~~CIFS~~ ~~*SMB2*~~ **SMB3** And Linux:
A Status Update
What works? What is coming soon?

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

Why SMB 3 – what does protocol improve on?

- **Addresses requirements of 4 key enterprise workloads and Improves**
  - Availability
    - Enable transparent client recover in the presence of Network Failure
    - Server Failure
    - Minimize failover time to reduce application stalls
    - Also allow planned, application initiated 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
  - Witness Notification Protocol
  - Clustered Client Failover
  - 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

Why SMB 3?

- **SMB 3 will be critical to Linux, and important to our customers**
  - Goes beyond traditional CIFS/SMB2 strengths (already the most popular file protocol) and addresses key enterprise workload requirements for virtualization, HPC, Database and Web
    - Adds significant performance and functional improvements. In many cases SMB 3 will be as fast as raw access to network block devices
- **Key SMB 3 features play well to Linux/Samba strengths on high end hardware**
  - Cluster friendly (cross-cluster shares, failover “witness protocol”, better HA features ...)
  - Larger i/o sizes
  - More parallel access, clients are more scalable
  - Metadata caching
  - Optional RDMA, and T10 enablement

What are our 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, 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 ...

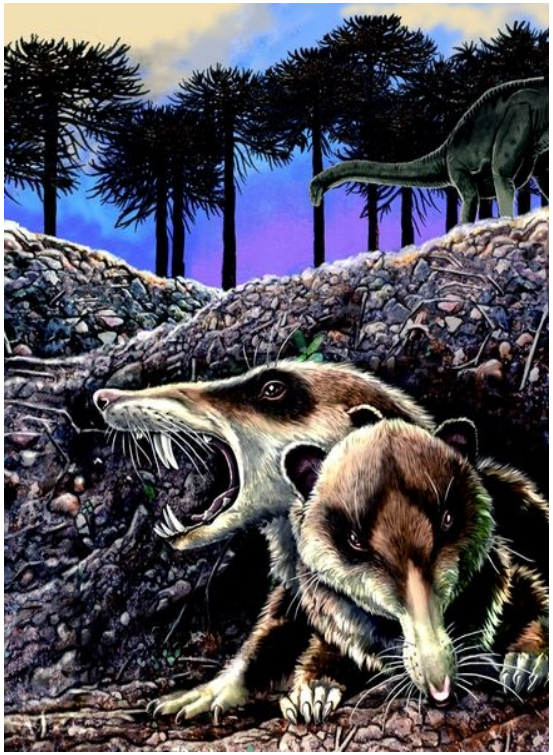
- Focus on SMB2.1 and SMB3 (SMB2.02 works, but lower priority)
  - Prototype and merge newer extensions faster (now that basic SMB2.1/SMB3 support in, and cifs.ko is MUCH more easily extensible)
- SMB3 faster than CIFS (leverage RDMA, multicredit, multichannel, leasing)
  - SMB3 remote file access near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

Development on Linux kernel clients and Samba is very active

- Kernel client (cifs.ko)
  - SMB2.1 support is in!
  - Current version is 2.0 (last year 1.78). Is visible via modinfo
    - In one year we have gone from kernel 3.4-rc6 to 3.9
  - 315 kernel changesets for cifs since 3.4-rc6, a very active year
  - More than 20 developers contributed
  - cifs continues to be one of the more active file systems
  - Development expected to grow more when SMB3 support improved, SMB2.1/SMB3 broadly matches cifs performance (which is quite good) and Unix Extensions for SMB2/SMB3 added

Kernel (including the cifs client) improving rapidly

- A year ago we had (3.4-rc6) “Saber-toothed Squirrel” (Cronopio dentiacutus)
  - http://news.nationalgeographic.com/news/2011/10/101102-saber-toothed-squirrel-fossils-paleontology-dinosaurs-science/



- Now we have 3.9 “Unicycling Gorilla”

Two Key Features we are working on (actually this week)

- Full Linux xattr support
  - Empty xattr (name but no value)
  - Case sensitive xattr values
  - Security (SELinux) namespace (and others)
- Copy offload (reflink, offload to server copy range)
  - New syscall coming, but cp already works with ioctl (to btrfs, ocfs2)
  - Samba support is in (thanks David)

