



SNIA DEVELOPER CONFERENCE



BY Developers FOR Developers

September 16-18, 2024
Santa Clara, CA

Demystifying Linux SMB mount options

Shyam Prasad, Bharath SM

Introduction

- Developers in the Azure Files team at Microsoft
- Focus on improved Linux customer experience with Azure Files
- Contribute to the Linux SMB and NFS ecosystem
- Core contributors to the Linux SMB client

Mount Options

- What is a mount option?
 - Tunable that allows altering the behavior of the filesystem that is mounted
- Why so many mount options on Linux SMB client?
 - Short answer: Workloads with different needs

Pros and Cons of having mount options

- **Pros:**

- Fine grained tunability
- Default values simplify usage
- Altered using remount command
- Control over experimental features

- **Cons:**

- Challenging for novice users
- Defaults may not be optimal
- Possible misconfigurations

Category of mount options

- Protocol version-specific
- Security
- Performance
- Data caching
- Metadata caching
- Access control
- Transport related
- Others

Protocol version-specific mount options

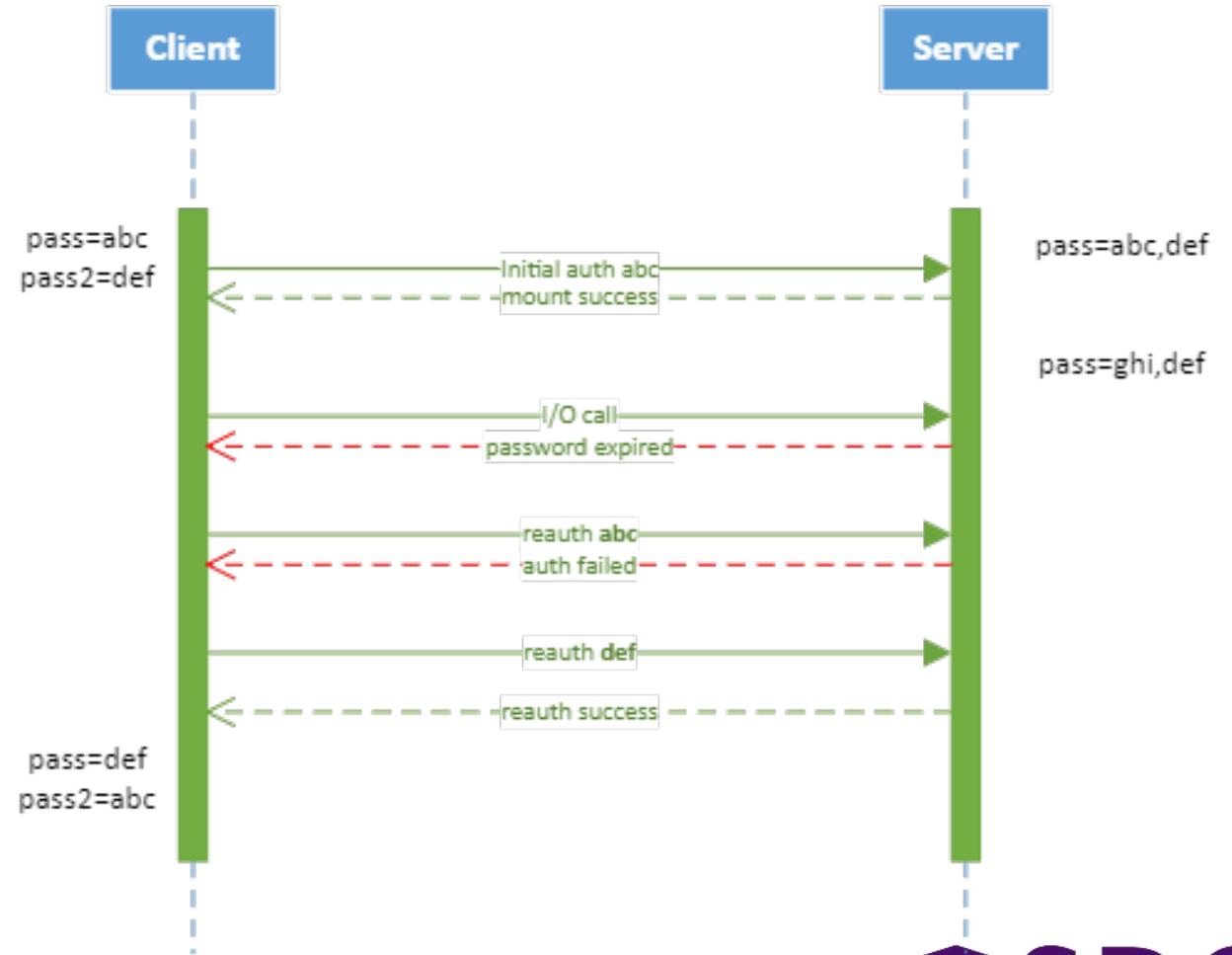
- **vers**
 - Allows selecting SMB dialect to be used. Default is negotiated with the server
- **resilienthandles**
 - Force use of resilient handles. Requires vers=2.1 or above
- **persistenthandles**
 - Force use of persistent handles. Requires vers=3 or above
- **handletimeout**
 - Set timeout on server for persistent/resilient handles
- **seal**
 - Force the use of encryption. Requires vers=3 or above

