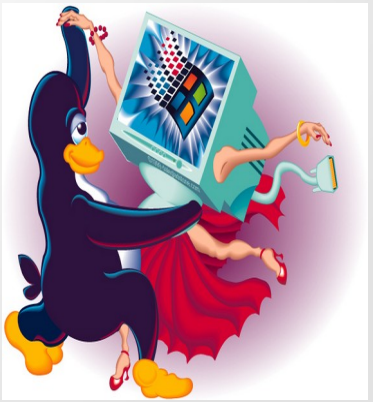
Linux Access to the Cloud and NAS reborn: Recent progress in SMB3.1.1

Steve French
Principal Software Engineer
Azure Storage - Microsoft



Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft Corporation
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.

Who am I?

- Steve French smfrench@gmail.com
- Author and maintainer of Linux cifs vfs (for accessing Samba, Windows and various SMB3/CIFS based NAS appliances)
- Also wrote initial SMB2 kernel client prototype
- Member of the Samba team, coauthor of SNIA CIFS Technical Reference,former SNIA CIFS Working Group chair
- Principal Software Engineer, Azure Storage: Microsoft

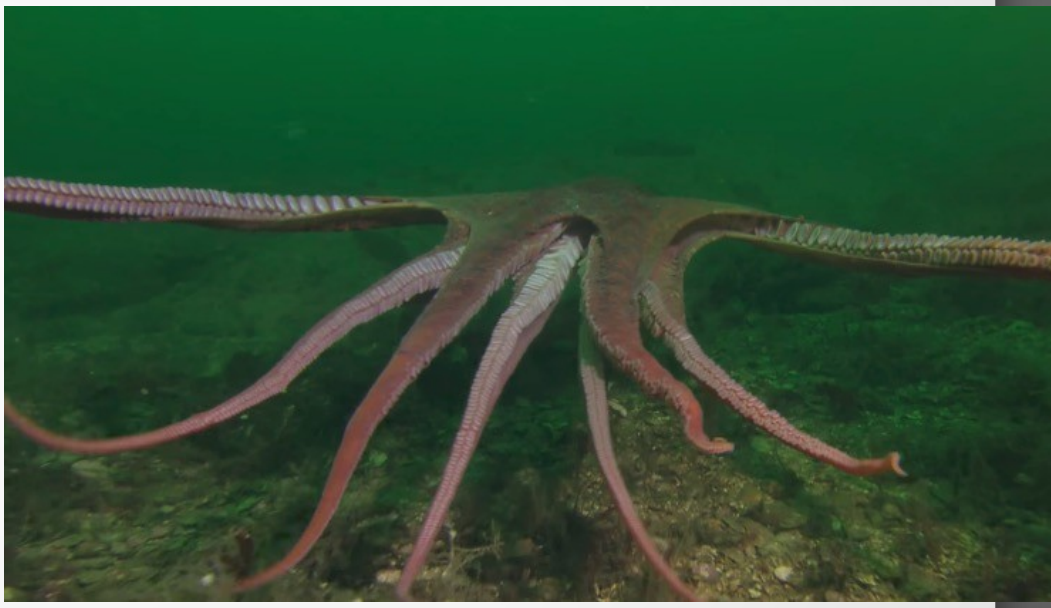
Outline

- General Linux File System Status – Linux FS and VFS Activity
- What are the goals?
- What's New – Key Feature Status
  - Kernel client
  - User space tools (cifs-utils)
  - Kernel server (new)
- Features expected soon
- Common Configuration Suggestions
- Testing Improvements

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

- A year ago Linux 5.0-rc4 “Shy Crocodile”

Last week:5.5 “Kleptomaniac Octopus”



Discussions driving some of the FS development activity

- New mount API
- Improving support for Containers
- Unification of credential mgmt - kernel keyring improved
- Better support for faster storage (NVME, RDMA)
- io_uring and better async i/o
- “The Linux Copy Problem”
- Shift to Cloud (longer latencies, object & file coexisting)

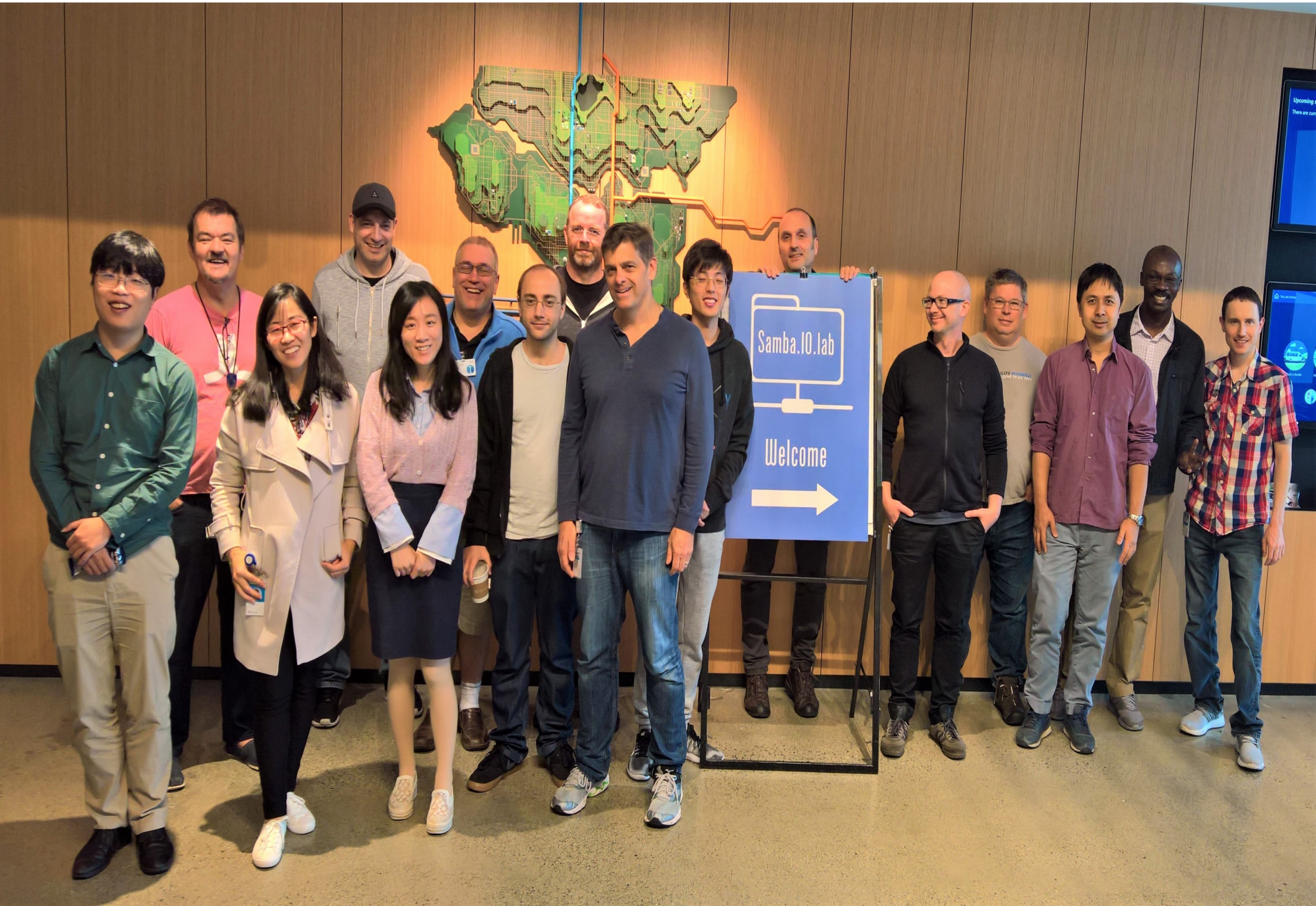
SMB on Linux reborn

- “*Version 5.0 marks the reborn, new improved SMB3 Client For Linux*”
- And it continues to improve rapidly, among the most active file systems on Linux for another year ...

