

A COMPARISON BETWEEN THE SAMBA3 AND LIKEWISE LWIOD FILE SERVERS

May 7, 2010

Outline



- Overview
- Architecture
- Feature Set
- Administrator Perspective
- Developer Perspective

Overview



- 18 years development
- GPLv3
- Personal copyright
- www.samba.org



- 1.5 years development
- GPLv2+
- Corporate copyright
- www.likewiseopen.org

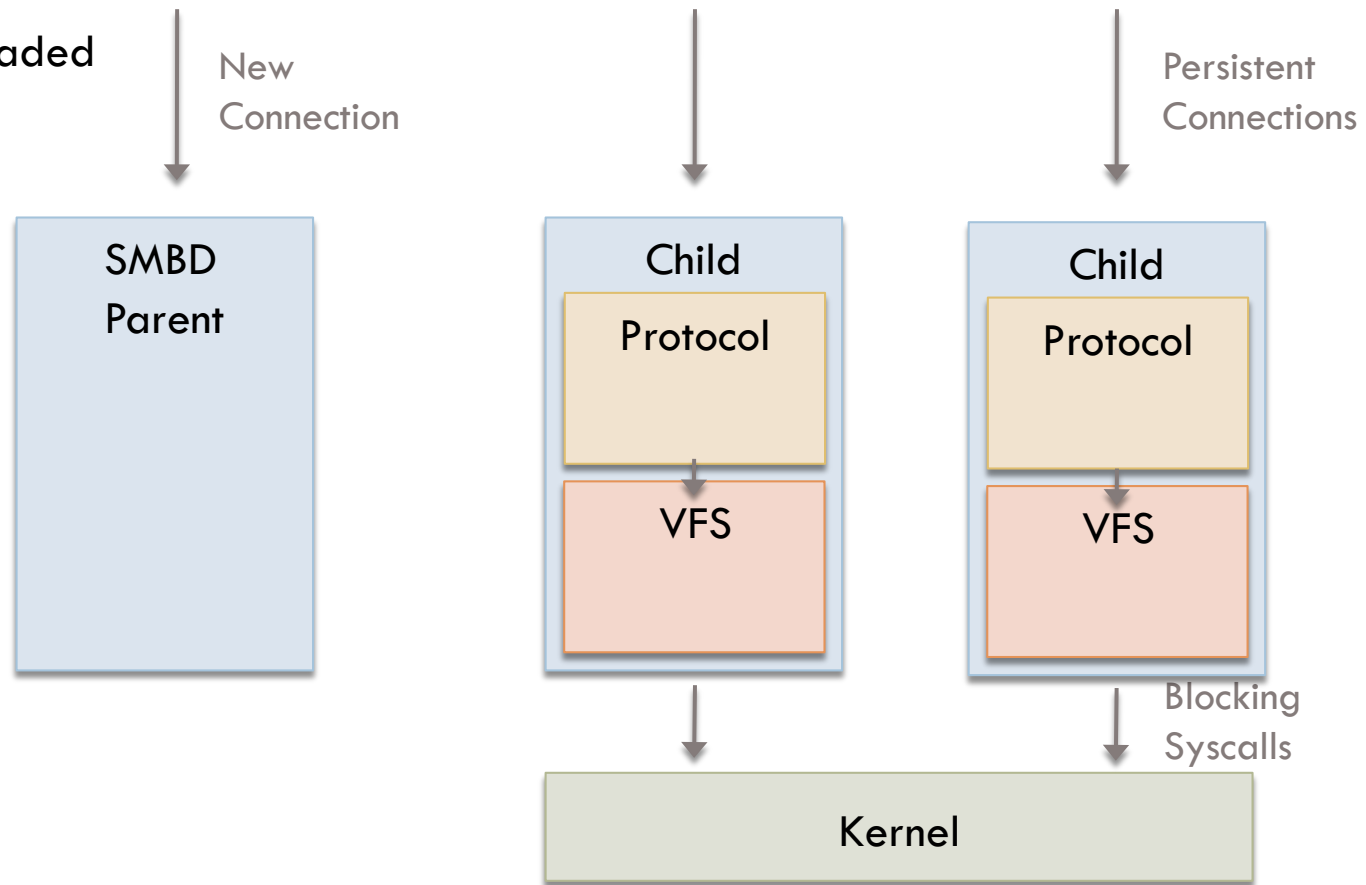
Comparison



- Git “master” branches as of April 17, 2010
- Focus on file server functionality
- Focus on Linux platform
- Only small discussion of:
 - Samba 4
 - winbindd
 - Isassd
- No discussion of:
 - Clustering

