SambaXP 2026

Better Remote Storage 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

Linux Kernel: A year ago and now …

- Now 7.0-rc7 Still “Baby Opossum Posse”
- Then 6.14
- 84267 changesets (non-merge commits) over period (up)
- 44,392 files changed (up more than 5%)
- 3.7 million insertions, 1.57 million deletions (up more than 5%)
- Is AI causing more frequent changes?
- Kernel is HUGE: 27.9 million lines of code

Linux Storage, Filesystem, MM, BPF summit

- Next month in Croatia. Last spring in Montreal

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

Some Linux FS topics of interest discussed recently

- Testing … testing … and more automated testing … (e.g. kdevops)
- Continue work on folios, netfs mapping layer, variable size pages, fscache, improving page caching
- Dcache (directory entry caching) improvements
- Improvements to delegations and changes to make filesystems opt in to nfs server support
- Improvements to openat2
- Leveraging eBPF more and more (not just dynamic tracing)
- Extending in kernel encryption: add support for QUIC for Linux (helps SMB3.1.1 kernel drivers and userspace drivers like Samba)
- fanotify (filesystem change notifications) improvements
- Shift to cloud and better support for faster storage (NVME) and net (RDMA/smbdirect)

Linux Filesystems Activity over past year (since 6.14 kernel)

- 6879 filesystem related changesets, 8.1% of total kernel changesets over this period even though only 4% of the lines of code! Decrease from last year but still lots of developer attention on filesystems …
  - Linux kernel fs are 1.19 million lines of code total (measured last week)
  - Bcachefs removed from 6.18 kernel (due to tension over development practices) and was very active and that accounts for some of the decrease
    - Kernel development follows strict practices for release timing
- Lots of progress over last few years!
  - Old ntfs fs removed (in favor of NTFS3), Reiserfs marked ‘obsolete’ and removed in 6.13 kernel, system V fs in 6.15, will JFS be next?
  - 6.14 adds support for uncached buffered i/o
  - 6.15 adds support for fanotify for mount/umount events
  - 6.18 introduced per-CPU memory caching (“Sheaves”) which helps folios
  - 7.0 improved large folio support, optimized close_range call, direct i/o improvements and huge perf improvement in file cache memory reclaim. New “statmount()” system call

Most Active Linux Filesystems over the past year

- VFS (mapping layer) 872 changesets (activity doubled!)
- The top filesystems and VFS dominate the activity
- Most active local fs are BTRFS 1229 (up a lot), f2fs 537 (up a lot), XFS 477 (down), ext4 358 (up a lot)
- Most active non-local fs is SMB3.1.1 (cifs.ko client) 582 (up a lot) and ksmbd (server) 286 (up a lot)
- Then NFSD (server) 266 and NFS (client) 303
  - cifs.ko also had many more lines changed. It has been a VERY active year for cifs.ko
- Other: ntfs3 (96), erofs (126, up), GFS2 (158 up), ceph (79, up)

SMB3.1.1 Activity was strong this year

- cifs.ko activity was strong, 582 changesets, up a lot
  - cifs is 67KLOC kernel code (not counting user space utilities)
  - 90% more than NFS, and more lines changed to. Lots of great contributions from many different developers (Thank you!)
- ksmbd activity up a lot
  - 29KLOC kernel code (not counting user space tools and libraries)

Note that Samba server (userspace) is over 4.1 million lines of code (orders of magnitude bigger than the kernel smbd server or any of the NFS servers) and is much more active, and includes key security tools and services (not just file server)

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, reduce traffic
  - 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 (3 recently)

e.g. memory management continues to improve. 238 fs related syscalls now, much more than POSIX had

Syscall name	Kernel Version introduced
Setxattrat, getxattrat, listxattrat, removexattrat	6.13
open_tree_attr	6.15
file_getattr, file_setattr	6.17
And new fs ioctl:	
FS_IOC_GETLBMD_CAP (returns metadata protection info capabilities)	6.17

Remember the interesting new fs: Bcachefs?

