SMB3.11: Recent Improvements in Linux Access to the Cloud, NAS and Popular SMB3 Based Systems

Steve French
Principal Software Engineer
Azure Storage - Microsoft
Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft 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, former SNIA CIFS Working Group chair

- Principal Software Engineer, Azure Storage: Microsoft
Outline

- General Linux Linux FS and VFS Activity and Status
- What are the goals?
- Key Feature Status
- Features under development, expected soon
- Performance overview
- POSIX compatibility and status of SMB3 Extensions
- Testing
Outline

• A year ago we had Linux kernel 4.13 “Fearless Coyote”

• Now kernel 4.19-rc5 “Merciless Moray”
What is driving file system activity?

- Proposed new mount and fsinfo API; extending ‘statx’
- Many critical evolving storage features:
  - Better support for faster storage
  - RDMA and low latency ways to access VERY high speed storage (e.g. NVMe), and faster/cheaper (10Gb → 40Gb→100Gb) ethernet
  - I/O priority
- Broadening use of copy offload (e.g. fix tools to use “copy_file_range” syscall) and making copy smart
- Cloud: longer latency, object & file coexist, strong sec
Activity since January 2018 (4.15 kernel)

- 3900 kernel file system changes since 4.15 kernel released, 6.8% of kernel overall (up). FS are important to Linux!
- Kernel is now 17.3 million lines of source code (measured last week with sloccount tool)
- 60+ Linux file systems. cifs.ko (cifs/smb3 client) among more active (#4 out of 60 and growing). More activity is good!
- BTRFS 801 changesets (up), most changesets of any fs related component
- VFS (overall fs mapping layer and common functions) 535
- XFS 507 (up)
- F2FS 313 (down)
- **cifs.ko** (CIFS/SMB2/SMB3 client) 276 (up more than 100%! And continuing to increase)
  - Has 48,652 lines of kernel code (not counting user space helpers and samba userspace tools)
- NFS client 212 (down)
- NFS server 72 (down). Linux NFS server is **MUCH** smaller than Samba server (or even CIFS or NFS clients).
- And various other file systems: EXT4 142, Ceph 116, GFS2 98, AFS 87 ...
- NB: Samba is as active as all Linux file systems put together (>4000 changesets per year) - broader in scope (by a lot) and also is user space not kernel. 3.4Million Lines of Code. **100x larger than the NFS server in Linux!**
Linux File Systems: talented developers

At Linux FS Summit in Utah in April
Samba team: Amazing group

Some at SMB3 I/O lab in Redmond last week …
What are our goals?

- Make SMB3/SMB3.11 and followons fastest, most secure general purpose way to access file data, whether in cloud, on premises or virtualized.
- Implement all reasonable Linux/POSIX features - so apps don’t have to know they are running on SMB3 mounts (vs. local).
- As Linux evolves, and need for new features discovered, can quickly add to kernel client and Samba.
Fixes and Features in progress last year ...

• Lots of completed work!

• Full SMB3.11 support!
• Statx (extended stat linux API returning additional metadata flags)
• Improved performance
• RDMA (smbdirect)
• Improved POSIX compatibility (see later talk!)
• Security improvements
• Multidialect support
• Snapshots
Exciting Year!

- Faster performance
- POSIX Extensions (finally)!
- SMB3.11, improved security
- LOTS of new features
35% more efficient mount & SMB3.11 works!
And SMB3.11 encryption works ... 

