

# multichannel / io\_uring

Status Update within Samba

Stefan Metzmacher <metze@samba.org>

Samba Team / SerNet

2021-09-28

<https://samba.org/~metze/presentations/2021/SDC/>

## Topics

- ▶ What is SMB3 Multichannel?
- ▶ Updates in Samba 4.15
- ▶ What is io-uring?
- ▶ io-uring for Samba
- ▶ Performance research, prototyping and ideas
- ▶ Questions? Feedback!

# What is SMB3 Multichannel?

- ▶ Multiple transport connections are bound to one logical connection
  - ▶ This allows using more than one network link
    - ▶ Good for performance
    - ▶ Good for availability reasons
  - ▶ Non TCP transports like RDMA (InfiniBand, RoCE, iWarp)
- ▶ All transport connections (channels) share the same ClientGUID
  - ▶ This is important for Samba
- ▶ An authenticated binding is done at the user session layer
  - ▶ SessionID, TreeID and FileID values are valid on all channels
- ▶ Available network interfaces are auto-negotiated
  - ▶ FSCTL\_QUERY\_NETWORK\_INTERFACE\_INFO interface list
  - ▶ IP (v4 or v6) addresses are returned together with:
    - ▶ Interface Index (which addresses belong to the same hardware)
    - ▶ Link speed
    - ▶ RSS and RDMA capabilities

SDC<sup>21</sup>

SAMBA<sup>+</sup>

Stefan Metzmacher

multichannel / io\_uring  
(3/21)

SerNet

## Last Status Updates (SDC 2020 / SambaXP 2021)

- ▶ I gave a similar talk at the storage developer conference 2020:
  - ▶ See <https://samba.org/~metze/presentations/2020/SDC/>
  - ▶ It explains the milestones and design up to Samba 4.13 (in detail)
- ▶ I gave a similar talk at the SambaXP 2021:
  - ▶ See <https://samba.org/~metze/presentations/2021/SambaXP/>
  - ▶ It explains the milestones and updates up to Samba 4.15 (in detail)

SDC<sup>21</sup>

SAMBA<sup>+</sup>

Stefan Metzmacher

multichannel / io\_uring  
(4/21)

SerNet

- ▶ Automated regression tests are in place:
  - ▶ socket\_wrapper got basic fd-passing support (Bug #11899)
  - ▶ We added a lot more multichannel related regression tests
- ▶ The last missing features/bugs are fixed (Bug #14524)
  - ▶ The connection passing is fire and forget (Bug #14433)
  - ▶ Pending async operations are canceled (Bug #14449)
- ▶ 4.15 finally has "server multi channel support = yes"
  - ▶ We require support for TIOCOUTQ (Linux) or FIONWRITE (FreeBSD)
  - ▶ We disable multichannel feature if the platform doesn't support this
    - ▶ See: Retries of Lease/Oplock Break Notifications (Bug #11898)

## What is io-uring? (Part 1)

- ▶ Linux 5.1 introduced a new scalable AIO infrastructure
  - ▶ It's designed to avoid syscalls as much as possible
  - ▶ kernel and userspace share mmap'ed rings:
    - ▶ submission queue (SQ) ring buffer
    - ▶ completion queue (CQ) ring buffer
  - ▶ See "[Ring in a new asynchronous I/O API](#)" on LWN.NET
- ▶ This can be nicely integrated with our async tevent model
  - ▶ It may delegate work to kernel threads
  - ▶ It seems to perform better compared to our userspace threadpool
  - ▶ It can also inline non-blocking operations

