

multichannel / io_uring

Status Update within Samba

Stefan Metzmacher <metze@samba.org>

Samba Team / SerNet

2021-05-05

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

Check for Updates

- ▶ Check for an updated version of this presentation here:
- ▶ <https://samba.org/~metze/presentations/2021/SambaXP/>

(draft)

- ▶ 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? (Part 1)

- ▶ 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

What is SMB3 Multichannel? (Part 2)

- ▶ IO ordering is important for multichannel
 - ▶ Requests can get lost between client and server
 - ▶ Responses can get lost between server and client
 - ▶ The client isn't able to know the difference
 - ▶ Replays contain the REPLAY flag in the SMB2 header
 - ▶ FILE_NOT_AVAILABLE indicates "please retry" to the client
 - ▶ Windows returns ACCESS_DENIED in some cases instead
 - ▶ In other cases Windows ignores a replay and deadlocks the client
 - ▶ I need to discuss this with Microsoft
 - ▶ See: Samba Bug #14449
- ▶ State changing operations need replay detection
 - ▶ They need to execute only-once
 - ▶ SMB2 Create uses a CreateGUID
 - ▶ SMB2 Lock uses an array with sequence numbers
 - ▶ Windows only supports this on resilient and persistent handles
 - ▶ Future Windows versions are supposed to fix that

What is SMB3 Multichannel? (Part 3)

- ▶ Write/Set operations only need a barrier
 - ▶ An epoch number is incremented on each channel failure
 - ▶ The current epoch number is part of each request
 - ▶ The server remembers the last seen epoch number
 - ▶ Non-REPLAY requests with stale epoch fail
 - ▶ REPLAY requests fail, when there are pending older epoch numbers
- ▶ Read/Get operations can be replayed safely
- ▶ Lease/Oplock break notifications should be retried
 - ▶ Break notifications wait for transport acks
 - ▶ On channel failures they are retried on other channels
 - ▶ Windows doesn't retry for oplocks, only leases

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

Updates in Samba 4.15

- ▶ 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 will hopefully have "server multi channel support = yes"
 - ▶ Currently it's still off by default, but may change before 4.15.0rc1
 - ▶ 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)
- ▶ I have unofficial backport for older branches
 - ▶ SerNet's SAMBA+ 4.14 includes the patches
 - ▶ "server multi channel support = no" is still the default

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 "[Ringing 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

io-uring for Samba (Part 1)

- ▶ 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, backport 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_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
 - ▶ Most likely to be backported to the 5.10 LTS kernel
- ▶ For admins and userspace developers...
 - ▶ '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
 - ▶ gdb may show worrying messages:
 - ▶ "warning: Architecture rejected target-supplied description"
 - ▶ But it seems they can be ignored and will be fixed soon

Performance research (SMB2 Read)

- ▶ Last October I was able to do some performance research
 - ▶ DDN was so kind to sponsor about a week of research on real world hardware
 - ▶ With 100GBit/s interfaces and two NUMA nodes per server.
- ▶ 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>

Performance with MultiChannel, sendmsg()

4 connections, ~3.8 Gbytes/s, bound by >500% cpu in total, sendmsg() takes up to 0.5 msecs

```
top - 05:51:18 up 2 days, 46 min, 2 users, load average: 5.42, 3.22, 1.52
threads: 823 total, 3 running, 800 sleeping, 0 stopped, 0 zombie
MemInfo: 8.8 Mb, 8.5 Mb, 8.8 Mb, 96.8 Mb, 8.8 Mb, 9.1 Mb, 8.2 Mb, 8.8 Mb
HiB Mem : 193824.1 total, 18229.6 free, 2817.5 used, 8276.3 buff/cache
Mem: 1824.0 total, 1824.0 free, 0 used, 185554.7 avail Mem
```