2019 Linux FS/MM summit (in Puerto Rico in May)

- Great group of talented developers

Samba Team meeting and testing in Redmond in the Fall



Most Active Linux Filesystems this year

- 5338 kernel filesystem changesets last year (since 5.0-rc4 kernel) (flat)
  - FS activity: 6.2% of overall kernel changes (which are dominated by drivers) up slightly as % of activity
  - Kernel is huge (> 19 million lines of code, measured Saturday)
- There are many Linux file systems (>60), but six (and the VFS layer itself) drive 69% of activity (btrfs, xfs and cifs are the three most active)
  - File systems represent 4.8% of kernel source code (940KLOC) but among the most carefully watched areas
- cifs.ko (cifs/smb3 client) activity is strong
  - #3 most active of all fs with 375 changesets!
  - 54.9KLOC (not counting user space cifs-utils which are 11.6KLOC and samba tools which are larger)

Linux File System Change Detail for past year (5.0-rc4 to now)

- BTRFS 959 changesets (down slightly)
- VFS (overall fs mapping layer and common functions) 958 (up significantly)
- XFS 581 (up)
- CIFS/SMB2/SMB3 client 375 (since 4.18 kernel activity has gone way up)
- F2FS 271 (down)
- NFS client 283 (down)
- Others: EXT4 243(up), Ceph 164, AFS 135, GFS2 118, OCFS2 108 ...
- NFS server 137 (flat). Linux NFS server **MUCH** smaller than CIFS or Samba
- NB: Samba is as active as all Linux file systems put together (> 5000 changesets last year, over 1200 in the file server component alone!) - broader in scope (by a lot) and also is user space not kernel. **98x larger than the NFS server in Linux! Samba now 3.4 million lines of code**

Linux SMB3 Kernel Server Change Detail for past year

- Kernel server
  - Module: cifsd/ksmbd.ko
  - Size: 24.5 KLOC
  - 2480 Changesets
- User space tools
  - ksmbd-tools
  - Size: 10.3 KLOC
  - 810 Changesets

FS Have a tough job! Responsible for > 200 of 850 syscalls. +12 since last year's SDC!

Syscall name	Kernel Version introduced
io_uring_enter	5.1
io_uring_register	5.1
io_uring_setup	5.1
move_mount	5.2
open_tree	5.2
fsconfig	5.2
fsmount	5.2
fsopen	5.2
fspick	5.2

What are the goals?

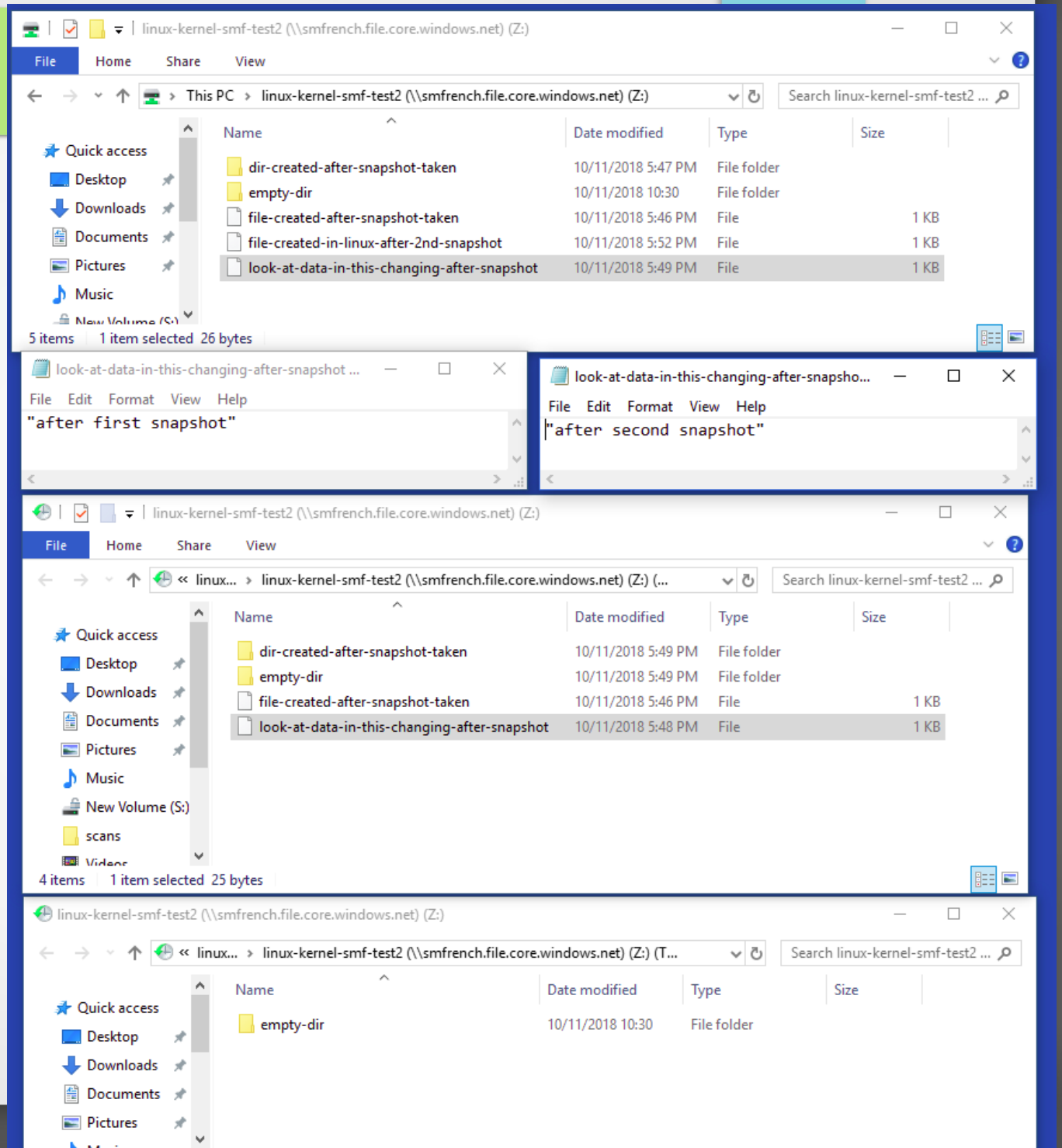
- Fastest, most secure general purpose way to access file data, whether in the cloud or on premises or virtualized
- Implement all reasonable Linux/POSIX features - so apps don’t know they run on SMB3 mounts (vs. local)
- As Linux evolves, and need for new features discovered, quickly add them to Linux kernel client and Samba

Examples of Great Progress we talked about last year!

- Reminders of some amazing progress …
- A few examples

Snapshots rock!

- Different mounts
- SS is Read-only
- Easy to use
- Example mount remote to cloud (Azure)



rsize and wsize increase

- Previous default 1MB
  - 4MB gave 1 to 13% improved performance to Samba depending on network speed, 1% better for read.
- Moved to 4MB in 4.20 kernel
- And default block size moved to 1MB (helps cp)

Examples of Great Progress more recently!

- This year ...

New Kernel Server “cifsd” arrives!