- ▶ Between userspace and filesystem (available from 5.1):
  - ▶ IORING\_OP\_READV, IORING\_OP\_WRITEV and IORING\_OP\_FSYNC
  - ▶ Supports buffered and direct io
- ▶ Between userspace and socket (and also filesystem) (from 5.8)
  - ▶ IORING\_OP\_SENDMSG, IORING\_OP\_RECVMSG
  - ▶ Improved MSG\_WAITALL support (5.12, backported to 5.11, 5.10)
  - ▶ IORING\_OP\_SPLICE, IORING\_OP\_TEE
  - ▶ Maybe using IORING\_SETUP\_SQPOLL or IOSQE\_ASYNC
- ▶ Path based syscalls with async impersonation (from 5.6)
  - ▶ IORING\_OP\_OPENAT2, IORING\_OP\_STATX
  - ▶ Using IORING\_REGISTER\_PERSONALITY for impersonation
  - ▶ IORING\_OP\_UNLINKAT, IORING\_OP\_RENAMEAT (from 5.10)
  - ▶ IORING\_OP\_MKDIRAT, IORING\_OP\_SYMLINKAT, IORING\_OP\_LINKAT (from 5.15)

### IORING\_FEAT\_NATIVE\_WORKERS (from 5.12)

- ▶ In the kernel...
  - ▶ The io-uring kernel threads are clone()'ed from the userspace thread
  - ▶ They just appear to be blocked in a syscall and never return
  - ▶ This makes the accounting in the kernel much saner
  - ▶ Allows a lot of restrictions to be relaxed in the kernel
- ▶ For admins and userspace developers...
  - ▶ They are no longer 'io\_wqe\_work' kernel threads
  - ▶ 'top' shows them as part of the userspace process ('H' shows them)
  - ▶ They are now visible in containers
  - ▶ 'pstree -a -t -p' is very useful to see them
  - ▶ They are shown as iou-wrk-1234, for a task with pid/tid 1234

- ▶ With Samba 4.12 we added "io\_uring" vfs module
  - ▶ For now it only implements SMB\_VFS\_PREAD,PWRITE,FSYNC\_SEND/RECV
  - ▶ It has less overhead than our pthreadpool default implementations
  - ▶ I was able to speed up a smbclient 'get largefile /dev/null'
    - ▶ Using against smbd on loopback
    - ▶ The speed changes from 2.2GBytes/s to 2.7GBytes/s
- ▶ The improvement only happens by avoiding context switches
  - ▶ But the data copying still happens:
    - ▶ From/to a userspace buffer to/from the filesystem/page cache
  - ▶ The data path between userspace and socket is completely unchanged
  - ▶ For both cases the cpu is mostly busy with memcpy

## Performance research (SMB2 Read)

- ▶ In October 2020 I was able to do some performance research
  - ▶ With 100Gbit/s interfaces and two NUMA nodes per server.
- ▶ At that time I focussed on the SMB2 Read performance only
  - ▶ We had limited time on the given hardware
  - ▶ We mainly tested with fio.exe on a Windows client
  - ▶ Linux kernel 5.8.12 on the server
- ▶ More verbose details can be found here:
  - ▶ <https://lists.samba.org/archive/samba-technical/2020-October/135856.html>



# IORING\_OP\_SENDMSG (Part 2)

The results vary heavily depending on the NUMA bouncing, between 5.0 GBytes/s and 7.6 GBytes/s

Monitoring 383 processes and 825 threads (interval: 5.0s)

