Mathias Dietz – IBM Research and Development, Mainz

Samba in a cross protocol environment

aka SMB semantics vs NFS semantics

Introduction

Mathias Dietz (IBM)

IBM Research and Development in Mainz, Germany

NAS architecture and development of IBM SONAS and Storwize V7000 Unified product.

Experience with Samba as part of the IBM SoFS offering since 2006 and the IBM OESV offering since 2003

IBM SONAS - Scale Out Network Attached Storage

Modular high performance storage with massive scalability and high availability. Supports multiple petabytes of storage for organizations that need billions of files in a single file system.

http://www.ibm.com/systems/storage/network/sonas/

IBM Storwize V7000 Unified

A virtualized storage system designed to consolidate block and file workloads into a single storage system for simplicity of management, reduced cost, highly scalable capacity and high availability.

http://www.ibm.com/systems/storage/disk/storwize_v7000/

Cross Protocol Use Cases

Some real world customer examples using multi-protocol access

1) **Movie rendering** where the workstations of 3D designers write the 3D models via SMB and a bunch of render servers access the 3D models with NFS

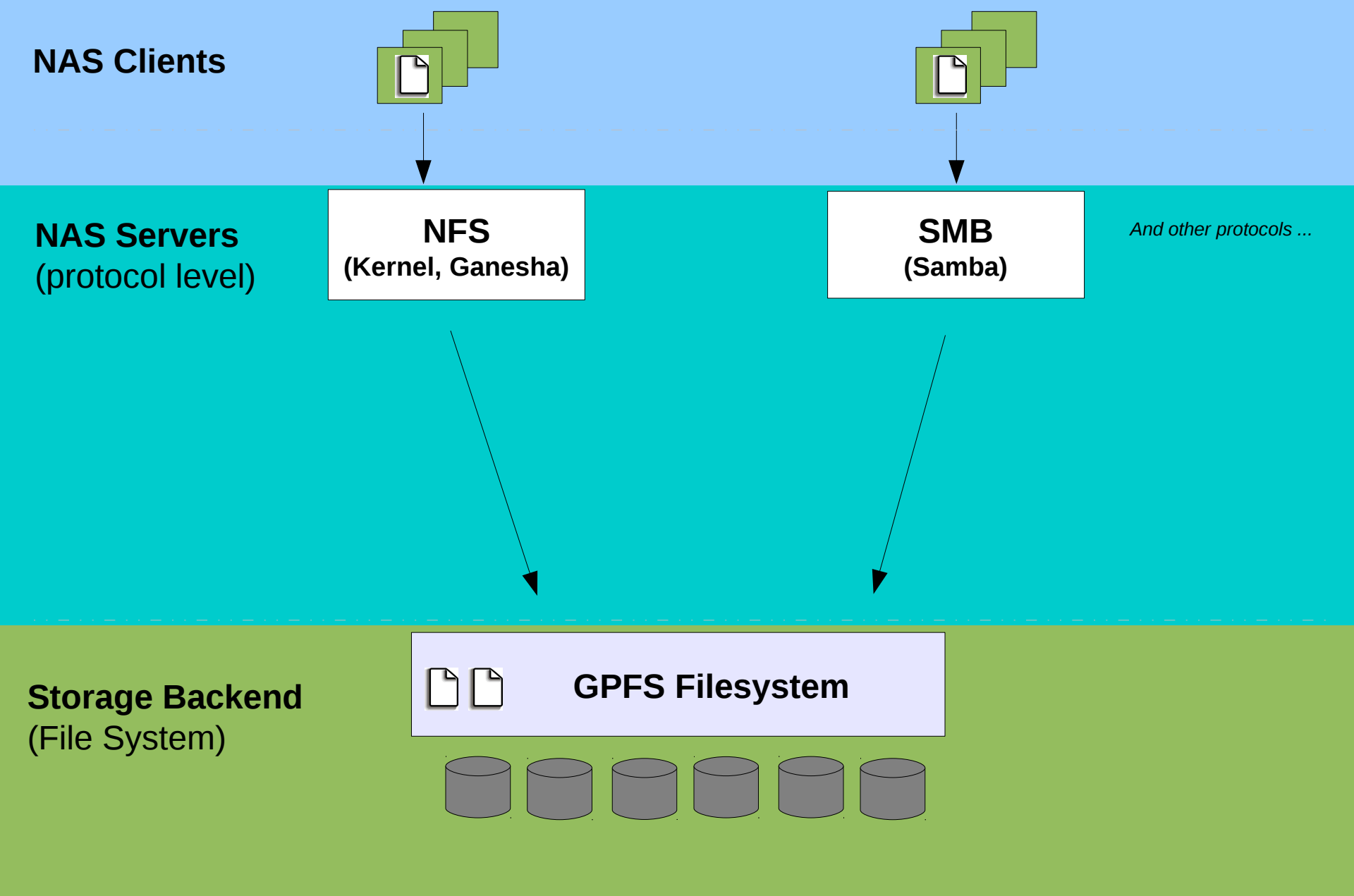
2) **Research Institute**: measurement data has been written by NFS and scientists access the data through SMB

