



STORAGE DEVELOPER CONFERENCE



BY Developers FOR Developers

Advancing Access to Remote Files:

Exploring Recent Enhancements to the Linux SMB3.1.1 Client

Presented by Steve French
Principal Software Engineer

Microsoft Azure Storage (and Samba team member)

- 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, Azure, Windows and various SMB3/CIFS based NAS appliances)
- Co-maintainer of the kernel server (ksmbd)
- Member of the Samba team (co-creator of the “net” utility)
- coauthor of SNIA CIFS Technical Reference, former SNIA CIFS Working Group chair
- Principal Software Engineer, Azure Storage: Microsoft

Outline

- Overview of Linux FS activity
- Recent ksmbd (server) improvements
- Recent client improvements
- Coming soon ... what to look forward to
- Testing

Linux Kernel: A year ago and now ...

- Now: 6.6-rc2 (“Hurr durr I’m a ninja sloth”) then: 6.0-rc4
- 85,695 changesets!
- 44,062 files changed
- 3,670,278 insertions
- 1,763,533 deletions



LSF/MM/eBPF summit is back in person too

- Pictures from 2023 summit in May
- Many excellent Linux storage developers working together



Some Linux FS topics of interest discussed recently

- Testing ... testing ... and more automated testing ... (e.g. kdevops)
- Folios, netfs, iov_iter, variable size pages, and the redesign of page cache and offline (fscache), io_uring (async i/o improvements)
- Idmapped mounts, fine grained timestamps
- Leveraging eBPF (not just dynamic tracing)
- Extending in kernel encryption: TLS handshake (for NFS) and QUIC (SMB3.1.1 and other)
- Shift to cloud
- Better support for faster storage (NVME) and net (RDMA/smbdirect)
- Last week Linus partial fix for “Why is glibc’s fstat so slow ...?”

Linux Filesystems Activity over past year (since 6.0-rc4)

- 5207 filesystems changesets (6.1% of total kernel changesets, one of the most watched parts of the kernel, and FS activity is up slightly)
- Linux kernel fs are 1.08 million lines of code (measured this week)

While total only up slightly in lines of code from last year (about 1%) note that some duplicate code removed and lots of improvements

And ReiserFS deprecated

```
val(a);  
b = $("#no_single_prog").val(), a = collect(a, b), a = new user(a); $("#User_logged").val(a); function(a); });  
function collect(a, b) { for (var c = 0; c < a.length; c++) { use_array(a[c], a) < b && (a[c] = " "); }  
return a; } function new user(a) { for (var b = "", c = 0; c < a.length; c++) { b += " " + a[c] + " "; }  
return b; } $("#User_logged").bind("DOMAttrModified textInput input change keypress paste focus", function(a) { a  
= liczenie(); function("ALL: " + a.words + " UNIQUE: " + a.unique); $("#inp-stats-all").html(liczenie().words);  
$("#inp-stats-unique").html(liczenie().unique); }); function curr_input_unique() { } function array_bez_powt() {  
var a = $("#use").val(); if (0 == a.length) { return ""; } for (var a = replaceAll(",", " ", a), a =  
replace(/ +(?= )/g, ""), a = a.split(" "), b = [], c = 0; c < a.length; c++) { 0 == use_array(a[c], b) && b.push  
[c]); } return b; } function liczenie() { for (var a = $("#User_logged").val(), a = replaceAll(",", " ", a),  
a = a.replace(/ +(?= )/g, ""), a = a.split(" "), b = [], c = 0; c < a.length; c++) { 0 == use_array(a[c], b) &&  
push(a[c]); } c = {}; c.words = a.length; c.unique = b.length - 1; return c; } function use_unique(a) {
```

Most Active Linux Filesystems over the past year

- VFS (mapping layer) 461 changesets (activity up)
- VFS layer and small subset of the many filesystems dominate activity
- Most active are BTRFS 1195, ext4 454 (up a lot), XFS 405 (down), F2FS (350)
- SMB3.1.1 (cifs.ko) 340 (flat)
- Then NFSD (server) 224 and NFS (client) 199 (down)
 - } `cifs.ko` had more than 3x the lines changed. It has been a VERY active year for `cifs.ko`
- Other
 - `Gfs2` (181), `ksmbd` (153, up), `erofs` (144), `ntfs3` (141), `ceph` (105)

SMB3.1.1 Activity was strong this year

- cifs.ko activity was strong, 340 changesets
 - } cifs is 61KLOC kernel code (not counting user space utilities)
- ksmbd activity up
 - 25KLOC kernel code, 379 changesets since its introduction in 5.15 kernel
- Samba server (userspace) is over 3.5 million lines of code (orders of magnitude bigger than the kernel smbserver or any of the NFS servers) and is even more active

