STORAGE DEVELOPER CONFERENCE
SDC 22
Fremont, CA
September 12-15, 2022
BY Developers FOR Developers

A SNIA Event

Accessing Files Remotely with Linux

Recent Progress with the SMB3.1.1 client and servers and where do we go from here?

Presented by: Steven 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, various SMB3/CIFS based NAS appliances and the Cloud (Azure)
  - Co-maintainer of the kernel server ("ksmbd")
- Member of the Samba team, coauthor of SNIA CIFS Technical Reference, former SNIA CIFS Working Group chair
- Principal Software Engineer, Azure Storage: Microsoft

Outline

- Overview of recent Linux FS activity
- Overview of SMB3
- Client Progress
  - Kernel
  - Utilities
- Server Progress
- Testing!
- Where do we go from here?

Back to in person: LSF/MM/eBPF summit helped a lot

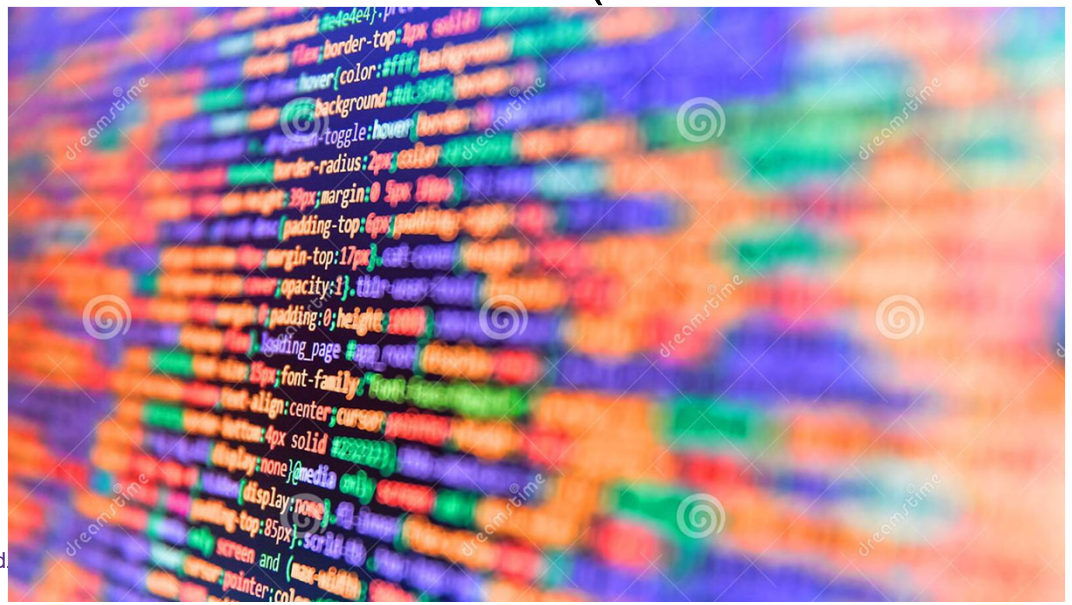
- May 2022
- Linux kernel storage and MM maintainers
- LWN has good summary of topics that were covered

Some Linux FS topics of interest from LSF and other recent discussions

- Folios, netfs and the redesign of the page cache
- Fscache redesign
- Improvements to statx and fsinfo and to inotify/fanotify
- Idmapped mounts
- Extending in kernel encryption: TLS handshake (for NFS) and QUIC (SMB3.1.1 and other)
- io_uring (async i/o improvements)
- Shift to cloud
- Better support for faster storage (NVME) and net (RDMA/smbdirect)

Linux Filesystems Activity over past year (since 5.14)

- 5608 filesystems changesets (up slightly) 6.7% of total kernel changesets, one of the most watched parts of the kernel (FS Activity up slightly as percentage of kernel activity)
- Linux kernel fs are 1.06 million lines of code (measured this week)



A year ago and now ...

- Now: 6.0-rc3 “Hurr durr I’ma ninja sloth”

- Then: 5.14 “Opossums on Parade”

Most Active Linux Filesystems over the past year

- VFS (mapping layer) 413 changesets (down a lot)
- Seven filesystems (and VFS layer itself) dominate the activity
- Most active are BTRFS 986 (activity increasing), XFS 507 (flat)
- NFS (306). Ext4 (270) and SMB3.1.1 (cifs.ko) (260 changesets)
  - It has been a VERY active year for cifs.ko
- Other:
  - ksmbd (new, added in the 5.15 kernel) (239), nfsd (216), ntfs3 (new, added in 5.15) 178, ceph (164), gfs2 (136)
- By lines changed: btrfs had most over this period, then in order xfs, ksmbd (new fs), ntfs3 (new fs), cifs.ko, nfs

SMB3.1.1 Activity was strong this year

- cifs.ko activity was strong, 260 changesets
  - cifs is now > 60KLOC kernel code (not counting user space utilities), 3% larger than a year ago
  - “git diff v5.14.. fs/cifs fs/smbfs_common | wc” is 22.8K
- ksmbd activity was also strong
  - Introduced in the 5.15 kernel, 27KLOC kernel code, 239 changesets since its introduction
- Samba server (userspace) is over 3.8 million lines of code (orders of magnitude bigger than the kernel smbd server or any of the NFS servers) and is even more active

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

- In-person plugfests are back!
- SMB3.1.1 plugfest collocated with SDC here
- Many exciting things being tested

Examples of recent kernel client features

Signing algorithm negotiation – allowing faster signing

- # modinfo cifs | grep signing
  parm: enable_negotiate_signing:Enable negotiating packet signing algorithm with server. Default: n/N/0 (bool)
- “insmod cifs.ko enable_negotiate_signing”

```
root@smfrench-ThinkPad-P52:~# cat /proc/fs/cifs/DebugData
Display Internal CIFS Data Structures for Debugging
---------------------------------------------------
CIFS Version 2.38
Features: DFS,FSCACHE,STATS,DEBUG,ALLOW_INSECURE_LEGACY,CIFS_POSIX,UPCALL(SPNEGO),X
WITNESS
CIFSMaxBufSize: 16384
Active VFS Requests: 0

Servers:
1) ConnectionId: 0x1 Hostname: localhost
Number of credits: 8190 Dialect 0x311 signed (AES-GMAC) nosharesock
TCP status: 1 Instance: 1
 Local Users To Server: 1 SecMode: 0x1 Req On Wire: 0
In Send: 0 In MaxReq Wait: 0

        Sessions:
        1) Address: 127.0.0.1 Uses: 1 Capability: 0x300047      Session Status: 1
        Security type: RawNTLMSSP  SessionId: 0x92cb01b signed (AES-GMAC)
        User: 0 Cred User: 0

        Shares:
        0) IPC: \\localhost\IPC$ Mounts: 1 DevInfo: 0x0 Attributes: 0x0
```


Signing changes

- 6.0 and earlier kernels /proc/fs/cifs/DebugData showed:
  Security type: RawNTLMSSP SessionId: 0x5f08b08 signed
- Now about 20% faster performance if workload not network constrained (thank you Enzo!)

