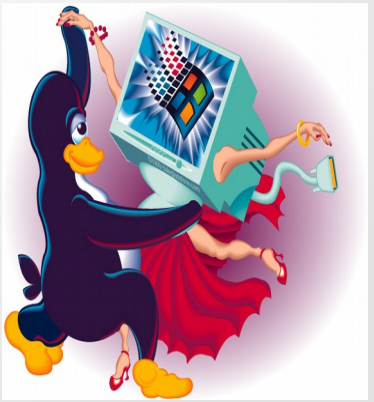
# Beyond SMB3: New Developments in the Linux SMB3 Implementation

Steve French
Principal Systems Engineer – Primary Data



# Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of Primary Data 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 and former SNIA CIFS Working Group chair
- Principal Systems Engineer: Primary Data

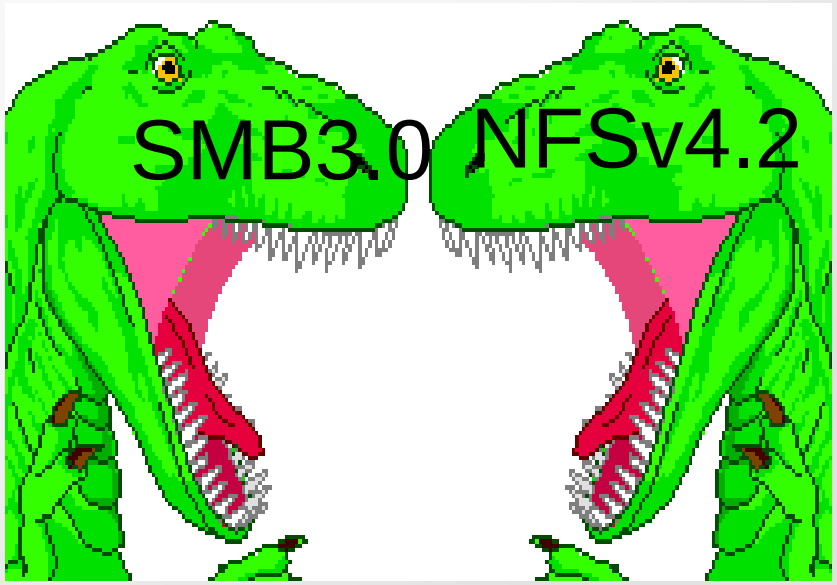
# Why file systems?

- Almost 50 years after the invention of the first File System, we care more than ever about how we store our data. The amount of data (largely unstructured) exceeded a Zettabyte in 2010 (IDC estimate), and continues to double every two to three years.
- Nearly all workloads depend on file systems. File Systems still matter more than ever with the explosion of “unstructured data” - in part due due to cloud, new web applications, video, audio.

# Why NAS (network file protocols? ... When could use SAN or object instead

- NAS is a superset of block (SAN) and object
  - But easier to manage
- NAS (now) can get 90+ of the performance of SAN with lower administrative costs and more flexibility
- And you get attributes at the right granularity (file/directory/volume)
  - Ownership information, easier to understand security, easy backup, useful info on application access patterns, intuitive archive/encryption/compression policy, quotas


- “Dinosaurs” created in same year reborn faster & strong
  - SMB3 (late 2012, Windows 8, Windows 2012 Server)
  - SMB3.02 (Windows 8.1, Windows 2012 R2)
  - NFSv4.1 (IETF spec approved 2010)
  - NFSv4.2 (coming soon)