- “mount -t cifs //server/share /mnt -o vers=3.11,seal”
- Thanks Aurelien!
Can load it as ‘smb3’ and even disable cifs

- Improving security: can disable cifs

```
root@smf-Thinkpad-P51:~# modprobe smb3 disable_legacy_dialects=1
root@smf-Thinkpad-P51:~# mount -t cifs //localhost/scratch /mnt1 -o vers=1.0,username=testuser,
mount error(22): Invalid argument
Refer to the mount.cifs(8) manual page (e.g. man mount.cifs)
root@smf-Thinkpad-P51:~# dmesg
[ 294.844994] FS-Cache: Netfs 'cifs' registered for caching
[ 294.845081] Key type cifs.spnego registered
[ 294.845084] Key type cifs.idmap registered
[ 297.769583] CIFS VFS: mount with legacy dialect disabled
```
Tracing with the new ftrace is so easy ...
Current List of CIFS/SMB3 tracepoints and an example of detail for one.
Example output: tracing mount and touch (create file) failure
Splice write fixed (also helps sendfile)

```
root@smf-Thinkpad-P51:~# gio copy /mnt1/trace.dat /mnt1/target.dat
Transferred 7.2 MB out of 7.2 MB (7.2 MB/s)
root@smf-Thinkpad-P51:~#
```
Statx (and cifs pseudoxattrs) and get/set real xattrs work

```bash
root@smf-Thinkpad-P51:/mnt1# setfattr file2 -n user.somexattr -v somevalue
root@smf-Thinkpad-P51:/mnt1# getfattr file2 -d
# file: file2
user.somexattr="somevalue"

root@smf-Thinkpad-P51:/mnt1# ~/statx/test-statx file2 2M
statx(file2) = 0
results=fdf
  Size: 0       Blocks: 0       IO Block: 16384   regular file
Device: 00:38   Inode: 13107206  Links: 1
Access: (0755/-rwxr-xr-x) Uid: 0    Gid: 0
Modify: 2018-06-05 02:39:25.088837500-0500
Change: 2018-06-05 02:39:25.088837500-0500
  Birth: 2018-05-31 18:06:01.644761500-0500
SMB3/CIFS Features by release (cont)

- 4.13 (27 changesets) September 3rd, 2017
  - Change default dialect to SMB3 from CIFS
  - SMB3 support for “cifsacl” mount option (and mode emulation)
  - Bug fixes

- 4.14 (37 changesets) November 12th, 2017
  - Bug fixes (especially for SMB2.1/SMB3 validate negotiate)
  - Default dialect changed to multidialect (SMB2.1, SMB3, SMB3.02)
  - Added xattr support for SMB2/SMB3

- 4.15 (6 changesets) – January 28, 2018
  - Minor bug fixes
SMB3/CIFS Features by release (cont)

- **4.16 (68 changesets)** – April 1
  - Add splice_write support
  - Add support for smbdirect (SMB3 rdma). Thanks Long Li!
- **4.17 (56 changesets)** - June 3
  - Bug fixes
  - Add signing support for smbdirect
  - Add support for SMB3.11 encryption, and preauth integrity
  - SMB3.11 dialect improvements (and no longer marked experimental)
- **4.18 (89 changesets!)** - August 12th