Security type: RawNTLMSSP SessionId: 0x5f08b08 signed (AES-GMAC)

Or if server doesn't support GMAC will fall back to:
Security type: RawNTLMSSP SessionId: 0x5f08b08 signed (AES-CMAC)

Example perf #s

- “dd if=/dev/zero of=/mnt/target bs=4M count=256”
- Signing (default prior to patches): 280MB/sec
- Signing (GMAC, with these patches): 310MB/sec
- Encryption (vers=3.0, CCM): 170MB/sec
- Encryption (vers=3.1.1 GCM): 1.1GB/sec
- (testing on my laptop yesterday)

Directory Caching Improvements

- Thanks to Ronnie Sahlberg, directory caching and use of directory leases (to improve metadata caching even more, and safely) is MUCH improved
- Huge perf win!
- Testing going now at plugfest
- Target - next merge window (6.1 Linux kernel)

Notice cached directory information with lease reduces requests needed (stat does not need to be sent)

```
root@smfrench-Virtual-Machine:~# ls /mnt/test/tmp ; stat /mnt/test/tmp/populate_root
populate_root
  File: /mnt/test/tmp/populate_root
  Size: 0               Blocks: 0          IO Block: 1048576 directory
Device: 2bh/43d Inode: 4222124650717072  Links: 2
Access: (0755/drwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2022-09-14 05:16:20.290862200 -0500
Modify: 2022-02-21 14:20:58.000000000 -0600
Change: 2022-02-21 14:21:24.698796900 -0600
 Birth: 2022-02-21 14:21:03.408893400 -0600
```


smb2

No.	Time	Source	Destination	Protocol	Length	Info
2	0.047976613	172.30.33.95	172.30.32.1	SMB2	416	Create Request File: tmp;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO;Close Request
3	0.048522239	172.30.32.1	172.30.33.95	SMB2	584	Create Response File: tmp;GetInfo Response;Close Response
5	0.048769215	172.30.33.95	172.30.32.1	SMB2	408	Create Request File: tmp;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO
6	0.049323137	172.30.32.1	172.30.33.95	SMB2	536	Create Response File: tmp;GetInfo Response
7	0.049490653	172.30.33.95	172.30.32.1	SMB2	322	Create Request File: tmp;Find Request SMB2_FIND_ID_FULL_DIRECTORY_INFO Pattern: *
8	0.049983206	172.30.32.1	172.30.33.95	SMB2	640	Create Response File: tmp;Find Response
9	0.050158418	172.30.33.95	172.30.32.1	SMB2	170	Find Request File: tmp SMB2_FIND_ID_FULL_DIRECTORY_INFO Pattern: *
10	0.050498847	172.30.32.1	172.30.33.95	SMB2	144	Find Response, Error: STATUS_NO_MORE_FILES
11	0.050653669	172.30.33.95	172.30.32.1	SMB2	160	Close Request File: tmp
12	0.050948621	172.30.32.1	172.30.33.95	SMB2	196	Close Response
14	5.763953339	172.30.33.95	172.30.32.1	SMB2	160	Close Request File: tmp
15	5.765041094	172.30.32.1	172.30.33.95	SMB2	196	Close Response

```
▾ SMB2 (Server Message Block Protocol version 2)
  ▸ SMB2 Header
  ▾ Create Response (0x05)
    ▸ StructureSize: 0x0059
      Oplock: No oplock (0x00)
    ▸ Response Flags: 0x00
      Create Action: The file existed and was opened (1)
      Create: Feb 21, 2022 14:21:03.405926900 CST
      Last Access: Sep 14, 2022 11:38:54.583445500 CDT
      Last Write: Feb 21, 2022 14:21:03.408893400 CST
      Last Change: Feb 21, 2022 14:21:03.408893400 CST
      Allocation Size: 0
      End Of File: 0
    ▸ File Attributes: 0x00000010
      Reserved: 00000000
    ▸ GUID handle File: tmp
      Blob Offset: 0x00000098
      Blob Length: 56
    ▸ ExtraInfo SMB2_CREATE_QUERY_ON_DISK_ID
▾ SMB2 (Server Message Block Protocol version 2)
  ▸ SMB2 Header
  ▾ Find Response (0x0e)
      [Info Level: SMB2_FIND_ID_FULL_DIRECTORY_INFO (38)]
    ▸ StructureSize: 0x0009
      Blob Offset: 0x00000048
      Blob Length: 282
    ▾ Info: 580000000000000000512728c6027d80112ca202b58c8d801e685728c6027d801e685728c…
      ▸ FileIdBothDirectoryInfo: .
      ▸ FileIdBothDirectoryInfo: ..
      ▸ FileIdBothDirectoryInfo: populate_root
```


Deferred close improvements

- Currently used for i/o patterns like open/read/close/open/read/close
- Extending to cover many more scenarios, greatly improving performance
- Handle cache (deferred close time) now configurable with new mount parm “closetimeo” (thank you Bharath!)

SMB1/CIFS deprecation

- SMB3.1.1 rocks …
- Gradually move the old, insecure dialects out of the default module used for SMB2.1/SMB3/SMB3.1.1, so easier to deprecate SMB1/CIFS

Multichannel improvements

- Requerying network interfaces, dynamically adjusting
- Reconnect improvements
- Performance improvements (thank you Shyam Prasad!)
- Soon will be enabled by default (when server supports multiple interfaces or RSS)

SMB Direct to-do (early observations) (thanks to Tom Talpey and Metze)

- Reduce SGE usage, and decrease maximum fragment size
  - Fails to operate on SoftiWARP provider
  - Needless memory usage
  - High SGE usage impacts performance
  - Patches submitted for review yesterday
- Fix RDMA “responder resources”, which do NOT apply to RDMA Writes
  - Significant performance limiter for bulk reads
- Fix sends to not wait for completion before returning
  - Stalls the pipeline, and costs significant context switching
- Use RDMA post-multiple to improve compound send efficiency
- Ensure packet kmem cache optimal packing (3x1364 == 4092)
- Review protocol parsing and state validation
  - E.g. ksmbd allows renegotiate (?), reassembles oversize segments (?)
- Hangs when shutting down with connection held
- Merge the two implementations: fs/cifs/smbdirect.[ch] and fs/ksmbd/transport_rdma.[ch]
  - Either refactor and merge, or consider metze’s alternative “smbdirect socket” driver