- ksmbd.ko and userspace helper utilities
- Thank you Namjae and team!
- See https://wiki.samba.org/index.php/Linux_Kernel_Server

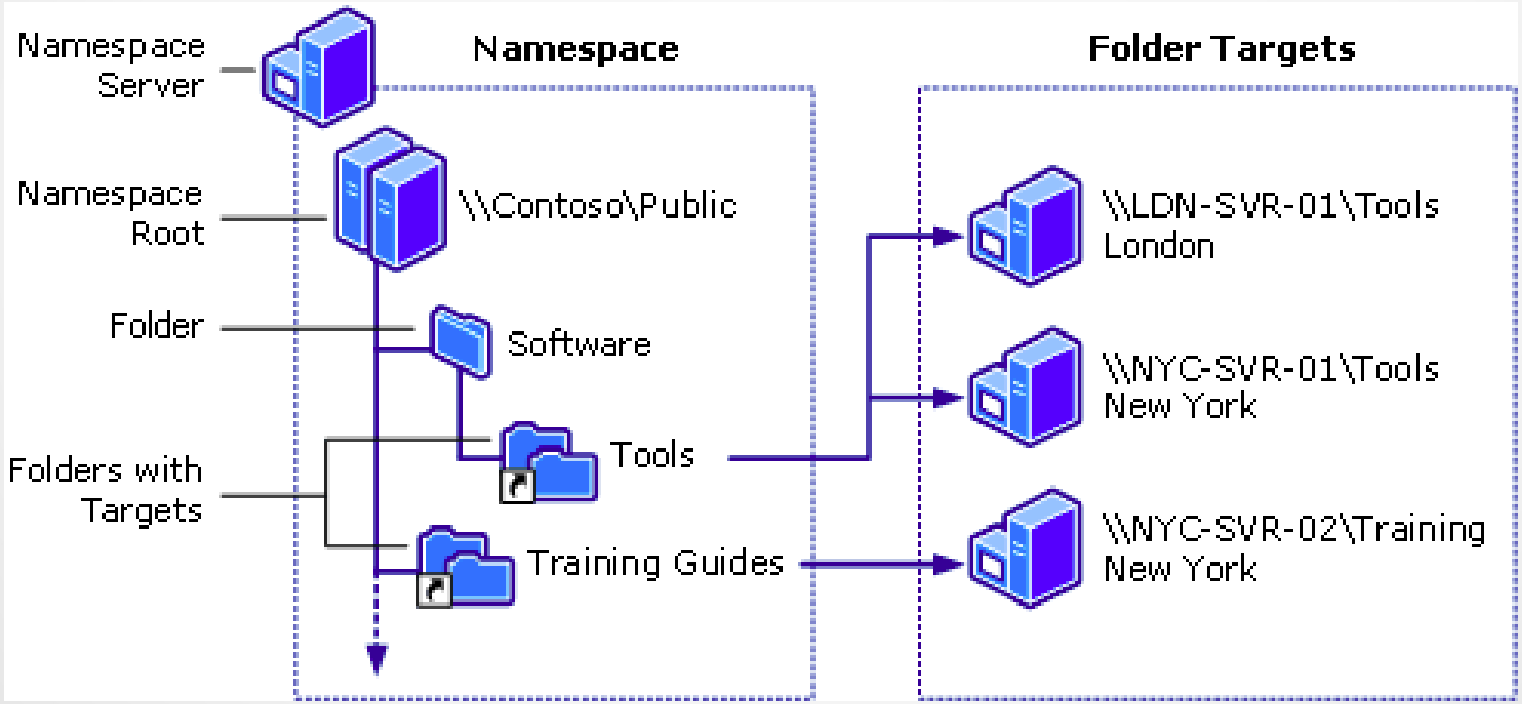
```
root@smfrench-ThinkPad-P52:/home/smfrench/ksmbd-tools# mount -t cifs //localhost/test /mnt -o u
ame=testuser,password=testpass
root@smfrench-ThinkPad-P52:/home/smfrench/ksmbd-tools# ls /mnt
0740dir  1GB          fio-testfile.0.0  newfile0764  test-432
0750dir  310          fio-testfile.1.0  somefile     test-433
0754dir  314-dir      fio-testfile.2.0  syscalltest  timestamp-test.txt
0760     dbench       fio-testfile.3.0  test-430
0765dir  dir-no-posix  fsx               test-431
root@smfrench-ThinkPad-P52:/home/smfrench/ksmbd-tools# ps -A | grep mbd
 3391 ?        00:00:00 usmbd
 3392 ?        00:00:00 usmbd
 3393 ?        00:00:00 ksmbd-tun0
 3394 ?        00:00:00 ksmbd-wlp0s20f3
 3395 ?        00:00:00 ksmbd-enp0s31f6
 3396 ?        00:00:00 ksmbd-lo
 3417 ?        00:00:00 ksmbd:48810
root@smfrench-ThinkPad-P52:/home/smfrench/ksmbd-tools# touch /mnt/newfile
root@smfrench-ThinkPad-P52:/home/smfrench/ksmbd-tools# mkdir /mnt/newdir
```

New (in kernel) server for SMB3

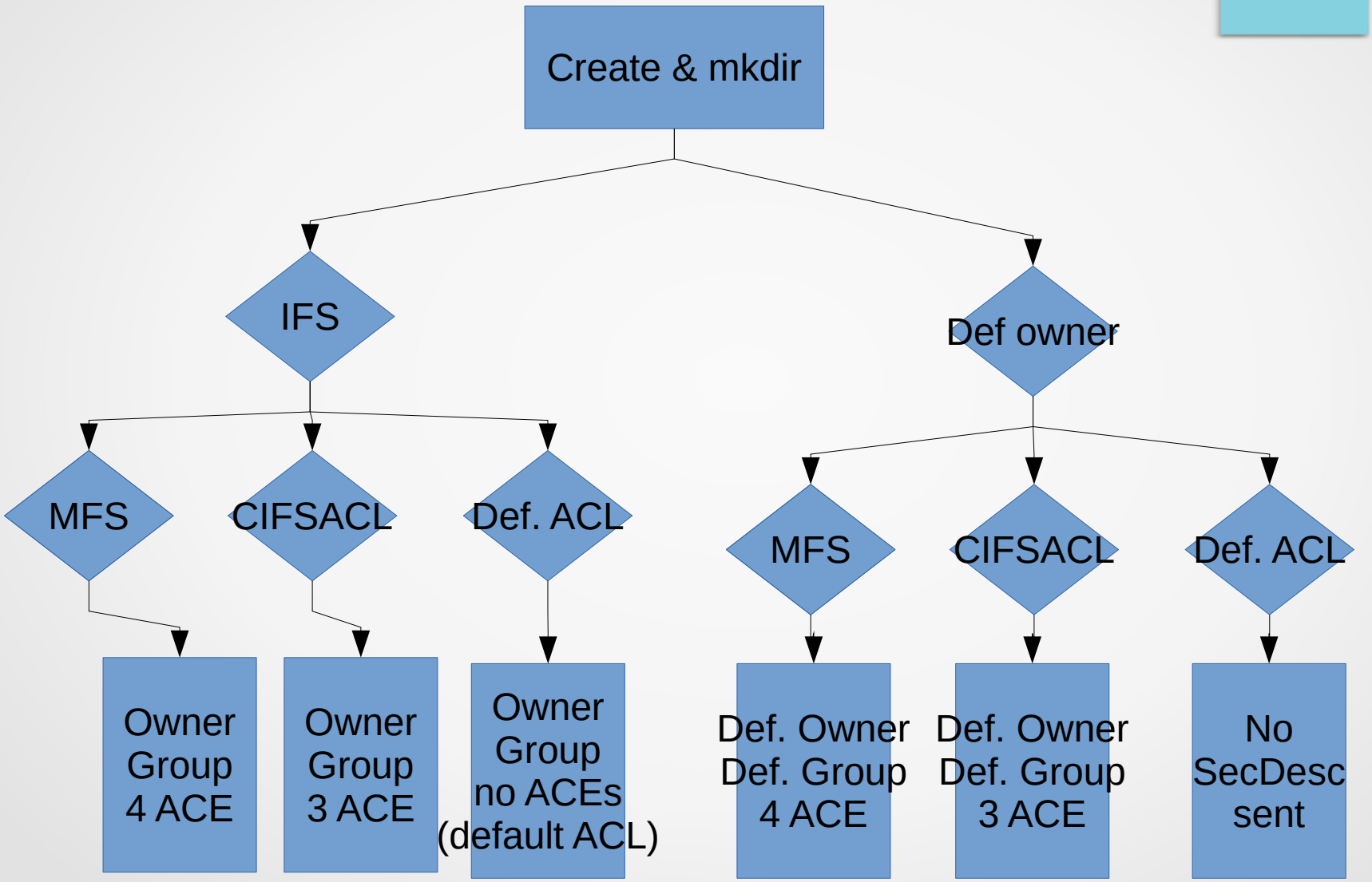
- Name of module: “ksmbd.ko”
- Name of source directory “cifsd” (to make it easier to find in the kernel fs directory, fs/cifsd will show up next to fs/cifs directory in the directory listing
- Name of daemons – begin with “ksmbd” to distinguish the “kernel” smb3 server from Samba (user space) whose processes are named “smbd”