- 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
- Key features they emphasized include: copy on write, full metadata and data check-summing, replication, advanced caching, data placement, snapshots, scalable, erasure coding
- But … huge number of fixes each release can be suspicious
- Bcachefs created a substantial amount of drama (detailed discussion on this last year at the LSF/Storage summit), including a brief ban of the lead developer, and then removal of bcachefs from the 6.18 and later kernels
- A good lesson in why following recommended practices is important.

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

- In-person plugfests are back!
- SMB3.1.1 Interoperability Lab restarted, colocated with SDC last 4 years
  - And at SambaXP now as well!
- Many exciting things being tested

Recent improvements in the kernel client

(cifs.ko)

Multichannel continues to improve

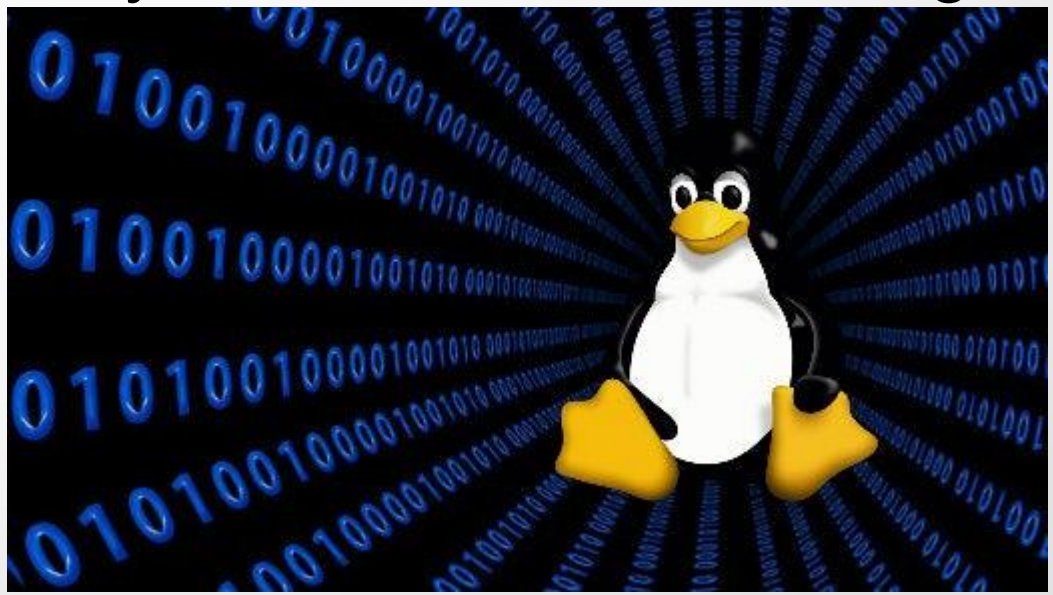
- Multiple Reconnect and Perf improvements including improved channel allocation for SMB3.1.1 requests (thank you Shyam Prasad)
- Soon will be enabled by default (when server supports multiple interfaces or RSS) now that we can asynchronously add channels at mount time (so mount is not slow when some advertised interfaces on server are not reachable and timeout), but multichannel is easy to enable via mount parm

Folios/Netfs/MM caching improvements

- Currently still working on:
  - Improvements to better getting more I/O in flight for sequential reads when multichannel
  - Better support for cached writes
  - Support for content crypto (in devel)

My favorite improvements

- The SMB3.1.1 Linux/POSIX Extensions are now production ready and in both Linux and Samba!
- Cool dir lease perf improvements
- And as Linux continues to evolve, we can continue to add new SMB3.1.1 features to deal with the new syscalls and fsctls and flags
- Keep up the testing!