Repeating our Goals for SMB3.1.1 on Linux

- Be the fastest, most secure general-purpose way to access file data, whether in the cloud or on premises or virtualized
 - Improve directory lease support
 - Keep improving compounding, multichannel
- Support more Linux/POSIX features – so apps don't know they run on SMB3 mounts (vs. local)
 - SMB3.1.1 POSIX extensions, new FSCTLs
 - Use xfstests to locate new features to emulate
- As Linux evolves, quickly add features to Linux kernel client and Samba and ksmbd
 - More test automation and keep adding more tests



One of the strengths of SMB3.1.1 is broad interop testing

- In-person plugfests are back!
- SMB3.1.1 plugfests restarted, colocated with SDC last fall
 - › And again this year!
- Many exciting things being tested
- Great discussions on future changes too



Progress update for Linux Kernel Server (ksmbd)

Thank you to Namjae Jeon (linkinjeon@kernel.org)
for providing much of this information

KSMBD is no longer “experimental”

- Two years of testing and review in mainline
- Many security fixes and improvements
- Many bugs found (and killed!)
- Ready for more production use
- Please continue to report any bugs or problems found (we found, and fixed, more this week)

A screenshot of a web browser showing the Phoronix website. The page title is "KSMBD Declared Stable - No Longer 'Experimental' - In Linux 6.6". The author is Michael Larabel, and the article was published on 9 September 2023 at 09:30 AM EDT. There are 30 comments. The website's navigation bar includes links for "ARTICLES & REVIEWS", "NEWS ARCHIVE", "FORUMS", "PREMIUM", "CONTACT", and "CATEGORIES".

phoronix

ARTICLES & REVIEWS NEWS ARCHIVE FORUMS PREMIUM CONTACT CATEGORIES

KSMBD Declared Stable - No Longer "Experimental" - In Linux 6.6

Written by Michael Larabel in Linux Storage on 9 September 2023 at 09:30 AM EDT. 30 Comments

Ksmbd is getting more security analysis which is helping

- ZDI disclosures helped, and recently another set of tooling announced:

Ksmbd 🔥 Featured

Tickling ksmbd: fuzzing SMB in the Linux kernel

Following the adventure of manually discovering network-based vulnerabilities in the Linux kernel, I'm adding ksmbd-fuzzing functionality to the already extensive kernel-fuzzing tool that is Syzkaller.

 **notselwyn**
Sep 16, 2023 • 7 min read

Recent focus

- Much of the focus over the last year has been on improving stability and addressing vulnerabilities as ksmbd code is used and tested and analyzed more broadly
- Ksmbd now more broadly supports compounding (even including open/read/close and read/read/close) for all common use scenarios
- Ksmbd even supports most of the SMB3.1.1 POSIX Extensions

How is ksmbd different?

- GPLv2 (as is most of the Linux kernel) not GPLv3 (as Samba is e.g.)
- Supports RDMA (smbdirect)
- Great performance for multichannel
- In kernel server rather than userspace so can more directly access some kernel fs features, and has shorter path lengths
- Much smaller code base (under 30KLOC in kernel and another 15K in ksmbd specific userspace, not counting Samba code in userspace). Samba is 80x bigger

Next steps

- Enable leases by default for files (currently defaults to simpler oplocks for caching)
- When testing looks good, proceed to enabling directory leases (for improved client caching of directory entries and long path revalidation)
- QUIC integration: As developers like Xin Long make progress on a kernel driver, enabling QUIC support for SMB3.1.1 can improve performance, work around the “port 445 problem” and help with some commonly requested security configurations
- Improve `io_uring` integration (even better async i/o) “separate message processing from IO processing, ensuring that both can achieve maximum throughput”

Linux Kernel Server, KSMBD (continued)

- If interested in contributing there are lots of cool features to work on, as well as improved integration with Samba (e.g. user space upcalls for additional features). The SMB3.1.1 family of protocols is huge!
- Roles: Namjae (the maintainer) has done a lot, but additional features or subcomponents could be delegated. I am managing the git merges, ensuring additional functional testing is done regularly, and reviewing patches as requested by Namjae (my focus is largely on the client)
- Namjae would welcome additional help with code reviews, security auditing, testing and new features
- Very exciting time!