Global Name Space – Much better! Thank you Paulo!

- Remember what DFS could do in Windows … now the Linux client can handle failover and DFS entry caching too



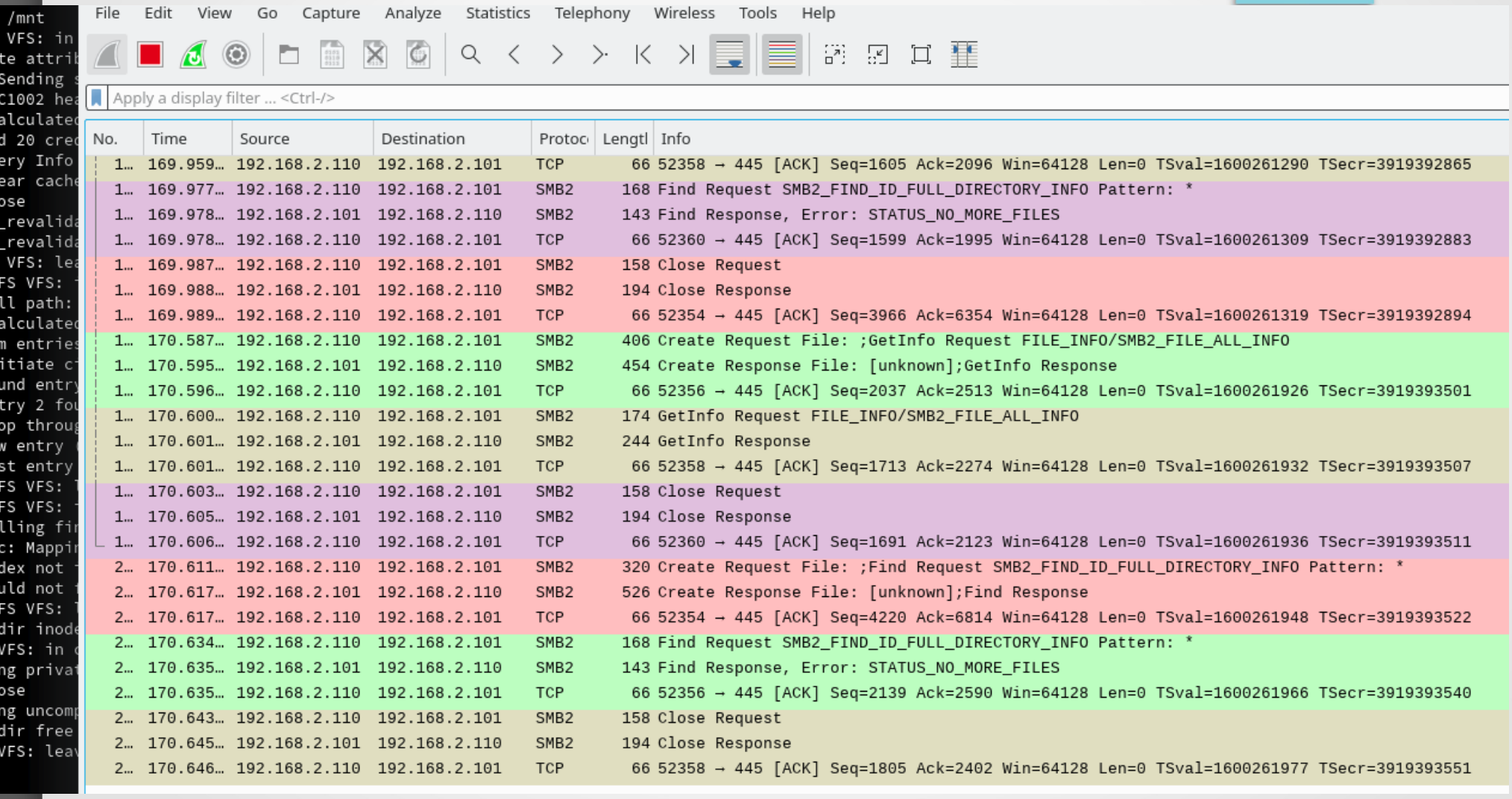
New security models: idsfromsid, modefromsid, cifsacl



Multichannel (in 5.5 kernel)

- Thank you Aurelien!
- Expected to be a big performance win …
- Working performance optimizations at plugfest this week