Dir leases are going to be huge help reducing I/O

- Simple example, ls a couple directories twice, a few seconds apart: “ls /mnt ; ls /mnt/test ; ls /mnt/scratch ; sleep 3 ; echo 0 > /proc/fs/cifs/Stats ; ls /mnt/test; ls /mnt/scratch”
- With directory leases, second ls for each directory took ZERO (!) roundtrips to server – no network traffic
- Max requests in flight: 0
  1) \\localhost\test
  SMBs: 0 since 2025-09-17 05:39:04 UTC
  Open files: 0 total (local), 1 open on server
  2) \\localhost\scratch
  SMBs: 0 since 2025-09-17 05:39:04 UTC
  Open files: 0 total (local), 1 open on server

Without directory leases – lots more I/O sent to server

- 30 requests sent to server instead of zero for the subsequent “ls” of the two directories
- Max requests in flight: 3
  1) \\localhost\test
  SMBs: 17 since 2025-09-17 05:41:01 UTC
  Open files: 0 total (local), 0 open on server
  Creates: 5 total 0 failed
  Closes: 5 total 0 failed
  QueryDirectories: 3 total 0 failed
  QueryInfos: 4 total 0 failed
  2) \\localhost\scratch
  SMBs: 13 since 2025-09-17 05:41:01 UTC
  Bytes read: 0 Bytes written: 0
  Creates: 4 total 0 failed
  Closes: 4 total 0 failed
  QueryDirectories: 2 total 0 failed
  QueryInfos: 3 total 0 failed

Debugging dir leases is easier now too

- Note: max_cached_dirs default is 16 but for most use cases can be set very safely on mount much higher
- # cat /proc/fs/cifs/open_dirs
  # Version:1
  # Format:
  # <tree id> <sess id> <persistent fid> <path>
  Num entries: 1
  0x98dc529b 0xbad60079 0x545c7ba5 valid file info, valid dirents
  Num entries: 16
  0xfe306dde 0xbad60079 0x20608b0e /test valid file info, valid dirents
  0xfe306dde 0xbad60079 0xca4f5361 /test-637 valid file info, valid dirents
  0xfe306dde 0xbad60079 0x8cf00b0b /ftrunc valid file info, valid dirents
  0xfe306dde 0xbad60079 0xa8af34cb /idmapped_mounts_1 valid file info, valid dirents
  0xfe306dde 0xbad60079 0x97a44c7b /test-565 valid file info, valid dirents
  0xfe306dde 0xbad60079 0xdc584179 /test-434 valid file info, valid dirents
  0xfe306dde 0xbad60079 0x772577c /test-433 valid file info, valid dirents
  0xfe306dde 0xbad60079 0x55bcbdab /fsx valid file info, valid dirents
  0xfe306dde 0xbad60079 0xd3a51a30 /tmp valid file info, valid dirents
  0xfe306dde 0xbad60079 0xa712c488 /21763/sub valid file info, valid dirents
  0xfe306dde 0xbad60079 0xddc966e3 /21763 valid file info, valid dirents

New mount parms

- upcall_target= (mount, app): helps with namespace use cases
- nativesocket
- nbsessinit: (force initialization of NetBIOS session even when not port 139)
- nounicode (relevant for smb1 only)
- pass2= (as synonym for “password2=“)
- symlink= *default, none, native, unix, mfsymlinks, sfu, nfs, wsl)
- symlinkroot= (adjust non-POSIX symlink target for absolute symlinks)
- reparse=none (allow disabling saving special files via reparse point)

Let’s make RDMA easy to use!

- The RDMA (“smbdirect”) code in the kernel client and server are being rewritten (thank you Metze!) to use common code in a new module
- This also allows (“smbdirect”) to be used for userspace tools and services (like Samba e.g.) but also makes it MUCH easier for apps to use RDMA (to benefit from low latency and high throughput) even if they don’t use SMB3.1.1

6.13 kernel (January 2025) (cifs.ko vers: 2.52)

- New mount option “upcall_target” to allow handling different namespaces (needed e.g. for some container workloads)
- Performance improvement to reuse deferred close file handles for writes (not just reads, and stat)
- Handle password rotation cases where password on mount has just expired, but password2 specified on mount is valid
- SMB3.1.1 Linux/POSIX extensions improvements (support for chmod added and for querying special file types)

6.14 kernel (March 2025) (cifs.ko vers: 2.53)

- Various netfs (data caching) enhancements and fixes, including better parallelizing reads
- Improved reparse point (special file handling)
- Multichannel channel selection perf improvement (thank you Shyam!)
- Can now negotiate IAKerb (if userspace tools configured to get it)
- New mount options “reparse=none” to allow disabling using reparse points to create special files, and “symlink=“ to allow choosing the way symlinks will be created by default for the mount

6.15 kernel (May 2025) (cifs.ko vers: 2.54)

- Added new pseudo-xattrs to allow retrieving just the SACL (e.g. system.smb3_ntsd_sacl) or owner (system.smb3_ntsd_owner) or DACL
- Improve RFC1001 handling (e.g. if connecting over port 139) and also various SMB1 fixes (Thank you Pali!)
- Multichannel fixes (Thank you Bharath et al!)
- Path name processing perf improvement (thank you Paulo!)