3) **Software development** team used NFS and SMB for accessing source code paths concurrently and build machines are using NFS to build the code.

We will talk about

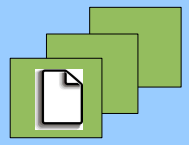
- Protocol interoperability
- Accessing the same data through multiple protocols concurrently or sequentially
- Focus on NFS and SMB
- Using the example of GPFS filesystem

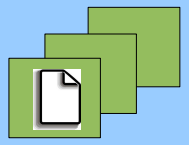
Cross Protocol - System Context



Cross Protocol - System Context

NAS Clients





NAS Servers
(protocol level)

NFS
(Kernel, Ganesha)

SMB
(Samba)

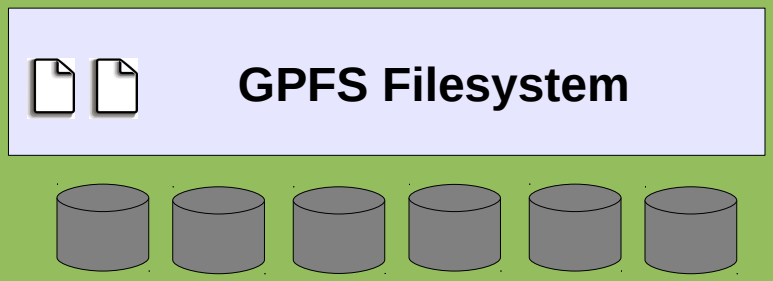
And other protocols ...

- **Locking (byte range)**
- **Share reservation**
- **Delegations**
- **Grace (lock reclaim)**
- **Sessions (4.1)**
- **v3/v4.x differences**

- **Authentication**
- **ID Mapping**
- **Recovery**
- **Notifications**
- **Extended Attributes**
- **Auditing & Tracing**
- **I/O & resource balancing**

- **Locking**
- **Share Modes**
- **Oplocks / Leases**
- **Durable / Persistent FH**
- **Streams**
- **Sessions**
- **Windows Attributes**
- **SMB 2.x version differences**

Storage Backend
(File System)



- **Locks**
- **Share Modes / Reservations**
- **GPFS Leases (oplocks)**
- **Extended attributes**
- **Windows attributes**
- **ACL**
- **...**