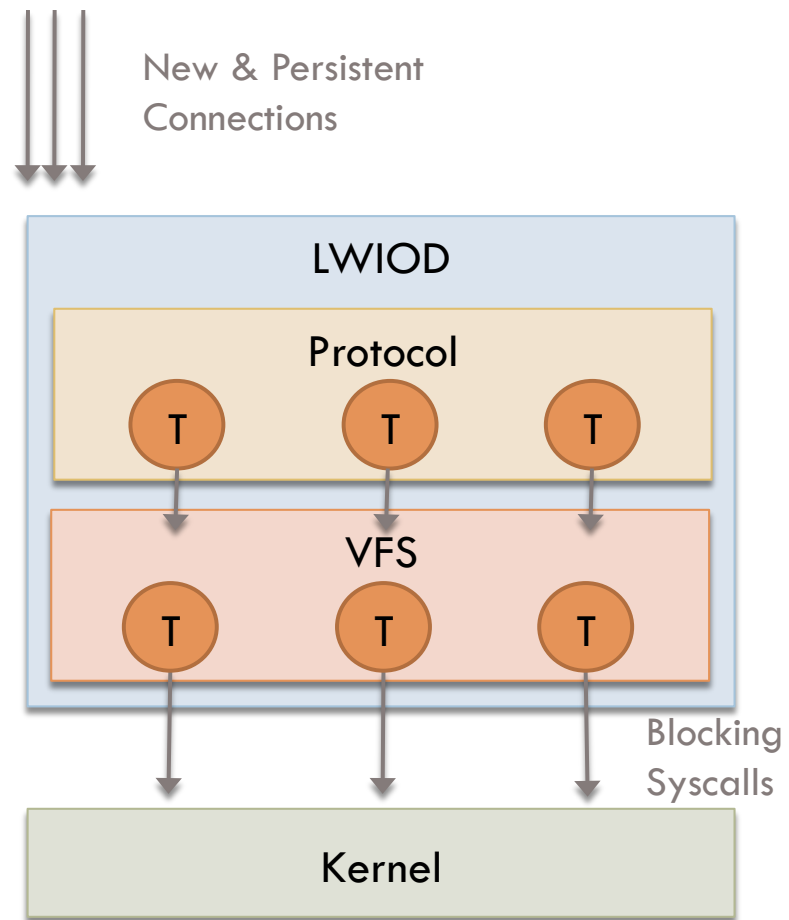
SMBD Architecture

- Multi-Process
- Single Threaded



LWIOD Architecture

- Single Process
- Multi-Threaded



Problem Classes



- Single Stream Read/Write
- Blocking Operations
 - Truncate of large file
- Simultaneous Sessions
 - IIS / Terminal Server
- Connection Throttling

Single Stream Read/Write



+ Simple, direct, I/O



+ Pipelined network I/O
written in parallel
– Thread context switch
overhead

- Depends on client network pipelining. Windows Vista or greater.
- Depends on kernel I/O interface: read()/write(), aio, sendfile(), splice().

Blocking Operations



- Network and file system interfaces are serialized



- + Parallel syscalls
- + Network I/O not blocked by file system operations
- + Greater responsiveness

- Extreme case: ECHO packets go unresponded and client disconnects.

Simultaneous Sessions



- I/O of user2 blocked by user1 from same client



- + All network I/O serviced in parallel

- Windows SMB client implementation multiplexes multiple sessions over single transport connection.

Connection Throttling



- + “max smbd processes”
- No distinction between idle and active connections
- New connection degradation all or nothing.