  - RDMA and Direct I/O improvements (Thank you Long Li!)
  - Bug fixes
  - SMB3 POSIX extensions (initial minimal set, open and negotiate context only. use ‘posix’ mnt parm)
  - Add “smb3” alias to cifs.ko (“insmod smb3”)
  - Allow disabling less secure dialects through new module install parm (disable_legacy_dialects)
  - Add support for improved tracing (ftrace, trace-cmd) – thanks to XFS developers for good ideas!
  - Cache root file handle, reducing redundant opens, improving perf (Thanks Ronnie!)
- **4.19-rc4 (65 changesets)** (4.19 expected to be released In late October)
- **For-next (next kernel, 4.20)** (26 changesets so far and many patches being added this week!)
Linux CIFS/SMB3 client bug status summary

- Bugzilla.kernel.org summary
- Bugzilla.samba.org summary
- Would love help to triage, and close out some of the bugs which are already fixed.
New Features!

- SMB3 … even better than before!
- smbdirect/RDMA
- Snapshot mounts
- Compounding
- Multichannel
- And more ...
SMBDIRECT – SMB3 and RDMA

- Thank you Long Li (slides courtesy of him)
- High Speed!
Test environment

• Hardware
  • Mellanox ConnectX-3 Pro 40G Infiniband
  • Mellanox SX6036 40G VPI switch
  • 2 x Intel E5-2650 v3 @ 2.30GHz
  • 128GB RAM

• Windows 2016 SMB Server
  • SMB Share on RAM disk

• Windows 10 client
  • Registry settings limits to 1 RDMA connection
SMB Read 40G Infiniband – comparing to Windows

I/O size at queue depth 16

MB/s

4K | 16K | 64K | 256K | 1M | 4M

SambaXP2018
Now
Windows 10
Snapshot mounts

- Want to compare backups?
- Look at previous versions?
- Recover corrupted data
- ...
- Demo
SMB2/SMB3 Compounding

(Slides courtesy of Ronnie Sahlberg at RedHat who is doing great work improving this)

- Hard work is done now and merged, and helps with “df” (statfs) for example. Compounding for 10 other additional operations are in for-next

- smb2 compounding is VERY flexible and there are a lot of places in cifs.ko where we will be able to use them to
  - improve performance
  - also make the client get slightly more posix like behavior from smb2.

- There are still many number of places where we should switch to using compounding.
### smb2

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000060</td>
<td>192.168.124.203</td>
<td>192.168.124.1</td>
<td>SMB2</td>
<td>198</td>
<td>Create Request File:</td>
</tr>
<tr>
<td>2</td>
<td>0.000064358</td>
<td>192.168.124.1</td>
<td>192.168.124.203</td>
<td>SMB2</td>
<td>222</td>
<td>Create Response File: [unknown]</td>
</tr>
<tr>
<td>3</td>
<td>0.001715177</td>