# And why Linux?

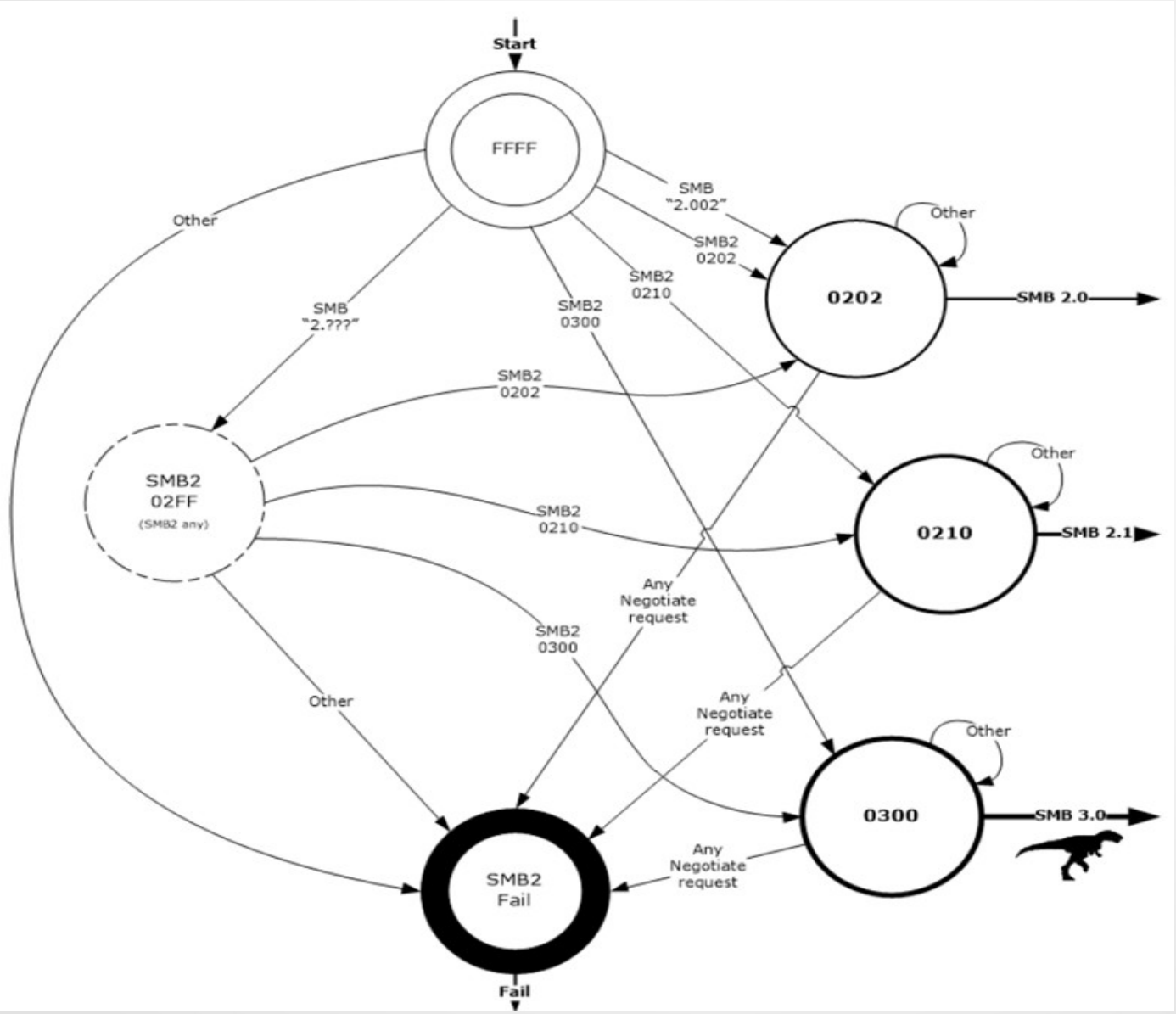
- Large Talented Community. Rate of improvement is unsurpassed
  - More than 75,000 changesets in the kernel last year, 4900 in the file system alone
  - Changes from over 1200 developers are added to the kernel each release
  - Development never stops – constant incremental improvements and fixes
  - The processes and tools (e.g. “git” distributed source code control) work
- Broad selection of file systems. More than 50 file systems to choose from including:
  - Local File Systems (ext4, xfs, btrfs, fat)
  - Cluster File Systems (ocfs2, gfs2)
  - Network File Systems (nfs, cifs/smb2/smb3, ceph)
  - Special Purpose File Systems
  - FUSE (user space file systems helper) enables many more (including Gluster and NTFS