NFSv4 Concepts vs SMB Concepts

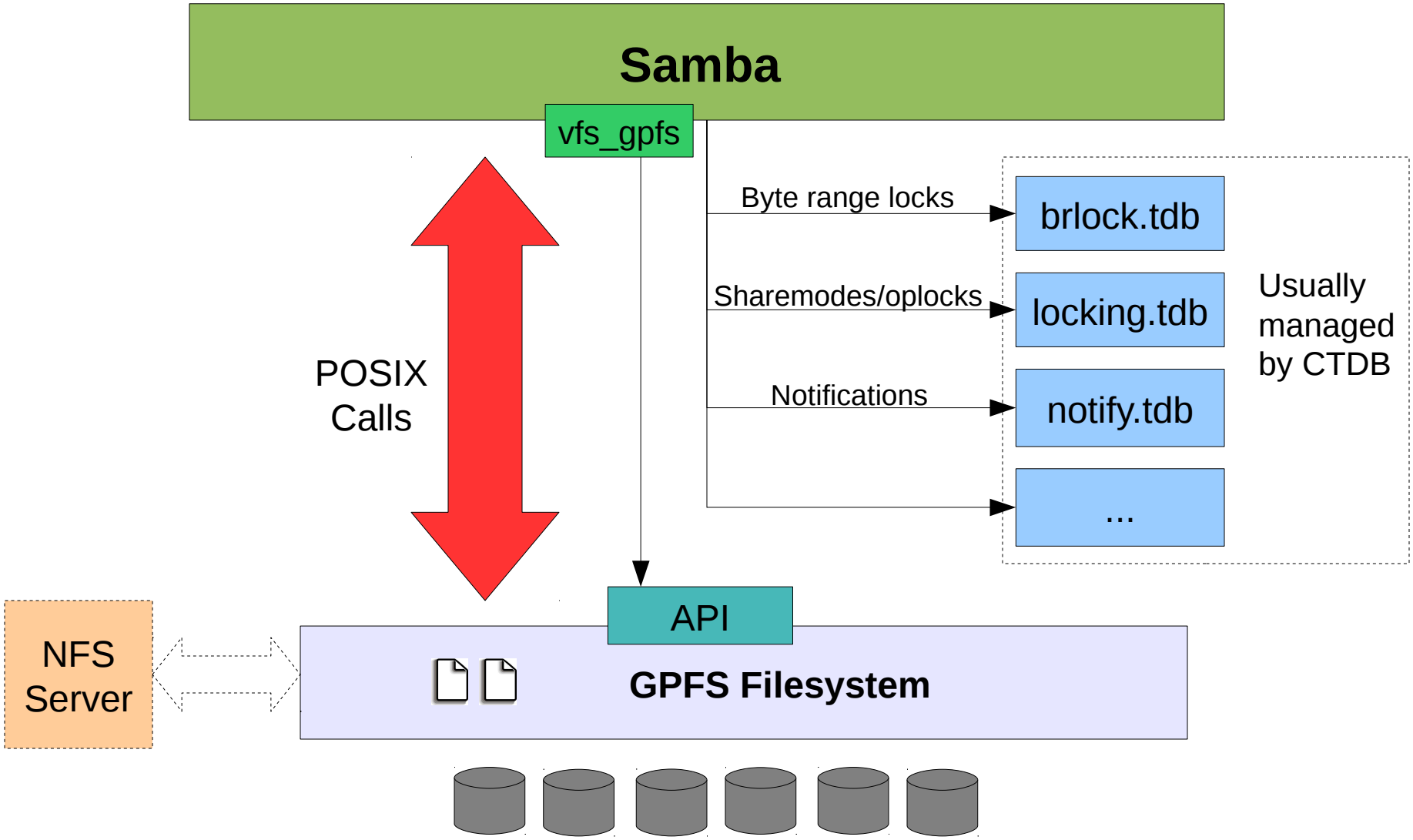
From a high level perspective Windows® semantics have a huge overlap with NFSv4.x semantics

– Byte range locks	~	Byte Range locks (NFSv3 POSIX fcntl)
– Windows Oplocks	~	NFSv4 Delegations
– Windows Sharemodes	~	NFSv4 Reservations
– Alternate Data Streams	~	NFSv4.1 Named Attributes
– SMB 2.1 Directory leases	~	NFSv4.1 Directory Leases
– SMB Notifications	~	NFSv4.1 Directory Change Notifications
– NTFS ACLs	~	NFSv4 ACL
– SMB2 Durable Open	~	Lock reclaim
– SMB3 Replay Detection	~	Duplicate reply cache

NFSv4 has many similarities with SMB

But there are many semantical differences when looking into the details !

