Linux: Shaping the Future of Network File Systems: A Status Update

Steve French
Principal Systems Engineer – Primary Data

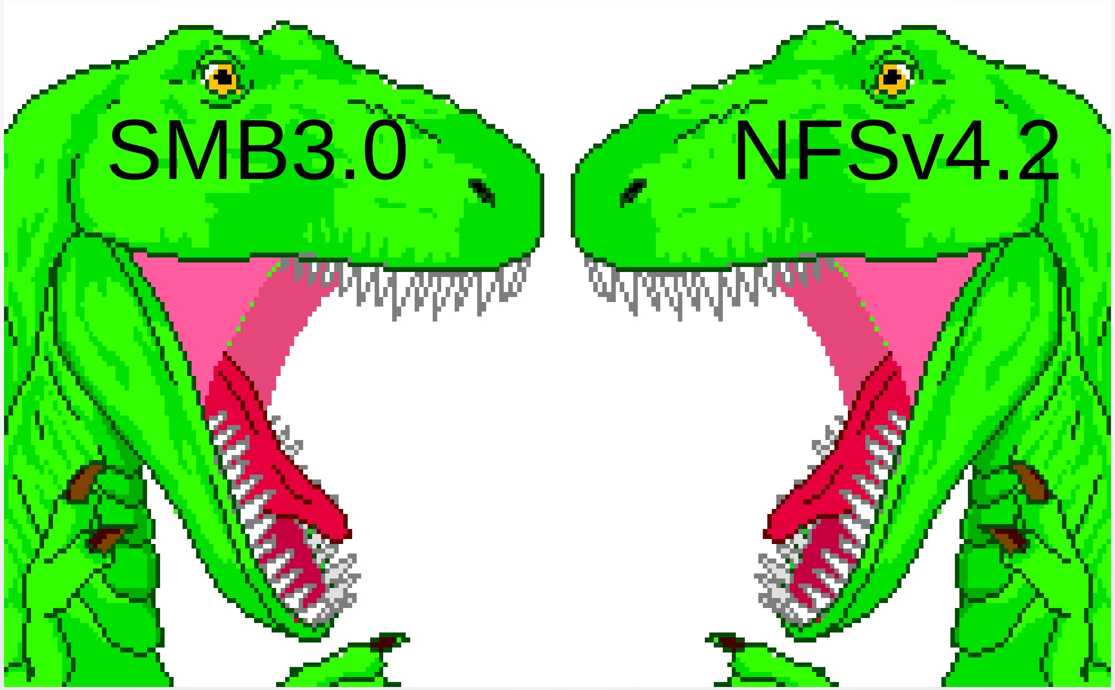
Legal Statement

- 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 SMB/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 Grou chair
- Work for Primary Data

- These “Dinosaurs” created in the same Orwellian year reborn – faster and stronger!



Why do we care about 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

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, and out of kernel: Lustre and IBM GPFS)
  - Network File Systems (nfs, cifs/smb2/smb3, ceph)
  - Special Purpose File Systems
  - FUSE (user space file systems helper) enables many more (including Gluster and NTF

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

Most Active Linux Filesystems

- 5277 filesystem changes since 3.9 kernel!
  - Linux kernel file system activity is continuing to be very strong
- Most active fs
  - Btrfs 883 changesets
  - VFS (overall fs mapping layer and common functions) 676
  - Xfs 524
  - Nfs client 396
  - Ext4 338
  - CIFS/SMB2/SMB3 client 220
  - Nfs server 200
- 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

New generation of network fs are born!