6.16 kernel (July 2025) (cifs.ko vers: 2.55)

- Big perf improvement for directory leases (repeated “ls” of a directory now reuses cached content, improving perf), and to set ParentLeaseKey when create/open directories & files (to avoid unneeded lease breaks), and dir lease fixes.
- netfs fixes (thanks David!) and improved netfs tracepoints
- Multichannel reconnect improvements (thanks Shyam!) and improve channel selection, and fix sec=krb5 (was doing single channel)
- Improve deferred close
- SMB3.1.1 POSIX should not remap chars, also fix ls file type parse
- Fix posix to not remap special characters
- Fix ‘ls’ failure due to invalid symlink, fix native symlink traversal

6.17 kernel (September 2025) (cifs.ko vers: 2.56)

- Dir lease perf improvement (kept open based on last access time) and important dir lease fix
- Improve POSIX symlink and socket handling (default to reparse tags)
- RDMA/smbdirect improvements
- Important fixes for rename and unlink races
- Improved debugging including add new tracepoints tracking leases
- Multichannel channel allocation fix
- Add new tracepoint for lease break notifications

6.18 kernel (November 2025) (cifs.ko vers: 2.57)

- Most active release for cifs.ko (174 changesets!), largely due to many smbdirect/rdma changes (thank you Metze!)
- Smbdirect changes include introducing struct smbdirect_socket and converting to use it in both client and server. Fix iWARP and also fix some disconnect problems
- Multichannel reconnect fix
- Crypto library improvements (improved perf)
- Debugging improvements

6.19 kernel (February 2026) (cifs.ko vers: 2.58)

- Add tracepoint for krb5 auth
- Reconnect backoff improvements
- Allow changing multichannel options on remount (thanks Rajasi!) and fix conflicting multichannel mount option ordering
- Misc cleanup patches
- Important fix for smbdirect credit handling regression

7.0 kernel (April 2026) (cifs.ko vers: 2.59)

- Very active release (2nd biggest for cifs.ko)!
- Many smbdirect/RDMA fixes including for credit handling issues, fixing client timeouts and useless network traffic, also added server support for mixed iWARP and ROCE interfaces
- Many security fixes, probably helped by increase in use of AI
  - AI tools now autoanalyze patches sent to the linux-cifs mailing list (e.g. https://sashiko.dev/#/?list=org.kernel.vger.linux-cifs)
- Deferred close umount fix
- Multichannel fixes, including channel addition is now async at mount
- Error mapping cleanup and perf improvements

7.1 kernel (expected June 2026) (cifs.ko vers: 2.60)

- Add support for O_TMPFILE (creating temporary files)
- Move smbdirect code (client and server) into smbdirect.ko helper module (7.2 kernel will make the smbdirect kernel driver useable by userspace applications)
- Fixing remount to handle properly all valid mount parms
- Dir lease perf improvements

Planned and in-progress Improvements (To-dos)

(cifs.ko)

Coming soon

- Support for O_TMPFILE (creating temp files – thank you Paulo)). Will be in 7.1-rc1
- Improving directory entry caching (better leveraging directory leases reduce metadata i/o)
- Improved use of compounding (e.g. for “ls” could save a roundtrip)
- Support for SMB3.1.1 over QUIC
- New common RDMA/smbdirect code, helper module shared by ksmbd, cifs.ko and userspace such as Samba server and tools (Thank you Metze!)
- SMB3.1.1 Compression support (thank you Enzo!)
- Improved MacOS SMB3.1.1 interop
- Improve content crypto (great idea David Howells has been investigating)
- Rewrite of the way cifs.ko calls TCP (improves perf, thank you David Howells)
- Support for additional auth mechanisms (to make it easier to move away from NTLMv2 when not domain joined and able to use sec=krb5)

