Linux Kernel Clients: A Year in Review
"Sheep on Meth"
to
"Flesh-Eating Bats with Fangs"

Steve French
IBM – Linux Technology Center

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 CIFS Technical Reference and former SNIA CIFS Working Group chair
- Architect: Filesystems/NFS/Samba for IBM LTC

- Last May Kernel version was 2.6.34:
- “Sheep on Meth”
  - 2.6.34 came out May 16th, 2010

- Now we have 2.6.39-rc6
- “Flesh-eating Bats with Fangs” (name since 2.6.36-rc8)

Development on Linux kernel clients is very active

- 355 kernel changesets for cifs since 2.6.34 (most active year ever)
- And … improvements to related tools have accelerated dramatically: e.g. over a few months last year 100+ changesets in mount helper and related utilities were merged
- For most of 2010 smb2 development accelerated …
  - A month after SambaXP last year, public git tree for smb2 development created on kernel.org
  - SMB2 module now moved from distinct smb2.ko prototype to optional experimental part of cifs.ko.
  - SMB2 has few dialects, no redundant commands, no buggy backlevel servers to workaround (yet), and name handling much simpler (UCS-16 only), and 32 bit status errors only)
- SMB2 development should accelerate again this summer

Just in case you forgot the goals ...

- Local/Remote Transparency
  - Most applications shouldn't notice or care if on remote mount vs. ext4
- Near perfect POSIX semantics to Samba servers (and those which implement POSIX extensions) and best effort semantics to Windows and other NAS filers
- Fast, efficient, full function gateway for accessing data from Linux which lives on Windows & NAS
- As reliable as reasonably possible over bad networks
- But what about SMB2? The protocol offers many improvements which will benefit Linux:
  - Add scalability that cifs can not handle (e.g. increased number of open files, more inodes)
  - Higher performance on fast networks:
    - very large i/o, flow control (credits)
  - And also slower networks
    - Better caching
    - Better operation compounding
  - Offer stricter data integrity guarantees than cifs can offer

The Team (More than 30 developers contributed)

- **CIFS (both Jeff and Steve were in top 25 Linux fs contributors again this year). Among the contributors to cifs this year were:**
  - Jeff Layton
  - Steve French
  - Shirish Pargaonkar, Pavel, Suresh
  - And many more
- **SMB2**
  - Steve French
  - Pavel
  - Shirish Pargaonkar

CIFS
Celebrating 8 years in the Linux Kernel

Busy year

- Now cifs.ko version 1.71 (Linux kernel 2.6.38-rc6)
- A year ago was CIFS version 1.62 (Linux kernel 2.6.33)
- Two years earlier huge amounts of Microsoft WSPP documentation released