SMBDIRECT Transport Improvements – RDMA for the world

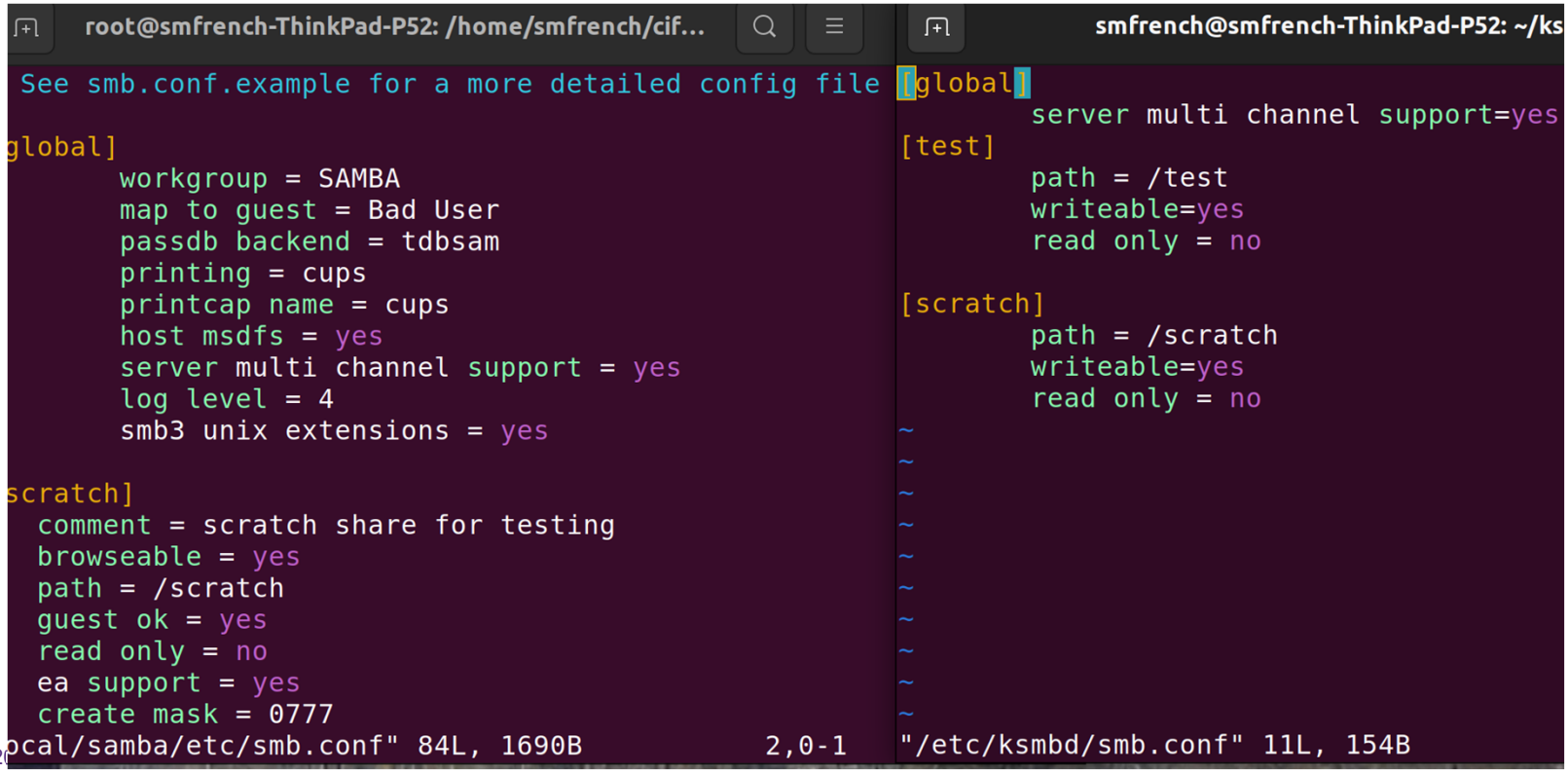
- SMBDIRECT is an abstraction layer for making RDMA useable more broadly. It has no SMB dependencies (SMB3 was just the first consumer of this generic transport layer, but it applies more broadly)
- Longer term plan is to:
  - Bring common from cifs.ko and ksmbd for RDMA into smbdirect.ko
  - Enable user space access to RDMA through smbdirect.ko so user space applications can benefit from the performance gains of RDMA
  - Improvements to this common module will benefit both client and server (and userspace)
- smbdirect.ko will provide
  - PF_SMBDIRECT sockets
  - Send message and receive message will get MSG_OOB messages for read and write offload, greatly improving performance and reducing CPU overhead
- (SMB independent) “echo server client” smbdirect tests under development to improve regression tests without requiring SMB
- Thanks to Metze for this work. Feedback and review and testing welcome.

SMB3.1.1 POSIX Extensions

- See talk on this at the SMB3.1.1 Plugfest this week
- Has been in Linux client for years & simpler than SMB1 Unix extensions
  - Great progress on Samba (client and server) and ksmbd server
  - Ongoing testing at plugfest with three servers and two clients

Setting up Samba and ksmbd shares are easy

NB: Samba requires “smb3 unix extensions = yes” in smb.conf



Mounting from the Linux kernel client

- Remember to add “posix” on mount command
- Also consider “mfsymlinks” if want client only symlinks

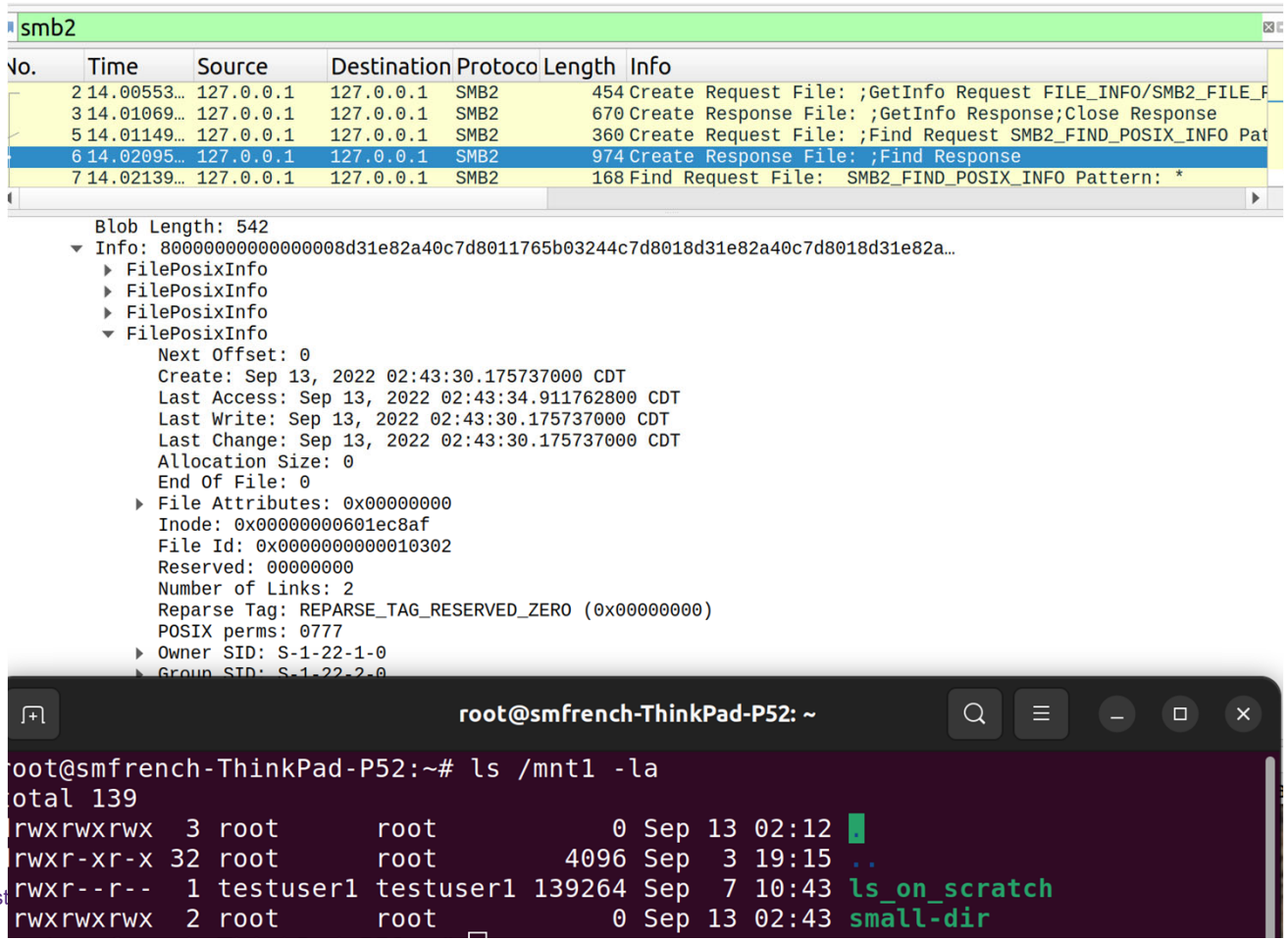
```
root@smfrench-ThinkPad-P52:/home/smfrench/mulder-posix/samba# mount -t cifs //loca
 /mnt1 -o username=testuser,password=testpass,mfsymlinks,posix
root@smfrench-ThinkPad-P52:/home/smfrench/mulder-posix/samba# cat /proc/fs/cifs/De
Display Internal CIFS Data Structures for Debugging
---------------------------------------------------
CIFS Version 2.38
Features: DFS,FSCACHE,STATS,DEBUG,ALLOW_INSECURE_LEGACY,CIFS_POSIX,UPCALL(SPNEGO),
WITNESS
CIFSMaxBufSize: 16384
Active VFS Requests: 0

Servers:
1) ConnectionId: 0x4 Hostname: localhost
Number of credits: 443 Dialect 0x311 posix
TCP status: 1 Instance: 1
Local Users To Server: 1 SecMode: 0x1 Req On Wire: 0
In Send: 0 In MaxReq Wait: 0

        Sessions:
        1) Address: 127.0.0.1 Uses: 1 Capability: 0x300047      Session Status: 1
        Security type: RawNTLMSSP  SessionId: 0x35ada477
        User: 0 Cred User: 0

        Shares:
```