- 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)

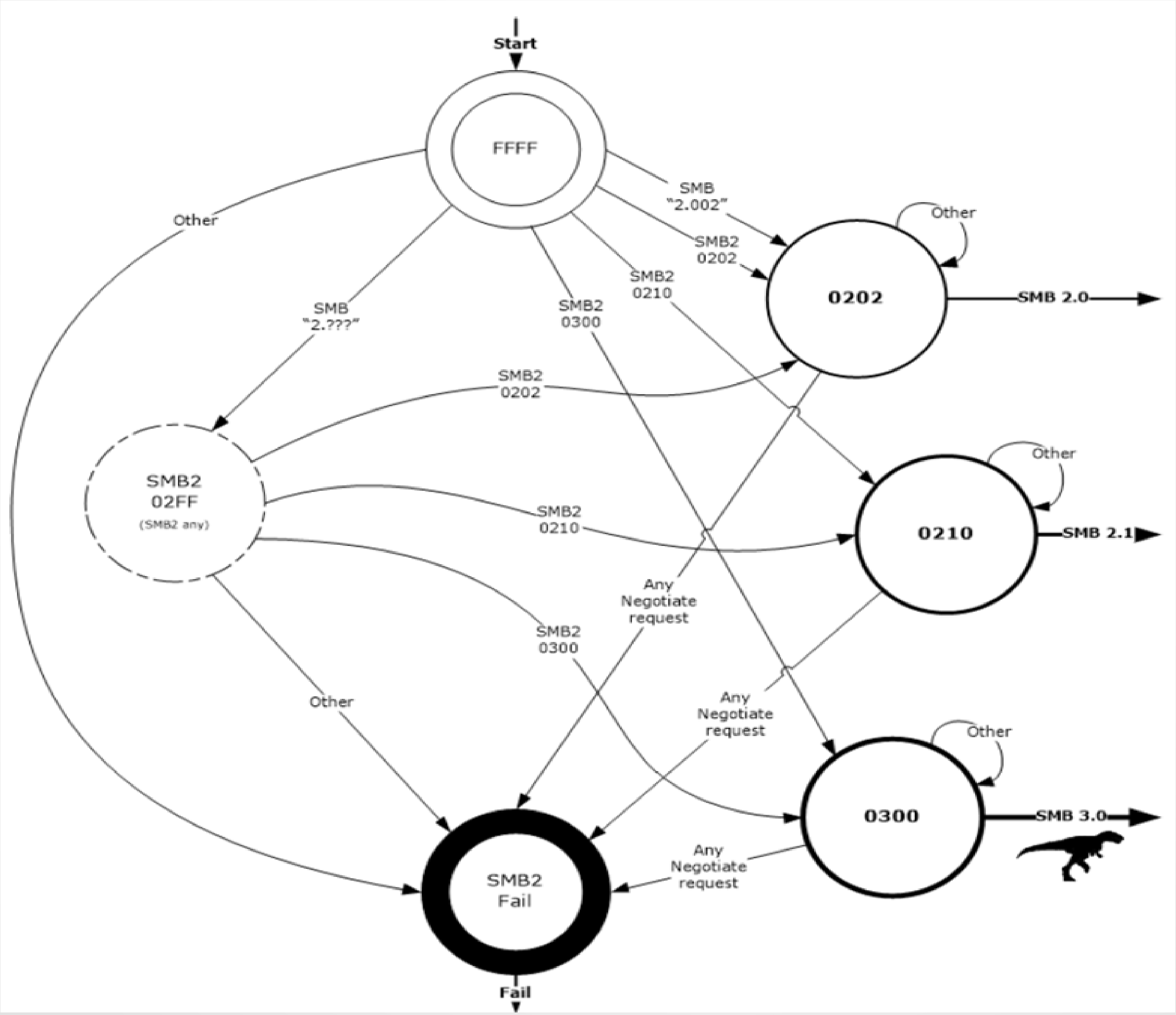
SMB3: Great Feature Set, Broad Deployment, Amazing Performance

- Introduction of new storage features in Windows causing one of most dramatic shifts in storage industry as companies rapidly move to support "SMB3" (aka SMB2.2)
- "SMB2.2 (CIFS) screams over InfiniBand" (Stora CH Blog)
- Is (traditional) SAN use going to die?
  - "Market trends show virtualization workloa moving to NAS" (Dennis Chapman, Techni Director NetApp, SNIA SDC 2011)
  - "Unstructured data (file-based) storage is growing faster than structured data" (Thom Pfenning, Microsoft GM, SNIA SDC 2011)
  - Customers prefer "file"
- SMB3 market share is obviously MUCH larger NFSv4.1 now but pNFS/NFSv4.1 deployment slow and NFSv4.2 coming soon ...