Linux Kernel Client improvements since last SambaXP

- Performance improvements
  - Faster vectored AIO (3.4) and forcedirectio faster (3.4 and 3.5)
  - Readahead to Samba faster (3.4)
  - Maximum number of requests to a server increased: 256 to 32K (3.4)
    - Most servers default to 50 simultaneous requests. requires update to smb.conf (Samba) or Windows server registry
  - SMB2.1 leases (caching) added
  - Added mount option Cache=
- CIFS is VERY fast now on Linux to Linux (not just to Windows)
  - Great way to transfer large files
  - Not just the client has improved, Samba server performance also faster now due to improved dispatch of reads and writes as well
- Other Improvements
  - SMB2.1 support added!
    - (followed by SMB2.02 and minimal SMB3 support)
  - “strictcache” mount option
  - Cifs.ko smaller footprint
- Security improvements
  - Ntlmv2 auth default (not NTLM)
  - Smb2.1 signing (better than cifs signing) implemented
  - Dfs security negotiation fixes


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 (and 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)
- 3.10pre 13 changes, cifs version 2.1
  - cifs module size reduced
  - Smb3 signing fixed (planned) (Shirish)
- 3.11 (planned)
  - Copychunk (refcopy ioctl, and syscall support) (Steve)
  - Full POSIX xattr support for cifs unix extensions (Steve and JRA)
  - Quota support (Satchin)
  - Auth and security settings config overhaul (Jeff and/or Steve)
  - Larger i/o sizes (including multicredit), compound ops (could be 3.12)
- 3.12 or 3.13 (tentative)
  - Persistent and durable handle support (Pavel)

cifs read continues to get faster

- cifs_iovec_read now collects/issues (larger) asynchronous reads. Primarily of use when a share is mounted with forcedirectio, or strictcache and the client doesn't have an oplock for the file being (in 3.5. From Jeff Layton)
- Big increase read performance there
- Test results from my low end KVM test rig to samba. Running simply:
  - $ dd if=./ddtest.out of=/dev/null bs=1M
- Results:
  - Unpatched 3.4-rc2 kernel -- rsize is always capped at 16k here:
    - 1073741824 bytes (1.1 GB) copied, 97.6394 s, 11.0 MB/s
  - Patched 3.4-rc2 kernel – rsize=1M:
    - 1073741824 bytes (1.1 GB) copied, 9.89869 s, 108 MB/s
  - Patched 3.4-rc2 – rsize=61440:
    - 1073741824 bytes (1.1 GB) copied, 13.4146 s, 80.0 MB/s

Continuing perf improvements from early 2012 (thank you Jeff Layton)

- Normal buffered large file read got MUCH faster on 3.2
  - 1GB file copy from server to /dev/null with dd (on slow kvm test system, would be more dramatic improvement over fast network to fast server)
  - prepatch, with 16k rsize:
    - 1073741824 bytes (1.1 GB) copied, 47.2119 s, 22.7 MB/s
  - postpatch, with 1M rsize:
    - 1073741824 bytes (1.1 GB) copied, 11.1602 s, 96.2 MB/s
  - postpatch, with 60k rsize:
    - 1073741824 bytes (1.1 GB) copied, 12.5183 s, 85.8 MB/s
- Readahead had been often limited to 128K due to kernel bdi defaults – but when mounted to Samba which supports MUCH larger reads this limits perf gain of readahead
  - Patch to improve this in 3.4 kernel (slow kvm guest mount to local Samba on host – see below)
  - $ dd if=./ddtest.out of=/dev/null bs=1M count=1024
  - Prepatch:
    - 1048576000 bytes (1.0 GB) copied, 28.1979 s, 37.2 MB/s
  - Postpatch:
    - 1048576000 bytes (1.0 GB) copied, 11.92 s, 88.0 MB/s

But we have work to do: today SMB2.1 is faster than cifs (for kernel client) in only a few cases (so far)

- Generally SMB2 performance would benefits 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)
- This is likely to improve a lot in 3.11 and on kernels

SMB2/SMB3 Kernel Client Status (Great work by Pavel fixing up SMB2/SMB2.1)

- Can negotiate all three dialects (test focus is on SMB2.1)
- SMB2.1 mounts work but still experimental
  - 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
- Little missing pieces (cifsacl) and some corner case (e.g. rename and delete of open files) need more testing
- No Unix Extensions defined yet
- SMB3 implementation very minimal
  - Smb3 signing needs to be added ASAP
- Need to take advantage of various optional features when server supports
  - Cluster enablement
  - Directory Leases
  - Per-share encryption
  - Durable/Persistent handles, Witness protocol (reliability)
    - Even if server support, persistent handles probably need to be mount option too?

