tester	0 s	'/shutdown-vm.sh win16-tester'
3	Shutting down fedora29-tester	1 s	'/shutdown-vm.sh fedora29-tester'
4	Shutting down ubuntu-btrfs-tester	0 s	'/shutdown-vm.sh ubuntu-btrfs-tester'
5	Restoring image for fedora29-tester	0 s	'/restore-image.sh fedora29-tester ...'
6	Rebooting fedora29-tester	26 s	'/reboot-vm.sh fedora29-tester ...'
7	Build xfstests on fedora29.vm.test	6 s	'ssh fedora29.vm.test ...'
8	Copy Files	1 s	'/copy-files.sh'
9	Build and install new kernel	50 s	'/build-kernel-rpms.sh revision: ...'
10	Rebooting fedora29-tester_1	57 s	'/reboot-vm.sh fedora29-tester ...'
11	Build cifsutils on fedora29.vm.test	30 s	'ssh fedora29.vm.test ...'
12	Initialize xfstests on fedora29.vm.test	1 s	'ssh fedora29.vm.test ...'
13	Run warmup smb3azure generic/024	12 s	'ssh fedora29.vm.test ...'
14	Run xfstest smb3azuremultichan cifs/006	5 s	'ssh fedora29.vm.test ...'
15	Run xfstest smb3azuremultichan cifs/100	4 s	'ssh fedora29.vm.test ...'
16	Run xfstest smb3azuremultichan cifs/103	8 s	'ssh fedora29.vm.test ...'
17	Run xfstest smb3azuremultichan cifs/105	5 s	'ssh fedora29.vm.test ...'

The screenshot shows the CIFS TESTING dashboard for build 222. It displays a list of 309 test runs, each with a unique ID, a description, and a duration. The tests are as follows:

Test ID	Description	Duration
286	Run xfstest smb3sambabtrfs generic/664	6 s 'ssh fedora29.vm.test ...'
287	Run xfstest smb3sambabtrfs generic/670	46 s 'ssh fedora29.vm.test ...'
288	Run xfstest smb3sambabtrfs generic/671	25 s 'ssh fedora29.vm.test ...'
289	Run xfstest smb3sambabtrfs generic/672	3:36 'ssh fedora29.vm.test ...'
290	Run xfstest smb3azuremultichan generic/676	13:35 'ssh fedora29.vm.test ...'
291	Run xfstest smb3azuremultichan generic/694	5 s 'ssh fedora29.vm.test ...'
292	Run xfstest smb3azuremultichan generic/696	5 s 'ssh fedora29.vm.test ...'
293	Run xfstest smb3azuremultichan generic/701	5 s 'ssh fedora29.vm.test ...'
294	Run xfstest smb3azuremultichan generic/708	5 s 'ssh fedora29.vm.test ...'
295	Run xfstest smb3samba generic/728	8 s 'ssh fedora29.vm.test ...'
296	Run xfstest smb3azuremultichan generic/732	5 s 'ssh fedora29.vm.test ...'
297	Run xfstest smb3azuremultichan generic/736	7:01 'ssh fedora29.vm.test ...'
298	Run xfstest smb3azuremultichan generic/737	7 s 'ssh fedora29.vm.test ...'
299	Run xfstest smb3azuremultichan generic/738	1:56 'ssh fedora29.vm.test ...'
300	Run xfstest smb3sambabtrfs generic/742	9 s 'ssh fedora29.vm.test ...'
301	Run xfstest smb3 git/0000	1:08 'ssh fedora29.vm.test ...'
302	Run xfstest smb3azure git/0002	16 s 'ssh fedora29.vm.test ...'
303	Run xfstest smb3mfs git/0003	35 s 'ssh fedora29.vm.test ...'
304	Run xfstest smb3azuremultichan git/0005	21 s 'ssh fedora29.vm.test ...'
305	Run xfstest smb3 git/0022	6 s 'ssh fedora29.vm.test ...'
306	Run xfstest smb3 git/0055	6 s 'ssh fedora29.vm.test ...'
307	Run xfstest smb3sambabtrfs git/3000	10 s 'ssh fedora29.vm.test ...'
308	Run xfstest smb3azuremultichan git/3909	17 s 'ssh fedora29.vm.test ...'
309	Check xfstests results on fedora29.vm.test	1 s 'ssh fedora29.vm.test ...'

