~~CIFS~~ ~~*SMB2*~~ ~~*SMB2.1*~~ ~~*SMB3*~~ **SMB3.02** And Linux: A Status Update

How do you use it?
What works?
What is Coming?

Steve French
Senior Engineer
SMB3 Architect - IBM Storage

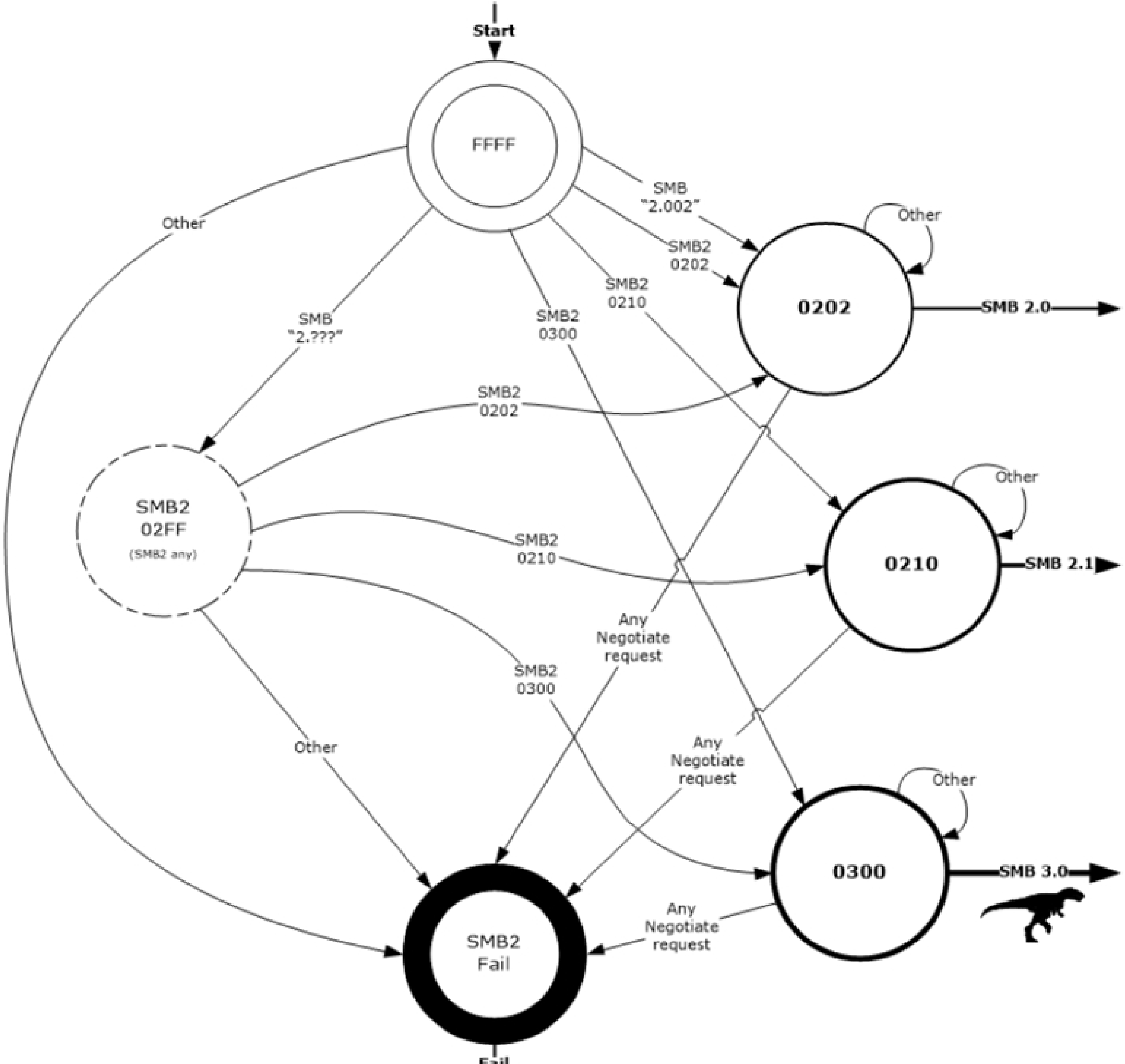
Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of IBM Corporation
- A full list of U.S. trademarks owned by IBM may be found at http://www.ibm.com/legal/copytrade.shtml.
- 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 or sfrench@us.ibm.com)
- Author and maintainer of Linux cifs vfs (for accessing Samba, Windows and various SMB/CIFS based NAS appliances)
- Wrote initial SMB2 kernel client prototype
- Member of the Samba team, coauthor of SNIA CIFS Technical Reference and former SNIA CIFS Working Group chair
- SMB3 Architect: IBM Storage