<td>192.168.124.203</td>
<td>192.168.124.1</td>
<td>SMB2</td>
<td>174</td>
<td>GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO File: [unknown]</td>
</tr>
<tr>
<td>5</td>
<td>0.000199699</td>
<td>192.168.124.1</td>
<td>192.168.124.203</td>
<td>SMB2</td>
<td>244</td>
<td>GetInfo Response</td>
</tr>
<tr>
<td>6</td>
<td>0.002740685</td>
<td>192.168.124.1</td>
<td>192.168.124.1</td>
<td>SMB2</td>
<td>158</td>
<td>Close Request File: [unknown]</td>
</tr>
<tr>
<td>7</td>
<td>0.000274182</td>
<td>192.168.124.1</td>
<td>192.168.124.203</td>
<td>SMB2</td>
<td>194</td>
<td>GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO File: [unknown]</td>
</tr>
<tr>
<td>8</td>
<td>0.000332539</td>
<td>192.168.124.203</td>
<td>192.168.124.1</td>
<td>SMB2</td>
<td>194</td>
<td>Close Response</td>
</tr>
<tr>
<td>9</td>
<td>0.000425636</td>
<td>192.168.124.1</td>
<td>192.168.124.203</td>
<td>SMB2</td>
<td>194</td>
<td>GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO File: [unknown]</td>
</tr>
<tr>
<td>10</td>
<td>0.005065979</td>
<td>192.168.124.1</td>
<td>192.168.124.1</td>
<td>SMB2</td>
<td>194</td>
<td>Closed Request File:</td>
</tr>
<tr>
<td>11</td>
<td>0.005326782</td>
<td>192.168.124.1</td>
<td>192.168.124.203</td>
<td>SMB2</td>
<td>194</td>
<td>Closed Request File:</td>
</tr>
<tr>
<td>12</td>
<td>0.006505853</td>
<td>192.168.124.203</td>
<td>192.168.124.1</td>
<td>SMB2</td>
<td>194</td>
<td>Closed Request File:</td>
</tr>
<tr>
<td>13</td>
<td>0.006694439</td>
<td>192.168.124.1</td>
<td>192.168.124.203</td>
<td>SMB2</td>
<td>194</td>
<td>Closed Request File:</td>
</tr>
<tr>
<td>14</td>
<td>0.006740080</td>
<td>192.168.124.203</td>
<td>192.168.124.1</td>
<td>SMB2</td>
<td>194</td>
<td>Closed Request File:</td>
</tr>
<tr>
<td>15</td>
<td>0.012183134</td>
<td>192.168.124.1</td>
<td>192.168.124.203</td>
<td>SMB2</td>
<td>454</td>
<td>Create Response File: [unknown]; GetInfo Response; Close Request</td>
</tr>
</tbody>
</table>

- Frame 14: 390 bytes on wire (3120 bits), 390 bytes captured (3120 bits) on interface 0
- NetBIOS Session Service
- SMB2 (Server Message Block Protocol version 2)
  - SMB2 Header
    - Create Request (0x05)
- SMB2 (Server Message Block Protocol version 2)
  - SMB2 Header
    - GetInfo Request (0x10)
- SMB2 (Server Message Block Protocol version 2)
  - SMB2 Header
    - Close Request (0x05)
API

- You create an array of requests. One request at a time and set if they are related or not.
- The result is an array of iovectors, one vector per request.
First a CREATE at [0]