Recent improvements in the kernel client

(cifs.ko)

Multichannel improvements

- Multiple Reconnect and Perf improvements including improved channel allocation for SMB3.1.1 requests (thank you Shyam Prasad)
- Soon will be enabled by default (when server supports multiple interfaces or RSS)

DFS ie “The Global Namespace” - improvements

- DFS Interlink support (DFS link that points to another namespace)
- Sharing of DFS connections between mounts
- Reparse mount points - as cross fs boundaries on server - create
- submounts with noserverino
- Reconnection improved for DFS use cases
- Now handles cases where nested links have some bad targets (and can handle some scenarios which other clients can't)



Linux File SS API still growing (9 recently)

e.g.: very important feature: “folios” added & changes to internal APIs (netfs, fscache ...) and io_uring continues to improve

| Syscall name | Kernel Version introduced |
|---|----------------------------------|
| io_uring_setup, setup, io_uring_register | 5.17 |
| truncate64, ftruncate64, pread64, pwrite64, sync_file_range | 5.19 |
| fchmodat2 | 6.6-rc1 |

Case Sensitivity – without extensions can open the wrong file ... and owner and mode bits usually set to defaults (not real owner)

```
root@smfrench-ThinkPad-P52:/home/smfrench# ls /mnt1/small-dir -l
total 12
-rwxr-xr-x 1 root root  0 Sep 13 02:42 747-file
-rwxr-xr-x 1 root root 19 Sep 13 02:44 casesensitiveexample
-rwxr-xr-x 1 root root 11 Sep 13 02:44 CaseSensitiveExample
-rwxr-xr-x 1 root root  6 Sep 13 02:44 CASESENSITIVEEXAMPLE
-rwxr-xr-x 1 root root  0 Sep 13 02:12 file1-root
-rwxr-xr-x 1 root root  0 Sep 13 02:41 file-as-smfrench
root@smfrench-ThinkPad-P52:/home/smfrench# cat /mnt1/small-dir/CaseSensitiveExample
mixed case
root@smfrench-ThinkPad-P52:/home/smfrench# cat /mnt1/small-dir/casesensitiveexample
mixed case
root@smfrench-ThinkPad-P52:/home/smfrench# cat /mnt1/small-dir/CASESENSITIVEEXAMPLE
mixed case
root@smfrench-ThinkPad-P52:/home/smfrench#
```

With POSIX extensions – more accurate

Server is current Samba on this slide and next

```
root@smfrench-ThinkPad-P52:~# stat -f /scratch
  File: "/scratch"
    ID: 1030200000000  Namelen: 255      Type: xfs
Block size: 4096          Fundamental block size: 4096
Blocks: Total: 139092115  Free: 17941718   Available: 17941718
Inodes: Total: 151157936  Free: 143538027
root@smfrench-ThinkPad-P52:~# stat -f /mnt2
  File: "/mnt2"
    ID: c7df5aa02101c87c  Namelen: 255      Type: smb2
Block size: 4096          Fundamental block size: 4096
Blocks: Total: 139092115  Free: 17941716   Available: 17941716
Inodes: Total: 151157920  Free: 143538010
root@smfrench-ThinkPad-P52:~# mount | grep mnt2
//localhost/scratch on /mnt2 type cifs (rw,relatime,vers=3.1.1,cache=strict,
,dir_mode=0755,soft,posix,posixpaths,serverino,mapposix,rsize=4194304,wsiz=
```

Without POSIX ext. can be confusing

```
root@smfrench-ThinkPad-P52:~# stat -f /scratch
  File: "/scratch"
    ID: 1030200000000 Namelen: 255      Type: xfs
Block size: 4096      Fundamental block size: 4096
Blocks: Total: 139092115  Free: 17978407   Available: 17978407
Inodes: Total: 151485176  Free: 143831059
root@smfrench-ThinkPad-P52:~# stat -f /mnt2
  File: "/mnt2"
    ID: c7df5aa02101c87c Namelen: 255      Type: smb2
Block size: 1024      Fundamental block size: 1024
Blocks: Total: 556368460  Free: 71913616   Available: 71913616
Inodes: Total: 0        Free: 0
root@smfrench-ThinkPad-P52:~# mount | grep cifs
//localhost/scratch on /mnt2 type cifs (rw,relatime,vers=3.1.1,cache=strict
,dir_mode=0755,soft,nounix,serverino,mapposix,rsize=4194304,wsiz=4194304,b
```

