SNIA DEVELOPER CONFERENCE

SDC 24

BY Developers FOR Developers

What’s new in Samba?

Ralph Böhme, Samba Team, SerNet

2024-09-16

What’s new in Samba?

Samba 4.20 and 4.21

4.20

- Support for Witness Protocol [MS-SWN]
- Initial experimental support for SMB3 UNIX Extensions

4.21

- LDAP TLS/SASL channel binding support
- Per-user and group "veto files" and "hide files"
- Automatic keytab update after machine password change
- New cephfs VFS module vfs_ceph_new

Samba 4.20 and 4.21

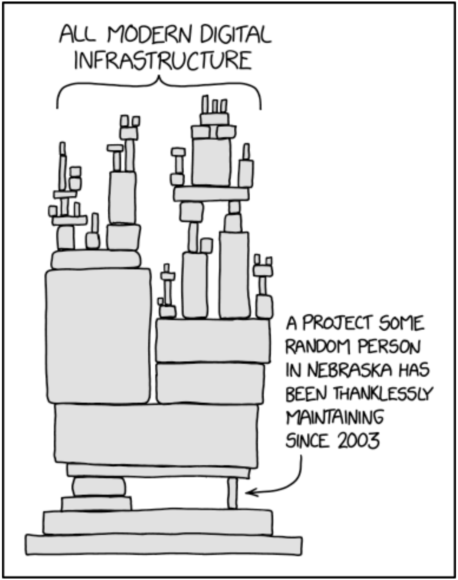
4.20

- Support for Witness Protocol [MS-SWN]
- Initial experimental support for SMB3 UNIX Extensions

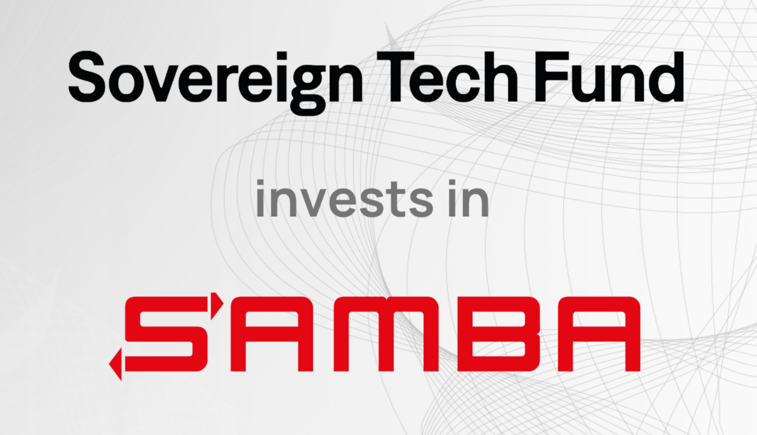
4.21

- LDAP TLS/SASL channel binding support
- Per-user and group "veto files" and "hide files"
- Automatic keytab update after machine password change
- New cephfs VFS module vfs_ceph_new

Modern Digital Infrastructure


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

STF Funding



Sovereign Tech Fund

invests in

SAMBA

STF, Samba & SerNet

Souvereign Tech Fund

- a German federal government funding program
- goal is to sustainably strengthen the open source ecosystem
- STF budget in 2023: 22 m €, 2024: 16 m €
- some funded projects: Gnome, FreeBSD, Log4j, PyPi, . . .
- SerNet applied to have Samba development funded
- STF invests 680k € into Samba via SerNet
- three SerNet Samba developers will work on 8 Samba features

Funded Features and Timeline

- Started 1st of September 2024
- 18 months project duration
- estimated development time of 2.25 years
- spread across three developers
- 8 large features . . . (see next slide)

STF funded Samba features

Features

- SMB3 UNIX Extensions
- Directory Leases
- Persistent Handles
- SMB over QUIC
- SMB Direct
- Performance with `io_uring`
- Netlogon Security Hardening
- SID Filtering

SMB3 UNIX Extensions

SMB3 UNIX Extensions in Samba

- Work in progress in kernel client and Samba server
- Volker has been busy in the background laying the foundations in Samba
  - many things do work
  - some things still require design discussion (special files, fchmod() and fchown())

SMB3 UNIX Extensions Specification

- Work is funded to finish the specification
- Recently we split the SMB3 UNIX Spec into three documents: `POSIX-SMB2`, `POSIX-FSA` and `POSIX-FSCC`
- `POSIX-FSCC` contains on-the-wire protocol changes and is mostly complete
- `POSIX-SMB2` is WIP and `POSIX-FSA` is mostly to be done