Samba on GPFS - Architecture



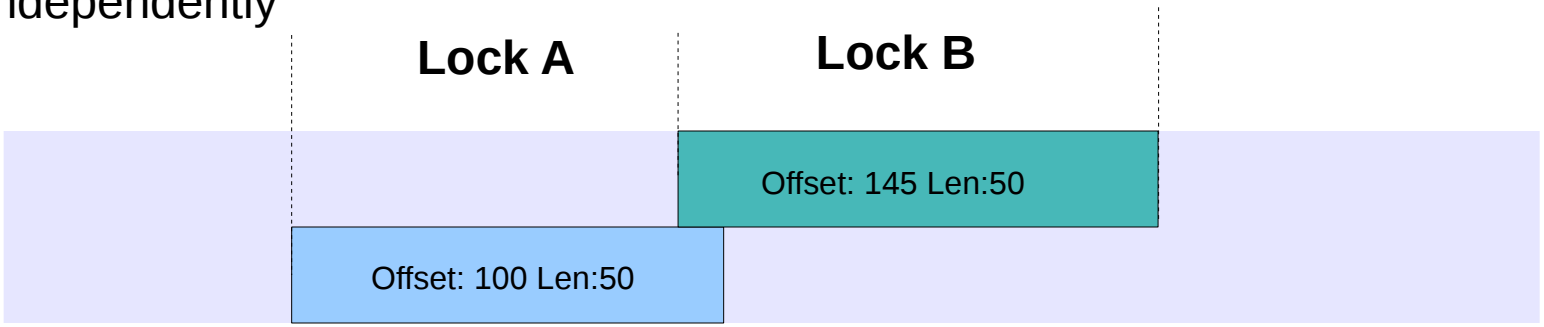
Byte Range Lock differences

- Different byte range lock semantics between **NFS & SMB**
- **NFS** locks (based on fcntl() call)
  - By default are Advisory
  - Mandatory locking is optional in NFSv4
  - Byte range locks can be merged/split
  - NFSv4 server might reject locks which are overlapping/split (NFS4ERR_LOCK_RANGE) , clients can then emulate the behavior (race condition)

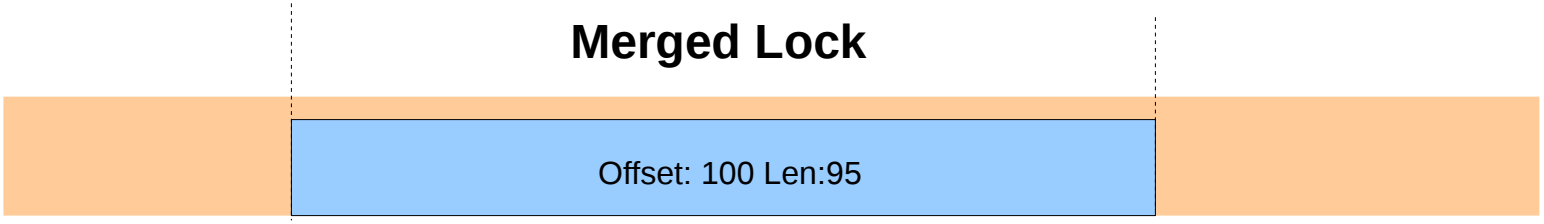
- **SMB** locks (SMB2 LOCK Request)
  - Are all mandatory (no advisory locks)
  - Supports stacking of locks instead of merging/splitting
  - An unlock range MUST be identical to the lock range. Sub-ranges cannot be unlocked.
  - A write lock can be acquired even if opened for read.

Byte Range Lock differences

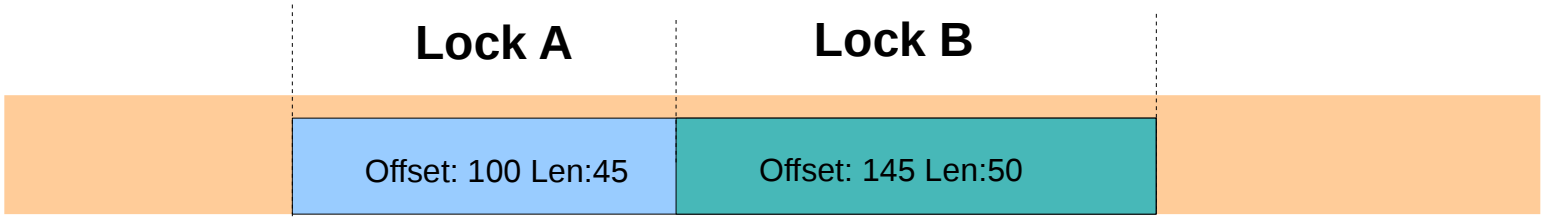
Windows keeps the locks separated (stacked), each lock can be unlocked independently