PID	USER	PR	NI	VT	RES	SHR	S	CPU	MEM	TIME+	COMMAND	
387378	root	20	0	0	0	0	0	0	0	0:05.40	io_wqe_worker-0	
387549	root	20	0	0	0	0	0	0	0	0:21.39	io_wqe_worker-0	
387550	root	20	0	0	0	0	0	0	0	0:21.45	io_wqe_worker-0	
387587	root	20	0	0	0	0	0	0	0	0:09.92	io_wqe_worker-1	
387554	root	20	0	0	663180	144024	10804	5	23.2	0.1	0:49.13	smbd
387556	root	20	0	0	663180	144024	10804	5	19.9	0.1	0:49.95	smbd
387559	root	20	0	0	663180	144024	10804	5	18.5	0.1	0:49.92	smbd
387563	root	20	0	0	663180	144024	10804	5	19.5	0.1	0:49.30	smbd
387557	root	20	0	0	663180	144024	10804	5	19.2	0.1	0:49.13	smbd
387560	root	20	0	0	663180	144024	10804	5	19.2	0.1	0:49.97	smbd
387561	root	20	0	0	663180	144024	10804	5	19.2	0.1	0:49.97	smbd
387534	root	20	0	0	663180	144024	10804	5	18.9	0.1	0:49.90	smbd
387576	root	20	0	0	663180	144024	10804	5	18.9	0.1	0:49.41	smbd
387562	root	20	0	0	663180	144024	10804	5	18.5	0.1	0:49.92	smbd
387538	root	20	0	0	663180	144024	10804	0	11.3	0.1	0:49.16	smbd
387552	root	20	0	0	0	0	0	0	0	0:12.25	io_wqe_worker-0	
437	root	20	0	0	0	0	0	0	0	0:03.18	ksmworker/0-2-resv	
387183	root	20	0	0	0	0	0	0	0	0:00.41	ksmworker/180-2-evnt	
387568	root	20	0	0	0	0	0	0	0	0:00.42	ksmworker/29-0-evnt	
387588	root	20	0	0	62964	5532	3998	0	0.3	0.0	0:00.12	top
1	root	0	0	0	24512	18952	6176	0	0	0	0:02.85	systemd
2	root	0	0	0	0	0	0	0	0	0	0:00.13	kthreadd
3	root	0	-20	0	0	0	0	0	0	0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	0	0	0	0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	0	0	0	0	0:00.00	ksmworker/0-0-ksm
10	root	0	-20	0	0	0	0	0	0	0	0:00.00	mm_percpu_wq
11	root	0	0	0	0	0	0	0	0	0	0:00.32	ksoftirq/0
12	root	0	0	0	0	0	0	0	0	0	0:00.32	ksoftirq/0
13	root	rt	0	0	0	0	0	0	0	0	0:00.43	migration/0
14	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/0
15	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/2
16	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/2
17	root	0	0	0	0	0	0	0	0	0	0:00.47	migration/1
19	root	0	-20	0	0	0	0	0	0	0	0:00.40	ksmworker/1-00-ksm
21	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/2
22	root	rt	0	0	0	0	0	0	0	0	0:01.37	migration/2
23	root	0	0	0	0	0	0	0	0	0	0:00.41	ksoftirq/2
24	root	0	-20	0	0	0	0	0	0	0	0:00.40	ksmworker/2-00-ksm
26	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/2
27	root	rt	0	0	0	0	0	0	0	0	0:01.39	migration/3

Administrator Windows PowerShell

```
cmd.exe /s /c netsh interface portx subport 0 add name=ethernet0
```

Run status group # (all jobs):

```
cmd.exe /s /c netsh interface portx subport 0 add name=ethernet0
```

Starting 2 threads

```
cmd.exe /s /c netsh interface portx subport 0 add name=ethernet0
```

Task Manager

Process: Performance Usage Details Services

CPU 52.27%

Memory 10510 MB (20%)

Ethernet 31.9 Gbps

Ethernet Mellanox ConnectX-6 Adapter

Throughput

31.9 Gbps

Adapter name: SLOT 4 Port 1

Connection type: Ethernet

IP address: 10.168.0.153

MAC address: N6D5a51555ccca4b510

SMBAA multichannel / io_uring Stefan Metzmacher (13/22) SerNet

IORING OP SENDMSG prototyped (Part 1)