SMB2/SMB3 Kernel Client Plans

- SMB2.1 no longer considered “experimental” by 3.11 or 3.12
  - Bugfixes and more testing
  - Test feedback welcome
- (SMB2.1 and) SMB3 expected to pass similar set of functional tests (to what cifs can do to Windows) by summer MS test event
  - Rename and delete of open files, acl support, symlinks need to be updated
- Current plan is only focus on 2.1 and SMB 3, even if SMB2.02 works
- 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
  - The latter will take longer. Not clear if we could (or should) pick different default to Samba and Windows.

What about NFS ...?

Not For Windows Only: “SMB3 Unix Extensions”

- Now that we know more, what do we need from SMB3 Unix Extensions?
  - Posix pathnames
  - Posix delete (unlink) 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

Historic shift occuring – What do we in Linux have to do?

- While we implement SMB 3 et al: Focus on our strengths (e.g. clustering and RDMA)
  - Samba has had GREAT clustering support for years (Tridge's ctdb was brilliant!)
  - Our RDMA stack is good, with broad driver support
  - Our server side copy (copychunk and T10 block copy) should be able to excel
  - We can be faster than Windows and non-Linux NAS
- Ensure SMB 3 Unix Extensions include all key Linux needs (POSIX locking, SELinux …)
  - Intend Unix Extensions to be small. Will be documented (see Chris Hertel at RedHat)
- Optimize BOTH ends of our Linux workloads (and not just for Apache, KVM)
  - Ensure I/O sizes, flow control are optimized on both SMB2.2 kernel client and Samba
  - Fix some VFS and client i/o bottlenecks so KVM can scale better on SMB2.2 mounts
- As Microsoft has shown with Hyper-V in Windows 8 – fixing migration of running images (KVM in our case) over SMB 3 is very powerful feature (may require VFS changes)
- Merge RichACLs so NFSv4 (4.1 actually) and Samba can get consistent access control
- Continue to take advantage of GREAT detailed protocol documentation on various loosely related protocols, and implement services, extensions and libraries to interoperate with branch cache (v2), Witness protocol (HA), Claims Based Access control (Hits Kerberos padata field, and RichACLs), and some of the CIM-like storage management features

Thank you for your time!!

Backing Charts – Older Material which may be of interest to some

Although network API closer to Windows than POSIX, CIFS and SMB2 not really Windows specific

- Mac, Solaris, Linux and most other operating systems have kernel clients. Solaris and Mac even use CIFS ACLs in-kernel. CIFS/SMB2 default for some Unix and all Windows.
- “Unix Extensions” developed by SCO, extended by HP and then Linux and Mac. Improve most “posix vs. windows” issues such as retrieving the Linux 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 the same approach as “Services for Unix”). Mount from Linux to Windows just works for most applications. NB: NFSv3 is not completely POSIX friendly
- Microsoft made SMB2 slightly more “unix friendly” so extensions for SMB2/3 will be smaller

SAMBA

opening windows to a wider world

CIFS/SMB2 is not just a file protocol – it addresses complete picture

- When comparing SMB2 to NFS or to another cluster file system remember the big picture
  - It is not just about files, but how to find them and manage them
  - SMB2 easily wins over alternatives when looking at easily managing, securing data
  - In IBM we run Samba/GPFS in GSA (our server farm). Easy and clients work out of the box
- SMB2 (and SMB 3) are integrated with
  - “DFS” (global namespace)
  - Advanced security features (not just Kerberos authentication)
  - A rich management model (DCE-RPC, LDAP and CIM based) which uses the same access control and authentication. Samba can handle thousands of types of administrative requests remotely to manage everything from desktop settings, to DNS, to file server exports via PowerShell and MMC (Windows) or use Linux's “net” and “samba-tool” and smbclient on Linux, or use the many third party systems management tools
- Domain Controller (central security and configuration server) ties resource configuration and security together in one place across multiple protocols (Kerberos, LDAP, DNS, DHCP, NTP, DCE/RPC, Print, SMB2)
  - Linux implementation based on Samba 4 looks great (check back at SambaXP this May for an update)
- Generally it is easier to manage SMB2 because the many existing tools just work, whether to Windows, or NAS, or Samba. Very rich management/admin environment