# Linux FS Community is talented (See us at the 2014 FS Summit)

# Most Active Linux Filesystems

- 4872 filesystem changes since 3.11 kernel!
  - Linux kernel file system activity is continuing to be very strong
- cifs.ko (cifs/smb3 client) among most active fs
  - Btrfs 820 changesets
  - VFS (overall fs mapping layer and common functions) 591
  - Xfs 532
  - Nfs client 403
  - Ext4 239
  - CIFS/SMB2/SMB3 client 210
  - Nfs server 368 (activity increasing, most are very recent, in last two releases)
- NB: Samba (cifs/smb2/smb3 server) is more active than all those put together since it is broader in scope (by a lot) and also is in user space not in kernel

# SMB3 Rocks


Start
FFFF
Other
SMB
"2.002"
SMB2
0202
SMB2
0210
SMB2
0300
SMB
"2.???"
0202
Other
SMB 2.0
SMB2
0202
SMB2
02FF
(SMB2 any)
0210
Other
SMB 2.1
SMB2
0210
Any
Negotiate
request
SMB2
0300
Other
Any
Negotiate
request
0300
Other
SMB 3.0
SMB2
Fail
Any
Negotiate
request
Fail

# Although network API closer to Windows than POSIX, CIFS and SMB3 not really Windows specific

- Mac, Solaris, Linux and most other operating systems have kernel clients. Solaris and Mac even use CIF ACLs in-kernel. CIFS/SMB2 default for some Unix and all Windows.
- CIFS “Unix Extensions” developed by SCO, extended by HP and then Linux and Mac. Improve most “pos vs. windows” issues such as retrieving Linux mode, POSIX ACL and POSIX locking
- CIFS Unix Extensions implemented in Samba and Linux kernel client among others.
- Unix Extensions are optional (when mounted to Windows, they are emulated instead, sometimes using th same approach as “Services for Unix”). Mount from Linux to Windows just works for most applications. N NFSv3 is not completely POSIX friendly but NFSv4.2 is close to complete mapping of Linux file operation
- For SMB3 Linux/POSIX extensions are under development (see later slides)
  - Microsoft made SMB2 slightly more “unix friendly” so extensions for SMB2 will be smaller
  - SMB3 Unix Extensions design in progress

# Current Versions (SMB3.0 vs. NFSv4.1)

- Both have borrowed from each other: NFSv4 in particular added various cifs features (including statefulness, and various security features)
- SMB3.0 and NFSv4.1 both include:
  - Kerberos authentication, packet signing, encryption
  - “RichACL” (CIFS ACLs)
  - Support for file transfers via RDMA
- NFSv4.1 includes optional pNFS (file or block or object) to spread network i/o load from a single client across a cluster
- But SMB3.0 and related protocols now include
  - Multipath, per-share encryption, better server side copy, support for copy on write files, claims based access control, branch caching (content addressable storage), volume shadow copy, improved cluster awareness and load balancing, T10 extensions, flow control on every response, application aware and also transparent failover

# Will NFSv4.2 Address SMB 3 gaps?

- See http://www.nfsv4bat.org/Documents/index.html for recent presentations
  - And http://datatracker.ietf.org/wg/nfsv4/ for official ietf standards documents
- NFSv4.2 specification does include some items to close gaps:
  - Server side file copy
  - “punch hole” support
  - Fadvise (indicate file access patterns) and allocate (space reservation) support
- And of course “pNFS” (optional in 4.1 and 4.2) does not have an equivalent in SMB3 although SMB3 does support clustering and a global name space SMB3 does not have ability to split a file across multiple data servers as NFS does
- And NFSv4.2 spec includes bug fixes (for NFSv4/NFSv4.1 spec problems)
- Fortunately 4.2 is a much smaller update than NFSv4.1 (1/7th the document size).
- But … SMB 3 already has MUCH wider deployment, and widely supported
- SMB 3 Unix Extensions are not complete yet (for complete Linux application interoperability)
- An interesting new optional pNFS layout type (“flexfiles”) has been proposed to IETF by some of my colleagues which expands NFS use cases and allow NFS data servers to not have to be as tightly coupled to the metadata server (see https://tools.ietf.org/html/draft-bhalevy-nfsv4-flex-files-03)