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)
SMB2 0210
0210
Other
SMB 2.1
Any Negotiate request
SMB2 0300
Other
Any Negotiate request
Other
0300
SMB 3.0
Any Negotiate request
SMB2 Fail
Fail

Why SMB 3 – what does protocol improve on?

- **SMB3 will be important to Linux and our customers. Plays well to Linux/Samba strengths on high end hardware**
  - Cluster friendly, larger I/O sizes, more scalable
- **Addresses requirements of key enterprise workloads and Improves**
  - Availability
    - Enable transparent client recover in the presence of Network or Server Failure
    - Minimize failover time to reduce application stalls, Also allow planned failover
  - Performance
    - Enable clients to aggregate available bandwidth across adapters transparently
    - Continue to increase efficiency on high bandwidth networks
    - Cluster enabled to allow for higher scalability
  - Traffic Reduction
    - Continue improving user perceived latency when working in a WAN environment
- **Key features:**
  - Multichannel
  - SMB over RDMA
  - Scale-Out Awareness
  - Per-share encryption (and security even better in other areas too ...)
  - Persistent Handles and Clustered Client Failover
  - Witness Notification Protocol
  - Directory Leasing and improved metadata caching
  - Branch Cache v2 (content addressable storage)
  - Support for Storage Features (TRIM, block size discovery, T10 etc)
  - Claims Based Access Control

What are our general goals ...

- Local/Remote Transparency
  - Most applications shouldn't notice or care if on remote mount vs. ext4
- Near perfect POSIX semantics to Samba (and those which implement POSIX extensions) and best effort to other non-POSIX NAS filers
- Fast, efficient, full function, secure method for accessing (from Linux) data which lives on Windows servers or other NAS
- As reliable as reasonably possible over bad networks
- Be able to read and set not just file data but also all reasonably important Windows metadata (for backup, archive, gateways and to help server migration)

What are our [SMB3] goals ...

- Focus on SMB2.1 and SMB3, SMB3.02 (SMB2.02 works, but lower priority)
  - Prototype and merge newer extensions faster (now that basic SMB2.1/SMB3 support in, and cifs.ko is MUCH more easily extensible)
- SMB3 faster than CIFS (leverage RDMA, multicredit, multichannel, leasing)
  - SMB3 remote file access near local file access speed (when RDMA)
    - Lot of work to do here (SMB2.1 leasing works though)
- [Good progress] Improve Samba server through cooperative testing
- [Good progress] Continue to cleanup many of the small design and code problems noticed after coding cifs (good progress here)
- [3.11 and later] Allow Higher Data Integrity guarantees, especially through use of durable/persistent handles (handle server failure without data loss)
- [Done] Set better security settings than would be possible with cifs (which supports many older, buggy servers), and take advantage of better signing and [future] per-share encryption
- [Beginning] Testbed for SMB3 Unix Extensions
- [Started] Set better, simpler default mount options than cifs