- + New connections limited by same thread pool as all other operations

Architecture Conclusion



- LWIOD **should** solve a number of scalability issues in specific workflows.
 - More graceful degradation under load.
- LWIOD **should** have less total resource utilization.
- Multi-threaded model forfeits some POSIX support.
 - Kernel access checks
 - fcntl() locking

Feature Set

	SMBD	LWIOD
SMB Dialect	<= NT LM 0.12	== NT LM 0.12
SMB2 Dialect	SMB 2.002	SMB 2.002
NetBios Transport	✓	✗
Share Level Security	✓	✗
Share Mode Locks	✓	✓
Oplocks	✓	✓
Change Notify	✓	✓
Byte Range Locks	○	○
Access Control Lists	○	○
Alternate Data Streams	✓	✗
Shadow Copy	✓	○

✓ Works

○ Mostly Works

✗ Doesn't Work

SMB Protocol Dialect / NetBios



+ Unix Extensions



+ SNIA Reference support

- Unsupported:

- Port 139

- NT4 clients

- OS X <= 10.4 clients

- No password-less login

SMB2



- SMB v1 equivalent support in both.
 - SMB2 2.002 == Windows Vista / 2008 Server
- No SMB2 only features:
 - Durable / Resilient Handles
 - Leases
 - Credits

Share Mode / Oplock / Change Notify



- + Linux kernel oplock partial support
- + Linux kernel notify support

- Both: Functionality implemented in user space.
- No cross-protocol support without kernel implementation.

Byte Range Locks



+ Can back with POSIX
locks

- ❑ Both: Implement Windows locking semantics in user space.
- ❑ Both: Use user space queue that doesn't guarantee fairness.
- ❑ Both: Provide strict locking semantics for SMB only traffic.

Access Control Lists

	SMBD	LWIOD
Access Checks	Kernel	User Space
Storage in Xattr	✓	✓
Map to Mode Bits	✓	✓
Map to POSIX ACL	✓	✗
Map to NFSv4 ACL	○	✗
NT4 Style ACL	✓	✗

- SMBD: xattr storage AND POSIX ACL
- LWIOD: xattr storage OR mode bits

Access Control Lists



- Best choice:
 - + Perfect POSIX support
 - + Perfect Windows
 - + get/set
 - Imperfect Windows
 - access check



- Best choice:
 - + Perfect Windows support
 - No POSIX support

Access Control Lists

- Both: Lossy conversion between NTFS ACL and mode bits / POSIX ACL
 - RWX != Full Control
 - Groups owning files
 - Deny aces and canonical order
- Both: Rely on SID to Unix ID mapping
 - Unmappable SIDs can't be stored
- NTFS ACLs to POSIX sucks no matter what you do!

Alternate Data Streams



+ ADS storage in hidden subdirectory tree

+ ADS storage in xattrs



- Support for `::$DATA` stream only

Shadow Copy



- + Allows enumeration / restore
- + Conversion of wire name to directory name

- POSIX VFS module lacks enumeration / restore

- Both: Support parsing of @GMT path format in protocol head.
- Both: Rely on underlying file system to implement snapshots.

Feature Set (continued)

	SMBD	LWIOD
Srvsvc	○	✓
DOS File Attributes	✓	✓
Distributed File System	✓	✗
Signing / Sealing	✓	✓
Privileges	✓	✗
Auditing	○	✗
IPv6	✓	✗
Copy Chunk	✗	✗

Non-File Server Functionality

	SMBD	LWIOD
Printing / spoolss	✓	✗
NetBios Name Server	✓	✗
NT4 Domain Server	✓	✗
Active Directory Server	○	✗

Feature Conclusion



- SMBD far more mature in number of features.
- No technical reason for lack of features in LWIOD simply a matter of development time and effort.
- LWIOD supports newer clients only.
- Some features cannot be perfectly duplicated on POSIX regardless of server implementation.
 - ACLs
 - BRL

Administering / Configuration



- Stored in .ini text file
- 374 Total Parameters
- 146 Share Parameters
- Alternate registry configuration storage



- Stored in registry
- 14 Parameters (lwiiod only)
- + MMC support by default
- Examples of missing config:
 - Access based enum
 - Share path substitution
 - Name mangling