# SMB3 Development activity continues

- Kernel client (cifs.ko)
  - SMB2, 2.1 and 3.0 (and even minimal 3.02) support are in!
  - SMB3 is MUCH faster for large file read/write now! In some cases now fastest way to copy files
  - Current version is 2.04 and is visible via modinfo (and in /proc/fs/cifs/DebugData)
    - In one year we have gone from kernel 3.11 to 3.17-rc4
  - Over 200 kernel changesets for cifs, a typical year, but recent activity increasing
  - More than 20 developers contributed
  - cifs continues to be one of the more active file systems in kernel
  - Big improvements in testing of cifs and smb3 for kernel client
- Samba server also continues to improve its SMB2 and SMB3 support
  - And not just the server … Smbclient (user space ftp like tools) supports SMB2

# Kernel (including cifs client) improving

- 14 months ago we had 3.10 “Unicycling Gorilla”

Now we have 3.17-rc4 “Shuffling Zombie Juror”

# Features in process

- SMB3 ACL support
- Recovery of pending byte range locks after server failure (we already recover successful locks)
- Investigation into additional copy offload (server side copy) methods
- Full Linux xattr support
  - Empty xattr (name but no value)
  - Case sensitive xattr values
  - Security (SELinux) namespace (and others)
- SMB3 Unix Extensions prototyping
- With Richard Sharpe's work on RDMA in the Samba server, is it time to push harder to do SMB3 RDMA on the kernel client?

# Improvements by release

- 3.7 97 changes, cifs version 2.0
  - SMB2 added: **support for smb2.1 dialect added!**
  - remove support for deprecated "forcedirectio" and "strictcache" mount options
  - remove support for CIFS_IOC_CHECKUMOUNT ioctl
- 3.8 60 changes, cifs version 2.0
  - ntlmv2 auth becomes default auth (actually ntlmv2 encapsulated in NTLMSSP)
  - **smb2.02 dialect support added** and smb3 negotiation fixed
  - don't override the uid/gid in getattr when cifsacl is enabled
- 3.9 38 changes, cifs version 2.0
  - dfs security negotiation bug fixes (krb5 security).  Rename fixes
- 3.10 18 changes, cifs version 2.01
  - cifs module size reduced
  - nosharesock mount option added
- 3.11 69 changes, cifs version 2.01
  - Various bug fixes: DFS, and workarounds for servers which provide bad nlink value
  - Security improvements (including SMB3 signing, but not SMB3 multiuser)
  - Auth and security settings config overhaul (thank you Jeff!)
  - SMB2 durable handle support (thank you Pavel!)
  - Minimal SMB3.02 dialect support

# Improvements by release (continued)

- 3.12 40 changes, cifs version 2.02: **SMB3 support much improved**
  - SMB3 multiuser signing improvements, (thank you Shirish!) allows per-user signing keys on ses
  - SMB2/3 symlink support (can follow Windows symlinks)
  - Lease improvements (thank you Pavel!)
  - debugging improvements
- 3.13 34 changes
  - Add support for setting (and getting) per-file compression (e.g. “chattr +c /mnt/filename”)
  - Add SMB copy offload ioctl (CopyChunk) for very fast server side copy
  - Add secure negotiate support (protect SMB3 mounts against downgrade attacks)
  - Bugfixes (including for setfacl and reparse point/symlink fixes)
  - Allow for O_DIRECT opens on directio (cache=none) mounts. Helps apps that require directio such as newer specsfs benchmark and some databases
  - Server network adapter and disk/alignment/sector info now visible in /proc/fs/cifs/DebugData
- 3.14 27 changes
  - Security fix for make sure we don't send illegal length when passed invalid iovec or one with invalid lengths
  - Bug fixes (SMB3 large write and various stability fixes) and aio write and also fix DFS referrals when mounted with Unix extensions