Security related mount options

- **sec**
 - Specifies the security mode to use for the SMB sessions
 - Mostly variants of KRB5 and NTLMv2
- **username/password**
 - Username and password credentials to be used for mount
 - With some security modes, password may be skipped
- **credentials**
 - Read user credentials from a file
 - Safer method to enter a password. File permissions need to be restricted
- **guest**
 - Uses guest user session to the server, if the server allows it

Security related mount options (contd)

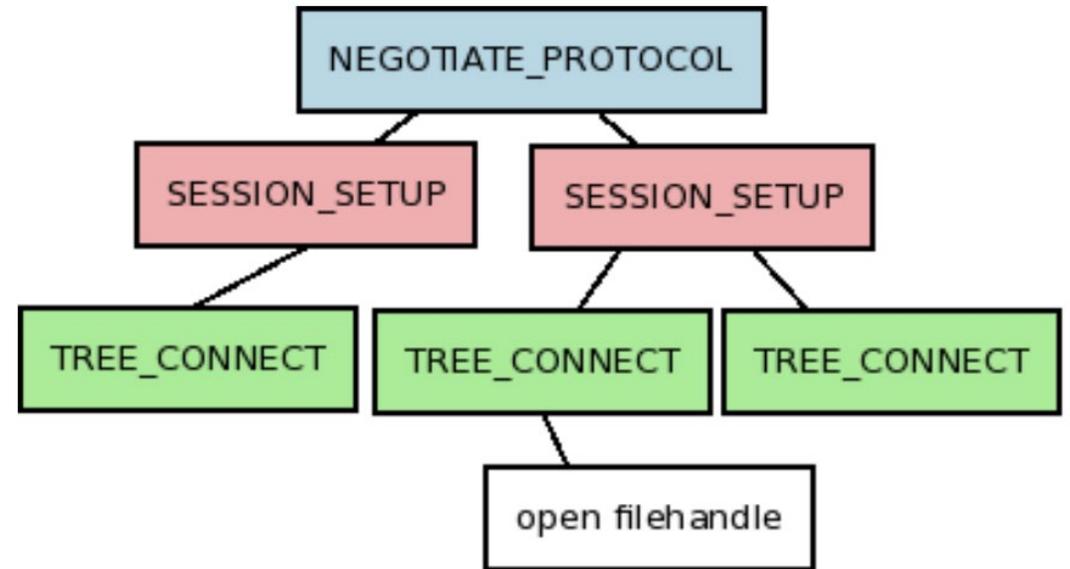
- **password2**
 - Newly introduced.
 - Improves mount availability
 - Specifies an alternative password
 - Useful when server supports multiple passwords
 - Can be updated using remount command



Security related mount options (contd)

multiuser

- Allows accessing the same mount point as multiple users
- Each user accessing the mount uses own SMB session
- More details: [Past presentation by Jeff Layton on multiuser.](#)



Security related module parameters

- **disable_legacy_dialects**
 - Disallow the use of vers=1.0 and vers=2.0
- **enable_gcm_256**
 - Enable the client to negotiate AES_256_GCM encryption with the server
- **require_gcm_256**
 - Require the server to support AES_256_GCM encryption
- **/proc/fs/cifs/SecurityFlags**
 - Allows disabling or enforcing security modes, encryption and signing