Performance Caught My Attention too … SMB3.0 is FAST

- At the SNIA Developer Conference three years ago... an amazing demo. SMB2.2 (aka SMB3) using RDMA reached over 3.7GB/sec throughput for a common database benchmark
  - And did not look as hard to setu
- It gets better … Microsoft demonstrated even faster performance more recently
- Storage Developers asking is “Network” (SMB3 File Access) now better than “Local”?

SMB3 Rocks


Start
FFFF
SMB "2.002"
Other
SMB2 0202
SMB2 0210
SMB "2.???"
SMB2 0300
Other
0202
SMB 2.0
SMB2 0202
Other
SMB2 02FF (SMB2 any)
SMB2 0210
0210
SMB 2.1
Any Negotiate request
SMB2 0300
Other
Any Negotiate request
Other
0300
SMB 3.0
Any Negotiate request
SMB2 Fail
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 operatio
- No Unix/Linux Extensions for SMB3 (yet)
  - Microsoft made SMB2 slightly more “unix friendly” so extensions for SMB2 will be smaller
  - SMB3 Unix Extensions design in progress

opening windows to a wider world

What about NFS?

- Once many distributed/network file systems, now down to two popular ones:
  - NFS (v3/v4/v4.1) and CIFS/SMB2/SMB3
- And some special purpose cluster specific fs (note that you couldn't run general applications o and Hadoop mounts so those aren't considered network file systems)
- NFS
  - Created by Sun 1984 (roughly the same time as SMB) who documented NFSv2 in 1989, a documented NFSv3 in 1995,
  - NFSv4 became an open internet standard relatively late: RFC 3530 in 2003. It heavily bor from cifs features (oplock, ACLs, DFS referrals)
  - An enormous update (more than 600 pages), NFSv4.1 became a standard in 2010, and in optional pNFS (parallel) support and better cifs/windows ACL interoperability
  - But NFS v4.1 Implementations have lagged, and even NFSv4 deployment slower than expected. Mos NFS users still use 17 year old NFS version 3, even though it is not fully posix compliant and can not d caching
  - Work continues on an opensource userpsace Ganesha NFS server with NFSv4.1/pNFS support, and a kernel pNFS kernel server (currently maintained out of kernel). Few pNFS clients other than Linux ex

SMB vs. NFS

- SMB1
  - Stateful
  - Per-user connections
  - Directly uses TCP (RFC1001)
  - Can be used as a transport for other protocols (DCE/RPC, print, systems management)
  - Originally optimized for DOS/OS2 then Windows
  - Rich: Lots of file operations

- NFSv3
  - Stateless (mostly). Idempotence allows operations to be repeated
  - Does not guarantee safe caching
  - Per-system connections
  - Runs over SunRPC transport protocol
  - Originally for Solaris
  - Minimal: Few file operations, almost a “lowest common denominator” across Unix

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

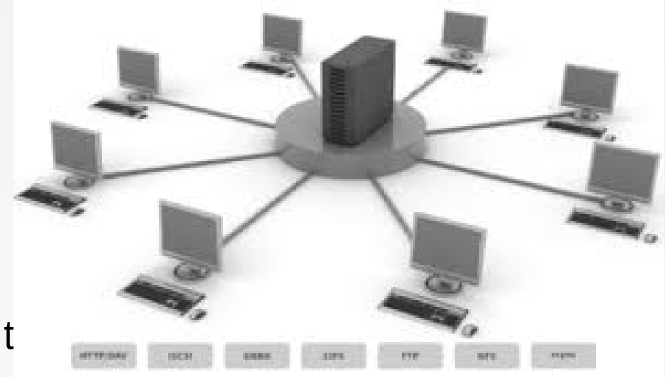
Will NFSv4.2 Address SMB 3 gaps?