Note exact mode bits and owner reported w/POSIX Extensions

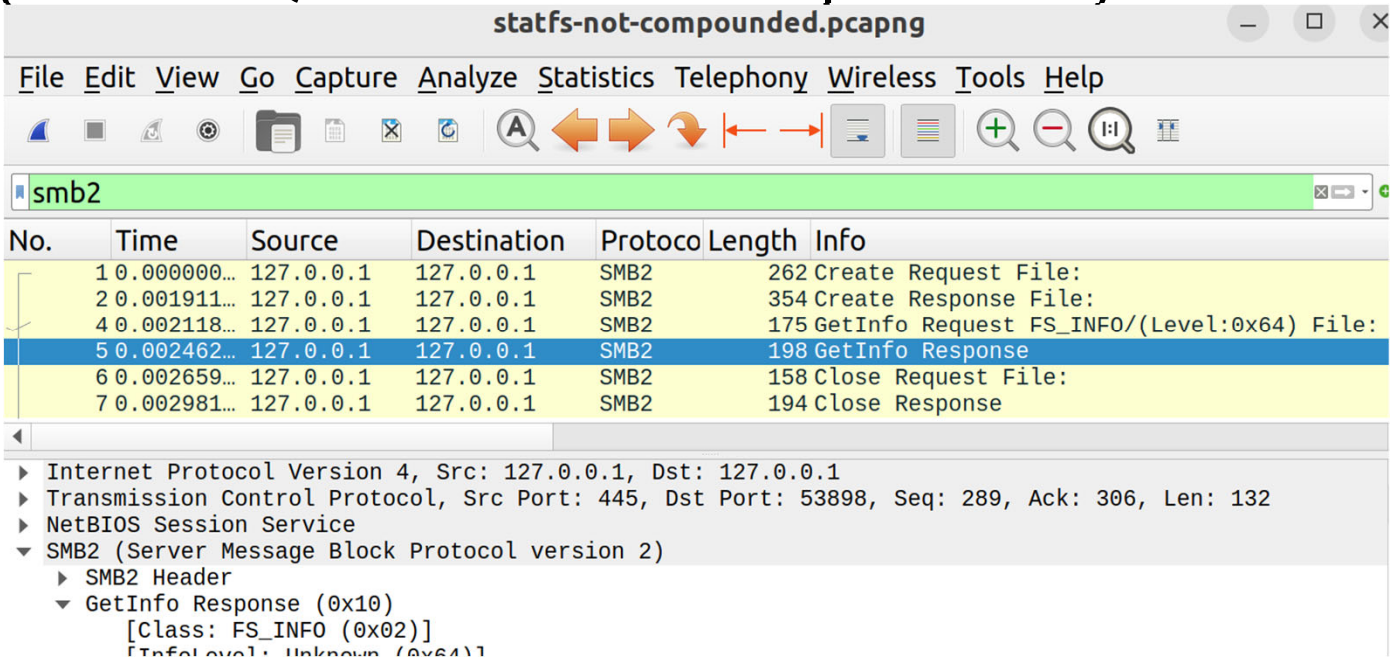


Query FS Info – includes additional posix fields

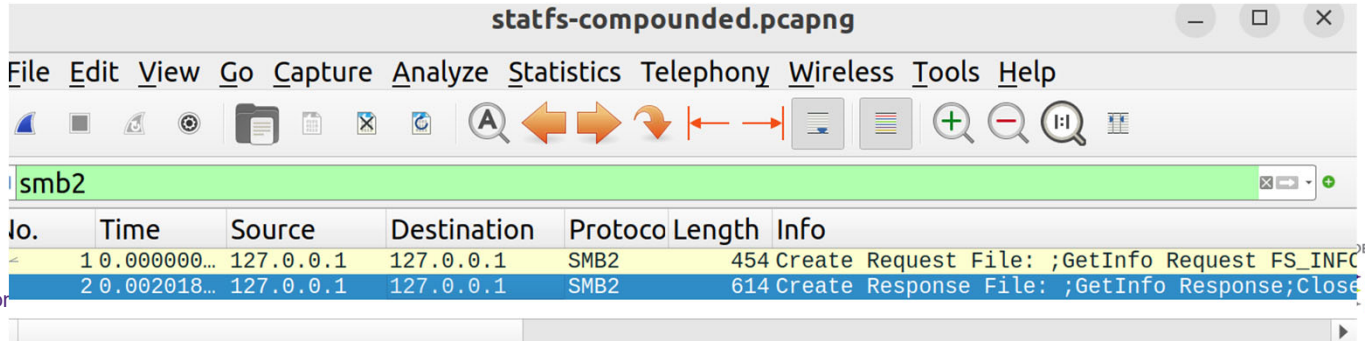
```
root@smfrench-ThinkPad-P52:/home/smfrench# mount -t cifs //local
-o username=testuser,password=testpass,mfsymlinks,posix
root@smfrench-ThinkPad-P52:/home/smfrench# stat -f /mnt1
  File: "/mnt1"
    ID: c7df5aa0f1e89eff Namelen: 255     Type: smb2
Block size: 4096       Fundamental block size: 4096
Blocks: Total: 139092115  Free: 48993190   Available: 48993190
Inodes: Total: 278320128  Free: 273211966
root@smfrench-ThinkPad-P52:/home/smfrench#
```


Better performance (POSIX QFS Info now compounded)

Before:



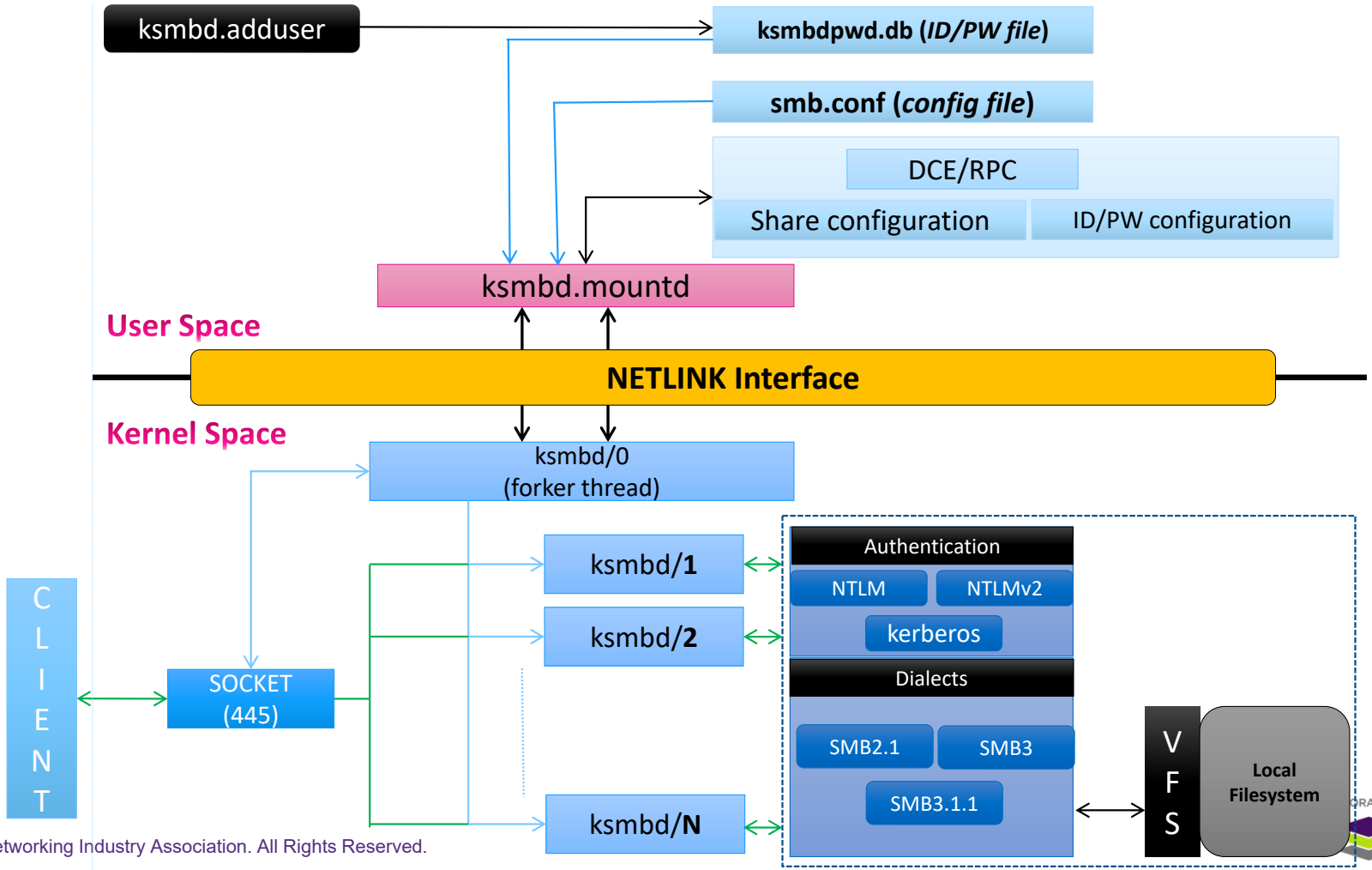
Now:



Recent ksmbd progress (kernel server)

Provided by Namjae Jeon (linkinjeon@kernel.org)

Architecture

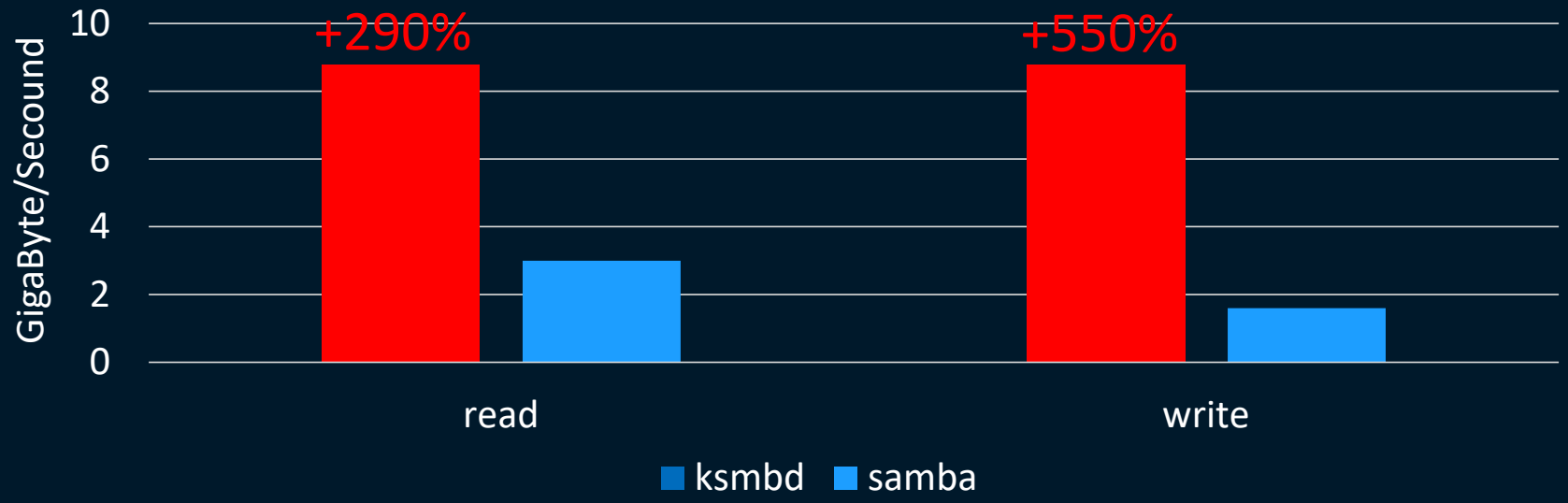


RSS(Receive Side Scaling) mode support

- ksmbd newly supports RSS mode
- Ziwei Xie(high-flyer) compared the performance samba and ksmbd on their test environment. Thanks Ziwei!
- In RSS mode, there is a performance difference of 3 times for read and 4 times for write on his setup.

Performance Comparison on Multichannel + RSS mode

*Test Environment
CPU : AMD EPYC 7H12 64-Core Processor
NIC : MCX653105A-HDAT
Client : Windows
Benchmark tool : FIO



ksmbd start to fully support smb-direct

- 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. smbd 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 NICs without configuration.
  - Server should send RDMA NIC info to client.
  - No need to specify RDMA NIC information to smb.conf.