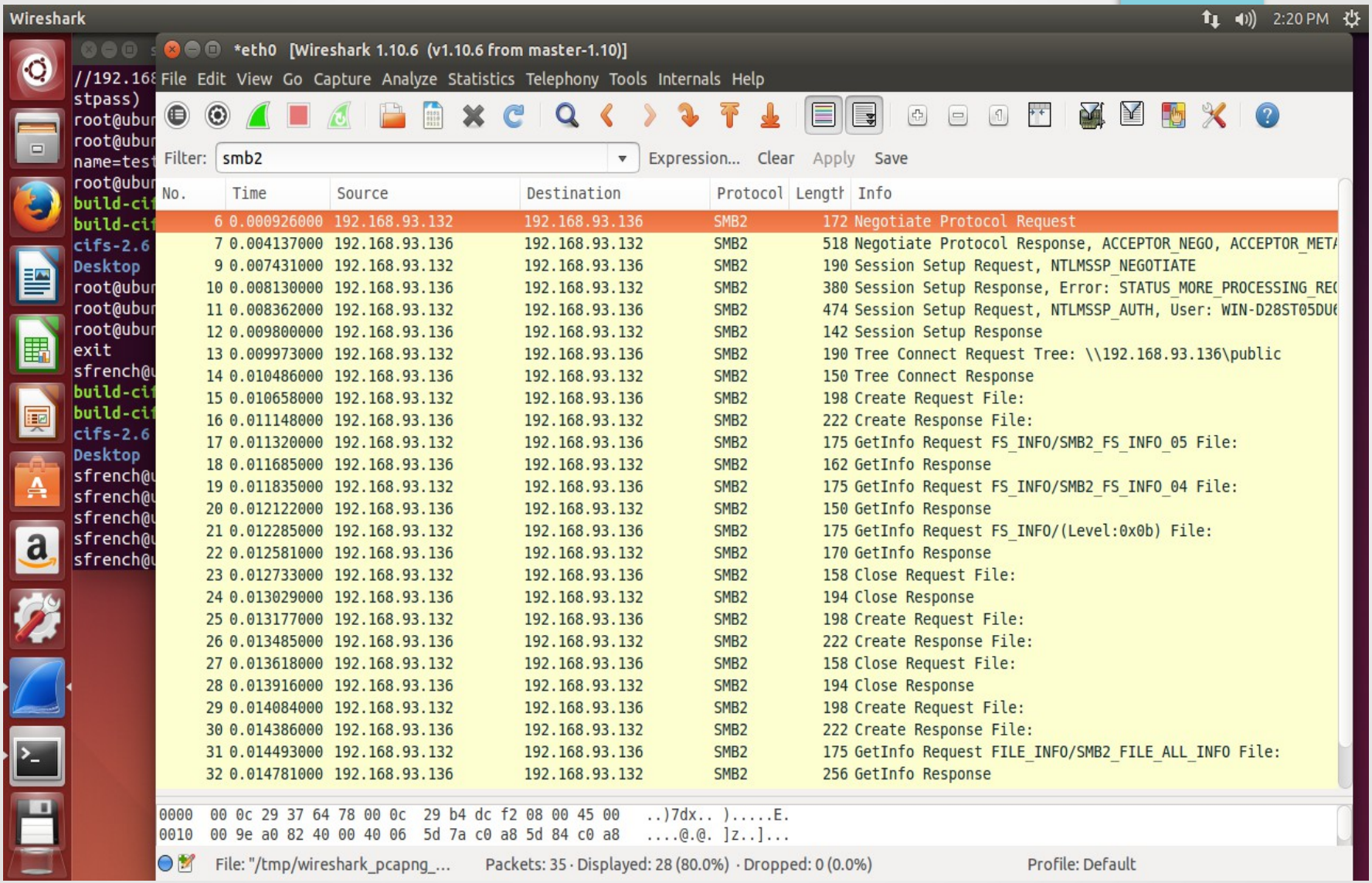
# Improvements by release (continued)