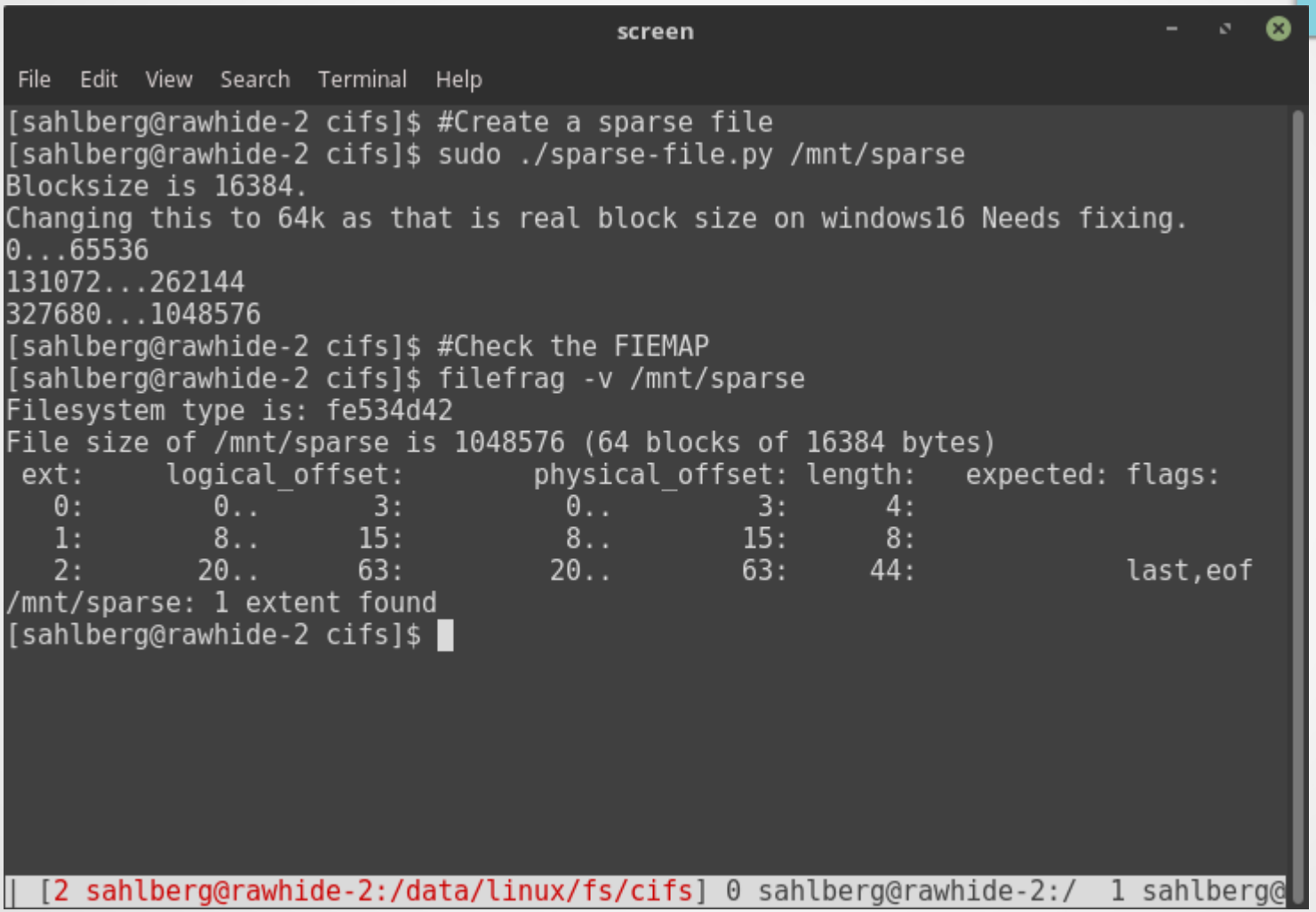
Trace using multichannel w/current cifs.ko



Compounding helps a lot – thanks Ronnie!

Added in so far:
1) update timestamps on existing file: touch /mnt/file" goes from 6 request/resp pairs to 4
2) delete file "rm /mnt/file" from 5 to 2
3) make directory "mkdir /mnt/newdir" 6 to 3
4) remove directory "rmdir /mnt/newdir" 6 down to 2
5) rename goes from 9 request/response pairs to 5 ("mv /mnt/file /mnt/file1")
6) hardlink goes from 8 to only 3 (!) ("ln /mnt/file1 /mnt/file2")
7) symlink with mfsymlinks enabled goes from 11 to 9 ("ln -s /mnt/file1 /mnt/file3")
8) query file information “stat /mnt/file” goes from six roundtrips down to 2
9) And get/set xattr, and statfs and more
10) In 5.6 kernel Ronnie added support for compounding readdir (9 requests down to 7 for typical directory, more gain for larger directories)

Sparse File Support (and other network fs can’t do this)!



Wireshark · Packet 1094 · wireshark_virbr0_20190605130142_6HSabS

```
▼ Ioctl Response (0x0b)
  ▶ StructureSize: 0x0031
    unknown: 0000
  ▶ Function: FSCTL_QUERY_ALLOCATED_RANGES (0x000940cf)
  ▶ GUID handle File: sparse
  ▼ In Data
      Offset: 0x00000070
      Length: 16
    ▼ Range
        File Offset: 0
        Length: 9223372036854775807
  ▼ Out Data
      Offset: 0x00000080
      Length: 48
    ▼ Range
        File Offset: 0
        Length: 65536
    ▼ Range
        File Offset: 131072
        Length: 131072
    ▼ Range
        File Offset: 327680
        Length: 720896
```

```
0000  52 54 00 c1 f8 ef 52 54  00 fd 53 ac 08 00 45 00   RT....RT ..S...E.
0010  00 e8 01 b2 40 00 80 06  7d 7b c0 a8 7c c6 c0 a8   ....@... }{..|...
0020  7c cb 01 bd a4 82 f2 c0  be e0 38 d3 89 13 80 18   |....... ..8.....
0030  08 1c e6 e7 00 00 01 01  08 0a 09 f5 f8 b8 7a 14   ........ ......z.
0040  92 4a 00 00 00 b0 fe 53  4d 42 40 00 01 00 00 00   .J.....S MB@.....
0050  00 00 0b 00 0a 00 01 00  00 00 00 00 00 00 0a 01   ........ ........
0060  00 00 00 00 00 00 34 52  00 00 05 00 00 00 21 00   ......4R ......!.
0070  00 1c 00 88 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 31 00  00 00 cf 40 09 00 59 44   ......1. ...@..YD
0090  33 00 22 00 00 00 bd 00  00 00 22 00 00 00 70 00   3."..... ..."...p.
00a0  00 00 10 00 00 00 80 00  00 00 30 00 00 00 00 00   ........ ..0.....
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 ff ff   ........ ........
00c0  ff ff ff ff ff 7f 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  01 00 00 00 00 00 00 00  02 00 00 00 00 00 00 00   ........ ........
00e0  02 00 00 00 00 00 00 00  05 00 00 00 00 00 00 00   ........ ........
00f0  0b 00 00 00 00 00                                  ......
```

No.: 1094 · Time: 9.906657148 · Source: 192.168.124.198 · Destination: 192.168.124.203 · Protocol: SMB2 · Length: 246 · Info: Ioctl Response FSCTL_QUERY_ALLOCATED_RANGES File: sparse

Close Help

Example test code using sparse files:

screen

File Edit View Search Terminal Help

```
def fiogetbsz(f):
        return struct.unpack('I', fcntl.ioctl(f, FIGETBSZ, struct.pack('I', 0)))
[0]

def main():
        if len(sys.argv) != 2:
                usage()

        # test mapping: filefrag -v and hdparm --fibmap
        # ioctl(3, FIGETBSZ, 0x55c339b16070)
        # ioctl(3, FS_IOC_FIEMAP, {fm_start=0, fm_length=1844674407370955\
        #        1615, fm_flags=0, fm_extent_count=292}
        f = os.open(sys.argv[1], os.O_CREAT|os.O_RDWR)
        bs = fiogetbsz(f)
        print 'Blocksize is %d.'  % bs
        if bs < 65536:
                print 'Changing this to 64k as that is real block size on window
s16 Needs fixing.'
                bs = 65536
        pwrite(f, 'a' * 1024 * 1024, 0)
        smb2_set_sparse(f, 1)
        smb2_set_zero_data(f, bs, bs * 2)
        smb2_set_zero_data(f, bs * 4, bs * 5)
        buf = smb2_query_allocated_ranges(f, 0, 1024 * 1024)
        while len(buf):
                r = struct.unpack_from('<2Q', buf, 0)
                print "%d...%d" % (r[0], r[0]+r[1])
                buf = buf[16:]

        os.close(f)


if __name__ == "__main__":
        main()
[sahlberg@rawhide-2 cifs]$
```