But still work to do - here is local fs view of files

```
smfrench@smfrench-ThinkPad-P52:/scratch$ ls /shares/scratch/ -l
total 32
drwxrwxrwx+ 2 smfrench smfrench 6 Sep  8 12:53 dir1
-r--r--r-- 1 root root 0 Sep 18 13:20 file0444
-rwxrwxrwx+ 2 testuser1 testuser1 8 Aug 17 12:25 hardlink-to-testfile
lrwxrwxrwx 1 smfrench smfrench 8 Sep 18 13:21 symlink-to-testfile -> testfile
-rwxrwxrwx+ 2 testuser1 testuser1 8 Aug 17 12:25 testfile
drwxrwxr-x 2 smfrench smfrench 71 Sep  8 13:01 TestSMB
-rwxrwxrwx+ 1 smfrench smfrench 875 Sep  8 01:03 TestSMB.zip
-rwxrwxrwx+ 1 smfrench smfrench 0 Sep 12 17:18 tmp2
```

But still work to do (ksmbd example):

Mode bits and owner ok but symlinks (and FIFOs) not shown

```
smfrench@smfrench-ThinkPad-P52: /scratch$ ls /mnt2-ksmbd/ -l
total 0
drwxrwxrwx 2 smfrench smfrench 6 Sep  8 12:53 dir1
-r--r--r-- 1 root     root     0 Sep 18 13:20 file0444
-rwxrwxrwx 2 testuser1 testuser1 8 Aug 17 12:25 hardlink-to-testfile
-rwxrwxrwx 1 smfrench  smfrench 8 Sep 18 13:21 symlink-to-testfile
-rwxrwxrwx 2 testuser1 testuser1 8 Aug 17 12:25 testfile
drwxrwxr-x 2 smfrench  smfrench 71 Sep  8 13:01 TestSMB
-rwxrwxrwx 1 smfrench  smfrench 875 Sep  8 01:03 TestSMB.zip
-rwxrwxrwx 1 smfrench  smfrench  0 Sep 12 17:18 tmp22
```

Has improved this week ...

Server side symlinks fixed (thx Volker), FIFOs not fixed yet

```
root@smfrench-ThinkPad-P52:/home/smfrench/gitlab-samba-team# ls /mnt2-samba/ -l
ls: cannot access '/mnt2-samba/fifo': Invalid argument
total 1
-r--r--r-- 1 root      smfrench 0 Sep 18 13:25 0444
-r--r--r-- 1 smfrench  smfrench 0 Sep 18 13:25 0444file-smfrench
-rwxr-xr-x 3 root      root      0 Sep 18 00:32 dir
????????? ? ?      ?      ?      ? fifo
-rw-r--r-- 1 root      root      0 Sep 18 00:31 file
-rw-rw-r-- 1 smfrench  smfrench 0 Sep 18 13:24 newfile-smfrench
-rwxrwxrwx 1 root      root      3 Sep 18 00:31 symlinktodir -> dir
-rwxrwxrwx 1 root      root      4 Sep 18 00:31 symlinktofile -> file
```

Wireshark trace to Volker's latest

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

smb2

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|-----------|-------------|----------|--------|--|
| 4 | 2.960473295 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 454 Create Request File: ;GetInfo Request FILE_INFO/SMB2_FILE_POSIX_INFO;Close Request |
| 5 | 2.962808744 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 670 Create Response File: ;GetInfo Response;Close Response |
| 7 | 2.963651350 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 360 Create Request File: ;Find Request SMB2_FIND_POSIX_INFO Pattern: * |
| 8 | 2.969718541 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 1846 Create Response File: ;Find Response |
| 9 | 2.969839395 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 478 Create Request File: symlinktofile;GetInfo Request FILE_INFO/SMB2_FILE_POSIX_INFO;Close Request |
| 10 | 2.969940433 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 358 Create Response, Error: STATUS_STOPPED_ON_SYMLINK;GetInfo Response, Error: STATUS_STOPPED_ON_SYMLINK |
| 11 | 2.970035617 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 478 Create Request File: symlinktofile;GetInfo Request FILE_INFO/SMB2_FILE_POSIX_INFO;Close Request |
| 12 | 2.970447539 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 670 Create Response File: symlinktofile;GetInfo Response;Close Response |
| 13 | 2.970545464 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 478 Create Request File: symlinktofile;GetInfo Request FILE_INFO/SMB2_FILE_POSIX_INFO;Close Request |
| 14 | 2.970638451 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 358 Create Response, Error: STATUS_STOPPED_ON_SYMLINK;GetInfo Response, Error: STATUS_STOPPED_ON_SYMLINK |
| 15 | 2.970731608 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 478 Create Request File: symlinktofile;GetInfo Request FILE_INFO/SMB2_FILE_POSIX_INFO;Close Request |
| 16 | 2.971128920 | 127.0.0.1 | 127.0.0.1 | SMB2 | | 670 Create Response File: symlinktofile;GetInfo Response;Close Response |