SMB3 UNIX Extensions

SMB3 UNIX Extensions in Samba

- Work in progress in kernel client and Samba server
- Volker has been busy in the background laying the foundations in Samba
  - many things do work
  - some things still require design discussion (special files, fchmod() and fchown())

SMB3 UNIX Extensions Specification

- Work is funded to finish the specification
- Recently we split the SMB3 UNIX Spec into three documents: `POSIX-SMB2`, `POSIX-FSA` and `POSIX-FSCC`
- `POSIX-FSCC` contains on-the-wire protocol changes and is mostly complete
- `POSIX-SMB2` is WIP and `POSIX-FSA` is mostly to be done

Directory Leases

Directory Leases

- Introduced in 2011
- Prototyped Directory Leases support in Samba in 2021
- "Most things work", writing tests
- Existing bug 13458: when deleting files Samba doesn't break H leases
  - Initial-delete-on-close when client disconnects tcon/session/TCP

Persistent Handles / Transparent Failover

Persistent Handles

- Also introduced in 2011
- Requires complex changes to file handle state handling and rigorous testing
- Prototype exists since 2018
  - see my presentations from SDC and SambaXP 2018
- SMB layer changes are trivial
- Durable Handles code lays groundwork, but many subtle changes needed:
  - `git diff ... 99 files changed, 5967 insertions, 393 deletions`
- Basic idea for the file handle state storage:
  - as before: use a non-replicating database for non-persistent opens
  - new: transparently store persistent open state in replicating database
  - Add a new flag to the database API store operation: `DBWRAP_FLAG_PER_REC_PERSISTENT`

SMB over QUIC

SMB over QUIC

- `IPPROTO_QUIC` coming to your friendly Linux socket API
- Samba continues to use the socket API with minimal changes for `IPPROTO_QUIC`
- For our automated tests we will need to extend `socket_wrapper` with support for `IPPROTO_QUIC`
- Possibly use userspace QUIC for client side support on older kernels

SMB Direct

SMB Direct

- Consolidate SMB direct support in the Linux kernel (both `cifs.ko` and `ksmbd` ship their own code)
- Expose it to userspace so `smbd` can use it
- Integrate SMB Direct support into smbd and smbclient
- Add automated SMB Direct functional testing

Performance with io_uring

Currently our single client and system IO throughput is CPU bound doing memcpy() in the kernel from user to kernel memory space

Make io_uring the default disk IO backend

- replace threadpool based disk IO with io_uring
- use preadv2(RWF_NOWAIT) to minimize latency for small IO

Performance with io_uring, cont.

Die memcpy, die!

- Use IORING_OP_[SENDMSG|RECVMSG] for higher single client performance
  - avoids blocking the smbd process in sendmsg()
  - still ends up doing memcpy() in the kernel, but adds some paralellisation
  - we can avoid one copy with IORING_OP_SENDMSG_ZC

One iouring op to rule them all: IORING_OP_SPLICE:

- completely avoids memcpy for disk and network IO path
- showstopper: data read from disk stored in pipe buffer is not stable
- other clients writing to the same blocks modifiy the pipe buffers
- semantics differ from `pread(fd, buf, ...) -> sendmsg(sfd, buf, ...)`
- only use if client has W lease (or exclusiv oplock or higher) ?

Performance with io_uring, cont.

Die memcpy, die!

- Use IORING_OP_[SENDMSG|RECVMSG] for higher single client performance
  - avoids blocking the smbd process in sendmsg()
  - still ends up doing memcpy() in the kernel, but adds some paralellisation
  - we can avoid one copy with IORING_OP_SENDMSG_ZC

One iouring op to rule them all: IORING_OP_SPLICE:

- completely avoids memcpy for disk and network IO path
- showstopper: data read from disk stored in pipe buffer is not stable
- other clients writing to the same blocks modifiy the pipe buffers
- semantics differ from
  `pread(fd, buf, ...) -> sendmsg(sfd, buf, ...)`
- only use if client has W lease (or exclusiv oplock or higher) ?

Netlogon Security Hardening

Netlogon Security Hardening

- MS-NRPC Netlogon security hardening
- Downgrade detection with `netr_LogonGetCapabilities()`
- Use Kerberos in Netlogin, avoid legacy NTLM crypto

SID Filtering

- Currently Samba doesn't implement SID filtering at security boundaries with trusts
- Mostly an Active Directory Feature
- Adds a security boundary between trusting forrests

Q&A

Q&A

Thank you!
Questions?

Ralph Böhme
slow@samba.org
rb@sernet.de