- 3.15 18 changes
  - Various minor bug fixes (include aio/write, append, xattr, and also in metadata caching)
- 3.16 25 changes
  - Allow multiple mounts to same server with different dialects
  - Authentication session establishment rewrite to improve gssapi support
  - Fix mapchars (to allow reserverd characters like : in paths) over smb3 mounts
- 3.17 65 changes (cifs version 2.04 – visible in modinfo)
  - Much faster SMB3 large read/write: including multicredit support (thank you Pavel!)
  - Many SMB3 fixes (found by newly updated automated fs tests: “xfstests”)
  - Directio allowed on cache=strict mounts
  - Fallocate/sparse file support for SMB3
  - Workaround problem with smb2.1 mounts to MacOS
- 3.18 (Some highlights of what to expect in next kernel)
  - SMB3 Emulated symlinks: Mfsymlink support for smb2.1/smb3 (complete).
  - SMB3 POSIX Reserved Character mapping: support for reserved characters e.g. * : ? < > etc. (complete)
  - ACL support for SMB3 (expected). Will be able to use cifsacl mount option for smb3 mounts
  - Multichannel? and SMB3 Unix extensions?

# Cifs-utils

- The userspace utils: mount.cifs, cifs.upcall,set/getcifsacl,cifscreds, idmapwb,pam_cifscreds
  - thanks to Jeff Layton for maintaining cifs-utils
- 31 changesets over the past year
  - Current version is 6.4
  - Includes various bugfixes (especially in setcifsacl util)
  - Dedicated kerberos keytab (other than system default) can be specified.
- Also of note: in 12/2012 Idmap plugin supportwas added (allows sssd, not just winbind, cached userid information to be used) in version 5.9 of cifs-utils

# SMB3.02 Mount to Windows



# Using SMB3

- Practical tips
  - Use -o vers=3.0 to Samba or Windows (or vers=3.02 to latest Windows, consider vers=2.1 to MacOS)
  - Mount options to consider
    - Cifsacl and mfsymlinks (3.18 or later kernel)
    - “sfu” option enables creation of FIFOs and char devices
    - And what about rsize/wsize?
- Restrictions
  - Case sensitivity
  - POSIX vs. Windows byte range locks, and unlink behavior

# SMB3 Kernel Client Status

- SMB3 support is solid (and FAST!), but lacks many optional features
- Badly needs Unix/Linux extensions for full posix app compatibility on Linux clients (and to compare with Apple's SMB2.1/SMB3 “AAPL” create context” which addresses a few POSIX compatibility issues)
- Can mount with SMB2.02, SMB2.1, SMB3, SMB3.02
  - Specify vers=2.0 or vers=2.1 or 3.0 or 3.02 on mount
  - Default is cifs but also mounting with vers=1.0 also forces using smb/cifs protocol
  - Default will change to SMB3 when Unix extensions available for SMB3, and performance and functional testing is as good or better

# SMB3 Kernel Status continued

- In:
  - SMB2.1 Lease support (improved caching)
  - SMB2 durable handles (improved data integrity)
  - Multicredit, fast large reads/writes
  - SMB3 signing (including for multiuser mounts)
    - Downgrade attack protection (secure negotiate)
  - Dynamic crediting (flow control)
  - Not SMB3 specific: Compressed files, copy offload
  - Windows 'NFS' symlinks (partial)