| PID    | Process       | Mem(K)  | IO(MB/s) | Mem % | CPU % | Private |
|--------|---------------|---------|----------|-------|-------|---------|
| 107538 | smbd          | 75.2    | 207316.8 | 0.0   | 5.88  | 2.4     |
| 107502 | io_uring_work | 12012.0 | 37483.2  | 0.3   | 3.97  | 0.7     |
| 107503 | io_uring_work | 10153.0 | 60312.4  | 0.2   | 5.28  | 0.7     |
| 107505 | io_uring_work | 5.0     | 50152.0  | 0.0   | 5.43  | 0.0     |
| 107533 | io_uring_work | 10084.0 | 21232.8  | 0.3   | 5.00  | 0.6     |
| 107578 | io_uring_work | 20.0    | 14615.0  | 0.0   | 3.73  | 0.2     |
| 104171 | hammer/33     | 0.0     | 10.0     | 0.0   | 2.73  | 0.0     |
| 104172 | hammer/22     | 0.0     | 10.0     | 0.0   | 2.73  | 0.0     |
| 107567 | io_uring_work | 0.3     | 775.0    | 0.0   | 5.95  | 0.0     |
| 107568 | hammer/1      | 11.1    | 28.2     | 0.4   | 8.43  | 0.0     |
| 107182 | hammer/10     | 0.0     | 11.3     | 0.1   | 2.20  | 0.0     |
| 107183 | hammer/11     | 0.0     | 11.3     | 0.1   | 2.20  | 0.0     |
| 107184 | hammer/12     | 0.0     | 11.3     | 0.1   | 2.20  | 0.0     |
| 107185 | hammer/13     | 0.1     | 1.6      | 0.1   | 1.90  | 0.0     |
| 107382 | hammer/21     | 0.0     | 10.0     | 0.0   | 5.00  | 0.0     |
| 104055 | hammer/23     | 0.0     | 28.0     | 0.0   | 2.13  | 0.0     |
| 107339 | hammer/27     | 0.0     | 10.0     | 0.0   | 5.57  | 0.0     |
| 2      | system        | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 3      | lsmbd         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 4      | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 5      | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 6      | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 10     | smc           | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 11     | smc           | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 12     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 13     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 14     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 15     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 16     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 17     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 18     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 19     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 20     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 21     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 22     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 23     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 24     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 25     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 26     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 27     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 28     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 29     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 30     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 31     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 32     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 33     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 34     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 35     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 36     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 37     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 38     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 39     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 40     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 41     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 42     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 43     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 44     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 45     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 46     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 47     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 48     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 49     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 50     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 51     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 52     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 53     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 54     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 55     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 56     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 57     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 58     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 59     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 60     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 61     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 62     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 63     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 64     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 65     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 66     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 67     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 68     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 69     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 70     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 71     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 72     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 73     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 74     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 75     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 76     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 77     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 78     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 79     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 80     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 81     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 82     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 83     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 84     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 85     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 86     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 87     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 88     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 89     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 90     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 91     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 92     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 93     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 94     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 95     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 96     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 97     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 98     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 99     | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |
| 100    | rsync         | 0.0     | 0.0      | 0.0   | 0.00  | 0.0     |

System: system, 2 kbthread, 3 rsync, 4 rsync, 5 rsync, 6 rsync, 10 smc, 11 smc, 12 rsync, 13 rsync, 14 rsync, 15 rsync, 16 rsync, 17 rsync, 18 rsync, 19 rsync, 20 rsync, 21 rsync, 22 rsync, 23 rsync, 24 rsync, 25 rsync, 26 rsync, 27 rsync, 28 rsync, 29 rsync, 30 rsync, 31 rsync, 32 rsync, 33 rsync, 34 rsync, 35 rsync, 36 rsync, 37 rsync, 38 rsync, 39 rsync, 40 rsync, 41 rsync, 42 rsync, 43 rsync, 44 rsync, 45 rsync, 46 rsync, 47 rsync, 48 rsync, 49 rsync, 50 rsync, 51 rsync, 52 rsync, 53 rsync, 54 rsync, 55 rsync, 56 rsync, 57 rsync, 58 rsync, 59 rsync, 60 rsync, 61 rsync, 62 rsync, 63 rsync, 64 rsync, 65 rsync, 66 rsync, 67 rsync, 68 rsync, 69 rsync, 70 rsync, 71 rsync, 72 rsync, 73 rsync, 74 rsync, 75 rsync, 76 rsync, 77 rsync, 78 rsync, 79 rsync, 80 rsync, 81 rsync, 82 rsync, 83 rsync, 84 rsync, 85 rsync, 86 rsync, 87 rsync, 88 rsync, 89 rsync, 90 rsync, 91 rsync, 92 rsync, 93 rsync, 94 rsync, 95 rsync, 96 rsync, 97 rsync, 98 rsync, 99 rsync, 100 rsync.