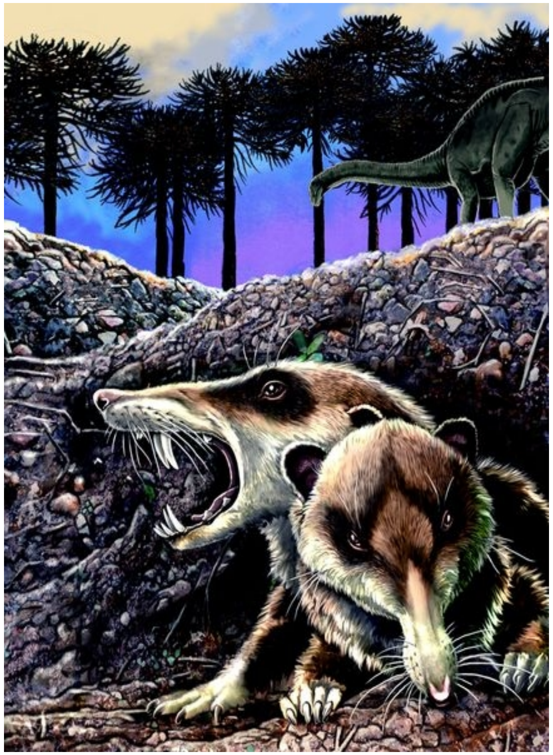
Development on Linux kernel clients and Samba is very active

- Kernel client (cifs.ko)
  - SMB2.1 and 3.0 (and even minimal 3.02) support are in!
  - Current version is 2.01 and is visible via modinfo (and in /proc/fs/cifs/DebugData)
    - In one year we have gone from kernel 3.6-rc5 to 3.11

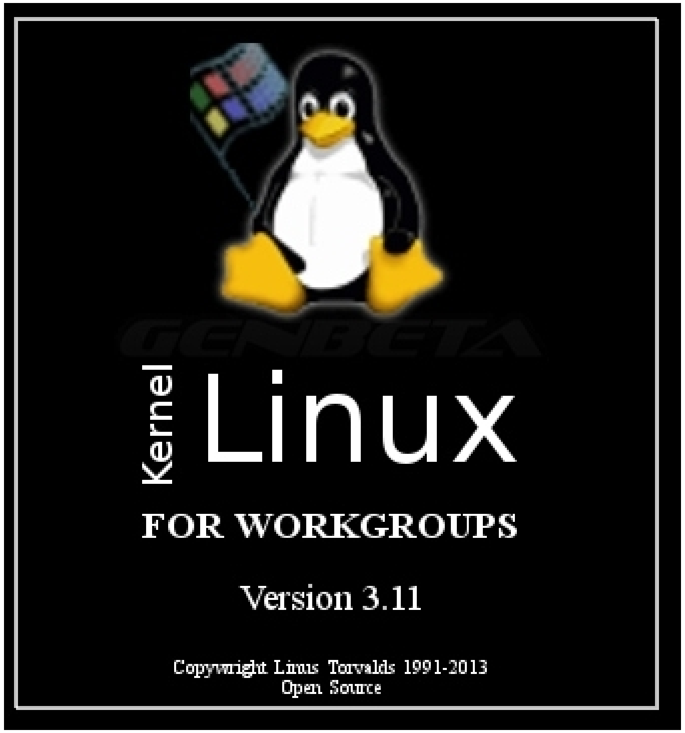
    308 kernel changesets for cifs, a very active year
  - More than 20 developers contributed
  - cifs continues to be one of the more active file systems in kernel
- Samba server also continues to improve its SMB2 and SMB3 support
  - And not just the server … Smbclient (user space ftp like tools) now support SMB2

Kernel (including the cifs client) improving rapidly

- A year ago we had (3.6-rc5) “Saber-toothed Squirrel” (Cronopio dentiacutus)
  - http://news.nationalgeographic.com/news/2011/10/101102-saber-toothed-squirrel-fossils-paleontology-dinosaurs-science/



- Now we have 3.11
  - “Linux for Workgroups” (soon to be “Suicidal Squirrel” for 3.12)
  - Almost 12,000 changesets in 3.11!