POSIX will merge overlapping locks



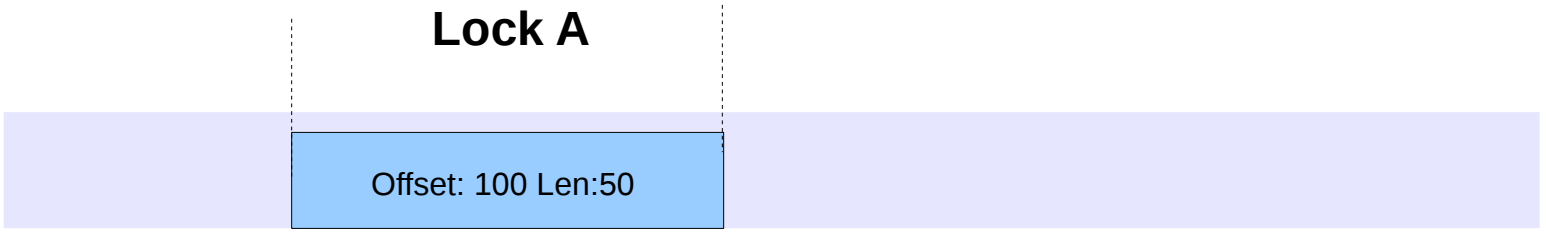
Overwrite lock A in case of different lock types (RDLCK,WDLCK)



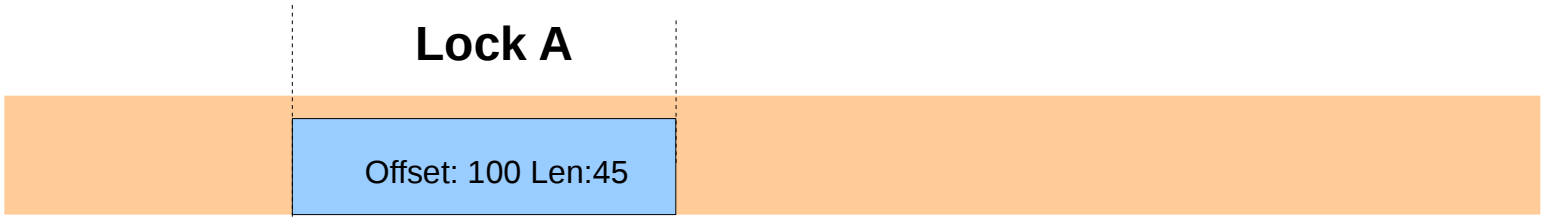
Byte Range Lock differences

After **unlocking Lock B** the semantical difference is clear:

Windows lock has a length of 50

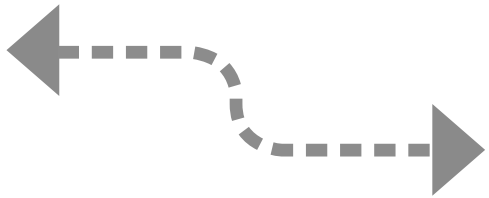


POSIX lock has length of 45 instead of 50 !!