Resource Monitor: Ethernet, 16.8 Mbps, 56.7 Gbps

SDC21 SAMBA+

Stefan Metzmacher

multichannel / io\_uring (13/21)

SerNet

# IORING\_OP\_SENDMSG (Part 3)

The major problem still exists, memory copy done by copy\_user\_enhanced\_fast\_string()

overhead: 178k of event 'cycles', 4000 Hz, event count (approx.): 8738159677 lost: 0/0 d

| overhead | shared object | Symbol                               |
|----------|---------------|--------------------------------------|
| 0.93%    | [kernel]      | [k] copy_user_enhanced_fast_string   |
| 0.90%    | [kernel]      | [k] show_file_read_iter              |
| 0.78%    | [kernel]      | [k] tcp_sendmsg_locked               |
| 0.76%    | [kernel]      | [k] find_get_entry                   |
| 0.75%    | [kernel]      | [k] get_page_from_freelist           |
| 0.74%    | [kernel]      | [k] list_del_entry_valid             |
| 0.69%    | [kernel]      | [k] native_queued_spin_lock_slowpath |
| 0.68%    | [kernel]      | [k] zone_spin_lock                   |
| 0.67%    | [kernel]      | [k] skb_release_data                 |
| 0.66%    | [kernel]      | [k] skx5e_sq_wait                    |
| 0.58%    | [kernel]      | [k] __free_pages_ok                  |
| 0.57%    | [kernel]      | [k] zone_spin_lock_irqsave           |
| 0.55%    | [kernel]      | [k] zone_watermark_ok                |
| 0.53%    | [kernel]      | [k] unlock_page                      |
| 0.52%    | [kernel]      | [k] copy_page_to_iter                |
| 0.51%    | [kernel]      | [k] find_get_entry                   |
| 0.51%    | [kernel]      | [k] __alloc_pages_node               |
| 0.51%    | [kernel]      | [k] skx5e_poll_tx_cq                 |
| 0.49%    | [kernel]      | [k] page_mapping                     |
| 0.48%    | [kernel]      | [k] sas_load                         |
| 0.47%    | [kernel]      | [k] shoox_getpage_gfp                |
| 0.47%    | [kernel]      | [k] __chop_object_size               |
| 0.47%    | [kernel]      | [k] tcp_wfree                        |
| 0.46%    | [kernel]      | [k] slab_free                        |
| 0.45%    | [kernel]      | [k] __sched_text_start               |
| 0.45%    | [kernel]      | [k] __free_one_page                  |
| 0.45%    | [kernel]      | [k] aark_page_accessed               |
| 0.45%    | [kernel]      | [k] bad_range                        |
| 0.44%    | [kernel]      | [k] tcp_wfree                        |
| 0.44%    | [kernel]      | [k] list_iter_advance                |
| 0.44%    | [kernel]      | [k] napi_irq_return_iret             |
| 0.44%    | [kernel]      | [k] tcp_write_smit                   |
| 0.44%    | [kernel]      | [k] __alloc_skb                      |
| 0.44%    | [kernel]      | [k] tasklet_action_common_isr.0      |
| 0.44%    | [kernel]      | [k] clear_page_errs                  |
| 0.44%    | [kernel]      | [k] do_syscall_64                    |
| 0.44%    | [kernel]      | [k] __transport_skb                  |
| 0.44%    | [kernel]      | [k] __skb_clone                      |
| 0.44%    | [kernel]      | [k] asocpy_errs                      |
| 0.44%    | [kernel]      | [k] memu_unlock                      |
| 0.44%    | [kernel]      | [k] __list_add_valid                 |
| 0.44%    | [kernel]      | [k] skx5e_sq_comp_init               |
| 0.44%    | [kernel]      | [k] tcp_ack                          |