4 connections, ~6.8 Gbytes/s, smbd only uses ~11% cpu, (io_wqe_work ~50% cpu) per connection, we still use >300% cpu in total

```
top - 05:45:38 up 2 days, 46 min, 2 users, load average: 0.03, 2.04, 1.61
threads: 823 total, 3 running, 800 sleeping, 0 stopped, 0 zombie
MemInfo: 8.8 Mb, 8.7 Mb, 8.8 Mb, 96.8 Mb, 8.8 Mb, 9.1 Mb, 8.2 Mb, 8.8 Mb
HiB Mem : 193824.1 total, 182194.6 free, 2782.6 used, 8276.9 buff/cache
Mem: 1824.0 total, 1824.0 free, 0 used, 185554.7 avail Mem
```

PID	USER	PR	NI	VT	RES	SHR	S	CPU	MEM	TIME+	COMMAND	
387577	root	20	0	0	0	0	0	0	0	0:05.40	io_wqe_worker-0	
387549	root	20	0	0	0	0	0	0	0	0:21.39	io_wqe_worker-0	
387550	root	20	0	0	0	0	0	0	0	0:21.45	io_wqe_worker-0	
387587	root	20	0	0	0	0	0	0	0	0:09.92	io_wqe_worker-1	
387554	root	20	0	0	663180	144024	10804	5	23.2	0.1	0:49.13	smbd
387556	root	20	0	0	663180	144024	10804	5	19.9	0.1	0:49.95	smbd
387559	root	20	0	0	663180	144024	10804	5	18.5	0.1	0:49.92	smbd
387563	root	20	0	0	663180	144024	10804	5	19.5	0.1	0:49.30	smbd
387557	root	20	0	0	663180	144024	10804	5	19.2	0.1	0:49.13	smbd
387560	root	20	0	0	663180	144024	10804	5	19.2	0.1	0:49.97	smbd
387561	root	20	0	0	663180	144024	10804	5	19.2	0.1	0:49.97	smbd
387534	root	20	0	0	663180	144024	10804	5	18.9	0.1	0:49.90	smbd
387576	root	20	0	0	663180	144024	10804	5	18.9	0.1	0:49.41	smbd
387562	root	20	0	0	663180	144024	10804	5	18.5	0.1	0:49.92	smbd
387538	root	20	0	0	663180	144024	10804	0	11.3	0.1	0:49.16	smbd
387552	root	20	0	0	0	0	0	0	0	0:12.25	io_wqe_worker-0	
437	root	20	0	0	0	0	0	0	0	0:03.18	ksmworker/0-2-resv	
387183	root	20	0	0	0	0	0	0	0	0:00.41	ksmworker/180-2-evnt	
387568	root	20	0	0	0	0	0	0	0	0:00.42	ksmworker/29-0-evnt	
387588	root	20	0	0	62964	5532	3998	0	0.3	0.0	0:00.12	top
1	root	0	0	0	24512	18952	6176	0	0	0	0:02.85	systemd
2	root	0	0	0	0	0	0	0	0	0	0:00.13	kthreadd
3	root	0	-20	0	0	0	0	0	0	0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	0	0	0	0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	0	0	0	0	0:00.00	ksmworker/0-0-ksm
10	root	0	-20	0	0	0	0	0	0	0	0:00.00	mm_percpu_wq
11	root	0	0	0	0	0	0	0	0	0	0:00.32	ksoftirq/0
12	root	0	0	0	0	0	0	0	0	0	0:00.32	ksoftirq/0
13	root	rt	0	0	0	0	0	0	0	0	0:00.43	migration/0
14	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/0
15	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/2
16	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/2
17	root	0	0	0	0	0	0	0	0	0	0:00.47	migration/1
19	root	0	-20	0	0	0	0	0	0	0	0:00.40	ksmworker/1-00-ksm
21	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/2
22	root	rt	0	0	0	0	0	0	0	0	0:01.37	migration/2
23	root	0	0	0	0	0	0	0	0	0	0:00.41	ksoftirq/2
24	root	0	-20	0	0	0	0	0	0	0	0:00.40	ksmworker/2-00-ksm
26	root	0	0	0	0	0	0	0	0	0	0:00.40	cpuhp/2
27	root	rt	0	0	0	0	0	0	0	0	0:01.39	migration/3