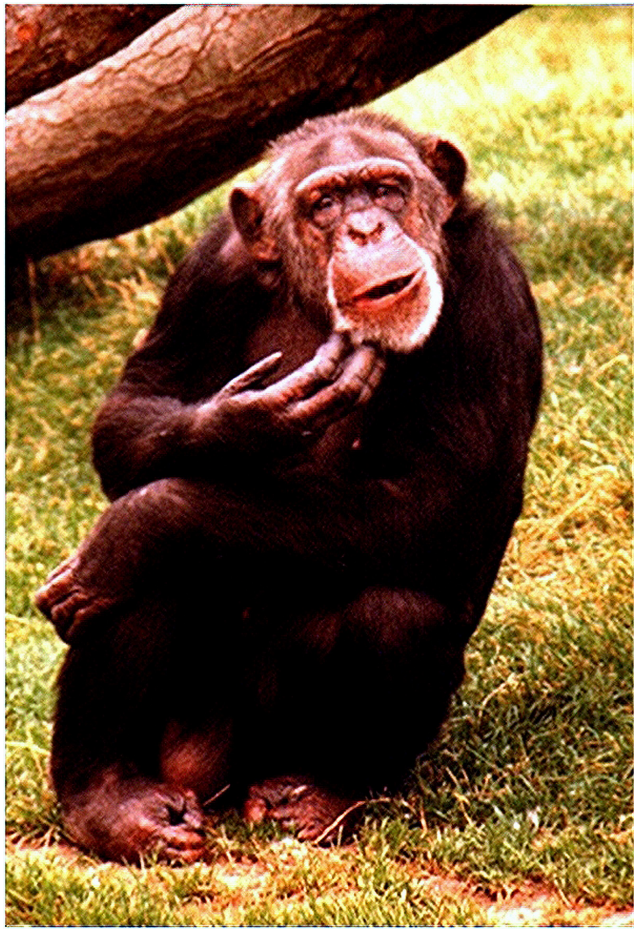
CIFS Progress

- Bugs in bugzilla (many which are now unreproducible/invalid or feature requests) against cifs and related tools:
  - kernel.org bugzilla 8 bugs open
  - samba.org bugzilla 50 bugs open
- Stability improvements:
  - "Strict cache" mode now available (write to cache only in exclusive oplock, read from cache when level II oplock or above) and cache invalidation improvements (Pavel)
  - Multiple potential buffer overruns fixed
  - DFS error handling and reconnection improvements
  - Stack space improvements, cleanups (including moving to kernel crypto API)
- New Features:
  - Get and Set native cifs acl xattr (Shirish)
  - UID mapping (upcall to map owner information from CIFS ACL SIDs to POSIX uid/gid)
  - "Raw NTLMSSP" (auth) support fixes
  - Additional /proc/fs/cifs/DebugData displayed
  - Maximum UserName and ShareName increased (to 256 and 80) to better match server limits

CIFS MultiUser Support and Improved Authentication

- NFS sends (at least) primitive Unix credentials via SunRPC but cifs uses default credentials (supplied at mount time) for all users
- CIFS had a “MultiUserMount” feature which was awkward to use
- Jeff Layton added (and I merged into mainline in 2.6.37) a much improved MultiUser Support for krb5 CIFS mounts (see his talk from SambaXP last year)
- (Following 5 slides courtesy of Jeff Layton)

What happens on file creation?



- File was created on server using mount credentials
- CIFS attempts to enforce permissions on client
- That can't fix ownership
- File is created but later ops fail!

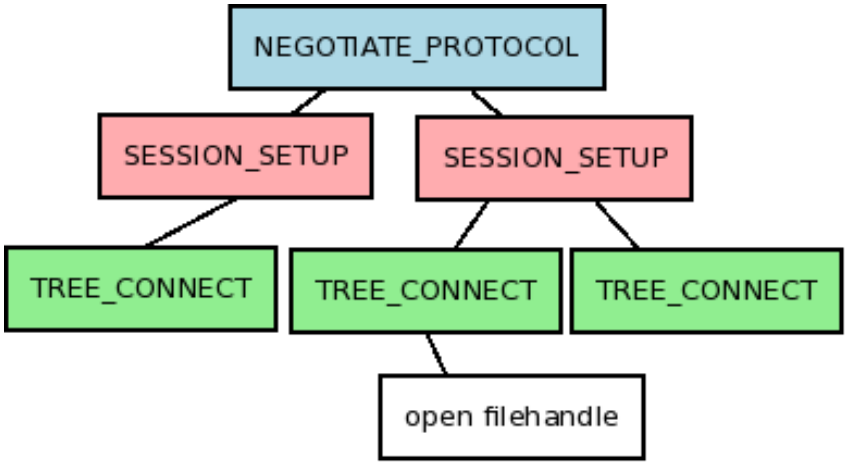
Why is it this way?

- CIFS protocol is **session-based**
- Credentials are handled per-session
- Linux CIFS only has single session per mount
- **Shared Credentials!**

The solution...