```c
oparms.tcon = tcon;
oparms.desired_access = FILE_READ_ATTRIBUTES;
oparms.disposition = FILE_OPEN;
oparms.create_options = 0;
oparms.fid = &fid;
oparms.reconnect = false;

rc = SMB2_open_init(tcon, &rqst[0], &oplock, &oparms, &srch_path);
if (rc)
    goto qfs_exit;
smb2_set_next_command(&rqst[0]);
```
Then a QUERY INFO at [1]

```c
rc = SMB2_query_info_init(tcon, &rqst[1], COMPOUND_FID, COMPOUND_FID,
                          FS_FULL_SIZE_INFORMATION,
                          SMB2_O_INFO_FILESYSTEM, 0,
                          sizeof(struct smb2_fs_full_size_info));

if (rc)
    goto qfs_exit;
smb2_set_next_command(&rqst[1]);
smb2_set_related(&rqst[1]);
```
Finally a CLOSE at [2]

```c
rc = SMB2_close_init(tcon, &rqst[2], COMPOUND_FID, COMPOUND_FID);
    if (rc)
        goto qfs_exit;
    smb2_set_related(&rqst[2]);
```
Send off the request

```c
rc = compound_send_recv(xid, ses, flags, 3, rqst,
                        resp_buftype, rsp_iov);
if (rc)
    goto qfs_exit;
```

rsp_iov returns an array of 3 response vectors.
Thank you Aurelien!

Made a lot of progress last week at the Samba test event

See example wireshark trace showing, 2\textsuperscript{nd} connection opened successfully and used by Linux client (to Windows 2016)
Multichannel (continued)

The client wants to bind to an existing session (smb2, ses_req, flags, session_binding), 1 byte.

The client then sends a Session Setup Request with the following flags:
- Flags: 1, Session Binding Request
- Security mode: 0x02, Signing required

The server receives the request and responds with a Session Setup Response, which includes:
- Status: Success
- Flags: 1, Session Binding Request
- Security mode: 0x02, Signing required

This indicates that a new channel has been established.

Main connection

New channel
Better HA: Reconnect improvements

- Resilient and persistent handles are supported, and reconnect continues to improve
- Some remaining items:
  - Add lock sequence number
  - Fix EAGAIN rc which can occur for pending ops which overlap a reconnect, and some reconnect bugs
  - Improve server to server failover
    - Allow alternate (failover) targets using DFS referrals
    - Witness protocol: server or share redirection
SMB3 and ACLs

- “cifsacl” mount option now supported for SMB3 for emulating mode bits via ACL
- Has various problems in practice emulating mode bits
- Alternatives (e.g. mount options) that we are testing this week
- And what about ACLs in the POSIX extensions ...
SMB3 Security Features

- SMB3.11 is no longer experimental, and works well
- SMB3.1.1 secure negotiate works (better than validate negotiate ioctl from SMB2.1 and SMB3)
- SMB3 and SMB3.11 Share Encryption works
  - AES128-CCM encryption algorithm is negotiated (AES128-GCM not supported yet for Linux client or Samba)
- And make it easy to disable cifs (vers=1.0)!
Passthrough ioctl ...

- Passthrough “query info” call (Thank you Ronnie!)
- Passthrough fsctl call (ioctl → smb3 fsctl) – prototype in progress
- Many interesting, useful features
  - Now we just need some python or C user space helpers to make them easier to use ...
Other Optional features

- statfs integration and new mount api integration
  - New API in Al Viro’s tree
- IOCTLs e.g. to list alternate data streams
  - NB: Querying data in alternate data streams (e.g. for backup) requires disabling posix pathnames (due to conflict with “:”)
- Clustering, Witness protocol integration
- DFS reconnect to different DFS server
- Performance features (see next slides)
- Other suggestions ...
POSIX Extensions for SMB3!

- See POSIX Extensions talk here!
- But here are some examples of improvements (even with current kernel, without all the extensions checked in)
CIFS Version 2.12
Features: dfs fsache lanman posix spnego xattr acl
Active VFS Requests: 0
Servers:
Number of credits: 16 Dialect 0x311 posix
1) Name: 127.0.0.1 Uses: 2 Capability: 0x300047 Session Status: 1 TCP status: 1
   Local Users To Server: 1 SecMode: 0x1 Req On Wire: 0
   Shares:
   0) IPC: \127.0.0.1\IPC$ Mounts: 1 DevInfo: 0x0 Attributes: 0x0
      PathComponentMax: 0 Status: 1 type: 0
      Share Capabilities: None Share Flags: 0x0
      tid: 0x4f5511db Maximal Access: 0x1f00a9

1) \localhost\test Mounts: 1 DevInfo: 0x20 Attributes: 0x1006f
   PathComponentMax: 255 Status: 1 type: DISK
   Share Capabilities: None Aligned, Partition Aligned, Share Flags: 0x0
   tid: 0x8579c31d Optimal sector size: 0x200 Maximal Access: 0x1f01ff

2) \localhost\test-no-posix Mounts: 1 DevInfo: 0x20 Attributes: 0x1006f
   PathComponentMax: 255 Status: 1 type: DISK
   Share Capabilities: None Aligned, Partition Aligned, Share Flags: 0x0
   tid: 0x1813a493 Optimal sector size: 0x200 Maximal Access: 0x1f01ff
MIDs:
Mode bits on create and case sensitive!