Administrator Windows PowerShell

```
cmd.exe /s /c netsh interface portx subport 0 add name=ethernet0
```

Run status group # (all jobs):

```
cmd.exe /s /c netsh interface portx subport 0 add name=ethernet0
```

Starting 2 threads

```
cmd.exe /s /c netsh interface portx subport 0 add name=ethernet0
```

Task Manager

Process: Performance Usage Details Services

CPU 18.27%

Memory 12712 MB (20%)

Ethernet 57.5 Gbps

Ethernet Mellanox ConnectX-6 Adapter

Throughput

57.5 Gbps

Adapter name: SLOT 4 Port 1

Connection type: Ethernet

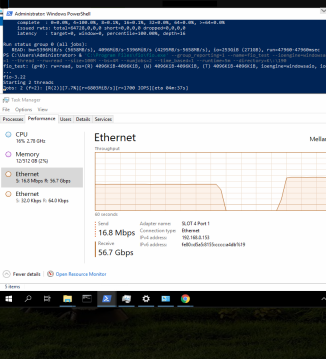
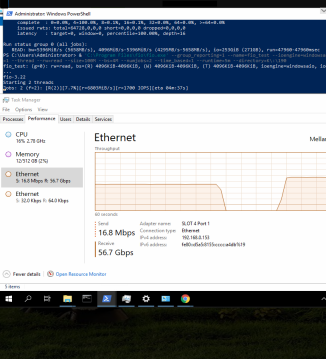
IP address: 10.168.0.153

MAC address: N6D5a51555ccca4b510

SMBAA multichannel / io_uring Stefan Metzmacher (14/22) SerNet

IORING_OP_SENDMSG prototyped (Part2)

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



```
Monitoring 783 processes and 825 threads (Interval: 5.0s)
PID      PPID      PMEM(K)  PMEM(%)  PMEM(GB)  CPU      %CPU     2dP
107032   io_wgk_wor 12812.0  0.0      0.012      0.3      2.77    0.2
107048   io_wgk_wor 18533.3  0.1      0.017      0.2      5.20    0.7
107053   io_wgk_wor 5.0      0.0      0.00012    0.0      0.00    0.0
107053   io_wgk_wor 14668.7  0.1      0.013      0.4      4.78    0.4
107079   io_wgk_wor 28.0     0.0      0.0003     0.0      0.23    0.2
107083   hawker/77  3.0      0.0      0.00003    0.0      0.00    0.0
108171   io_wgk_wor 3.3      0.0      0.00003    0.0      0.03    0.0
107092   io_wgk_wor 72.0     0.0      0.00072    0.0      0.00    0.0
107068   samtop    11.1     0.0      0.00111    0.4      0.69    0.0
107102   hawker/70  0.0      23.2     0.1      2.28    0.0     0.0
107038   hawker/67  0.0      28.0     0.1      2.72    0.0     0.0
107103   hawker/68  0.1      3.0      0.1      3.00    0.0     0.0
107032   hawker/71  0.0      18.0     0.1      1.80    0.0     0.0
106005   hawker/71  0.0      28.0     0.1      2.73    0.8     0.0
107000   hawker/57  0.0      18.0     0.0      1.80    0.0     0.0
107000   system    0.0      0.0      0.00000    0.0      0.00    0.0
2       kthruad   0.0      0.0      0.00000    0.0      0.00    0.0
6       rca_psr_3 0.0      0.0      0.00000    0.0      0.00    0.0
14       spdy0     0.0      0.0      0.00000    0.0      0.00    0.0
18       sm_worq_w 0.0      0.0      0.00000    0.0      0.00    0.0
11       koofing/0 0.0      0.0      0.00000    0.0      0.00    0.0
12       rca_wshd  0.0      0.0      0.00000    0.0      0.00    0.0
14       spdy0     0.0      0.0      0.00000    0.0      0.00    0.0
15       spdy0     0.0      0.0      0.00000    0.0      0.00    0.0
18       migration 0.0      0.0      0.00000    0.0      0.00    0.0
19       koofing/1 0.0      0.0      0.00000    0.0      0.00    0.0
21       spdy0     0.0      0.0      0.00000    0.0      0.00    0.0
22       migration 0.0      0.0      0.00000    0.0      0.00    0.0
23       koofing/2 0.0      0.0      0.00000    0.0      0.00    0.0
24       hawker/7 0.0      0.0      0.00000    0.0      0.00    0.0
25       koofing/3 0.0      0.0      0.00000    0.0      0.00    0.0
26       migration 0.0      0.0      0.00000    0.0      0.00    0.0
27       koofing/4 0.0      0.0      0.00000    0.0      0.00    0.0
28       koofing/3 0.0      0.0      0.00000    0.0      0.00    0.0
29       hawker/3 0.0      0.0      0.00000    0.0      0.00    0.0
31       spdy0     0.0      0.0      0.00000    0.0      0.00    0.0
32       migration 0.0      0.0      0.00000    0.0      0.00    0.0
33       koofing/4 0.0      0.0      0.00000    0.0      0.00    0.0
35       spdy0     0.0      0.0      0.00000    0.0      0.00    0.0
37       migration 0.0      0.0      0.00000    0.0      0.00    0.0
38       koofing/5 0.0      0.0      0.00000    0.0      0.00    0.0
39       koofing/5 0.0      0.0      0.00000    0.0      0.00    0.0

<- Memory for sorting: 1[0MB], 2[1MB], 3[0MB], 4[CPU], 5[0MB] ->
CPU - system CPU utilization

CPU: 0%  Mem: 0%  PerfMon: 1%  28 Processes: 0%  Mem: 0%
```