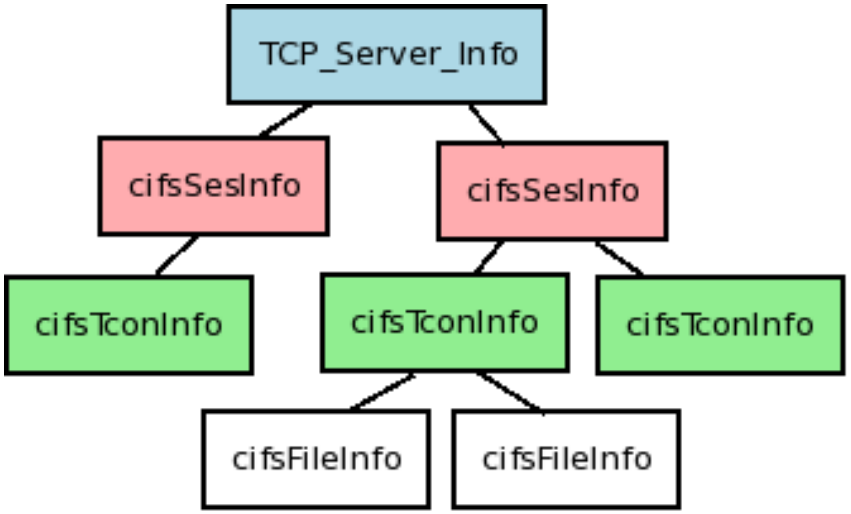
- Each user should use their own credentials
- Have multiple sessions per mount
- Establish sessions on an as-needed basis
- Let the server handle permissions
- **Goal:** Easy as NFS

Protocol hierarchy



- The CIFS protocol has a hierarchy of sorts
- NEGOTIATE
- SESSION_SETUP
- TREE_CONNECT
- Open filehandles
- Other path-based ops

Basic Object Hierarchy



- TCP_Server_Info (socket)
  - per socket
  - NEGOTIATE
- cifsSesInfo (credential)
  - SESSION_SETUP
- cifsTconInfo (share)
  - TREE_CONNECT
- cifsFileInfo (file)
  - open filehandles

CIFS User Space Utilities

- Split into new git tree (cifs-utils) last year to ease maintainability (big success – lots of fixes)
  - Fixes to cifs.upcall (for DNS resolution and kerberos authentication support, getting krb5 tickets and handling spnego)
  - Fixes to mount.cifs (mount helper)
    - Much easier to build and more stable
    - Mounts by less privileged users now safer
  - Since SambaXP have mount.smb2 helper (sharing code with mount.cifs.c) to handle differences between cifs/smb2 (smb2 also has fewer mount options)

FS Cache

- Oplock allows client caching. Linux has a page cache mechanism which is heavily utilized by cifs
- FS Cache allows cifs to use Dave Howell's file system caching mechanism to cache files to disk.
- Especially useful on slow networks or very heavily loaded servers, where reading from disk is faster than reading from the server

Better handling of request timeout/reconnect and slow responses

- Problem: some servers have highly variable response time on writes and can take minutes to complete an SMBwrite under some conditions – yet the traditional solution: dropping the session and reconnecting is not ideal if server is still able to respond to other requests
- SMBecho now used to poll server when no response received from any request in interval (and a request would otherwise timeout) – allowing us to give the server longer to respond if it can still process echo
- Allows us to better continue to keep session up if server still responding to requests
- Ctl-C of process now causes use to send cancel for slow request (on server0

Networking and Interface Selection

- DNS resolver now split out from cifs in helper routine
  - Used also now by nfs
  - Will be used by smb2 as well
- CIFS can now select which network interface (e.g. wireless vs. wired) that a cifs mount (tcp connection request) is sent over (Ben Greear)
  - Merged into cifs-2.6.git (in 2.6.37)
  - Similar patch in smb2
- Cifs mounts can use network namespaces (work better in containers)

New Symlink Support (Metze)

- Especially for servers which don't support Unix Extensions, need alternatives for creating symlinks
  - SFU/SUA style is ok
  - On recent Windows servers SMB2 style symlinks may also be possible
- Now have patches to support MacOS style symlinks (in 2.6.37)
  - Allow creating/reading symlinks to servers, like Windows, which don't support the Unix Extensions
  - This type of symlink is not typically read by the server (e.g. Windows) so is relatively safer to create than those which can be followed by server side applications (since the client can't follow these to server paths outside of the share)

Authentication and Security Improvements

- Caching uid information using the Kernel Credential Keyring
  - Another Google Summer of Code project
  - User space improvements merged
  - Waiting on fixes to kernel patches (possible for 2.6.37)
- Shirish's auth improvements
  - With these patches (raw) NTLMv2 authentication now fixed to Windows 7 and Windows Vista
    - More recent Windows is pickier than Windows XP and Windows 2003 about format of NTLMv2
  - Fixes NTLMv2 in NTLMSSP as well
  - CIFS now uses kernel crypto routines (e.g. for MD4 and MD5) instead of cifs specific implementation, reducing code/module size
  - Lanman auth e.g. to os/2 (regression from ntlmssp fix sideeffect) now fixed.
  - “Can now get a file's native cifs acl from user space (system.cifs_acl). If a DACL has entries for ACEs for SID Everyone and Authenticated Users, factor in mask in respective entries during calculation of permissions for all three, user, group, and other.”