# SMB3 Kernel Status continued

- TODO
  - ACLs for SMB2/SMB3
  - 3 types symlinks: Windows, Windows 'NFS' and 'MF"
    - Only mfsymlinks are complete (partial support for other 2)
  - POSIX/Unix extensions (see recent work by Volker)
  - Optional features:
    - Multichannel (started) and RDMA
    - Persistent handles
    - Witness protocol, improved cluster reconnection
    - Encrypted share support
    - ODX Copy Offload support (but can do CopyChunk)

# SMB3 POSIX Extensions

**In progress.  Discussions this week.**

# SMB3 Performance considerations

- Informal perf results 3.16-rc4 (Ubuntu) client. Server Windows 8.1.  VMs on same host (host disk is fairly fast SSD).
  - Copy to server performance increased about 20% percent (similar with or without conv=fdatasync)
  - dd if=/dev/zero of=/mnt/targetfile bs=80M count=25
  - 1st run copy to empty directory, 2nd run copy over target, (pattern repeated multiple times) averaging results
  - New code (with Pavel's patches)
  - ----------
  - CIFS 167MB/s
  - SMB3 200MB/s

  - Existing code (without his patches)
  - ----------
  - SMB3 166MB/s
  - CIFS 164.5MB/s

# More SMB3 Performance

- For large file reading SMB3 performance with Pavel's patches increased 76% over existing SMB3 code
- dd of=/dev/null if=/mnt/targetfile bs=80M count=25 (mounting and unmounting between attempts to avoid caching eff on the client)
- New code (with Pavel's patches)
- ----------
- CIFS 114MB/s
- SMB3 216MB/s

- Existing code (without his patches)
- ----------
- SMB3 123MB/s
- CIFS 110MB/s

# More SMB3 Performance Linux->Linux

- client Ubuntu with 3.16-rc4 with Pavel's patches, srv Fedora 20 (3.14.9 kernel Samba server version 4.1.9)
- dd if=/mnt/testfile of=/dev/null bs=50M count=30
- testfile is 1.5GB existing file, unmount/mount in between each large file copy to avoid any caching effect on client (although server will have cached it)

- SMB3 averaged 199MB/sec reads (copy from server)
- CIFS averaged 170MB/sec reads (copy from server)
- NFSv3 averaged 116MB/sec (copy from server)
- NFSv4 and v4.1 averaged 110MB/sec (copy from server)

- Write speeds (doing dd if=/dev/zero of=/mnt/testfile bs=60M count=25) more varied but averaged similar speeds for copy to server for both NFSv3/v4/v4.1 and SMB3 (~175MB/s)
- NB: Additional NFS server and client scalability patches have recently been added to kernel (it is possible that they may help these cases)

# Testing … testing … testing

- One of the goals for this summer was to improve automated testing of cifs.ko
  - Multiple cifs bugs found, test automation much improved, approximately 5  bugs/features remain to be fixed for full xfstest compatibility
  - See https://wiki.samba.org/index.php/Xfstesting-cifs
- Functional tests:
  - Xfstest is the standard file system test bucket for Linux
    - Runs over local file systems, nfs, and now cifs/smb3
      - Found multiple bugs when ran this first
    - Challenge to figure out which tests *should* work (since some tests are skipped when run over nfs and cifs)
  - Other functional tests include cthon, dbench, fsx. Cthon also has recently been updated to better support cifs
- Performance/scalability testing
  - Specsfs works over cifs mounts (performance testing)
  - Big recent improvements in scalability of dbench (which can run over mounts)
  - Various other linux perf fs tests work over cifs (iozone etc.)
  - Need to figure out how to get synergy with iostats/nfsstats/nfsometer

# XFSTEST current status

- Multiple server bugs found to
- Client bugs:
  - Generic tests: 011 (dirstress), 023 and 245 (rename), 075/091/127/263 (fsx failures fallocate related), 239 (need ACLs), 313 (timestamps)

- The Future of SMB is very bright
- Continued improvement over 30 years
- Here's to another 30 years!

# Thank you for your time