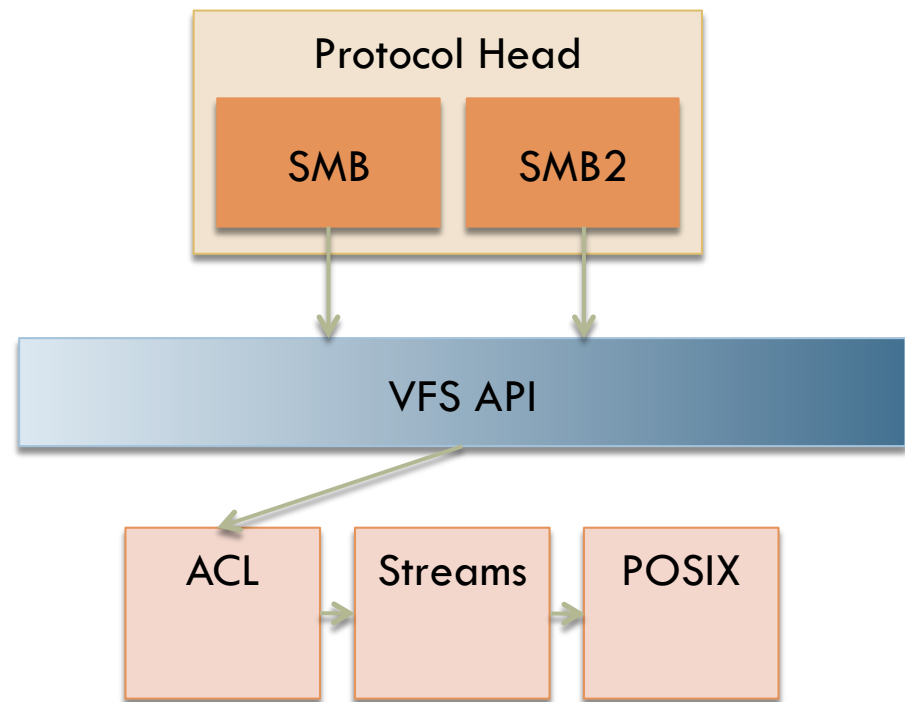
Developer Perspective



- Both: Provide division between protocol marshalling and kernel file system interface.
- Both: OEM's task is to implement a VFS module
- Wildly different coding style

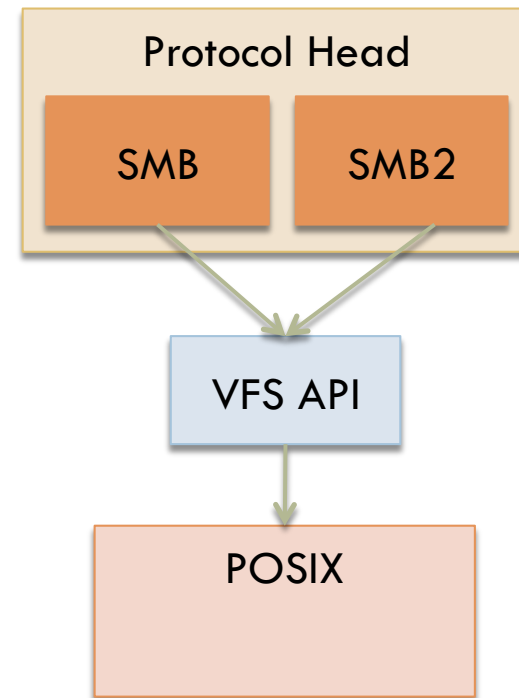
SMBD VFS

- Hybrid Interface
 - POSIX / NTFS / Other
- 113 functions
- Chained modules
- OS specific modules
 - irixacl, hpuxacl, zfsacl
- ~30 modules
 - Partial API



LWIOD VFS

- IOMGR Interface
 - ▣ Abstract Device Driver API
- 16 functions
 - ▣ Includes IOCTL
 - ▣ Every call can go async
- No chaining or filtering
- Generic POSIX module
 - ▣ No OS specific functionality yet
 - ▣ Reference implementation
- 1 Module



Questions?



Contact

Steven Danneman
sdanneman@isilon.com
steven@samba.org