SAMA

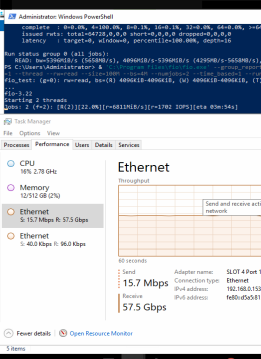
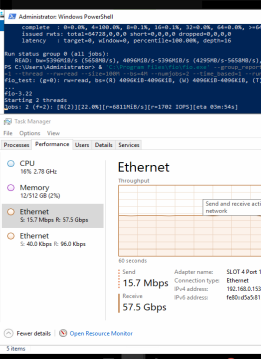
Stefan Metzmacher

multichannel / io_uring
(15/22)

SerNet

IORING_OP_SENDMSG prototyped (Part3)

The major problem still exists, memory copy done by copy_user_enhanced_fast_string()



```
samples: 1786 of event 'cycles', 4600 Hz, Count count (approx.): 8738153677 lost: 0/8
overhead
Shared object
0.01% [kernel] k] copy_user_enhanced_fast_string
0.28% [kernel] k] shmem_file_read_iter
0.73% [kernel] k] tcp_sendmsg_locked
0.14% [kernel] k] find_get_entry
0.24% [kernel] k] get_page_from_freelist
0.93% [kernel] k] __list_del_entry_valid
0.93% [kernel] k] native_wmem_spin_lock_slowpath
0.93% [kernel] k] raw_spin_lock
0.68% [kernel] k] skb_release_data
0.58% [kernel] k] mlx5_sq_exit
0.38% [kernel] k] __free_pages_ok
0.37% [kernel] k] raw_spin_lock_irqsave
0.35% [kernel] k] __zone_watermark_ok
0.33% [kernel] k] __list_lock_page
0.30% [kernel] k] copy_page_to_iter
0.31% [kernel] k] find_lock_entry
0.31% [kernel] k] __alloc_pages_node_mask
0.30% [kernel] k] __alloc_mllt_cq
0.29% [kernel] k] page_sapping
0.28% [kernel] k] xas_load
0.27% [kernel] k] shmem_getpage_sfp
0.25% [kernel] k] __check_object_size
0.23% [kernel] k] tcp_wfree
0.22% [kernel] k] __slab_free
0.21% [kernel] k] __sched_text_start
0.20% [kernel] k] __free_one_page
0.20% [kernel] k] mark_page_accessed
0.20% [kernel] k] bad_range
0.19% [kernel] k] tcp_data_insert
0.19% [kernel] k] iov_iter_advance
0.19% [kernel] k] native_ign_return_iter
0.18% [kernel] k] tcp_writ_exit
0.17% [kernel] k] __alloc_skb
0.16% [kernel] k] tasklet_action_common_isr.0
0.15% [kernel] k] clear_page_errs
0.14% [kernel] k] do_sync_irq
0.14% [kernel] k] __tcp_transmit_skb
0.13% [kernel] k] __skb_clone
0.13% [kernel] k] sock_page_errs
0.13% [kernel] k] sock_select
0.12% [kernel] k] __list_add_valid
0.12% [kernel] k] mlx5_qp_comp_int
0.11% [kernel] k] tcp_ack
```