Now 77 smb3 dynamic tracepoints (a year ago was 20 ...)!

```
root@smf-Thinkpad-P51:~# ls /sys/kernel/debug/tracing/events/cifs
enable                  smb3_hardlink_done          smb3_query_dir_err                  smb3_ses_expired
filter                  smb3_hardlink_enter         smb3_query_info_compound_done       smb3_set_eof_done
smb3_close_err          smb3_hardlink_err           smb3_query_info_compound_enter      smb3_set_eof_enter
smb3_cmd_done           smb3_lease_done             smb3_query_info_compound_err        smb3_set_eof_err
smb3_cmd_enter          smb3_lease_err              smb3_query_info_done                smb3_set_info_compound_done
smb3_cmd_err            smb3_lock_err               smb3_query_info_enter               smb3_set_info_compound_enter
smb3_credit_timeout     smb3_mkdir_done             smb3_query_info_err                 smb3_set_info_compound_err
smb3_delete_done        smb3_mkdir_enter            smb3_read_done                      smb3_set_info_err
smb3_delete_enter       smb3_mkdir_err              smb3_read_enter                     smb3_slow_rsp
smb3_delete_err         smb3_open_done              smb3_read_err                       smb3_tcon
smb3_enter              smb3_open_enter             smb3_reconnect                      smb3_write_done
smb3_exit_done          smb3_open_err               smb3_reconnect_with_invalid_credits smb3_write_enter
smb3_exit_err           smb3_partial_send_reconnect smb3_rename_done                    smb3_write_err
smb3_falloc_done        smb3_posix_mkdir_done       smb3_rename_enter                   smb3_zero_done
smb3_falloc_enter       smb3_posix_mkdir_enter      smb3_rename_err                     smb3_zero_enter
smb3_falloc_err         smb3_posix_mkdir_err        smb3_rmdir_done                     smb3_zero_err
smb3_flush_err          smb3_query_dir_done         smb3_rmdir_enter
smb3_fsctl_err          smb3_query_dir_enter        smb3_rmdir_err
```

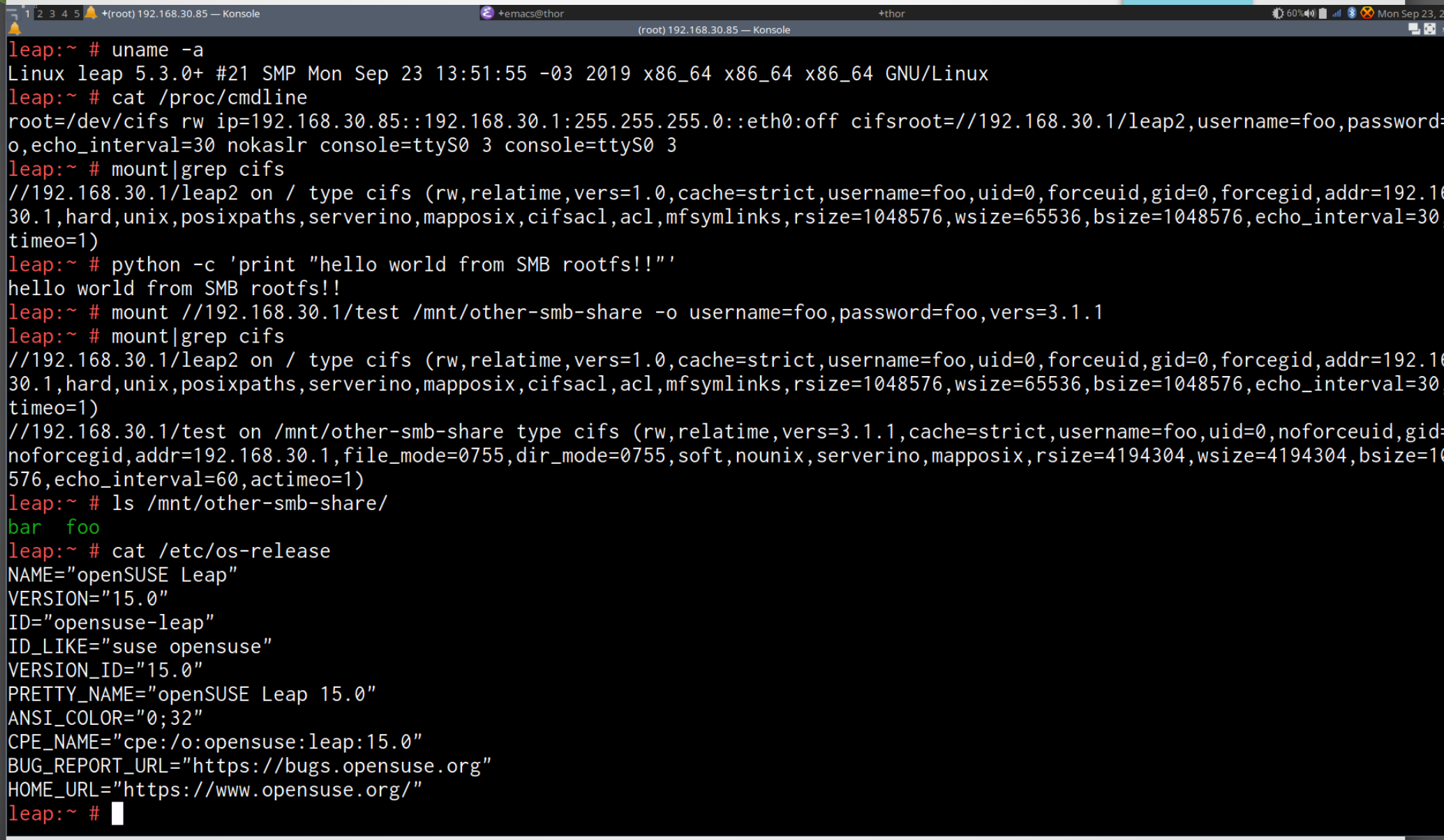
GCM Fast

- Can more than double write perf! 80% for read
- Works with Windows, and with complementary recent changes to Samba server, mounts to Samba also benefit (a lot)
- In 5.3 kernel

An example

- On this laptop … mounting to current Samba
  - Large writes 3x faster!
  - Large reads 2.5x faster
- Newly added SMB3.1.1 GCM support gives HUGE improvement in performance for large I/O

Boot diskless systems via cifs.ko!



Thank you Paulo!

- - Require ipconfig to set up network stack prior to mounting the SMB root filesystem:
  * E.g., "... ip=dhcp cifsroot=//localhost/share,..."

- - Current limitations:
  * no IPv6 support
  * default to insecure dialect SMB1 due to SMB1+UNIX extensions[1] (lack of SMB3+ POSIX extensions), although it can be changed through "cifsroot=" option. Fixes in progress for this to work with SMB3+
  * Init scripts that may fail due to unrecognized new cifsroot option

NetName context added

- In 5.3 kernel
- May help load balancers in the future (and debug tools too)

Can now view detailed info on open files (not just “lsof” output)

```
Sample output from "cat /proc/fs/cifs/open_files"

 # Version:1
 # Format:
 # <tree id> <persistent fid> <flags> <count> <pid> <uid> <filename> <mic
 0x5 0x800000378 0x8000 1 7704 0 some-file 0x14
 0xcb903c0c 0x84412e67 0x8000 1 7754 1001 rofile 0x1a6d
 0xcb903c0c 0x9526b767 0x8000 1 7720 1000 file 0x1a5b
 0xcb903c0c 0x9ce41a21 0x8000 1 7715 0 smallfile 0xd67
```

RDMA – Performance Improved

- Thank you Long Li! Many fixes/improvements
- No longer CONFIG_EXPERIMENTAL for smbdirect (RDMA) on Linux kernel client as of 5.3 kernel

4.20 (70 Changesets, December 23rd) cifs.ko internal module version 2.14

- RDMA and direct i/o performance improvements (add direct i/o to smb3 file ops)
- Much better compounding (create/delete/set/unlink/mkdir/rmdir etc.), huge perf improvements for metadata access
- Additional dynamic (ftrace) tracepoints
- Add */proc/fs/cifs/open_files* to allow easier debugging
- Slow response threshold is now configurable
- Requested rsize/wsize larger (4MB vs. 1MB)
- Query Info IOCTL passthrough (enables new “smb-info” tool to display useful metadata in much detail and also ACLs etc.), and allow ioctl on directories
- Many Bug Fixes (including for krb5 mounts to Azure, and fix for OFD locks, backup intent mounts)

5.0 (82 changesets) March 3rd, 2019 (cifs.ko internal module version 2.17)

- SMB3.1.1 requested by default (ie is now in default dialects list)
- DFS failover support added (can reconnect to alternate DFS target) for higher availability and

  DFS referral caching now possible, cache updated regularly (Thank you Paulo)
- Support for reconnect if server IP address changes (coreq change in user space implemented in latest version of cifs-utils) (Thank you Paulo!)
- Performance improvement for get/set xattr (compounding support extended)
- Many Bug Fixes (24 important enough for stable) including for large file copy in cases where network connection is slow or interrupted, reconnect fixes, and fix for OFD lock support. The buildbot is really helping improve cifs.ko code quality!