Coming soon

- Rajasi has sent a set of patches to check all mount parameters and ensure that they are correctly handled on remount. Good work!
- Shyam has a large set of directory lease patches that solve some key performance problems. Very exciting!
  - Send two querydir requests compounded together when doing “ls” to avoid an extra roundtrip on all but large directories (as the 2nd querydir will return rc)
  - Increase size of querydir buffers (currently only 16K) to reduce roundtrips
  - Increase maximum cached dirs (currently defaults to 16 although changeable)
  - Better use pagecache to cache directory entries
  - Handle cases where files are created on a client in a directory with a dir lease (to be able to still leverage a lease when files added or deleted)

Recent Debugging Improvements

- Two more dynamic trace points added
- Many more debugging tools now available (see Meetakshi's presentation at SDC for more details)
- Pseudo file for debugging dir leases added: /proc/fs/cifs/open_dirs and also can now clear directory cache with new module parm:

  "echo 1 > /sys/module/cifs/parameters/drop_dir_cache"

To 130 now (2 more eBPF trace points added)

```
root@smfrench-ThinkPad-P16s-Gen-2:~# ls /sys/kernel/tracing/events/cifs
cifs_flush_err                  smb3_falloc_err                 smb3_nblk_credits               smb3_query_wsl_ea_compound_done smb3_shutdown_done
cifs_fsync_err                  smb3_flush_done                 smb3_notify_done                smb3_query_wsl_ea_compound_enter smb3_shutdown_enter
enable                          smb3_flush_enter                smb3_notify_enter               smb3_query_wsl_ea_compound_err  smb3_shutdown_err
filter                          smb3_flush_err                  smb3_notify_err                 smb3_read_done                  smb3_slow_rsp
smb3_add_credits                smb3_fsctl_err                  smb3_open_done                  smb3_read_enter                 smb3_smbd_connect_done
smb3_adj_credits                smb3_get_reparse_compound_done  smb3_open_enter                 smb3_read_err                   smb3_smbd_connect_err
smb3_clone_done                 smb3_get_reparse_compound_enter smb3_open_err                   smb3_reconnect                  smb3_tcon
smb3_clone_enter                smb3_get_reparse_compound_err   smb3_oplock_not_found          smb3_reconnect_detected         smb3_tcon_ref
smb3_clone_err                  smb3_hardlink_done              smb3_overflow_credits           smb3_reconnect_with_invalid_credits smb3_tdis_done
smb3_close_done                 smb3_hardlink_enter             smb3_partial_send_reconnect     smb3_rename_done                smb3_tdis_enter
smb3_close_enter                smb3_hardlink_err               smb3_pend_credits               smb3_rename_enter               smb3_tdis_err
smb3_close_err                  smb3_hdr_credits                smb3_posix_mkdir_done           smb3_rename_err                 smb3_too_many_credits
smb3_cmd_done                   smb3_insufficient_credits       smb3_posix_mkdir_enter          smb3_rw_credits                 smb3_unlink_done
smb3_cmd_enter                  smb3_ioctl                      smb3_posix_mkdir_err            smb3_ses_expired                smb3_unlink_enter
smb3_cmd_err                    smb3_kerberos_auth              smb3_posix_query_info_compound_done smb3_ses_not_found          smb3_unlink_err
smb3_connect_done               smb3_key_expired                smb3_posix_query_info_compound_enter smb3_set_credits           smb3_unsupported_ioctl
smb3_connect_err                smb3_lease_ack_done             smb3_posix_query_info_compound_err smb3_set_eof                 smb3_wait_credits
smb3_copychunk_done             smb3_lease_ack_err              smb3_qfs_done                   smb3_set_eof_done               smb3_waitff_credits
smb3_copychunk_enter            smb3_lease_break_enter          smb3_query_dir_done             smb3_set_eof_enter              smb3_write_done
smb3_copychunk_err              smb3_lease_not_found            smb3_query_dir_enter            smb3_set_eof_err                smb3_write_enter
smb3_credit_timeout             smb3_lock_err                   smb3_query_dir_err              smb3_set_info_compound_done     smb3_write_err
smb3_eio                        smb3_mkdir_done                 smb3_query_info_compound_done   smb3_set_info_compound_enter    smb3_zero_done
smb3_enter                      smb3_mkdir_enter                smb3_query_info_compound_enter  smb3_set_info_compound_err      smb3_zero_enter
smb3_exit_done                  smb3_mkdir_err                  smb3_query_info_compound_err    smb3_set_info_err               smb3_zero_err
smb3_exit_err                   smb3_mknod_done                 smb3_query_info_done            smb3_set_reparse_compound_done
smb3_falloc_done                smb3_mknod_enter                smb3_query_info_enter           smb3_set_reparse_compound_enter
smb3_falloc_enter               smb3_mknod_err                  smb3_query_info_err             smb3_set_reparse_compound_err
```