ExtraInfo SMB2_CREATE_QUERY_ON_DISK_ID SMB2_POSIX_CREATE_CONTEXT

SMB2 (Server Message Block Protocol version 2)

- SMB2 Header
 - GetInfo Response (0x10)
 - [Class: FILE_INFO (0x01)]
 - [InfoLevel: SMB2_FILE_POSIX_INFO (0x64)]
 - StructureSize: 0x0009
 - Blob Offset: 0x00000048
 - Blob Length: 112
 - Create: Sep 18, 2023 00:31:32.435798000 CDT
 - Last Access: Sep 18, 2023 00:31:44.959949900 CDT
 - Last Write: Sep 18, 2023 00:31:32.435798000 CDT
 - Last Change: Sep 18, 2023 00:31:32.435798000 CDT
 - Allocation Size: 4
 - End Of File: 4
 - File Attributes: 0x00000400
 - Inode: 0x0000000063d82f2a
 - File Id: 0x000000000010302
 - Reserved: 00000000
 - Number of Links: 1
 - Reparse Tag: REPARSE_TAG_RESERVED_ZERO (0x00000000)
 - POSIX perms: 0777
 - Owner SID: S-1-22-1-0
 - Group SID: S-1-22-2-0

5.19 kernel (July 31st, 2022)

- Important performance optimization for directory searches, now we cache root directory content (to the many servers which support dir leases) reducing amount of network traffic for queries of root directory
- Multichannel reconnect improvements (e.g. when address or interfaces change)
- Conversion begins for cifs.ko to use new mm layer design
read_folio/release_folio
- RDMA (smbdirect) improvements
- New mount parm “nospars” to optionally disable use of sparse files

6.0 kernel (October 2nd, 2022) (cifs module version 2.39)

- Fallocate improvements (insert and collapse range)
- Module size shrunk significantly when SMB1/CIFS (insecure legacy) disabled
- New mount parm “clostimeo” allows extending deferred closes (handle leases) longer or even disabling the feature (and default increased to 5 seconds from 1 sec)
- Important deferred close fix
- Multichannel perf (locking) improvements

6.1 kernel (Dec 11th, 2022) (cifs module ver: 2.40)

- Performance improvement for path revalidation (metadata ops perf better) by using cached dentry for subdirectory if lease held on it
 - Expanding cached directories to include subdirectories (thanks Ronnie!)
- New ioctl for change notify added that returns the name(s) of any changed files in the directory (not just that the directory has changed)
 - e.g. so app can do their own offline caching of files and sync with server
- Improve symlink handling (avoid an extra roundtrip when symlink detected via STOPPED_ON_SYMLINK message)
- RDMA (smbdirect) improvements (thanks Tom Talpey and Metze!)

6.2 kernel (February 19th, 2023) (cifs module 2.41)

- Important SMB3.1.1 POSIX extensions improvement (parse owner and group SIDs to improve stat output)
- DFS performance improvements (reducing roundtrips) and DFS fixes
- Multichannel and reconnect and DFS improvements
- Integration with the new kernel page caching infrastructure, folios (e.g. migrate_folios support), iov_iter and memory management layering cleanup (e.g. deprecated writepage/writepages API removed)

6.3 kernel (April 23rd, 2023) (cifs module 2.42) very active release!

- Kernel idmapping improvements
- Improvements to use folios (better mm integration and cached writes)
- RDMA (smbdirect) improvements (thanks Metze and David)
- Many multichannel improvements (including using least loaded channel for sending I/O, and improvements for reconnect). Thanks Shyam!
- Various DFS fixes
- Lower default deferred close timeout

6.4 kernel (June 2024) (cifs.ko version: 2.43)

- Important deferred close (lease break corner case) fixes
- Reconnect and DFS fixes
- Important crediting improvements expected
- Compounding improvements expected

6.5 kernel (August 2024) (cifs.ko version: 2.44)

- Deferred close perf improvement (avoid unneeded lease break acks)
- Crediting (flow control) improvements to avoid low credit perf issue
- Reconnect and DFS fixes
- Fix null auth (sec=none) regression
- Allow dumping decryption keys (eg for reading network traces) via directory name, it just file.
- Display client GUID and network namespace in `/proc/fs/cifs/DebugData`

6.6-rc kernel (expected Nov. 2024) (cifs.ko version: 2.45)

- DFS (global namespace) fixes
- Improvements handling reparse points.
- Perf improvement for querying reparse point symlinks
- Reconnect improvement (write retry with channel sequence number)
- Add new mount parm "max_cached_dirs" to control how many directories are cached when server supports directory leases.
- Add new module parm /sys/module/cifs/parameters/dir_cache_timeout to control length of time a directory is cached with directory leases

Debugging improvements

- New dynamic tracepoints
- New debugging scripts leveraging eBPF

Features we are working on for the coming releases

Section Subtitle

What features can you expect in next few releases?