Data caching mount options

- **cache**

- Dictates cache coherency mode to be used for the mount
- `cache=strict` is to enable full data consistency
- `cache=loose` will cause the client to read cached data even without lease (not recommended)
- `cache=none` is to disable caching entirely (direct I/O)
- `cache=ro` is similar to read-only mount
- `cache=singleclient` assumes that the share is accessed only from this client

- **nolease**

- Force the client to not request server lease on file open

- **nobrl**

- Disables sending SMB byte-range-lock requests to server

Metadata caching mount options

- **acregmax**
 - File attributes caching time
- **acdirmax**
 - Directory attributes caching time
- **actimeo**
 - When set, sets both acregmax and acdirmax to this value
 - Defaults to 1 second

Performance related mount options

- **multichannel**
 - Enables support for SMB multichannel feature
 - Allows resiliency and parallelism for parallel workloads
 - Recent enhancement to channel allocation policy
 - Implemented channel sequence number for ensuring ordering
- **max_channels**
 - Set the max number of channels to establish with the server

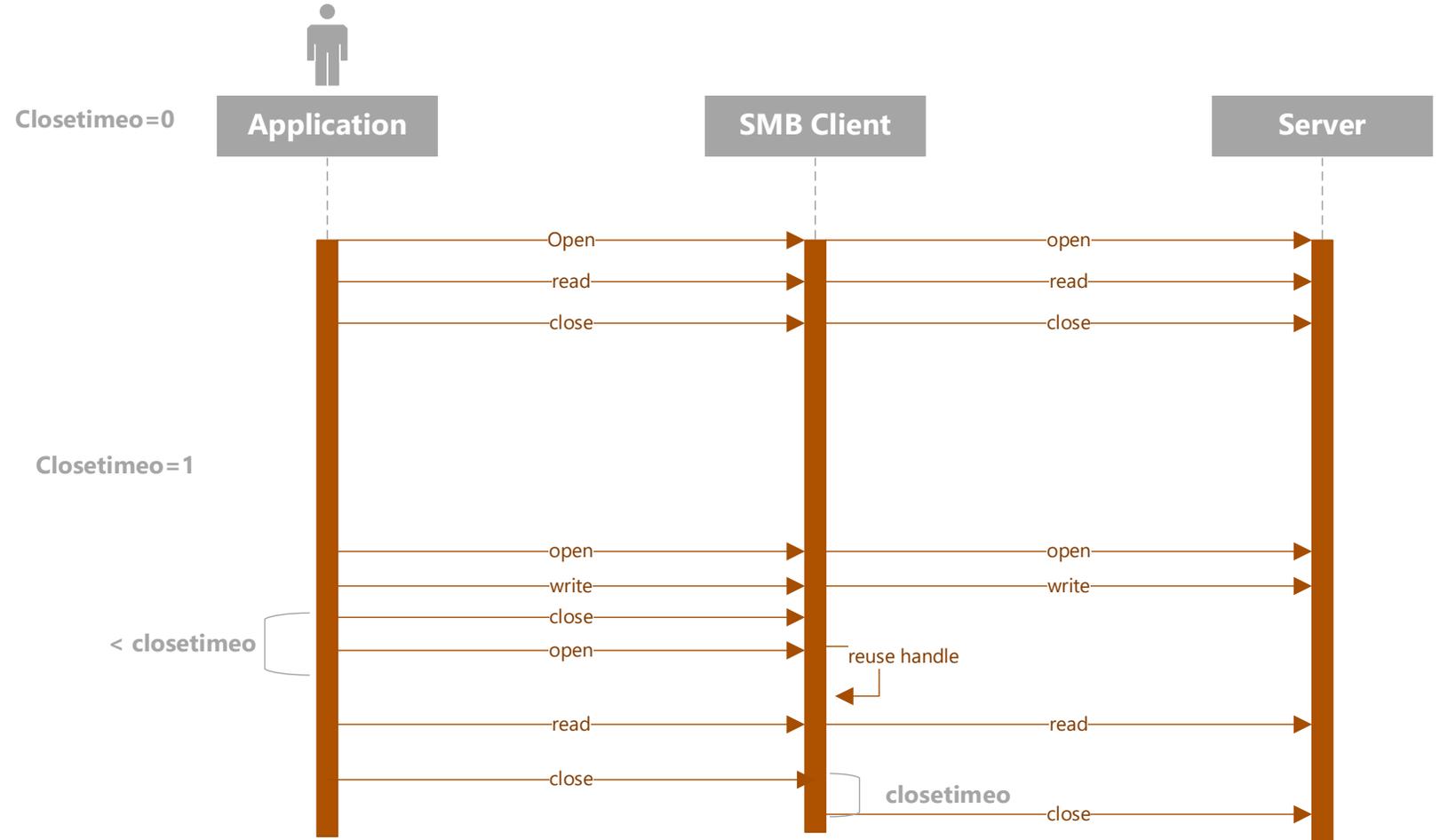
Performance related mount options (contd)