- 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 others in NAS community (EMC and NetApp) support it
- NFSv4.2 implementations likely to lag (a lot) behind SMB 3 (NB: some still working on NFSv4.1)
- On the other hand … SMB 3 Unix Extensions are not complete yet (needed for complete Linux application interoperability)
- Although not likely to be as widely used outside of Linux as SMB 3, the amount of NFS4.2 usage in pure Linux (ie Linux client mounts to Linux server) will depend in part on the quality of the respective implementations (SMB3 vs. NFSv4.2) not just the protocol features

What about other Network, Cluster, Distributed, Cloud … File Systems?

- Cluster File Systems (GPFS, Luster, GFS2, OCFS2)
  - Narrow target market Can run well on either one or only a few operating systems
    - Support not included with shipping PCs (Windows and Mac)
  - They will continue to have a place in the server room “behind” the servers for the more common protocols (SMB2, NFS, some cases Web).
  - With SMB 3 and pNFS so fast - don't mount clusterfs directly (let Samba srv do it)
  - They will continue to have a place in HPC, but more limited due to SMB2 and pNFS
    - NB: “Watson” did use GPFS, but behind NFS servers that HPC nodes mounted
- OpenAFS, GlusterFS, “CloudFS”
  - Performance issues (not just due to usual problems with user space file systems)
  - Narrow target market
    - Can run well on either one or only a few operating systems
    - Support not included with shipping PCs (Windows and Mac)
  - They don't tie into enterprise archive, data retention, security tooling on NAS
- Virtfs
  - Great for Linux guest to Linux host mounts (on the same box) due to virtio (could do virtio for SMB2 if desired) and stron affinity
- For the vast majority of use cases – it comes down to NFS or CIFS/SMB3

What about using block devices instead of file?



- SAN vs. NAS, block vs. network file access
- Why use file? Well ... it is cheaper AND
  - Easier to administer and setup: Users and Admins understand t
  - Security is easier on files (which have owners and ACLs)
  - Setting up Data Retention (Backup) and dedup correctly is easier on files
  - Application Access patterns can be optimized better or
  - More flexible (block devices emulated as files on NAS)

- Why use block?
  - Faster (?!) - is that still true?
  - Redirectors and file servers had implementation problems (inode locking, aio) which hindered scaling, especially with virtualization and database
- But what if Network File (with SMB 3) were now as fast as block ...? Would SAN move to more of a niche role sitting behind the cluster fs on high end NAS filers?

SMB3 is Everywhere

- Quoting Dan Shearer*“The costs and risks of having a non-default filesystem for Windo* *are rarely worth small performance advantages (at best.)”*
- A monopoly in operating system market gave us a ubiquitous network file protocol (SMB2/SMB3)
  - Well documented
  - Well tested
  - Frequent interoperability test events
  - Great broadly available functional test cases, network analyzer support
- Protocol broadly adopted by others
- But Linux has MANY advantages to allow us to implement SMB 3 efficiently

Update from SambaXP and NFS Bakeoff

- See the presentations from the SMB3 and NFS tracks at the annual Storage Developer Conference at http://www.snia.org/events/storage-developer/archive
- For NFS see recent and archived presentations from various conferences at:
  - http://www.nfsv4bat.org/Documents/index.html
  - And http://datatracker.ietf.org/wg/nfsv4/ for official ietf standards documents
- For SMB3
  - see http://www.sambaxp.org
  - Microsoft channel 9, msdn and Jose Barreto's blog at blogs.technet.com

NFS: continued progress

Areas address by NFSv4, NFSv4.1 and pNFS:

Security

Uniform namespaces

Statefulness & Sessions

Compound operations

Caching; Directory & File Delegations

Parallelisation; Layouts & pNFS

Client support for NFSv4.1 in 2.6.39 kernel (file layout kernel and Fedora 15), objects 3.0, blocks 3.1 kernel)