Four Features we are working on (actually this week)

- Recovery of pending byte range locks after server failure (we already recover successful locks)
- Full Linux xattr support
  - Empty xattr (name but no value)
  - Case sensitive xattr values
  - Security (SELinux) namespace (and others)
- Copy offload (aka reflink, ask server to copy a file range for us to improve perf)
  - New syscall coming, but cp already works with ioctl (to btrfs, ocfs2)
  - Samba support is in (thanks David)
- SMB3 Unix Extensions prototyping

Improvements by release

- 3.4 58 changes, cifs version 1.78
  - handle “sloppy” mount option
  - **Faster**
    - readahead don't cap ra_pages at the same level as default_backing_dev_info
    - Respect negotiated MaxMpxCount (allows more reqs in flight, if server supports)
- 3.5 42 changes, cifs version 1.78
  - add a deprecation warning to CIFS_IOC_CHECKUMOUNT ioctl
  - remove legacy MultiuserMount option
  - Add “cache=” option and display in /proc/mounts
  - add deprecation warnings to strictcache and forcedirectio mount options
- 3.6 64 changes, cifs version 1.78
  - atomic open improvement added to VFS (and cifs)
- 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

Improvements by release (continued)

- 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
- 3.12-pre 25 changes (so far), cifs version 2.02: **SMB3 support much improved**
  - SMB3 multiuser signing improvements (thank you Shirish!)
  - SMB2/3 symlink support (can follow Windows symlinks)
  - Lease improvements (thank you Pavel!)
  - Pending Byte range lock recovery (planned), debugging improvements (planned)

- 3.13 (planned)
  - Copychunk (refcopy ioctl, and syscall support) (Steve)
  - Full POSIX xattr support for cifs unix extensions (Steve and JRA)
  - Quota support? (Satchin)
  - Larger i/o sizes (including multicredit), compound ops
  - Per-share encryption?
  - Multichannel?
  - SMB3 Unix Extensions?
- 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?

CIFS Performance much improved … But we have work to do: today SMB2.1 is faster than cifs (for kernel client) in only a few cases

- CIFS is FAST! due to some good work by Jeff Layton and RedHat (included in 3.4 kernel)
- Generally SMB2 performance would benefit from three factors
  - Larger i/o sizes
  - credit based flow control (easier to achieve more parallelism)
  - Improved caching model
- But for kernel client today, the only key performance benefit is SMB2.1 leases
  - Cifs currently using larger i/o sizes (especially to Samba)
  - Cifs using fewer requests in some common code paths
    - Cifs is faster for stat (queryinfo), usually one path based request instead of 3 ie open/query/close (need to add compounding support to kernel client for smb2.1)
- SMB3 Performance likely to improve a lot in 3.13 and later kernels

SMB2/SMB3 Kernel Client Status

- Can negotiate all four dialects (test focus is on SMB2.1, SMB3, SMB3.02 only)
- SMB2.1 and SMB3 work reasonably well
  - Basic file/directory operations work
  - Passes most functional tests (not quite as good as cifs to Windows)
  - Slower than cifs to Samba (no Unix Extensions, and using smaller i/o) but should improve a lot with multicredit and compounding
  - Can follow symlinks, can leverage durable handles and file leases, signing works
- Little missing pieces (cifsacl) and some corner case (e.g. rename and delete of open files) need more testing
- SMB3 Unix Extensions prototyping beginning
- Need to take advantage of various optional features when server supports
  - Cluster enablement, persistent handles and Witness protocol (reliability)
  - Directory Leases
  - Per-share encryption
  - Multichannel and RDMA

SMB2/SMB3 Kernel Client Plans

- SMB2.1 no longer considered “experimental” by 3.12
  - Bugfixes and more testing
  - Test feedback welcome
- (SMB2.1 and) SMB3 expected to pass similar set of functional tests (to cifs)
  - Rename and delete of open files, acl support do need to be updated
- Focus on newer dialects not Vista's SMB2.02
- Fewer mount options than cifs (simpler), and more strict defaults
- To move from cifs to smb3 protocol as default we need:
  - SMB3 faster and more reliable than cifs to Windows (and non-Samba NAS)
  - SMB3 Unix Extensions defined, implemented in Samba and client

