SerNet

Samba/CTDB/GPFS

April 24, 2013

Volker Lendecke

SerNet GmbH, Göttingen - Berlin

SMB: Server Message Block

- Protocol grown over a long time
- Origins based on MS-DOS system calls
  - „Int 0x21 on the network“
- Drive D: Mapped over the network
- Semantics tracable to single tasking DOS
  - Applications expected to be the only file openers

SMB 1, 2, 3

- SMB1: The only Protocol up to Windows Vista
- Evolutionary development from MS-DOS to Windows 2003:
  - NTFS-Semantics, Unicode File Names
- Hundreds of client and server implementations
- SMB2 (Windows Vista): New implementation
- SMB3 (Windows 8 / 2012): Substantial new features
  - Scalability, High Availability

GPFS: Cluster file system

- GPFS presents a Posix view across nodes on a shared set of disks
- Born in the Multimedia space, grown to HPC, expanded to general file system tasks
- Many extensions
  - Snapshots
  - Rich ACLs
  - Interoperability (Windows client)

Samba: Protocol translation

- Samba sits between the Windows and the Posix worlds
  - SMB protocol carries a lot of Windows semantics
  - Opening files very Windows-like (more later...)
- Smbd is the most prominent daemon translating Windows to Posix semantics
  - All file operations need to live with Posix semantics
- Samba's VFS is a pluggable module interface
  - All Posix calls can be intercepted
  - VFS extensions exist for advanced file systems

vfs_gpfs

- GPFS can server SMB shares without a module
  - Samba works with „just Posix“
- GPFS provides extensions for functionality and performance
  - Special API, bypassing the standard Linux Kernel interfaces
  - Library is GPL compatible

Opening a File in Posix

- Check the path
  - Do all directories exist?
  - x-Bit permissions on complete path?
- File exists?
  - Permissions sufficient?
  - -> File gets opened
- For every single process that's pure read operations
  - Easy to parallelize

Open a file in SMB

- In general, similar operations
  - Path check (case insensitive file names)
  - Permission check (ACLs)
- Share Modes
- Windows CreateFile API, Parameter dwShareModes:
  - If this parameter is zero and CreateFile succeeds, the file or device cannot be shared and cannot be opened again until the handle to the file or device is closed.
- Every open must know of all other opens

Samba Architecture

- Single Threaded, Multi-Process smbd
- Threads for async pread/pwrite
- Every client opens basically one TCP connection
  - One smbd for each client
- Protocol allows for many user sessions and share connects over one TCP connection

Share mode implementation

- Every Samba process knows of all open files
- Metadata for open handles held in shared memory
- Trivial database tdb
  - Multi-writer key/value database
- Share mode database indexed by device/inode
- Other metadata, for example „delete on close flag“ held in that database

Case Insensitive File Names

- In Posix. Test.txt and tEst.TxT are different files
- Windows and SMB see those as just one
- When Windows opens a file with wrong upper/lower case, Samba has to list all files
- When Windows creates a file, Samba has to prove that it does not exist in a different combination → search again
- GPFS offers a getrealfilename API
  - Case insensitive search
  - No directory listing required anymore

ACLs

- Posix: rwxrwxrwx
  - Extremely limited, but understandable
- Posix ACLs: rwx for supplementary users and groups
  - Simple inheritance
- Windows ACLs:
  - More than a dozen separate permissions
  - Complex inheritance rules
  - Files can be owned by groups
- NFSv4 ACLs
  - Almost, but not quite as Windows

ACLs on GPFS

- GPFS has several ACL modes
  - Posix only, NFSv4 only, mixed
- All with a separate, non-standard API
  - Well, there is no such thing as an NFSv4 ACL API
- vfs_gpfs provides access to NFSv4 ACLs

Windows attributes

- Samba has to store extra attributes:
  - Read-Only, Archive, System, Hidden
- Historically, mapped to „x“ bits
- With „store dos attributes = yes“ Samba stores that data in a posix extended attribute
  - Xattrs historically slow in GPFS
- Special API to store Windows attributes in GPFS Inode
  - A lot faster
  - Archive in the future with automatic semantics

Leases, Share Modes

- NFSv4 provides much of Windows semantics
  - GPFS has support for share modes and leases to support NFSv4
- NFSv4 and SMB share modes and oplocks don't fully match
- Gpfs:leases = no and gpfs:share modes = no is a very common configuration for SMB-only shares

Performance

- GPFS is made for large files
  - GPFS blocksize large (1M not uncommon)
- Small file workload can be slow
- Fcntl locks don't scale
- Large files: Async I/O

vfs_preopen

- Stream out video with one file per frame
  - A few megabytes at most
  - Opening a file takes some milliseconds, streaming does not work due to latencies
- Frame Files are numbered
  - vfs_preopen will fork processes that open and start reading the next files
  - Files are pre-cached
- 300 Mbyte/sec demonstrated with small file workload

Fcntl locks

- Posix has advisory locks
  - Locking byte ranges does not block read/write
- Windows does mandatory locks
  - Slightly different semantics (locks are not merged)
- Cross-protocol locking: Match Windows locks to Posix locks
  - Every SMB read/write request must query GPFS locks
  - GPFS is slow for high numbers of fcntl locks
- „posix locking = no“ for SMB only exports

Async I/O

- GPFS is very good for scaling disks, files and threads
- Given enough processes, GPFS can keep tons of disks busy
  - SMB2 clients send parallel reads
- Samba's core is single threaded, only 1 outstanding read system call by default
  - Samba 3.6 forks helper processes
  - 4.0 spawns threads for higher performance

Possible future development

- Better metadata integration
  - Share modes, leases
  - ACLs (Claims based acls anyone?)
- NFS interoperability
  - Locking grace period after node failure
- Support for better durable / persistent file handles
  - File system needs to block file access while we're not there

Kontakt

SerNet

Volker Lendecke, VL@sernet.de

SerNet GmbH
Bahnhofsallee 1b   Schützenstr. 18
37081 Göttingen   10117 Berlin

tel +49 551 370000-0   +49 30 5 779 779 0
fax +49 551 370000-9   +49 30 5 779 779 9

http://www.sernet.de