Performance test environment

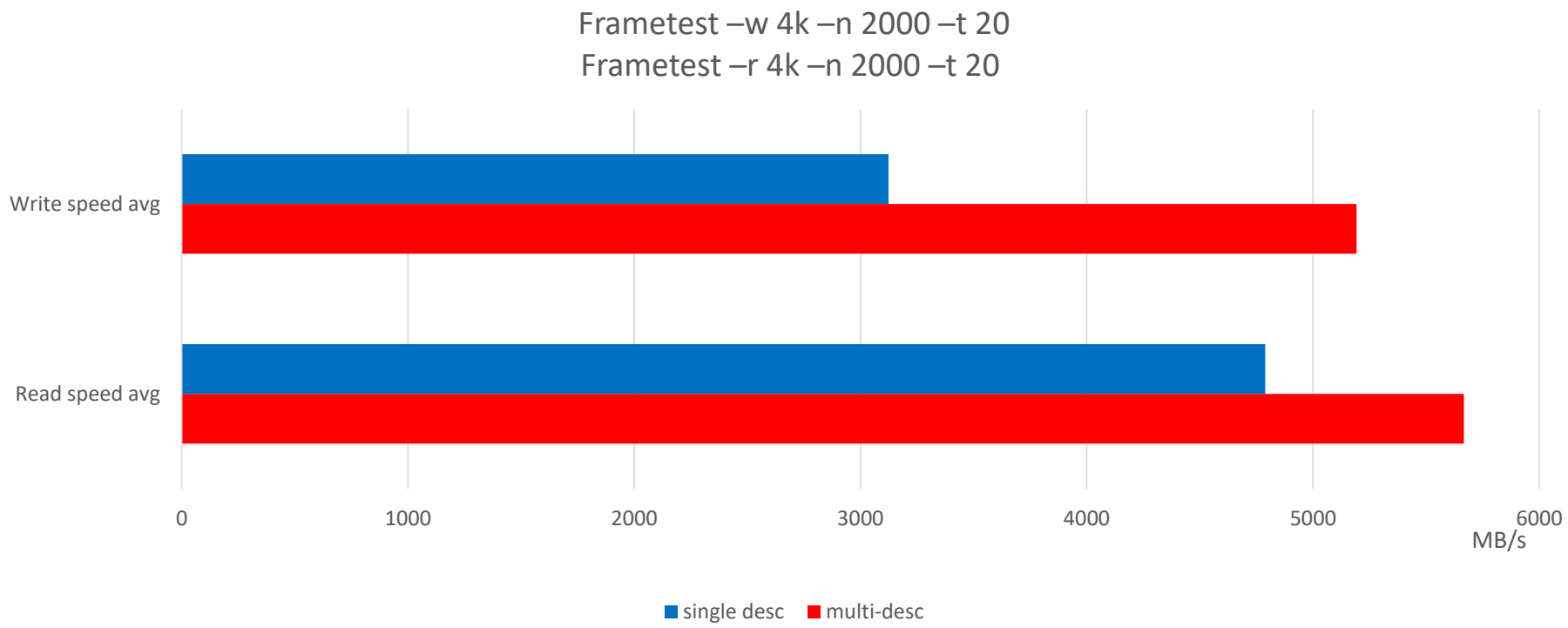
- Benchmark tool
  - Framtest (https://support.dvsus.com/hc/en-us/articles/212925466-How-to-use-frametest)
- Server

CPU: intel Silver 4114 x 2
DRAM: 512GB
NVMe SSD: Kioxia CM6 1.9T x 9(mdadm raid0 with XFS)
NIC: MCX516A-CCAT