NFS

- NFSv4.2 many years in the planning
  - Blocked on approval of RFC3530 bis (NFSv4 document update) but expected this year
    - RFC3530 better defines what happens when no username@domain mapping available (can send numeric uids now). Various clarifications
  - New features: security labels, virtualization features (server side copy, io_advise, sparse files)
- New layout types have been proposed: e.g. flexfiles
  - Flexfiles allows for some very interesting use cases see e.g https://datatracker.ietf.org/doc/draft-bhalevy-nfsv4-flex-files/?include_text=1
  - Allows data servers to not have to be tightly coupled to the pNFS metadata server
- FedFS (Global name space)
- rpcsecgss_v3 (security improvements)

NFS Linux implementation

- Linux NFS server scalability dramatically improved recently
  - Thanks to Jeff Layton among others
- Linux PNFS kernel server implementation is still out of mainline tree but code is available (see the work of Benny Halevy and others http://wiki.linux-nfs.org/wiki/index.php/PNFS_Development_Git_tree)
- Great work on improving stability and performance of NFS client (Trond a others)
- Linux client has minimal NFSv4.2 support (basically a config option to allo negotiating it, with most optional features disabled)

Linux CIFS/SMB2/SMB3 Kernel client

- Update on recent progress

Development activity continues

- Kernel client (cifs.ko)
  - SMB2, 2.1 and 3.0 (and even minimal 3.02) support are in!
  - Current version is 2.03 and is visible via modinfo (and in /proc/fs/cifs/DebugData)
    - In one year we have gone from kernel 3.9 to 3.15-rc5
  - 220 kernel changesets for cifs, a typical year
  - More than 20 developers contributed
  - cifs continues to be one of the more active file systems in kernel
- Samba server also continues to improve its SMB2 and SMB3 support
  - And not just the server … Smbclient (user space ftp like tools) support SMB2

Kernel (including cifs client) improving

- A year ago we had 3.9 “Unicycling Gorilla”

- Now we have 3.15 “Shuffling Zombie Juror”

Features in process

- SMB3 Large i/o and multicredit perf improvements (Pavel)
- Auth cleanup, rewrite to improve gss auth support (Sachin)
- 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 MF symlink support
- 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