Testing improvements

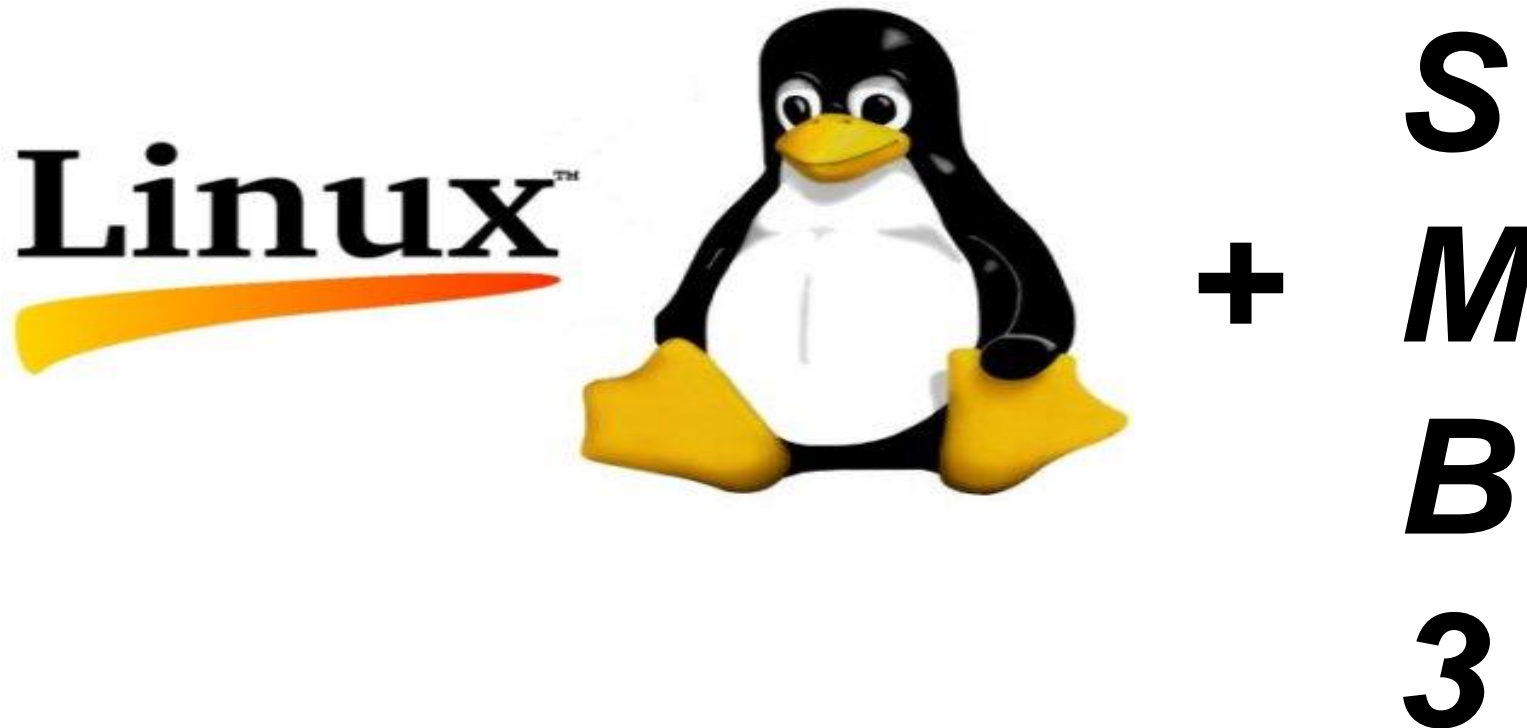
- The ‘buildbot’ has been invaluable in spotting bugs, but was out of commission for part of last year, and takes more developer time
- Investigating alternatives like “lisa” and kdevops to help optimize testing, reduce regressions, do new features faster, improve perf, and also more rapidly get patches backported
- Also trying to investigate how to better auto-run against branches like vfs next branch to reduce regression risk for changes outside of cifs.ko
- Fortunately there are multiple other ‘bots’ that already run tests regularly against cifs.ko
- We welcome new tests – whether added to industry standard Linux “xfstests” or cifs.ko or cifs-utils specific tests or other (e.g. we have added various of the ‘git functional tests’ to our test bots).

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> and test results
 - <http://smb311-linux-testing.southcentralus.cloudapp.azure.com/#/>
- Linux CIFS client <https://wiki.samba.org/index.php/LinuxCIFS>
- For ksmbd also see recent update on kernel server:
https://sambaxp.org/fileadmin/user_upload/sambaxp2024-Slides/sxp24-Jeon-ksmbd_status.pdf
- 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)

Thank you for your time

- Future is very bright!





Please take a moment to rate this session.

Your feedback is important to us.