Performance improvements

- Actimeo (inode metadata revalidate interval) now tunable (default 1 second) – changing default helpful for slow networks or for stricter timestamp guarantees
- VFS performance improvements in dcache locking
  - Cifs implementation to take advantage of this merged, should help perf
- CIFS locking was very granular already but final “big kernel lock” usage was removed
- Handle based calls on query (stat) not always being used. When we have appropriate file handle available use more efficient handle based calls (faster)
- CIFS Async Write patches under review, possible merged for 2.6.40

Near Term CIFS Futures

- (partial list, more welcome)
  - SMB2 merge
  - Code cleanup, moving some experimental future code (Jeff/Steve)
  - Nfsd over cifs fixes (partially worked already, Shirish debugging)
  - Encrypted tcon support (optional POSIX Extension which Samba server already supports) ie per-share network encryption
  - Documentation update
    - New man page for non-mount cifs issues
  - Async write (Jeff)
  - Improve default security: NTLMv2 will become the authentication default (moving from ntlm) starting in 2.6.41

SMB2
Network File System of the Future

History

- Prototype begun about 3 years ago then rewritten a year later
- Found various spec problems and bugs at MS test events
- Work restarted 1/2010 with Jeremy Bongio helping (improved read support)
- Pavel in Google Summer last summer – fast async write.
- Tested at 2009 and 2010 Microsoft plugfests, and SNIA SDC 2010
- Tested at 2010 and 2011 Connectathon (many file operations working, added hardlinks and rename and jra fixed some server bugs in those areas too)
- Cthon 2011 decision to merge with common code into cifs.ko (from distinct smb2.ko) big rewrite needed.
- On server side: Samba server 3.6-pre now SMB2 'feature complete' (including addition of some perf features, sendfile e.g.) for original smb2 dialect (not windows 7 newer smb2.1 dialect)

New Design

- SMB2 prototype was in distinct git tree (smb2.git on kernel.org) and built distinct kernel binary smb2.ko (and created pseudofiles in /proc/fs/smb2)
- At SDC 2010 a “cifs_common” (kernel library common to smb2 and cifs) prototyped - reduced smb2 code size about 5% (e.g. connection establishment) was expected to grow to 20%+ common code
- To address comments from Jeff Layton and others at Connectathon 2011 change from two binaries (smb2.ko and cifs.ko) to one (cifs.ko)
  - Smb2 (mostly) in distinct C files in fs/cifs – only built when experimental SMB2 support explicitly requested in build. “vers=smb2” on mount not new fstype (-t smb2)
  - Good: increases common code dramatically. Makes it easier to review for those familiar with cifs. Longer term maintenance easier
  - Bad: requires lots of rework, many trivial changes, much higher risk of smb2 changes breaking cifs, cifs global structures grow slightly to accommodate greater smb2 limits

Preliminary Performance Results

- Generally SMB2 benefits from three factors
  - Larger i/o sizes
  - credit based flow control (easier to achieve more parallelism)
  - Improved caching model
- email from Jeremy Bongio (February)
  - To server on local vm (cat remote-file > /dev/null) current Linux kernel clients to Samba server
  - SMB2 readpages about 20% faster already for medium to large file read (should be relatively even faster over physical network)
  - smb2
    - 166K 0.0342 sec
    - 200000K 11.012 sec
  - CIFS
    - 166K 0.0476 sec
    - 200000K 13.230 sec
- Pavel's async write code
  - For writing medium/large files to Linux/Samba server, performance ~30%better than NFS (Linux to Linux) and more dramatic compared to cifs

Performance results continued

- (Quoting Pavel's post from September 2010)
- SMB2 can be faster than NFS and CIFS on Linux. I got the following results after 10^5 writing each of 4096 bytes of data.
-
-
- SMB2 : ~0m 29s
- NFS : ~0m 40s
- CIFS : ~1m 28s
-
- The test environment:
- Client - Ubuntu 9.10 with 2.6.33 kernel on VirtualBox.
- Server - Fedora 12 with 2.6.32 kernel, Samba 4.0.0alpha11.
-
- Machines were connected through bridge.
-
- test is rather synthetic but it shows at least the one side of SMB2 advantages.