5.1 (86 changesets) May 5th 2019 (cifs.ko internal module version 2.19)

- “fsctl passthrough” support improved: allows tools like cifs-utils to easily query any info available over SMB3 fsctl or query_info
- New mount parm “handletimeout” to allow persistent/resilient handle behavior to be configurable
- Allow fallocate zero range to expand a file
- Improve perf: cache FILE_ALL_INFO for the shared root handle
- Improve perf: default inode block size reported as 1MB (NB: 4MB rsize/wsize)
- Cleanup mknod and special file handling (thank you Aurelien)
- Support guest mounts over smb3.1.1
- Add many dynamic trace points to ease debugging and perf analysis
- Bug fixes (23 important enough for stable). Adding even more tests to the buildbot really helped. Multiple fixes for ‘crediting’ (SMB3 flow control) – thank you Pavel!

5.2 (64 changesets, so far) July 7 2019. cifs.ko version 2.20

- Bug fixes (11 important enough for stable)
- Improved perf: sparse file support now allows fiemap, SEEK_HOLE and SEEK_DATA (helps cp to Samba e.g.)
- Add support for fallocate ZERO_RANGE
- Support “fsctl passthrough” for cases where send (write) data in SMB3 fsctl – allows user space tools to do more!

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

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

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

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

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

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

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

5.6 kernel – what is now merged (for April release) cifs.ko version 2.25

- “modefromsid” mount option much improved to set better ACL at file create time
- Add support for fallocate mode 0 for non-sparse files
- Allow setting owner info, DOS attributes and creation time from user space backup/restore tools (Thank you Boris Protopopov)
- Readdir performance optimization (add compouding support for readdir, cuts roundtrips for typical ls from about 9 to 7) (Thank you Ronnie)

5.6 kernel – what is in progress (testing here at SMB3 Plugfest)

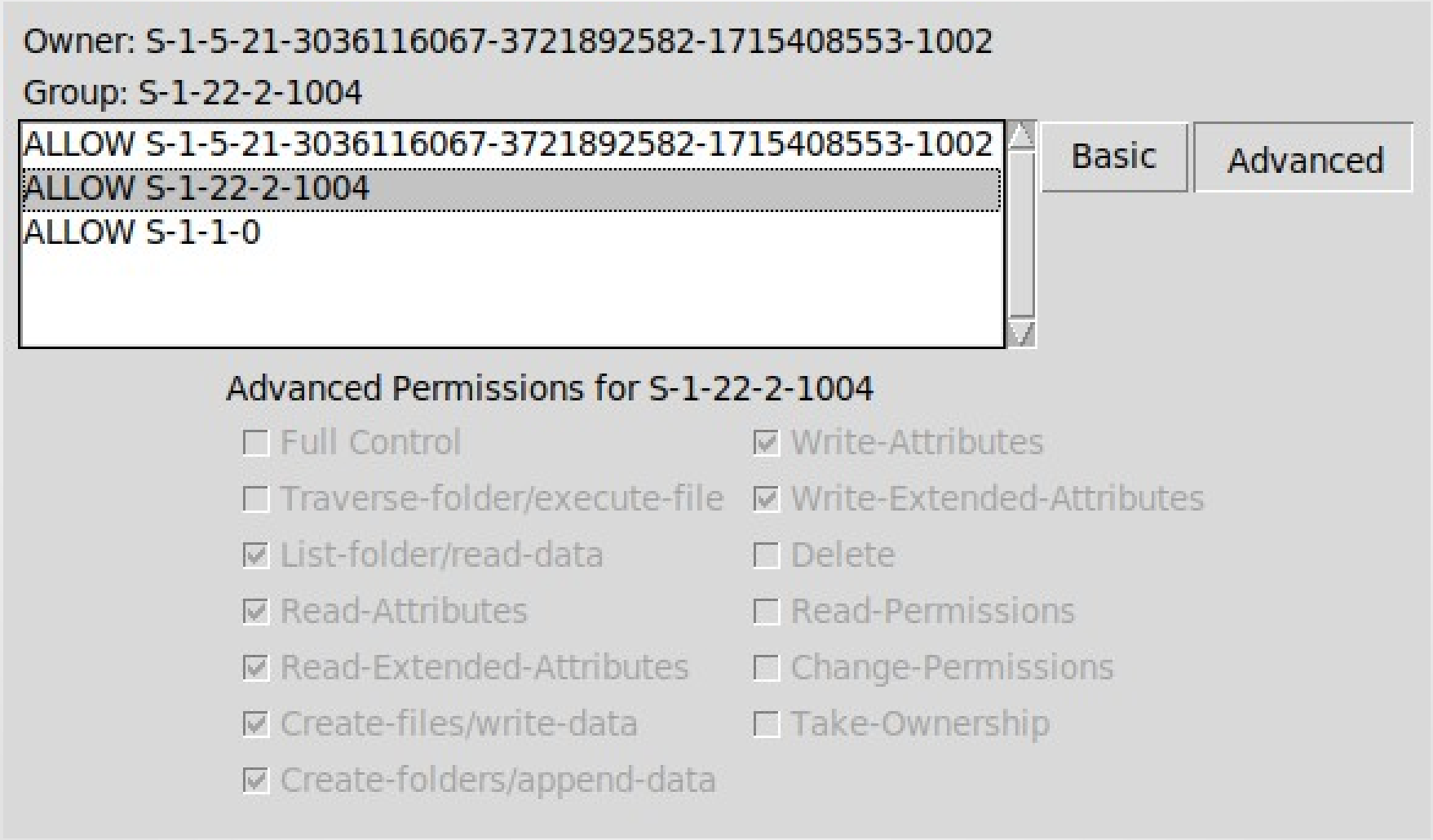
- Multichannel perf improvements
- Readdir improvements for modefromsid and cifsacl (so mode bits don’t get overwritten by default mode in readdir)
- Change notify support
- Swap over SMB3

Cifs-utils improvements

- Smbinfo rocks!
- Smbinfo rewritten in python
- Easy to extend
- New quota tools

Cifs-utils now even has a GUI!

- secddesc-ui.py



cifs-utils

- With pass-through SMB3 fsctl and query-info (and set-info) now possible it is easy to write user space tools to get any interesting info from the server
- Would love more contributions!
- Recently added python to make it easier to contribute
- Look at smbinfo from cifs-utils for examples

Recent example of how these are used

- With pass-through ioctl can now get quota information
  - New userspace helper tool, smb2quota.py, to display quota information for Linux SMB client file system
  - Will be part of cifs-utils
  - Thank you Kenneth D'souza!
- Let’s add more!