Byte Range Lock differences

To achieve cross protocol consistency



- Lock semantics must be adapted
  - Samba automatically maps between the SMB lock requests and the POSIX locking of the filesystem.
- Byte range locks from NFS clients must be honored by SMB clients and vice versa
  - Pass lock requests down to the filesystem (posix locking = yes)
  - NFS locks are mandatory for CIFS clients (CIFS treats all fcntl locks as mandatory)
  - CIFS locks are advisory only for NFS clients

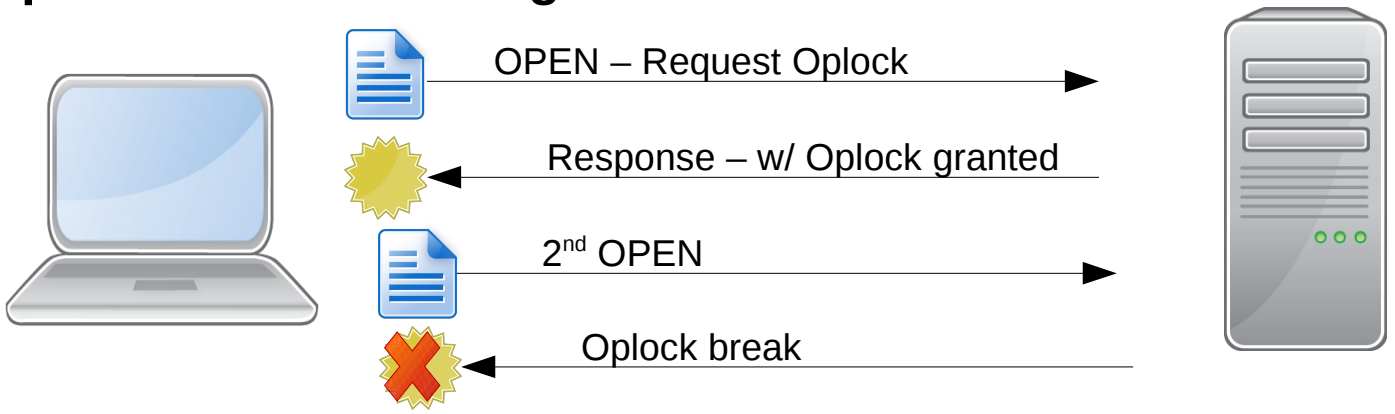
Oplocks / Leases vs Delegations

SMB Opportunistic Locks and NFS Delegations allow a client to cache data locally

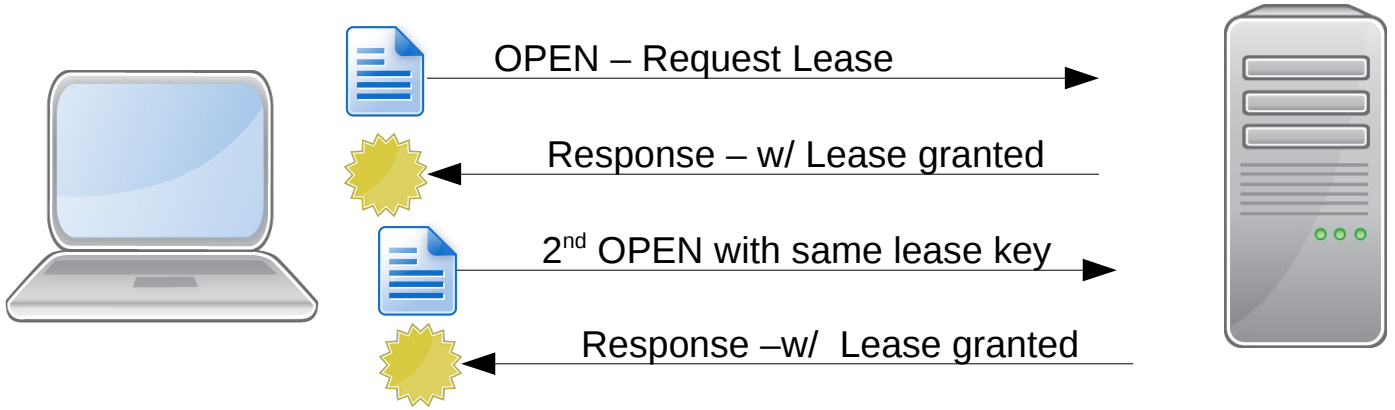
- **NFSv4 Delegations**
  - NFSv3 does not support delegations
  - NFSv4 delegations are READ or WRITE
  - NFSv4 allows delegations on files or Named Attributes
- **SMB Oplocks**
  - Level 1 (exclusive), Level 2 (read), Batch (exclusive)
  - Allows downgrade to Level 2 oplock (SMB2 OPLOCK_BREAK)
  - SMB allows oplocks on Files or Streams
- **SMB 2.1 Leases**
  - Lease key - other processes using this key are permitted access without breaking the lease (shared leases)
  - More oplock types (R,RH,RW,RWH)
  - Allows upgrade and downgrade