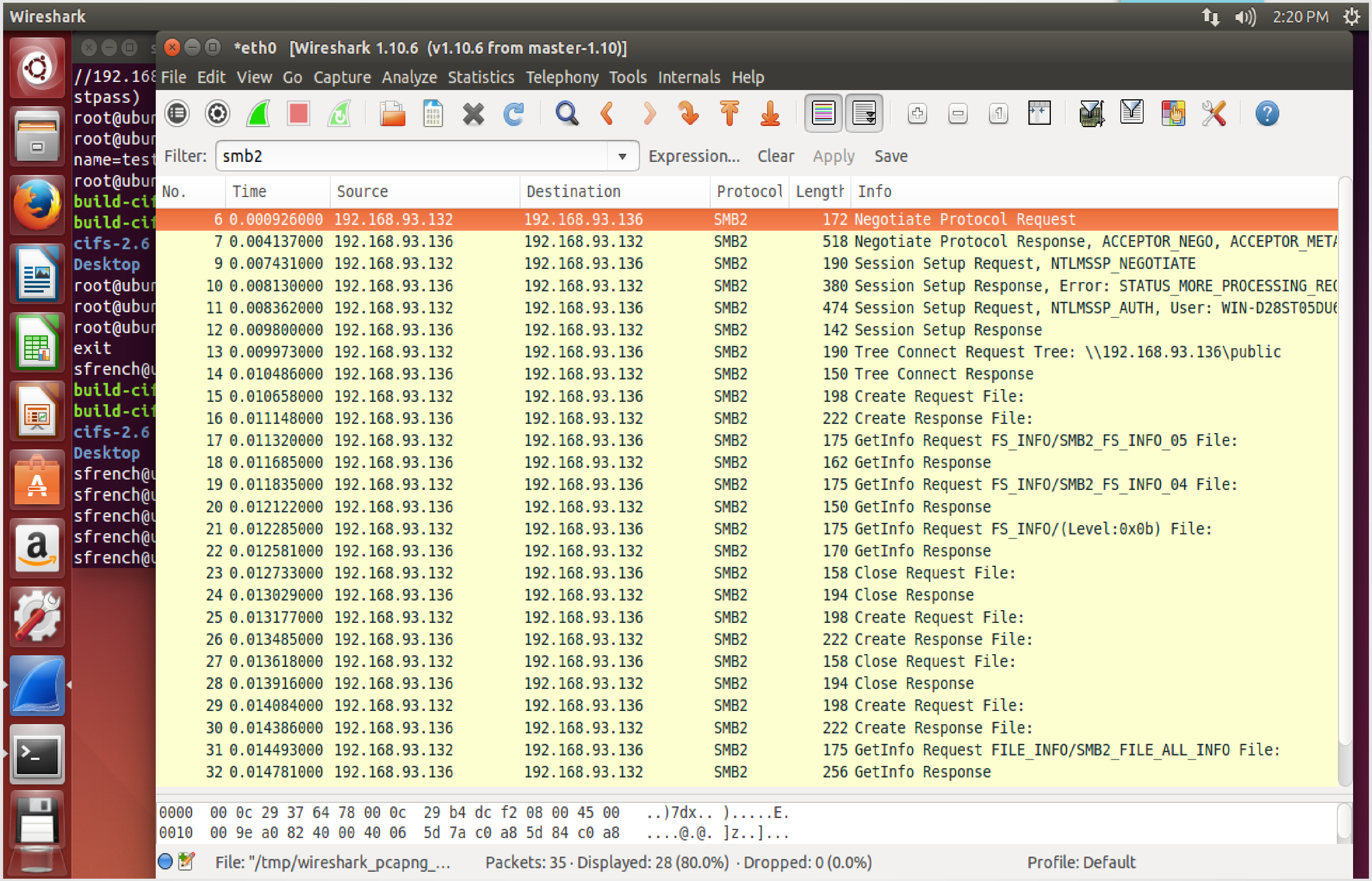
Improvements by release (continued)

- 3.15 17 changes
  - Various minor bug fixes (include aio/write, append, xattr, and also in metadata caching)
- Changes planned for 3.16 (or soon thereafter)
  - Allow multiple mounts to same server with different dialects
  - Authentication session establishment rewrite to improve gssapi support
- 3.17 plan to add higher performance large read/write, SMB2/SMB3 multicredit support

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 6.3.1
  - 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



Wireshark
2:20 PM
*eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

```
//192.168
stpass)
root@ubur
root@ubur
name=test
root@ubur
build-cif
build-cif
cifs-2.6
Desktop
root@ubur
root@ubur
root@ubur
exit
sfrench@u
build-cif
build-cif
cifs-2.6
Desktop
sfrench@u
sfrench@u
sfrench@u
sfrench@u
sfrench@u
```