Performance Results (Cont) – SMB2 twice as fast

- At SDC 2010 to jra's Samba 3.6-pre server mount from current Linux smb2 and cifs kernel clients
-
- time dd if=/dev/zero of=/smb2mnt/file1 bs=1M count=100
- 100+0 records in
- 100+0 records out
- 104857600 bytes (105 MB) copied, 1.31032 s, 80.0 MB/s
-
- time dd if=/dev/zero of=/cifsmnt/file2 bs=1M count=100
- 100+0 records in
- 100+0 records out
- 104857600 bytes (105 MB) copied, 2.64377 s, 39.7 MB/s
-
-

What is SMB2?

- Default filesystem protocol for Windows (since 2008).
  - Samba must support it well as a server
  - Current server implementation functional, but layered over SMB/CIFS rather than optimized, missing full implementation of some new protocol features
- Rapidly becoming most common network file system protocol
  - To get data most efficiently from other systems (not just Windows but most NAS), we must have great support for SMB2 kernel client (not just server) for Linux as well
  - To make Linux (and other Unix) even better (to Samba e.g.) adding minor extensions to Linux client/server
  - SMB2 better than CIFS for use by Linux for getting files from NAS
- What is it?
  - SMB2 is not simply a new dialect of SMB/CIFS
  - Protocol features continue to improve (already have an SMB2.1)
  - SMB2 enables new performance,security and reliability features
    - larger i/o sizes, caching improvements, credits, compounding, async operations, scalability greatly increased etc.
    - Features have synergy with Samba/Linux's strengths

SMB2 Implementation Design Goals

- Faster than CIFS
- Improve Samba server through cooperative testing
- Cleanup many of the small design and code problems noticed after coding cifs
- Experiment with features that are too risky to do in stable cifs
- Allow Higher Data Integrity guarantees through use of the new SMB2 protocol features in this area
- Set default security settings to higher level than would be possible with cifs (which supports many older, buggy servers)
- Testbed for Unix Extensions

SMB2 Kernel Client Priorities & Plans

- **Basic compatibility:** Enough function so Linux kernel client passes at least as many functional tests as Linux CIFS client (to Windows)
- **Basic compatibility:** Samba server passes basic functional tests (from Linux or Windows SMB2 clients)
- **Optimizations for Samba:**
  - Define "POSIX SMB2 Protocol Extensions"
    - Reserve command codes, information levels (with Microsoft)
    - Get agreement with Samba SMB2 server developers on extensions
    - Extend VFS and/or protocol in order to implement ALL local fs interfaces over SMB2: Maximize app compatibility
  - Ensure SMB2 is faster than CIFS to Samba/Linux
- **Community acceptance:**
  - Expand community
  - redesign problematic, hard to maintain areas in CIFS kernel client;
  - add SMB2 kernel client to linux-next then mainline (experimental feature within cifs.ko)

SMB2 Current Status

- Basic support (experimental) target 2.6.40 or 2.6.41
  - Mount parsing and kconfig for smb2 in linux-next
  - Error handling, stats, protocol ops reviewed expected to be in linux-next within a few weeks
  - Whether to share transport structures (including mids) with cifs under discussion
  - Rework of inode and file ops next (decide whether to use existing implementation of prototype, or share more code with cifs vfs handling)

Unix Extensions Redux: SMB2

Purpose of Unix Extensions

- Provide exact file semantics from Linux, Solaris, MacOS and other POSIX clients
  - Real world compatibility requires few operations outside of the posix spec (getattr for example)
- A network file system must provide transparency – look like a local file system, and not break common applications
- Compensations for non-POSIX style operations must not harm data integrity, and have only limited impact on performance
- Where reasonable Unix Extensions should be designed so the could be implementable by Windows and others non-Unix/Linux servers
- Work in progress (with MS) to do similar extensions for SMB2 (fortunately can be smaller in scope for SMB2 than was needed for CIFS)

- http://www.unixsmb2.org/docs.php For information about these extensions
-

Thank you for your time!!