Oplocks / Delegations vs Leases

SMB Oplocks / NFSv4 Delegations*



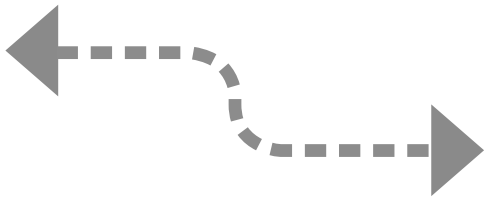
SMB 2.1 Leases



*NFSv4 seem to have the same limitation, but this can be client dependent

Oplocks , Leases and Delegations

To achieve cross protocol consistency



- Oplocks/Delegations must be delegated to the filesystem
  - GPFS can manage READ and WRITE oplocks
    - Samba option gpfs:leases=yes
  - Kernel limitation prevents cross-protocol Level II Oplocks
    - A read lease can be placed only on a file descriptor that is opened read-only.
    - GPFS lease code is based on Kernel implementation, so same restrictions apply
    - Level II oplocks should be turned off (level2 oplocks=no)
- SMB 2.1 Leases
  - Samba does not support them yet
  - Not supported by Kernel and GPFS
    - Thus cannot be used in cross protocol environments

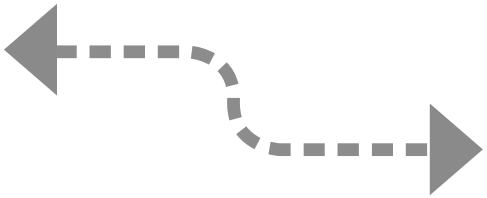
Share Modes / Share Reservations

SMB Share Modes and Share Reservations allows the client to control which operations are allowed for other applications on the same file

- **SMB Share Modes**
  - SHARE_READ and SHARE_WRITE modes
  - SMB allows explicit SHARE_DELETE share mode
    - indicates that other opens are allowed to delete or rename the file
  - Mandatory
  - Allowed on Files and Directories
- **NFS Share Reservations**
  - NFSv4 DENY READ,WRITE,BOTH or NONE
  - Mandatory
  - NFSv3 does not support share reservations

POSIX open() does not support share modes, Samba uses flock() when “kernel share modes” option is set to yes

Share Modes / Share Reservations



To achieve cross protocol consistency

- Share modes must be propagated to the filesystem
  - GPFS can manage READ and WRITE share modes
    - Samba option gpfs:sharemodes=yes
  - DELETE share mode is not supported by GPFS
    - vfs_gpfs module maps DELETE to READ+WRITE
    - Semantic is not 100% correct
- Currently the Samba sharemode handling is not atomic (multiple calls)
  - Race conditions might occur if NFSv4 server requests share reservations as well
  - GPFS 3.5 introduced a new createFile() API
    - Atomic call to get sharemode/oplock/initial ACL during open()
    - Not implemented in Samba yet
- With NFS V3 reading a file is not denied even with DENY_ALL mode
- Directory share modes are not visible to NFS

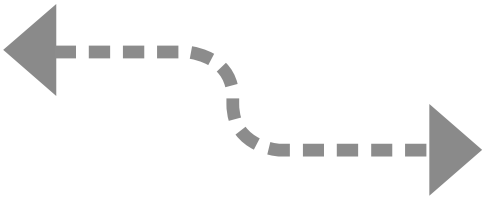
Change Notifications

Clients can register to get notified when a file's data or metadata under a given subtree is changed

- **SMB Change Notifications**
  - SMB2 CHANGE_NOTIFY
  - Can be recursive
  - Heavily used by Windows explorer
- **NFS Directory Notifications**
  - Not recursive
  - Not supported by NFSv3 and NFSv4.0
    - Support added with NFSv4.1

Samba uses the notify.tdb to register notification requests and send out Notifications when changes through Samba are detected