SAMA

Stefan Metzmacher

multichannel / io_uring
(16/22)

SerNet

IORING_OP_SENDMSG/SPLICE prototyped (Part1)

16 connections, ~8.9 GBytes/s, smbld ~5% cpu, (io_wq_work 3%-12% cpu filesystem->pipe->socket), only ~100% cpu in total.

The Windows client was still the bottleneck with "Set-SmbClientConfiguration -ConnectionCountPerRssNetworkInterface 16"

File Name	Size	Progress	Speed	Time	Source	Destination
111111.rast	20	0	0	0	0	0
111009.rast	20	0	0	0	0	0
111008.rast	20	0	0	0	0	0
111007.rast	20	0	0	0	0	0
111010.rast	20	0	0	0	0	0
111012.rast	20	0	0	0	0	0
111015.rast	20	0	0	0	0	0
111022.rast	20	0	0	0	0	0
111094.rast	20	0	0	0	0	0
111097.rast	20	0	0	0	0	0
111092.rast	20	0	0	0	0	0
111088.rast	20	0	0	0	0	0
111086.rast	20	0	0	0	0	0
111089.rast	20	0	0	0	0	0
111112.rast	20	0	0	0	0	0
111011.rast	20	0	0	0	0	0
111005.rast	20	0	0	0	0	0
111015.rast	20	0	0	0	0	0
111010.rast	20	0	0	0	0	0
111082.rast	20	0	0	0	0	0
111089.rast	20	0	0	0	0	0
111083.rast	20	0	0	0	0	0
111053.rast	20	0	0	0	0	0
111076.rast	20	0	0	0	0	0
111059.rast	20	0	0	0	0	0
111051.rast	20	0	0	0	0	0
111049.rast	20	0	0	0	0	0
111048.rast	20	0	0	0	0	0
111046.rast	20	0	0	0	0	0
111047.rast	20	0	0	0	0	0
111045.rast	20	0	0	0	0	0
111044.rast	20	0	0	0	0	0
111043.rast	20	0	0	0	0	0
111042.rast	20	0	0	0	0	0
111041.rast	20	0	0	0	0	0
111040.rast	20	0	0	0	0	0
111039.rast	20	0	0	0	0	0
111038.rast	20	0	0	0	0	0
111037.rast	20	0	0	0	0	0
111036.rast	20	0	0	0	0	0
111035.rast	20	0	0	0	0	0
111034.rast	20	0	0	0	0	0
111033.rast	20	0	0	0	0	0
111032.rast	20	0	0	0	0	0
111031.rast	20	0	0	0	0	0
111030.rast	20	0	0	0	0	0
111029.rast	20	0	0	0	0	0
111028.rast	20	0	0	0	0	0
111027.rast	20	0	0	0	0	0
111026.rast	20	0	0	0	0	0
111025.rast	20	0	0	0	0	0
111024.rast	20	0	0	0	0	0
111023.rast	20	0	0	0	0	0
111022.rast	20	0	0	0	0	0
111021.rast	20	0	0	0	0	0
111020.rast	20	0	0	0	0	0
111019.rast	20	0	0	0	0	0
111018.rast	20	0	0	0	0	0
111017.rast	20	0	0	0	0	0
111016.rast	20	0	0	0	0	0
111015.rast	20	0	0	0	0	0
111014.rast	20	0	0	0	0	0
111013.rast	20	0	0	0	0	0
111012.rast	20	0	0	0	0	0
111011.rast	20	0	0	0	0	0
111010.rast	20	0	0	0	0	0
111009.rast	20	0	0	0	0	0
111008.rast	20	0	0	0	0	0
111007.rast	20	0	0	0	0	0
111006.rast	20	0	0	0	0	0
111005.rast	20	0	0	0	0	0
111004.rast	20	0	0	0	0	0
111003.rast	20	0	0	0	0	0
111002.rast	20	0	0	0	0	0
111001.rast	20	0	0	0	0	0