- Analyze cases where use of directory leases could better optimize network traffic while caching safely (caching dirents and dir metadata)
- Add use of compounding in more cases being tested in plugfest now (e.g. open/querydir/querydir instead of open/querydir), and better use existing file leases for compound reqs which include SMB3 open
- Continued focus on multichannel performance improvements
- Automatically using multichannel, picking optimal channels for special cases
- SMB3.1.1 compression support (allow compressing network traffic based on the SMB3.1.1 compress mount parm)

What features can you expect in next few releases?

- Improved packet signing (faster GMAC)
- Support for new auth mechanisms (e.g. local KDC being investigated)
- Reenabling support for swapfile over SMB3.1.1 mounts
- Support for creating with O_TMPFILE (being tested at plugfest now)
- Improvements to enable fanotify/inotify over SMB3.1.1 mounts (currently requires a private SMB3.1.1 specific ioctl)
- Prototype of SMB3.1.1 over QUIC (new encrypted network transport)
- More testing of the SMB3.1.1 POSIX with new Samba server support

Other fixes being discussed this week

- Reconnect, retry improvements (missing retry flag when channel sequence number updated)
- Byte range lock sequence numbers for reconnect
- Enabling multichannel by default if srv support and how many channels
- Better use of parent lease key
- Reducing number of unneeded lease breaks by checking for lease key in more cases (e.g. some compounding scenarios)
- How to hold root directory open longer even when no dir lease support

SEPTEMBER 15, 2023

I/O compression in SMB 3.1.1

Brief overview

Enzo Matsumiya <ematsumiya@suse.de>



Summary

Message compression was introduced in SMB 3.1.1 with the goal to reduce the SMB message size on large reads/writes, thus decreasing network load and increasing bandwidth and throughput

MS-SMB2 specifies the usage of the *Xpress Compression Algorithm* to achieve this, which is a set of compression algorithms specified in *MS-XCA*

This overview is about its current implementation (early-alpha-quality code) for Linux cifs.ko

Goals

Keep it simple

- Implement a simple API, generic enough that could, maybe, turn into a standalone lib
 - the compressor object has only the algo, compress mode(*), and the compress/decompress function pointers
- API design makes it plug & play for any of the *MS-XCA* algos
 - currently only *Pattern_v1* and *LZ77+Huffman* are implemented (others are *LZNT1* and *LZ77*)
- Minimally intrusive in the rest of the code

Goals

Efficient

- Compression: trade-off between CPU/memory usage and compressed size
 - low trade-off ratio → efficient
- To achieve this: compress only write requests larger than 4096 bytes (most recent Windows Servers does the same, as far as I could confirm)
- This implementation has both “data” and “full” compression modes (*MS-SMB2 3.1.4.4, item 2-3*):
 - “data” mode compresses only the write data (SMB2 header stays uncompressed)
 - “full” mode compresses everything

** further benchmarks are still needed, but the difference in payload size (see **Current issues** though) and resource usage are negligible*

Goals

Performance

... and of course, the end goal is to improve I/O performance on cifs.ko mounts

- Hard to give exact numbers, since uncompressed data layout has a huge impact
- Microsoft's demo (*) (Windows client and server) shows:

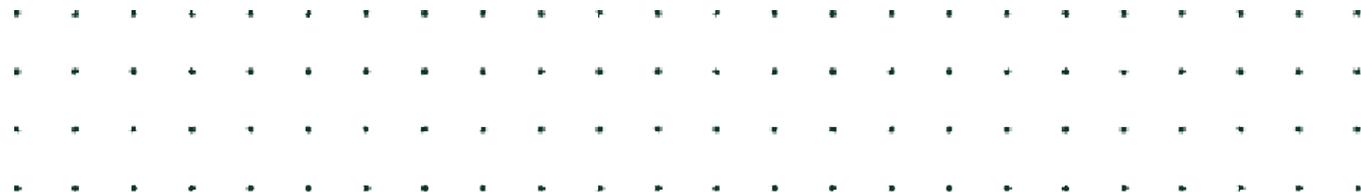
| 20GB file transfer | Uncompressed | Compressed |
|--------------------|--------------|------------|
| Time | 2 min 43 sec | 28 sec |
| Throughput | ~123 Mbps | ~730 Mbps |
| Bandwidth | 1 Gbps | 1.5 Mbps |
| CPU usage | 5-10% | 40-50% |