System: system, 2 kbthread, 3 rsync, 4 rsync, 5 rsync, 6 rsync, 10 smc, 11 smc, 12 rsync, 13 rsync, 14 rsync, 15 rsync, 16 rsync, 17 rsync, 18 rsync, 19 rsync, 20 rsync, 21 rsync, 22 rsync, 23 rsync, 24 rsync, 25 rsync, 26 rsync, 27 rsync, 28 rsync, 29 rsync, 30 rsync, 31 rsync, 32 rsync, 33 rsync, 34 rsync, 35 rsync, 36 rsync, 37 rsync, 38 rsync, 39 rsync, 40 rsync, 41 rsync, 42 rsync, 43 rsync, 44 rsync, 45 rsync, 46 rsync, 47 rsync, 48 rsync, 49 rsync, 50 rsync, 51 rsync, 52 rsync, 53 rsync, 54 rsync, 55 rsync, 56 rsync, 57 rsync, 58 rsync, 59 rsync, 60 rsync, 61 rsync, 62 rsync, 63 rsync, 64 rsync, 65 rsync, 66 rsync, 67 rsync, 68 rsync, 69 rsync, 70 rsync, 71 rsync, 72 rsync, 73 rsync, 74 rsync, 75 rsync, 76 rsync, 77 rsync, 78 rsync, 79 rsync, 80 rsync, 81 rsync, 82 rsync, 83 rsync, 84 rsync, 85 rsync, 86 rsync, 87 rsync, 88 rsync, 89 rsync, 90 rsync, 91 rsync, 92 rsync, 93 rsync, 94 rsync, 95 rsync, 96 rsync, 97 rsync, 98 rsync, 99 rsync, 100 rsync.

Resource Monitor: Ethernet, 15.7 Mbps R, 57.5 Gbps

SDC21 SAMBA+

Stefan Metzmacher

multichannel / io\_uring (14/21)

SerNet







## Improvements for transfers with SMB3 signing

- ▶ Samba 4.15 has support for AES-128-GMAC signing:
  - ▶ This is also available in recent Windows versions
  - ▶ It's based on AES-128-GCM (but only with authentication data)
  - ▶ The gnutls library is able to provide:
    - ▶ ~6 GBytes/s for AES-128-GCM
    - ▶ ~10 GBytes/s for AES-128-GMAC
- ▶ For SMB3 signing/encryption we use:
  - ▶ IORING\_OP\_SPLICE from a file into a (splice)pipe
  - ▶ IORING\_OP\_TEE from the (splice)pipe to a 2nd (tee)pipe
  - ▶ IORING\_OP\_READ from the (tee)pipe into a userspace buffer
    - ▶ (vmsplice might work even better)
  - ▶ The userspace buffer is only used to calculate the signing signature
  - ▶ IORING\_OP\_SENDMSG and IORING\_OP\_SPLICE are used in order to avoid a copy back to the kernel
- ▶ For a SMB2 read test I removed the signing check in smbclient:
  - ▶ The performance changed from ~3 GBytes/s before
  - ▶ To ~5 GBytes/s using the IORING\_OP\_TEE trick
    - ▶ With smbclient still being the bottleneck at 100% cpu

SDC<sup>21</sup>

SAMBA<sup>+</sup>

Stefan Metzmacher

multichannel / io\_uring  
(19/21)

SerNet

## Future Improvements

- ▶ recvmmsg and splice deliver partial SMB packets to userspace
  - ▶ I tested with AF\_KCM (Kernel Connection Multiplexor) and an eBPF helper
  - ▶ But MSG\_WAITALL is the much simpler and faster solution
  - ▶ I also prototyped a SPLICE.F\_WAITALL
  - ▶ eBPF support in io-uring would also be great for optimizations
- ▶ It also seems that socket->pipe->filesystem:
  - ▶ Does not implement zero copy for all cases
  - ▶ Maybe it's possible to optimize this in future
- ▶ In the end SMB-Direct will also be able to reduce overhead
  - ▶ My smbdirect driver is still work in progress...
  - ▶ With the IORING\_FEAT\_NATIVE\_WORKERS feature it will be possible glue it to IORING\_OP\_SENDMSG

SDC<sup>21</sup>

SAMBA<sup>+</sup>

Stefan Metzmacher

multichannel / io\_uring  
(20/21)

SerNet

- ▶ Feedback regarding real world testing would be great!
- ▶ Stefan Metzmacher, [metze@samba.org](mailto:metze@samba.org)
- ▶ <https://www.sernet.com>
- ▶ <https://samba.plus>

Slides: <https://samba.org/~metze/presentations/2021/SDC/>



Stefan Metzmacher

multichannel / io\_uring  
(21/21)

SerNet