Configuration hints for kernel client

- For cifs rsize defaults to 60K (Windows) / 1M (Samba) with max 127K (Windows) / 16 M (Samba)
- Wsize is similar except 64K for Windows
- Smb2 we default to 64K rsize and wsize
- For cifs the maximum number of simultaneous requests is controlled by: cifs_max_pending (32K) and (on the server in smb.conf) max mux (which can be raised well above its default of 50 if kernel 3.4 or later cifs mounts). For smb2 dynamic crediting is used instead so this is only for cifs
- Cache= mount parameter (cache=loose faster than the default of strict in a few cases)
- For sequential write or read (without reuse) mount option cache=none can be faster
- If packet signing is needed to be enabled, Smb3 signing should be faster
- Generally cifs is faster for current cifs.ko than smb2/smb3, and faster still if unix extensions enabled (choose smb2.1 or smb3 with “vers=2.1” or “vers=3.0” on mount

Configuration Hints (continued)

- Cifs.ko is great for testing
  - “nosharesock” can be useful to separate each user's connection on different socket. Can simulate multiple clients from one physical cliet
  - If unix extension bug with a server can mount without cifs unix with “nounix” on mount. Also useful for testing more “windows-like” behavior
  - Can choose different dialects with “nosharesock” (cifs) on /mnt1 then mount “nosharesock,vers=2.1” on /mnt2 then mount “nosharesock,vers=3.0” on /mnt3 etc. and negotiate different dialects from one physical client even to one server
  - Lots of debug information, stats available (see pseudo files in /proc/fs/cifs). CONFIG_CIFS_STATS2 build option allows even more to be displayed

Not For Windows Only: “SMB3 Unix Extensions”

- Now that we know more, what do we need from SMB3 Unix Extensions?
  - Posix pathnames
  - Case sensitive path name matching
  - Posix delete (unlink) and rename behavior
  - Posix create/mkdir (small additional create/open context)
  - Minor extensions for stat and statfs
    - Not clear whether usual workarounds for mode and ownership is acceptable
  - Posix (advisory) byte range locking
  - Xattr improvements
    - Create empty xattrs (value length is zero)
    - Case sensitive xattr names
    - Other namespaces other than user. (security, SELinux in particular)
    - (perhaps) xattrs on symlinks and special files
  - Nice to have:
    - Symlinks (non-admin symlinks can be emulated using “Minshall-French” client side symlinks)
    - Ability to create/query Unix special files
    - “Posix ACLs”

- List is manageable size

- Addition of extensible create contexts will help make extensions even smaller

Optional POSIX SMB3 Features

- These were the POSIX flags for CIFS (ones in bold are considered, bold/underlined most important/required)
  - **CIFS_UNIX_FCNTL_CAP** 0x00000001 /* support for fcntl locks */
  - **CIFS_UNIX_POSIX_ACL_CAP** 0x00000002 /* support getfacl/setfacl */
  - **CIFS_UNIX_XATTR_CAP** 0x00000004 /* support new namespace */
  - **CIFS_UNIX_EXTATTR_CAP** 0x00000008 /* support chattr/chflag */
  - **CIFS_UNIX_POSIX_PATHNAMES_CAP** 0x00000010 /* Allow POSIX path chars */
  - **CIFS_UNIX_POSIX_PATH_OPS_CAP** 0x00000020 /* Allow new POSIX path based calls including posix open and posix unlink */
  - CIFS_UNIX_LARGE_READ_CAP 0x00000040 /* support reads > 128K */
  - CIFS_UNIX_LARGE_WRITE_CAP 0x00000080
  - **CIFS_UNIX_TRANSPORT_ENCRYPTION_CAP** 0x00000100
  - CIFS_UNIX_TRANSPORT_ENCRYPTION_MANDATORY_CAP 0x00000200
  - CIFS_UNIX_PROXY_CAP 0x00000400 /* Proxy cap: 0xACE ioctl and QFS PROXY call */
- SMB3 Create context being considered for these to allow client to request among the three or more desired POSIX features, along with an infolevel to allow the capabilities to be queried on a volume or share

Thank you for your time!!