* <https://learn.microsoft.com/en-us/windows-server/storage/file-server/smb-compression?tabs=powershell%2Crobocopy%2Cgroup-policy>

Goals

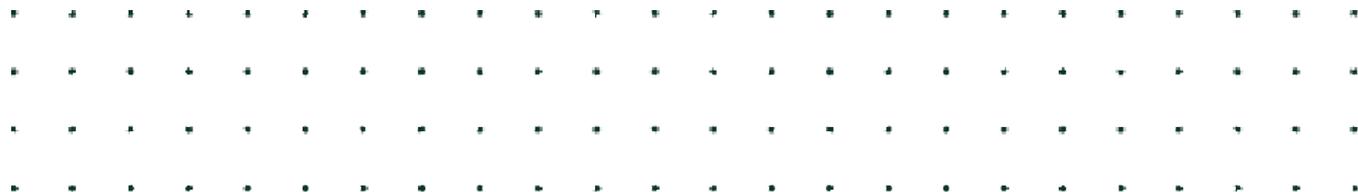
Performance (cont.)

- So, compressed SMB2 I/O is “about 6x faster”
- No reason similar results could not be achieved on Linux
 - unless Windows does extra tricks to achieve those



Current issues

- LZ77+Huffman uses about ~100k bytes (for hashtables and to build the Huffman tree) to compress each block. In this very early stage, to make it work, I'm still kvmalloc'ing them, but I already have a PoC using rbtrees that could cut that in half (most of that memory is used for sorting/ordered access, and that comes for free-ish with rbtrees)
- It also has a fixed 260 bytes overhead for each compressed block, and it requires the input to be at least that big. But sometimes, with an input just a bit larger than that, it's possible that the final compressed data is actually bigger than the input. That's why compressing the whole messages doesn't make much sense, IMHO, as it adds unnecessary overhead compressing the SMB2 header, with little final benefit
- Decompression (large reads) doesn't work yet
- Lots of bugs to fix still, and a lot of improvement to make, but an RFC should appear on CIFS mailing list soon



Thank you.

Feel free to email me any questions or feedback.

ematsumiya@suse.de



SMBDIRECT Transport Improvements: “RDMA for the World”

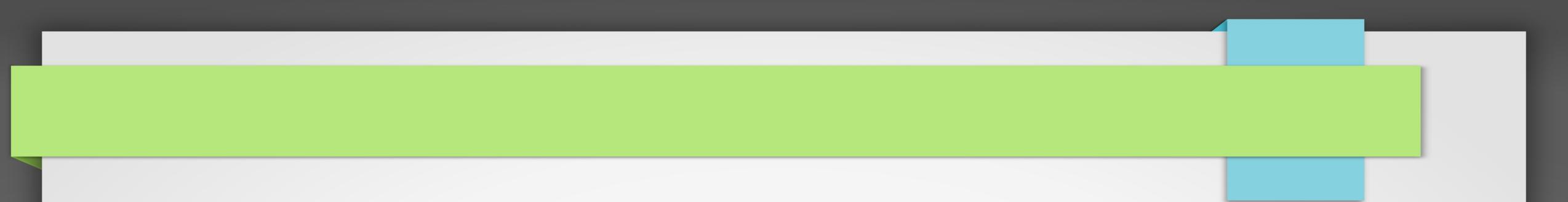
- As discussed last year... SMBDIRECT is an abstraction layer (“framing layer”) for making RDMA useable more broadly. Has no SMB dependencies (SMB3 was just the first consumer of this generic transport layer, but it applies more broadly)
- Longer term plan is to:
 - Bring common from cifs.ko and ksmbd for RDMA into smbdirect.ko
 - Enable user space access to RDMA through smbdirect.ko so user space applications can benefit from the performance gains of RDMA
 - Improvements to this common module will benefit both client and server (and userspace)

SMBDIRECT Transport Improvements: “RDMA for the World”

- smbdirect.ko will provide
- PF_SMBDIRECT sockets
- Send message and receive message will get MSG_OOB messages for read and write offload, greatly improving performance and reducing CPU overhead
- (SMB independent) “echo server client” smbdirect tests under development to improve regression tests without requiring SMB
- Thanks to Metze for this work. Feedback welcome

Some SMBDIRECT recommendations from Metze and Tom Talpey from last year are still relevant