Recent improvements in the user space tools

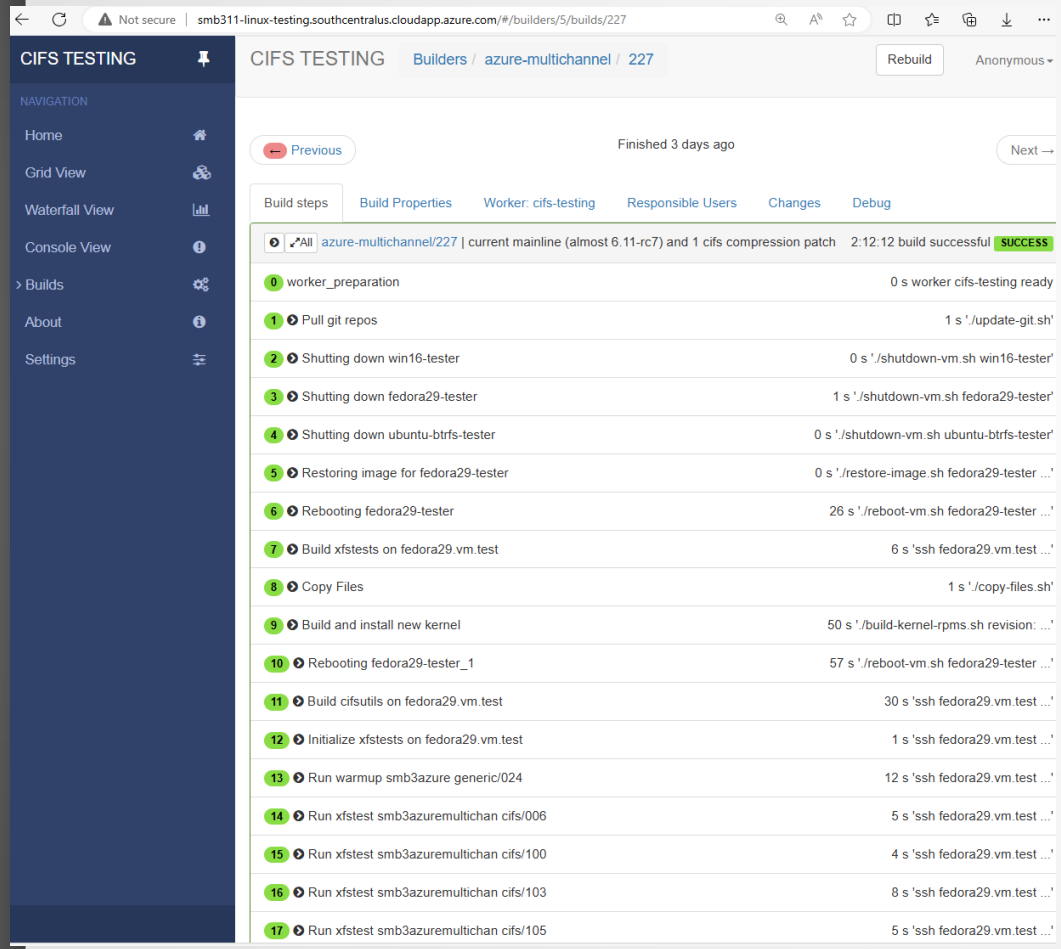
(cifs-utils)