Filter: smb2

```
No.  Time         Source          Destination     Protocol Length Info
   6 0.000926000 192.168.93.132 192.168.93.136 SMB2      172 Negotiate Protocol Request
   7 0.004137000 192.168.93.136 192.168.93.132 SMB2      518 Negotiate Protocol Response, ACCEPTOR_NEGO, ACCEPTOR_META
   9 0.007431000 192.168.93.132 192.168.93.136 SMB2      190 Session Setup Request, NTLMSSP_NEGOTIATE
  10 0.008130000 192.168.93.136 192.168.93.132 SMB2      380 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQ
  11 0.008362000 192.168.93.132 192.168.93.136 SMB2      474 Session Setup Request, NTLMSSP_AUTH, User: WIN-D28ST05DU6
  12 0.009800000 192.168.93.136 192.168.93.132 SMB2      142 Session Setup Response
  13 0.009973000 192.168.93.132 192.168.93.136 SMB2      190 Tree Connect Request Tree: \\192.168.93.136\public
  14 0.010486000 192.168.93.136 192.168.93.132 SMB2      150 Tree Connect Response
  15 0.010658000 192.168.93.132 192.168.93.136 SMB2      198 Create Request File:
  16 0.011148000 192.168.93.136 192.168.93.132 SMB2      222 Create Response File:
  17 0.011320000 192.168.93.132 192.168.93.136 SMB2      175 GetInfo Request FS_INFO/SMB2_FS_INFO_05 File:
  18 0.011685000 192.168.93.136 192.168.93.132 SMB2      162 GetInfo Response
  19 0.011835000 192.168.93.132 192.168.93.136 SMB2      175 GetInfo Request FS_INFO/SMB2_FS_INFO_04 File:
  20 0.012122000 192.168.93.136 192.168.93.132 SMB2      150 GetInfo Response
  21 0.012285000 192.168.93.132 192.168.93.136 SMB2      175 GetInfo Request FS_INFO/(Level:0x0b) File:
  22 0.012581000 192.168.93.136 192.168.93.132 SMB2      170 GetInfo Response
  23 0.012733000 192.168.93.132 192.168.93.136 SMB2      158 Close Request File:
  24 0.013029000 192.168.93.136 192.168.93.132 SMB2      194 Close Response
  25 0.013177000 192.168.93.132 192.168.93.136 SMB2      198 Create Request File:
  26 0.013485000 192.168.93.136 192.168.93.132 SMB2      222 Create Response File:
  27 0.013618000 192.168.93.132 192.168.93.136 SMB2      158 Close Request File:
  28 0.013916000 192.168.93.136 192.168.93.132 SMB2      194 Close Response
  29 0.014084000 192.168.93.132 192.168.93.136 SMB2      198 Create Request File:
  30 0.014386000 192.168.93.136 192.168.93.132 SMB2      222 Create Response File:
  31 0.014493000 192.168.93.132 192.168.93.136 SMB2      175 GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO File:
  32 0.014781000 192.168.93.136 192.168.93.132 SMB2      256 GetInfo Response
```

```
0000  00 0c 29 37 64 78 00 0c  29 b4 dc f2 08 00 45 00   ..)7dx.. ).....E.
0010  00 9e a0 82 40 00 40 06  5d 7a c0 a8 5d 84 c0 a8   ....@.@. ]z..]...
```

File: "/tmp/wireshark_pcapng_... Packets: 35 · Displayed: 28 (80.0%) · Dropped: 0 (0.0%) Profile: Default

SMB3 Kernel Client Status

- SMB3 support is solid, but lacks many optional features
- Badly needs Unix/Linux extensions for full posix app compatibility on Linux clients
- 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)
  - 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''
  - 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)
    - Large Read/Write support (in progress) and compound ops

SMB3 POSIX Extensions

- Existing FILE_UNIX_INFO_BASIC (from cifs unix extensions, **bold** are those needed in SMB3)
  - __le64 EndOfFile;
  - __le64 NumOfBytes;
  - __le64 LastStatusChange; /*SNIA specs DCE time for the 3 time fields */
  - __le64 LastAccessTime;
  - __le64 LastModificationTime;
  - **__le64 Uid;**
  - **__le64 Gid;**
  - **__le32 Type;**
  - **__le64 DevMajor;**
  - **__le64 DevMinor;**
  - **__le64 UniqueId;**
  - **__le64 Permissions;**
  - **__le64 Nlinks;**

Testing … testing … testing

- One of the goals for this summer is to improve automated testing of cifs.ko
- Functional tests:
  - Xfstest is the standard file system test bucket for Linux
    - Runs over nfs, I created patch to run over 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
- 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

Thank you for your time

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