Change Notifications



To achieve cross protocol consistency

- GPFS does not offer a API to register cluster wide change notifications
- Use Inotify for cross protocol change notifications on a single node
  - Samba option “kernel change notify=yes”
  - Not recursive
- For cluster wide change notifications
  - Use vfs_notify_fam together with Garmin
  - Listen to Volker Lendeckes Talk about “Scalable file change notify”

Timestamps and Windows Attributes

- Different file time stamps for Windows and NFS
  - SMB
    - creation time (birth time),
    - write time (modification of data)
    - access time (last access)
  - NFS
    - change time (meta data change)
    - modify time (content change)
    - access time (last access)

- Additional Windows Attributes
  - SYSTEM
  - HIDDEN
  - READ-ONLY
  - ARCHIVE

Timestamps and Windows Attributes

- Birth time and Windows attributes can be stored in GPFS
  - using the gpfs:winattr=yes option
- **Cross protocol considerations**
  - Windows Attributes are not seen by NFS
  - Read-only has no effect on NFS reads
    - Map read-only can be used in addition, but not with NFSv4 ACLs
  - Hidden has no effect on NFS
  - Archive flag will be set even if modified with NFS
  - Birth time is automatically filled by GPFS when a file is created

NFSv4 Grace Period

NFSv4 Servers can trigger a grace period to allow lock recovery. During this grace period clients can reclaim their locks but no new locks will be granted.

- This opens a window for **lock stealing across protocols**
  - Samba is unaware of NFS recovery.
  - During NFS recovery, Samba can grab a lock in the conflicting mode, resulting denial of NFS reclaim.
- **Possible solutions (not implemented yet)**
  - Make Samba aware of the grace period
    - Not granular enough – impact to CIFS/SMB IO
  - Disallow conflicting locks during grace
    - Filesystem would need to know NFS recovery information

NOTE: the term 'lock' covers all kinds of locks/share reservations/delegations.

More

- NFS - Kerberos PAC decoding
  - Ganesha can forward Kerberos PAC to winbind to decode group membership
  - Samba Machine account should be used to avoid keytab mismatch
- SMB Delete on close
  - Delete on close semantics are not known to NFS
  - Files are still visible until finally deleted
- Durable File handles
  - Samba implements durable file handles, but only if all cross-protocol functions are disabled
- Case insensitivity
  - GPFS has gpfs_get_realfilename() API to allow case-insensitive lookup
- Alternate Data Streams
  - Stream are not supported by GPFS, but Extended Attributes can be used (64k limit)
- ID mapping
  - Complex topic – could easily fill another 1h presentation
- ACL - Listen to SambaXP Talk - Recent improvements in using NFS4 ACLs with Samba (Alexander Werth, IBM)

Outlook

To achieve better cross protocol support **more operations must move down into the file system**.
Therefore the file system needs more NTFS semantics

- POSIX interface is not sufficient
- Samba POSIX mappings must be bypassed (Extend VFS Layer ?)

- **Possible GPFS / vfs_gpfs improvements**
  - Atomic create file call (bypass Samba create file)
  - SMB Locking semantics in GPFS
    - SMB style byte range locks
    - FILE_SHARE_DELETE flag
    - Level II oplock support (independent from kernel)
    - SMB Leases support
    - Mandatory locking support
  - Durable/Persistent File handle support in GPFS
  - SIDs in NFSv4 ACLs
  - Delete on close support in GPFS
  - More .....

Questions ?

Thank you !

Legal Information and Trademark

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:
IBM, IBM Logo, on demand business logo, GPFS
The following are trademarks or registered trademarks of other companies.
Linux is a registered trademark of Linus Torvalds.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
POSIX is a registered trademark of the IEEE
Storwize and the Storwize logo are trademarks or registered trademarks of Storwize Inc., an IBM Company.
* All other products may be trademarks or registered trademarks of their respective companies.

IBM may not offer the products, services or features discussed in this document and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
The information on the new products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information on the new products is for informational purposes only and may not be incorporated into any contract. The information on the new products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of any features or functionality described for our products remains at our sole discretion.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.