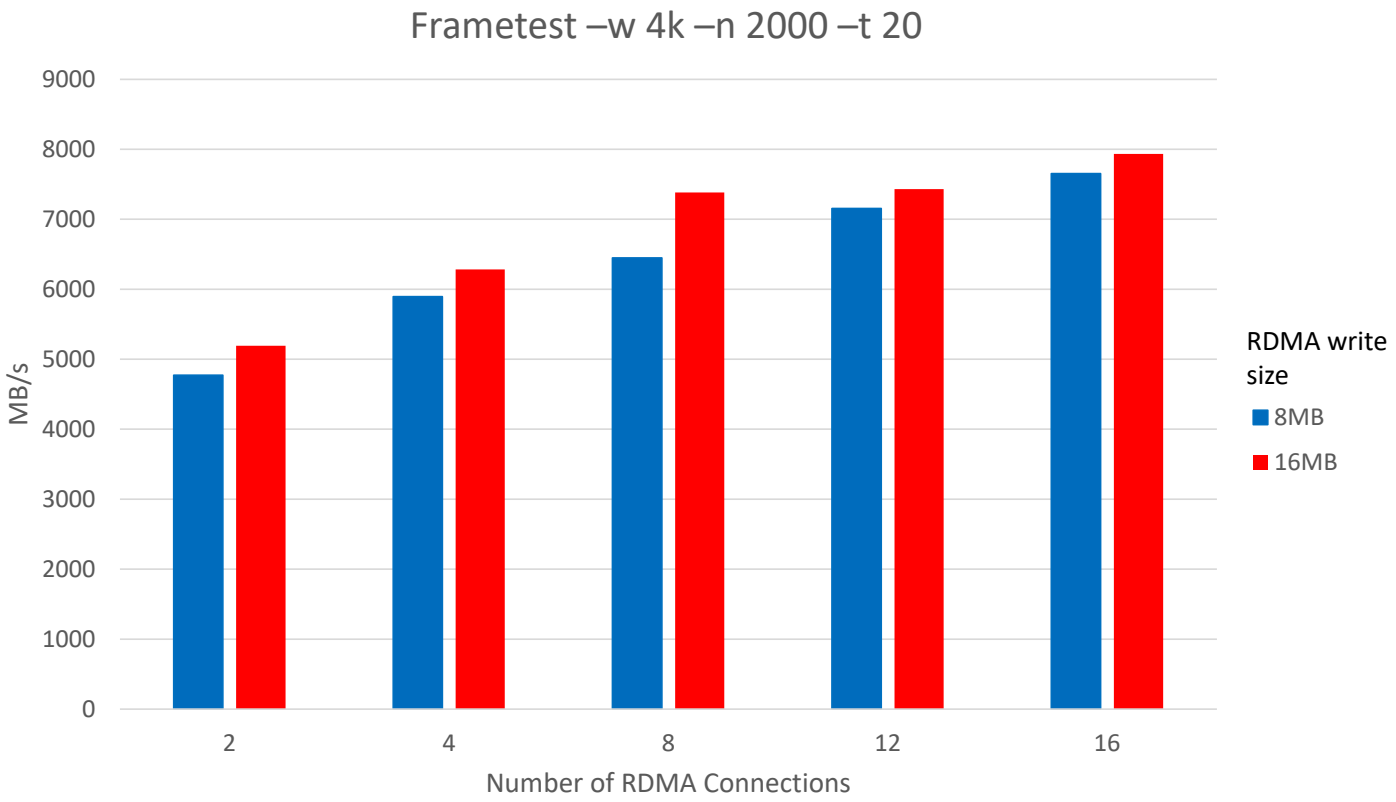
- Client

CPU: intel Silver 4215 x 2
DRAM: 64GB
NIC: MCX516A-CCAT

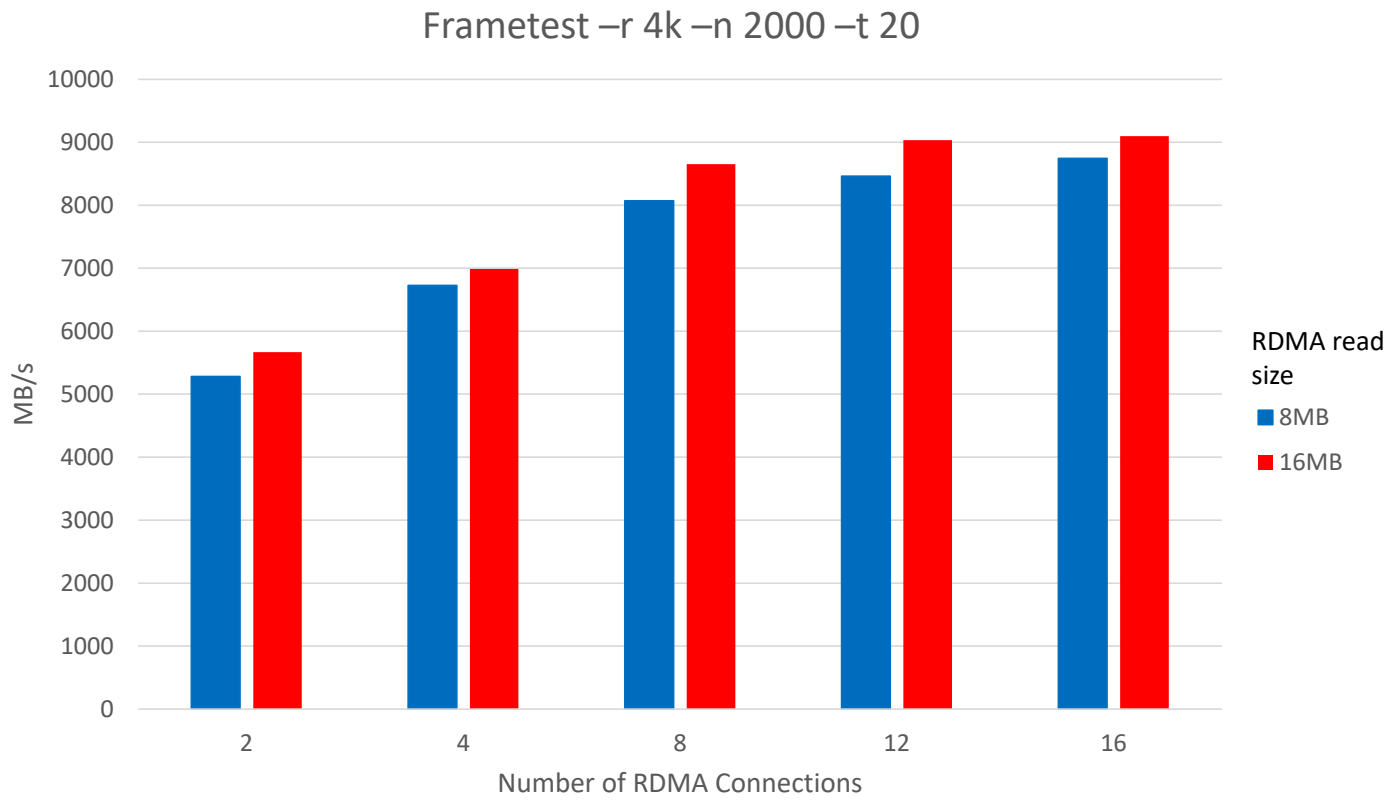
Performance comparison between single and multi-descriptor.



RDMA write performance per number of connections

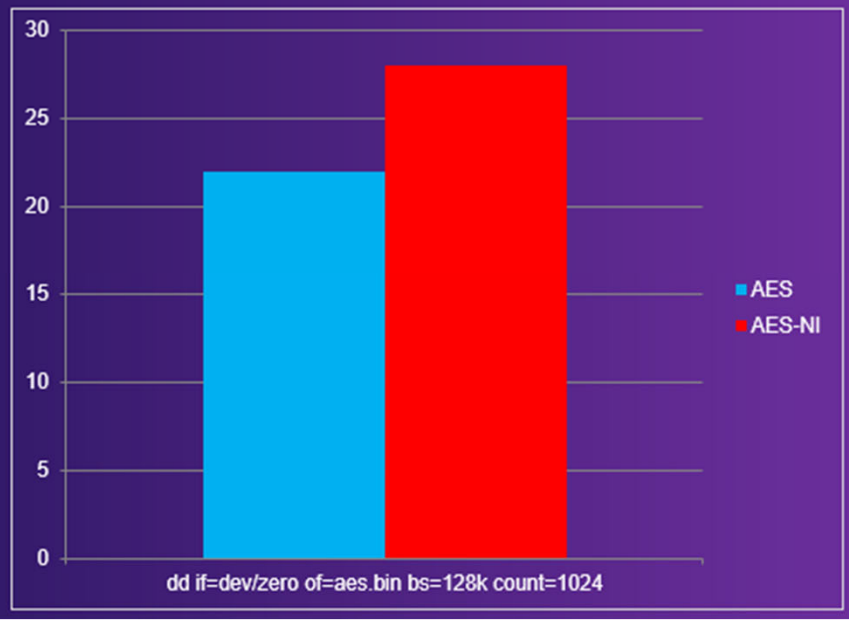


RDMA read performance per number of connections



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



Quality Improvements

- ksmbd has been used by many more users since merged into mainline.
- Improved multiple security issues reported by International Software Vulnerability Initiative.
- Issues related to SMB-Direct(RDMA) and Multi-channel(Multi-port NIC) feature have been improved as enterprise server users start to use ksmbd (initial users were mostly embedded and small footprint devices)
- Improvements from reviews by kernel maintainers (vfs dentry race issue, soft-iWARP issue, etc.)
- Testing continues to improve, adding additional regression tests as new features are enabled and as ksmbd gets used more widely

Future Plan

- Directory leases (WIP)
- SMB2 notify (WIP)
- Durable handle v1/v2 feature (WIP)
- Add ksmbd status option to show statistics using ksmbd.control
  - Processed requests, session info(user info, number of credits and more), session list, openfiles, NIC info.
- Config backend (Recently get a request, make the configuration interface available remotely over the WINREG RPC interface)

Linux Kernel Server, KSMBD (continued)

- If interested in contributing there are lots of cool features to work on, as well as improved integration with Samba (e.g. user space upcalls for additional features). The SMB3.1.1 family of protocols is huge!
- 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
- Also See the linux-cifs mailing list for more info
- Very exciting time!

Client progress (cifs.ko)

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”

5.17 kernel (March 20th), 51 changesets, cifs.ko ver 2.35

- Add support for new fscache (offline files caching mechanism)
- Send additional NTLMSSP info (including module and OS version) for improved debugging
- DFS (thank you Paulo!) and ACL fixes
- Restructuring of multichannel code

5.18 kernel (May 22nd), 40 changesets, cifs.ko ver 2.36

- Important performance improvement (reuse cached file handle for various common operations like stat and statfs if available), greatly reducing metadata operations (like open/close)
- Important fscache (offline file caching) and DFS improvements
- cross mount reflink now supported, which can dramatically improve copy performance from one share to another (on the same server) if they support duplicate extents.

5.19 kernel (July 31st, 2022) 57 patches

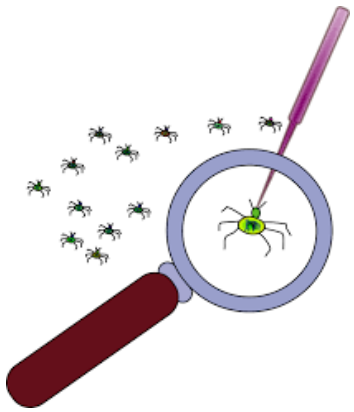
- Important performance optimization for directory searches, now we cache the root directory content (to the many servers which support directory leases) reducing amount of network traffic for queries in the root directory
- Multichannel reconnect improvements (e.g. when address or interfaces change)
- Periodic requerying of network interfaces
- New mount parm “nosparse” (to work around servers with problems with partial sparse file support)
- RDMA (smbdirect) improvements

6.0-rc5 kernel (release expected early October, 2022) (cifs module version 2.38). 48 patches so far

- Fallocate improvements (insert and collapse range)
- Module size shrunk significantly when SMB1/CIFS (insecure legacy) disabled
- New mount parm “closetimeo” to allow extending deferred closes (handle leases) longer (and default increased to 5 seconds from 1 sec)
- Multichannel perf (locking) improvements

Tracing continues to improve ...