```
root@Ubuntu-17-Virtual-Machine:/mnt# ~/create-4-files-with-mode-test
root@Ubuntu-17-Virtual-Machine:/mnt# cd /mnt1
root@Ubuntu-17-Virtual-Machine:/mnt1# ~/create-4-files-with-mode-test
root@Ubuntu-17-Virtual-Machine:/mnt1# ls /test /test-no-posix -la
/test:
total 12
drwxrwxrwx 3 root root 4096 May 31 16:55  1
-rwxr-xr-- 1 testuser testuser 0 May 31 16:55 0700
-rwxr--r-- 1 testuser testuser 0 May 31 16:55 0770
-rwxrwxr-x 1 testuser testuser 0 May 31 16:55 0775
drwxr-xr-x 2 sfrench sfrench 4096 Mar 24 10:34 tmp
/test-no-posix:
total 8
drwxrwxrwx 2 root root 4096 May 31 16:55  1
-rwxr-xr-- 1 testuser testuser 0 May 31 16:55 0700
-rwxr--r-- 1 testuser testuser 0 May 31 16:55 0770
-rwxr-xr-- 1 testuser testuser 0 May 31 16:55 0775
root@Ubuntu-17-Virtual-Machine:/mnt1# mkdir UPPER
root@Ubuntu-17-Virtual-Machine:/mnt1# touch upper
root@Ubuntu-17-Virtual-Machine:/mnt1# cd /mnt
root@Ubuntu-17-Virtual-Machine:/mnt# mkdir UPPER
root@Ubuntu-17-Virtual-Machine:/mnt# touch upper
root@Ubuntu-17-Virtual-Machine:/mnt# ls /test /test-no-posix
/test:
0700 0770 0775  tmp  upper  UPPER
/test-no-posix:
0700 0770 0775  UPPER
```
Rename works with POSIX extensions!
Googling NFS vs. SMB3 (or Samba) ... first result said:

"As you can see NFS offers a better performance and is unbeatable if the files are medium sized or small. If the files are large enough the timings of both methods get closer to each other. Linux and Mac OS owners should use NFS instead of SMB. Sadly Windows users are forced to use SMB ..."
Is NFS really always faster than Samba...
MYTHBUSTERS
There are cases where SMB3 to Samba is faster

- Localhost (network shouldn’t be an issue. Default Ubuntu Samba server vs. NFS kernel server. Default parms. Comparing NFSv3, NFSv4.2 and cifs.ko (SMB3.02 dialect is default)
- fio with the read/write job file: SMB3 12.5% faster to Samba (than NFSv4.2 server) for random reads and SMB3 12.8% faster for writes
- For sequential: SMB3 31.8% faster for read, 31.2% faster for write (and not just because of stricter sync)
- Even simple DD command with large file i/o shows SMB3 can be faster Linux to Linux for write than NFS
- For the many cases where NFS is faster … let’s take a look after compounding and improved handle caching … (and don’t forget “smbdirect” and huge progress with RDMA, hard to beat 95%+ utilization with SMB3/RDMA!)
... 1st test I tried SMB3 wins by 29% over NFS (defaults, localhost mounts)
Maybe coincidence so let's try fio … (at 1am!)

- Standard fio random read/write i/o job file, localhost Samba vs. NFS, using all defaults
- `/mnt2: fio ~/fio/fio-rand-RW.job`
- SMB3 20% faster than NFS for read, 21% for write
Still a lot of work to do though! SMB3 Performance WIP: great features ... but only if we implement them ...

- Key Features
  - Compounding (in 4.18, more will be in 4.20)
  - Large file I/O (looks good, let’s continue to optimize)
  - File Leases
    - Lease upgrades
  - Directory Leases (complete for root directory, to be extended ...)
  - Handle caching (under investigation)
  - Crediting (very helpful feature)
  - I/O priority
  - Copy Offload
  - Multi-Channel (in progress)
    - And optional RDMA (much improved, will be even better in 4.20)
  - Linux specific protocol optimizations possible too ...
Conclusion … When is SMB3 good?

- When need nice security …
- Workloads where performance with lots of large directories is not an obstacle (pending improvements to leasing and compounding in cifs.ko)
- Workloads which do not depend on case sensitivity (common unfortunately) and do not depend on advisory locking or delete of open files (more rare) … pending POSIX extensions in Samba etc.
- Where you can take advantage of smbdirect (RDMA)
- Where global namespace (DFS) helps
- Where rich features of SMB3 (snapshots, encrypted/compressed files, persistent handles) are helpful …
- And of course … to the cloud (Azure) and Macs and Windows and … not just Samba
Testing ... testing ... testing

- See xfstesting page in cifs wiki
  https://wiki.samba.org/index.php/Xfstesting-cifs
- Easy to setup, exclude file for slow tests or failing ones
- XFSTEST status update
  - Bugzillas
  - Features in progress
  - Automating improvements
Thank you for your time

- Future is very bright!

+ SMB3
Additional Resources to Explore for SMB3 and Linux

- Linux CIFS client https://wiki.samba.org/index.php/LinuxCIFS
- Samba-technical mailing list and IRC channel
- And various presentations at http://www.sambaxp.org and Microsoft channel 9 and of course SNIA … http://www.snia.org/events/storage-developer
- And the code:
  - https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/fs/cifs
  - For pending changes, soon to go into upstream kernel see:
    - https://git.samba.org/?p=sfrench/cifs-2.6.git;a=shortlog;h=refs/heads/for-next