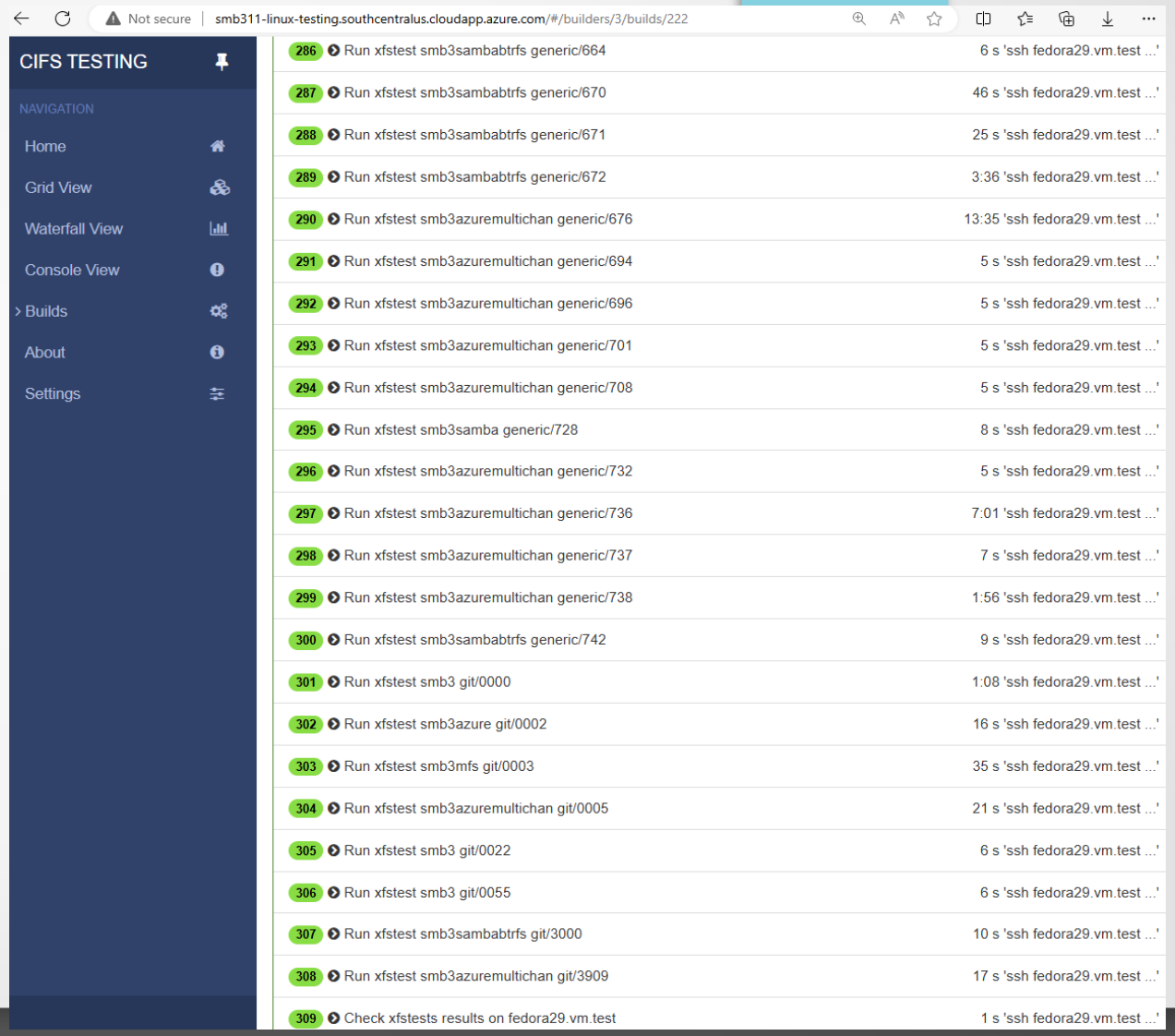
- Example of features that will be in 7.6: add “smbinfo notify” subcommand
- cifs-utils version 7.3, 7.4, 7.5 (minor updates) March and June 2025
  - Includes retry mount on EINPROGRESS fix and unspecified upcall target defaults to app not host namespace. Strongly recommend running cifs-utils 7.4 or later
- Example of features in 7.2
  - Add support for password rotation (e.g. “password2=“)
  - Add support for cifs.upcall to choose type of namespace resolution (“upcall_target”)
  - Allow setting timeout for keys in cifscreds
  - Improved documentation
- Various new tools being considered (new ideas welcome) for mount configuration, improved debugging and more

Testing Improvements

Test … test … test ...