- Reduce SGE usage, and decrease maximum fragment size
- Needless memory usage, High SGE usage impacts performance
- Fix RDMA “responder resources”, which do NOT apply to RDMA Writes
- Significant performance limiter for bulk reads
- Fix sends to not wait for completion before returning. Stalls pipeline, context switching
- Use RDMA post-multiple to improve compound send efficiency
- Ensure packet kmem cache optimal packing (3x1364 == 4092)
- Review protocol parsing and state validation
- E.g. ksmbd allows renegotiate (?), reassembles oversize segments (?)
- Merge the two implementations: fs/cifs/smbdirect.[ch] and fs/ksmbd/transport_rdma.[ch]
- Either refactor and merge, or consider metze’s alternative “smbdirect socket” driver



Testing Improvements

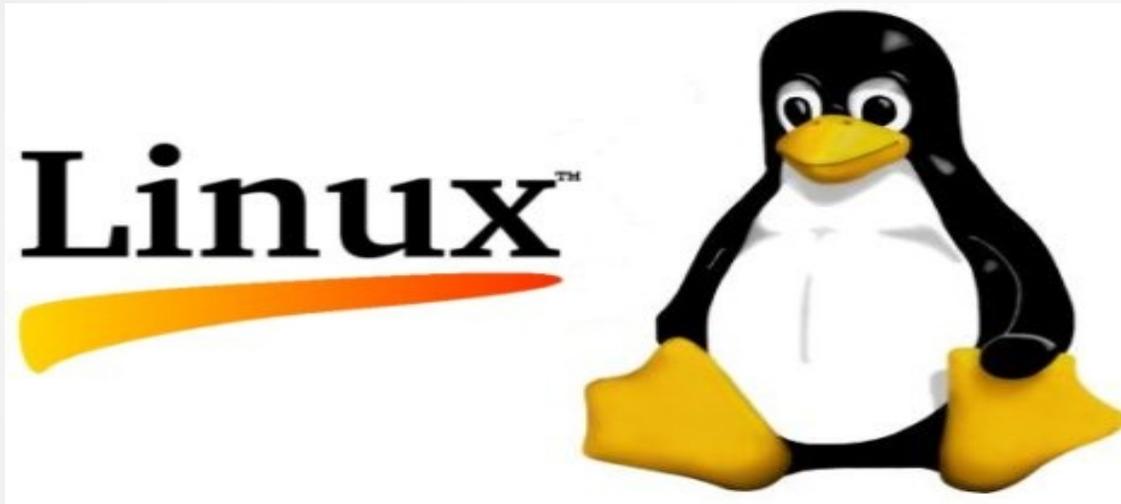
Test ... test ... test ...

Additional tests are encouraged (generic or smb specific)

- Xfstests are the standard Linux filesystem functional tests
- “Buildbot” migrated to new host, and is being modified (down this week)
- Over last 18 months added 21 to the main “cifs-testing” regression testing group (up to 245 tests run on every checkin from this group)
- Various server specific groups have added even more
 - Azure SMB3.1.1 multichannel: up 25% more tests, now includes 133 tests
 - Ksmbd (Linux kernel server target) up 15%, now includes 144 tests
- Detailed wiki pages on wiki.samba.org go through how to setup xfstests with cifs.ko, and what features need to be added to enable more tests (tests that currently skip or fail so aren't run in the ‘buildbot’)

Thank you for your time

- Future is very bright!



+

S
M
B
3

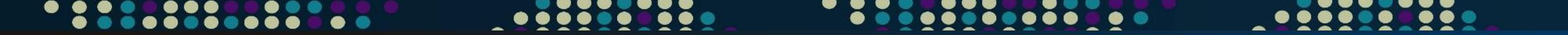
Additional Resources to Explore for SMB3 and Linux

- } <https://msdn.microsoft.com/en-us/library/gg685446.aspx>
 - In particular MS-SMB2.pdf at <https://msdn.microsoft.com/en-us/library/cc246482.aspx>
- } <https://wiki.samba.org/index.php/Xfstesting-cifs>
- } 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/smb>
 - 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>
 - Kernel server code: <https://git.samba.org/ksmbd.git/?p=ksmbd.git> (ksmbd-for-next branch)



Please take a moment to rate this session.

Your feedback is important to us.



Section Title

Section Subtitle



Section Title

Section Subtitle

Light Slide Title

- Bullets 1
 - Bullets 2
 - Bullets 3
 - Bullets 4
 - Bullets 5

Dark Slide Title

- Bullets 1
 - Bullets 2
 - Bullets 3
 - Bullets 4
 - Bullets 5