- **handlecache**
 - When the server supports dir leases, enables client caching of directory contents
- **max_cached_dirs**
 - When handlecache is enabled, this specifies the number of directories to cache the contents.
- **dir_cache_timeout (module parameter)**
 - When handlecache is enabled, this specifies the timeout for the cached contents

Performance related mount options (contd)

■ closetimeo

- Sets the timeout for deferred closes
- Helps reduce round-trips to server
- Recent optimization related to lease downgrade



Performance related mount options (contd)

- **rsize/wsize**
 - Override the max data size for read and write requests to the server
 - By default negotiated with the server
- **rasize**
 - Override the max readahead size for the filesystem
- **esize**
 - If specified, offload the decryption of the read response above this threshold
- **bsize**
 - Override the block size advertised by the filesystem

Access control related mount options

- **file_mode/dir_mode**
 - Overrides file and directory permissions on files and directories
- **posix/unix/linux (requires server support)**
 - Enables UNIX extensions, allowing more UNIX-friendly behavior
- **uid/gid**
 - Override the default uid/gid ownership for files/dirs on the mount point
- **cruid**
 - Used with `sec=krb5`
 - Cred cache for this UID instead of **uid** mount option
 - Useful when mounting is delegated to another user

Access control related mount options (contd)

- **idsfromsid/modefromsid**
 - Encodes modebits and file owner UID/GID in special SIDs on file ACLs
 - Useful when running Linux workloads on Windows-style servers
- **cifsac1**
 - Translates Linux identities to Windows-style SIDs
 - Translates Linux permissions to NT ACLs
 - Needs idmapping to be configured
 - Useful when interop is needed between Windows and Linux clients
- **noperm**
 - Disables access checks on the client

Transport related mount options

- **max_credits**
 - Allows limiting the number of credits consumed on a connection
- **echo_interval**
 - Period of sending SMB keepalive requests to the server. Default value 60s
- **hard**
 - Causes client to indefinitely retry server requests even in case of network or server outage
- **soft**
 - Causes client to retry on some network outage related errors limited number of times
- **retrans**
 - Max number of retries to server when soft mounts are used

Transport related mount options (contd)

- **ip/port**
 - Specify the IPv4/IPv6 address or custom port number for the server
 - If ip not specified, DNS resolution of the hostname extracted from the UNC will be used
- **srcaddr**
 - Specify a binding address on the client
- **nosharesock**
 - Do not share sockets with any existing connections to the same server
- **tcpnodelay**
 - Disables Nagle algorithm in TCP layer
 - Could improve performance
- **rdma**
 - Use RDMA as transport instead of TCP

Other mount options

- **snapshot**
 - Mount a specific snapshot version on the server instead of the active share
- **mfsymlinks**
 - For servers that do not support symlinks, this allows emulating symlinks using regular files
- **nostrictsync**
 - When set, avoids sending FLUSH command to the server
- **sloppy**
 - Allows mounts to succeed even when some mount options are not recognized
 - Helpful for writing portable mount commands
 - Eg: mount sloppy, closetimeo=30

References

- man page for mount.cifs
- https://www.snia.org/sites/default/orig/SDC2012/presentations/Revisions/JeffLayton_Multiuser%20Mounts%20with%20Linux%20CIFS_revision.pdf



Thank you