Sample output from smb2quota

```
/smb2quota.py -t /test
 Amount Used | Quota Limit | Warning Level | SID
 70.0 kiB    | 16.0 EiB    | 16.0 EiB      | S-1-5-32-544
 27.0 kiB    | 500.0 MiB   | 450.0 MiB     | S-1-5-21-3363399803-746912020-2622272238-1001
 4.0 MiB     | 16.0 EiB    | 16.0 EiB      | S-1-5-18
# ./smb2quota.py -c /test
S-1-5-32-544,71680,18446744073709551615,18446744073709551615
S-1-5-21-3363399803-746912020-2622272238-1001,27648,524288000,471859200
S-1-5-18,4220928,18446744073709551615,18446744073709551615
# ./smb2quota.py -l /test
SID:S-1-5-32-544
Quota Used:71680
Quota Threshold Limit:NO_WARNING_THRESHOLD
Quota Limit:NO_LIMIT
SID:S-1-5-21-3363399803-746912020-2622272238-1001
Quota Used:27648
Quota Threshold Limit:471859200
Quota Limit:524288000
SID:S-1-5-18
Quota Used:4220928
Quota Threshold Limit:NO_WARNING_THRESHOLD
Quota Limit:NO_LIMIT
```

Common Configuration Options – Suggested use cases

- Frequently recommended
  - mfsymlinks
  - noperm
  - dir_mode=, file_mode=, uid=, gid=
- Sometimes recommended
  - cifsacl,idsfromsid or (now 5.6 and later) modefromsid
  - actime=
  - sec=krb5
  - seal
  - sfu
  - hard
  - nostrictsync (and also cache=)

Testing … testing … testing

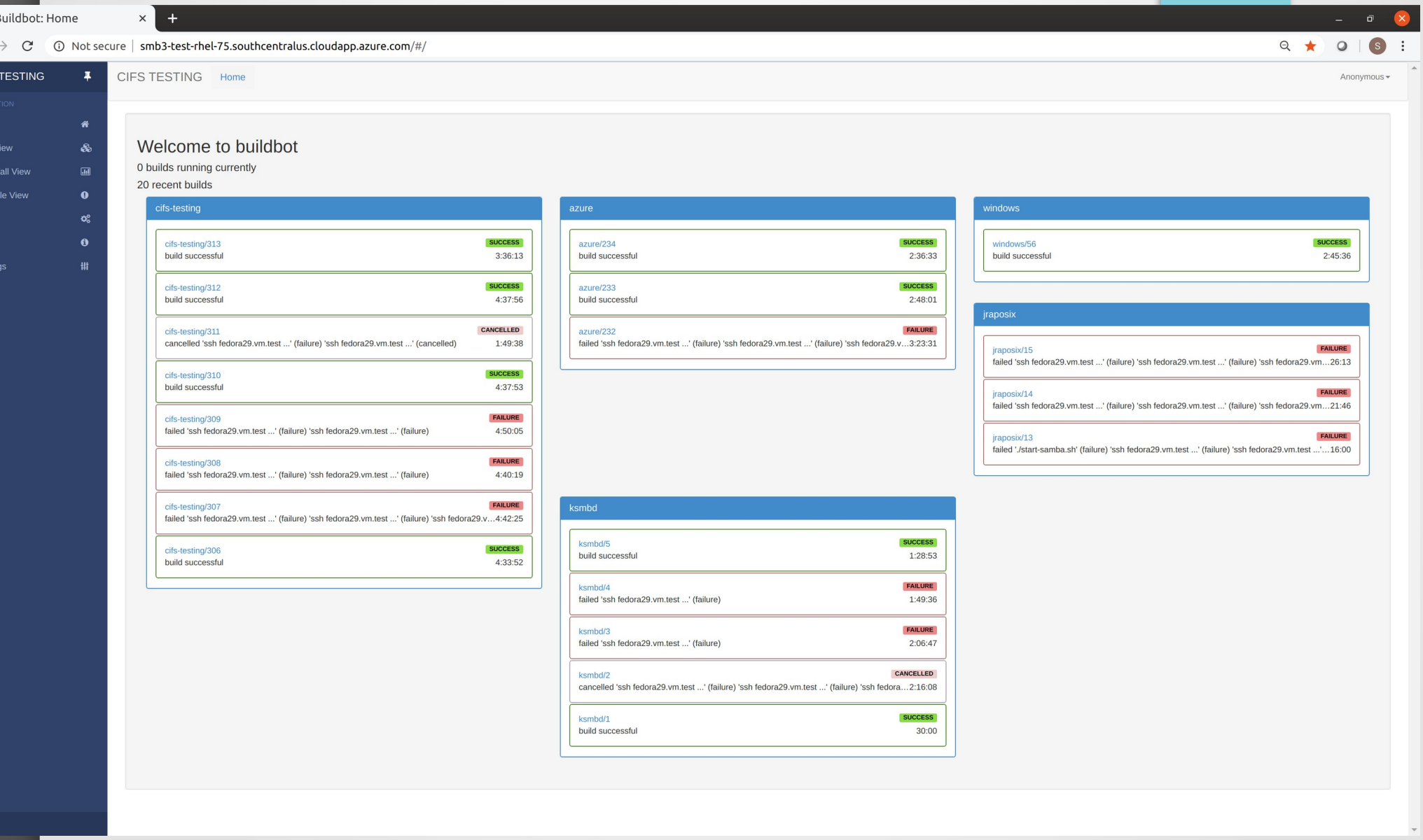
- The “buildbot” - automated regression testing! Thank you Paulo, Ronnie and Aurelien. See:

  http://smb3-test-rhel-75.southcentralus.cloudapp.azure.com
- See xfstesting page in cifs wiki https://wiki.samba.org/index.php/Xfstesting-cifs
- Easy to setup, exclude file for slow tests or failing ones
- Huge improvement in XFSTEST – up to 127 groups of tests run over SMB3 (more than run over NFS)! And more being added every release

Thanks to the buildbot – Best Releases Ever for SMB3!

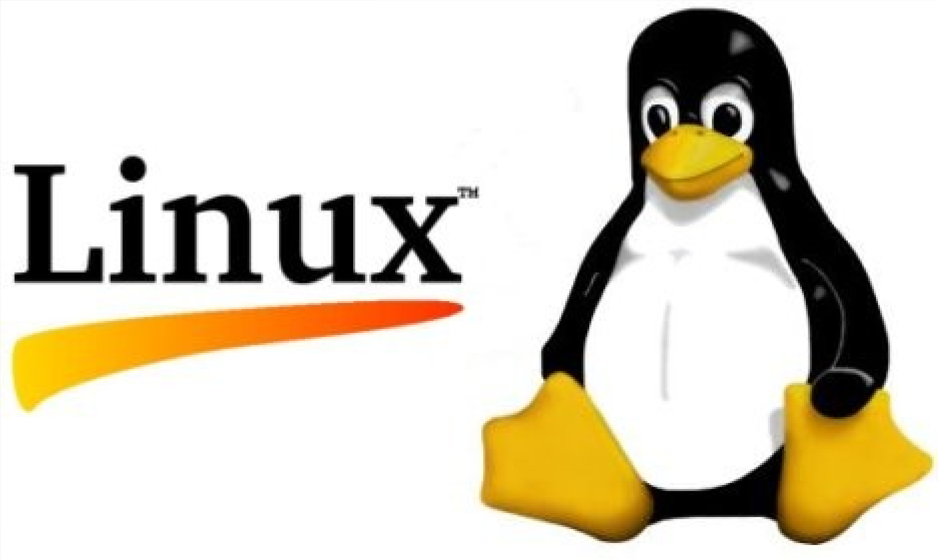
- Prevents regressions
- Continues to improve quality

Buildbot now has even more targets



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