Stefan Metzmacher multichannel / io_uring SerNet

smbclient IORING_OP_SENDMSG/SPLICE (network)

4 connections, ~11 GBytes/s, smbld 8.6% cpu, with 4 io_wq_work threads (pipe to socket) at ~200% cpu each.

smbclient is the bottleneck here too

File Name	Size	Progress	Speed	Time	Source	Destination
111061.rast	20	0	0	0	0	0
111060.rast	20	0	0	0	0	0
111059.rast	20	0	0	0	0	0
111058.rast	20	0	0	0	0	0
111057.rast	20	0	0	0	0	0
111056.rast	20	0	0	0	0	0
111055.rast	20	0	0	0	0	0
111054.rast	20	0	0	0	0	0
111053.rast	20	0	0	0	0	0
111052.rast	20	0	0	0	0	0
111051.rast	20	0	0	0	0	0
111050.rast	20	0	0	0	0	0
111049.rast	20	0	0	0	0	0
111048.rast	20	0	0	0	0	0
111047.rast	20	0	0	0	0	0
111046.rast	20	0	0	0	0	0
111045.rast	20	0	0	0	0	0
111044.rast	20	0	0	0	0	0
111043.rast	20	0	0	0	0	0
111042.rast	20	0	0	0	0	0
111041.rast	20	0	0	0	0	0
111040.rast	20	0	0	0	0	0
111039.rast	20	0	0	0	0	0
111038.rast	20	0	0	0	0	0
111037.rast	20	0	0	0	0	0
111036.rast	20	0	0	0	0	0
111035.rast	20	0	0	0	0	0
111034.rast	20	0	0	0	0	0
111033.rast	20	0	0	0	0	0
111032.rast	20	0	0	0	0	0
111031.rast	20	0	0	0	0	0
111030.rast	20	0	0	0	0	0
111029.rast	20	0	0	0	0	0
111028.rast	20	0	0	0	0	0
111027.rast	20	0	0	0	0	0
111026.rast	20	0	0	0	0	0
111025.rast	20	0	0	0	0	0
111024.rast	20	0	0	0	0	0
111023.rast	20	0	0	0	0	0
111022.rast	20	0	0	0	0	0
111021.rast	20	0	0	0	0	0
111020.rast	20	0	0	0	0	0
111019.rast	20	0	0	0	0	0
111018.rast	20	0	0	0	0	0
111017.rast	20	0	0	0	0	0
111016.rast	20	0	0	0	0	0
111015.rast	20	0	0	0	0	0
111014.rast	20	0	0	0	0	0
111013.rast	20	0	0	0	0	0
111012.rast	20	0	0	0	0	0
111011.rast	20	0	0	0	0	0
111010.rast	20	0	0	0	0	0
111009.rast	20	0	0	0	0	0
111008.rast	20	0	0	0	0	0
111007.rast	20	0	0	0	0	0
111006.rast	20	0	0	0	0	0
111005.rast	20	0	0	0	0	0
111004.rast	20	0	0	0	0	0
111003.rast	20	0	0	0	0	0
111002.rast	20	0	0	0	0	0
111001.rast	20	0	0	0	0	0

Stefan Metzmacher multichannel / io_uring 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
- ▶ For SMB3 signing/encryption we may use:
 - ▶ IORING_OP_TEE with vmsplice could be used in order to still allow IORING_OP_SPLICE from/to the filesystem
 - ▶ vmsplice may also need to be optimized and added to io-uring
 - ▶ With eBPF support in io-uring we might be able to offline signing/encryption
- ▶ In the end SMB-Direct will also be able to reduce overhead
 - ▶ My smbdirect driver is still work in progress...

Questions? Feedback!

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