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





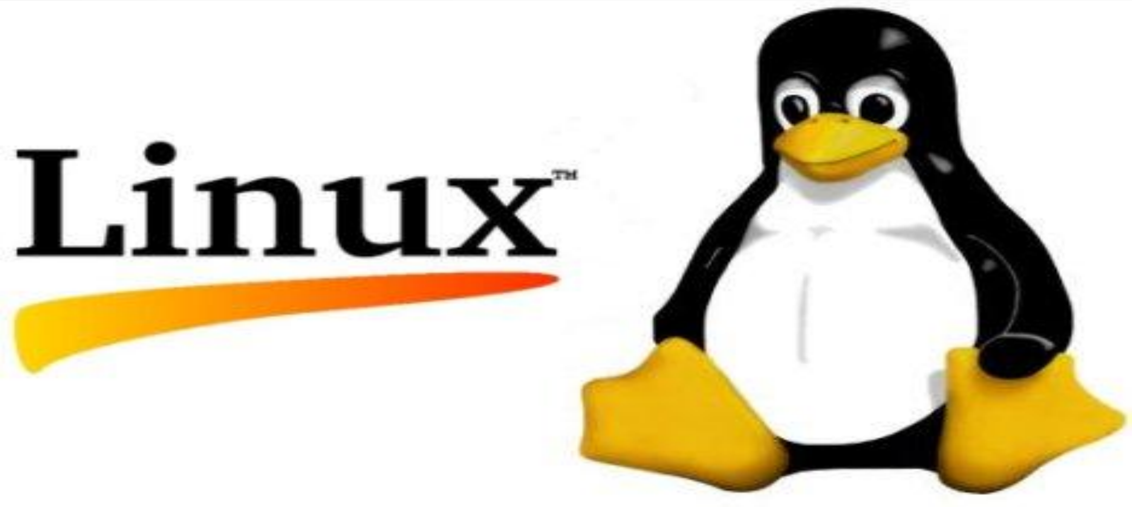
Additional tests are encouraged (generic or smb specific)

- Xfstests are the standard Linux filesystem functional tests
- Over last ten months added 13 to main “cifs-testing” regression testing group (311 tests run on every checkin from this group) Various server specific groups have added even more (also have added two new test groups)
- Azure SMB3.1.1 multichannel: added 12 more tests, now 169 tests
  - e.g. http://smb311-linux-testing.southcentralus.cloudapp.azure.com/#/builders/8/builds/274
- Windows server target added 12 more tests, now includes 156 tests
  - e.g. http://smb311-linux-testing.southcentralus.cloudapp.azure.com/#/builders/6/builds/78
- 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’)

- The ‘buildbot’ has been invaluable in spotting bugs, but was out of commission for part of last year, and takes more developer time
- There are alternatives like “lisa” (thank you Piyush!) and kdevops to help optimize testing, reduce regressions, do new features faster, improve perf, and also more rapidly get patches backported. Lisa is now starting to be used to verify backports.
- 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 ‘git functional tests’ to our test bots). See https://github.com/bharathsm-ms/xfstests-dev tests/cifs directory e.g. (for cifs specific tests, cifs/100 through cifs/312)

Thank you for your time

- Future is very bright!



+ S
M
B
3

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
- Samba-technical mailing list
- And various presentations at http://www.sambaxp.org and Microsoft Learn (learn.microsoft.com) and of course SambaXP archives and 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
  - Userspace client tools code: https://git.samba.org/?p=cifs-utils.git (master branch)
  - Kernel server code: https://git.samba.org/ksmbd.git/?p=ksmbd.git (ksmbd-for-next branch)