- Added more than 11additional dynamic tracepoints



```
root@smfrench-ThinkPad-P52:/sys/kernel/debug/tracing/events/cifs# ls
cifs_flush_err       smb3_flush_enter             smb3_pend_credits                    smb3_rmdir_done
cifs_fsync_err       smb3_flush_err               smb3_posix_mkdir_done                smb3_rmdir_enter
enable               smb3_fsctl_err               smb3_posix_mkdir_enter               smb3_rmdir_err
filter               smb3_hardlink_done           smb3_posix_mkdir_err                 smb3_ses_expired
smb3_add_credits     smb3_hardlink_enter          smb3_posix_query_info_compound_done  smb3_set_credits
smb3_adj_credits     smb3_hardlink_err            smb3_posix_query_info_compound_enter smb3_set_eof
smb3_close_done      smb3_hdr_credits             smb3_posix_query_info_compound_err   smb3_set_eof_done
smb3_close_enter     smb3_insufficient_credits    smb3_query_dir_done                  smb3_set_eof_enter
smb3_close_err       smb3_lease_done              smb3_query_dir_enter                 smb3_set_eof_err
smb3_cmd_done        smb3_lease_err               smb3_query_dir_err                   smb3_set_info_compound_done
smb3_cmd_enter       smb3_lease_not_found         smb3_query_info_compound_done        smb3_set_info_compound_enter
smb3_cmd_err         smb3_lock_err                smb3_query_info_compound_enter       smb3_set_info_compound_err
smb3_connect_done    smb3_mkdir_done              smb3_query_info_compound_err         smb3_set_info_err
smb3_connect_err     smb3_mkdir_enter             smb3_query_info_done                 smb3_slow_rsp
smb3_credit_timeout  smb3_mkdir_err               smb3_query_info_enter                smb3_tcon
smb3_delete_done     smb3_nblk_credits            smb3_query_info_err                  smb3_too_many_credits
smb3_delete_enter    smb3_notify_done             smb3_read_done                       smb3_wait_credits
smb3_delete_err      smb3_notify_enter            smb3_read_enter                      smb3_waitff_credits
smb3_enter           smb3_notify_err              smb3_read_err                        smb3_write_done
smb3_exit_done       smb3_open_done               smb3_reconnect                       smb3_write_enter
smb3_exit_err        smb3_open_enter              smb3_reconnect_detected              smb3_write_err
smb3_falloc_done     smb3_open_err                smb3_reconnect_with_invalid_credits  smb3_zero_done
smb3_falloc_enter    smb3_oplock_not_found        smb3_rename_done                     smb3_zero_enter
smb3_falloc_err      smb3_overflow_credits        smb3_rename_enter                    smb3_zero_err
smb3_flush_done      smb3_partial_send_reconnect  smb3_rename_err
root@smfrench-ThinkPad-P52:/sys/kernel/debug/tracing/events/cifs# ls | wc
     99      99    1832
```


eBPF is amazing ...

- See Brendan Gregg's website
- Also see e.g. https://wiki.samba.org/index.php/LinuxCIFS_troubleshooting
- Can be as simple to do as "trace-cmd record -e cifs"
  - And then "trace-cmd show" in another window
- Let us know if suggestions on other debugging tracepoints that would be helpful
- And don't forget about *proc/fs/cifs/Stats, proc/fs/cifs/open_files and proc*/fs/cifs/DebugData ...

Coming soon ...

New features under development for SMB3.1.1 on Linux

What features can you expect in next few releases?

- Extending use of directory leases to improve metadata caching (currently limited to root directory)
- SMB3.1.1 compression support (allow compressing network traffic based on the SMB3.1.1 compress mount parm)
- statx to return additional SMB3.1.1 attributes like “offline”
- Improvements to enable fanotify/inotify over SMB3.1.1 mounts (currently requires a private SMB3.1.1 specific ioctl)
- Reenabling swap file support over SMB3.1.1 mounts
- Prototype of SMB3.1.1 over QUIC (new encrypted network transport)
- More perf improvements for folios, cache, parallel i/o, multichannel
- More testing of the SMB3.1.1 POSIX extensions to ksmbd (and Samba server). New fsctls or optional info levels as Linux syscalls continue to evolve
- If interested in helping with any of these let me and/or linux-cifs list know!

Testing Improvements

Section Subtitle

Additional tests are encouraged

- Xfstests are the standard Linux filesystem functional tests
- Added 20+ to the main “cifs-testing” regression testing group (up to 246 tests run on every checkin from this group)
- Will be adding more and more RDMA (smbdirect) tests
- Various server specific groups have added even more
  - Azure SMB3.1.1 multichannel: up 25% more tests, now includes 133 tests
  - Ksmbd (Linux kernel server target) up 15%, now includes 144 tests
- There are detailed wiki pages on wiki.samba.org going 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’)

Recent improvements – cifs-utils

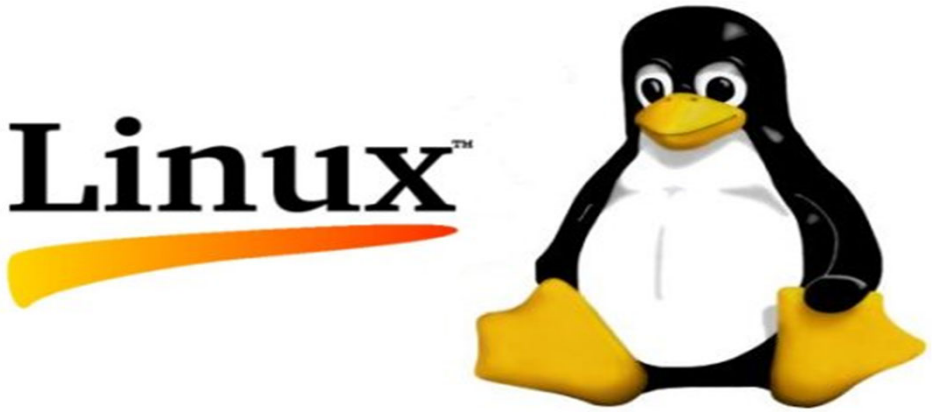
Userspace tools

Improved user space tools (cifs-utils)

- cifs-utils 6.14 released in Sept
  - Add commands to view Alternate Data Streams
  - setcifsacl improvements
  - Improved debugging (keydump)
- Cifs-utils 6.15 released in April
- More recently 7.0 released August 11th
  - Add support for gss-proxy (improving krb5 credential retrieval)
  - Improve support for Heimdal, not just MIT kerberos
  - Misc. bug fixes

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/ksmbd.git/?p=ksmbd.git (ksmbd-for-next branch)

Please take a moment to rate this session.

Your feedback is important to us.

Older slides for additional information

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 shutdown 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
- And 5.15 kernel
  - Important deferred close (handle lease) bug fixes
  - Support for weaker authentication (NTLMv1 and LANMAN) removed
  - (And experimental kernel server, ksmbd, merged)

