

# NFS version 4.1

Spencer Shepler, Sun Microsystems

Mike Eisler, Network Appliance

Dave Noveck, Network Appliance

# Contents

- Comparison of NFSv3 and NFSv4.0
- NFSv4.1 Fixes and Improvements
  - ACLs
  - Delegation Management
  - Parallel Opens
  - Asynchronous Blocking Locks
  - Callbacks
- NFSv4.1 New Features
  - Sessions
  - Trunking
  - pNFS
  - Directory Delegations and Notifications
  - Non-regular file delegations
  - Global Namespace
  - Data Retention
- Status
- Links

# Comparison of NFSv3 and NFSv4

## NFSv3

- A collection of protocols (file access, mount, lock, status)
- Stateless
- UNIX-centric, but seen in Windows too
- Deployed with weak authentication
- 32 bit numeric uids/gids
- Ad-hoc caching
- UNIX permissions
- Works over UDP, TCP
- Needs a-priori agreement on character sets

## NFSv4

- One protocol to a single port (2049)
- Lease-based state
- Supports UNIX and Windows file semantics
- Mandates strong authentication
- String-based identities
- Real caching handshake
- Windows-like access
- Bans UDP
- Uses a universal character set for file names

# NFSv4.1 Fixes and Improvements

## Access Control Lists (ACLs)

- NFSv4.0 ACLs are derived from Windows 2000
- NFSv4.1 ACLs add better (read: Windows-like) support for ACL inheritance
- Like Windows, NFSv4.1 ACLs now explicitly separate “discretionary” ACL (dACL) on a file from its “security” ACL (sACL)
  - dACL: ALLOW/DENY Access Control Entries (ACEs)
  - sACL: AUDIT/ALARM ACEs

# NFSv4.1 Fixes and Improvements

## Delegation Management

- Delegation Re-Acquisition
  - NFSv4.0 tied delegations to OPEN
  - Only by polling via another OPEN could a client re-acquire delegation
  - NFSv4.1 adds a separate WANT\_DELEGATION operation to allow client have delegation “pushed” via callback when available
- Delegation Recall
  - Often a server wants to recall delegations due to general resource constraints
  - Server has no idea which delegations to recall
  - NFSv4.1 adds a callback that asks client to retain up to N delegations and return the rest

# NFSv4.1 Fixes and Improvements

## Parallel Opens

- NFSv4.0 requires a given “open owner” to serialize all OPENS on the same file
- NFSv4.1 allows parallel OPEN, and includes a generation number to indicate which OPEN was executed last

# NFSv4.1 Fixes and Improvements

## Asynchronous Blocking Locks

- NFSv4.0 requires client to poll for a blocking byte range lock
- NFSv4.1 adds a notification callback to indicate when a lock is available

# NFSv4.1 Fixes and Improvements

## Callbacks

- NFSv4.0 requires callbacks to recall delegations
  - A separate connection path was used
  - Not firewall friendly
- NFSv4.1 requires clients to create connections and hand them to servers to be used for callbacks



## NFSv4.1 New Features Sessions

- NFSv[234] do not support exactly once semantics
  - Approximated with a reply cache that has indefinite size
- Approximation drawbacks:
  - The unbounded size makes persistence across failover and reboot impractical
  - Prone to cache misses
  - Prone to false positives
    - bad clients
    - reply caches that are too big



## NFSv4.1 New Features Sessions (continued)

- NFSv4.1 achieves exactly once semantics via a “slot table”
- Slot table created when session created
- Specific number of slots
- NFSv4.0 requests consist of a list of operations in the COMPOUND request
  - NFSv4.1 mandates each COMPOUND start with a SEQUENCE operation that carries a slot number and sequence number (per slot)
- Each slot corresponds to a request in progress or a completed request
- Each request is either a retry of the last executed request, or a new request with a sequence number exactly one higher then previously executed
  - This series OK:
    - SEQUENCE slot 1, seq# 2, SEQUENCE slot 1, seq#2, SEQUENCE slot 1, seq#3
  - This series not OK:
    - SEQUENCE slot 1, seq# 2, SEQUENCE slot 1, seq#4, SEQUENCE slot 1, seq#5



## NFSv4.1 New Features Sessions – Slot Table Example

slot number	0	1	2	...	29	30	31
sequence number	100	200	150	...	25	125	175
cached reply	SEQ., PUTFH, WRITE	SEQ., PUTFH, RENAME	SEQ., PUTFH, OPEN, GETFH, GETATTR	...	SEQ., PUTFH, REMOVE	SEQ., PUTFH, READ	SEQ., PUTFH, LOOKUP



## NFSv4.1 New Features Session Management

- CREATE\_SESSION
  - negotiates the size of the slot table
  - negotiates persistence of slot table, enabling true exactly once semantics (EOS) through server reboot
- DESTROY\_SESSION
  - Client can indicate explicitly when it no longer needs a session (e.g. when unmounting)

Thus reply cache size and lifetime are bounded



## NFSv4.1 New Features Trunking

- `BIND_CONN_TO_SESSION`
  - Allows client to bind additional connections to the session
  - New connections may be to alternate network interfaces of a multi-homed server
- Trunking over multiple sessions also permitted
  - Enables the use of storage clusters where multiple server nodes have paths or replicas to the same data



## NFSv4.1 New Features Parallel NFS (pNFS)

- pNFS allows servers to stripe data of regular files across multiple storage devices
- A pNFS server consists of:
  - A metadata server (MDS) that implements the full NFSv4.1 protocol
  - One or more storage devices
- A pNFS client is an NFSv4.1 client that is prepared to directly access storage devices
- The pNFS client finds out about storage devices from the MDS via a new LAYOUTGET operation

# NFSv4.1 New Features

## pNFS (continued)

- LAYOUTGET returns a layout that describes the striping pattern for a given file
- layouts are recallable which allows pNFS servers to re-stripe a file if desired or necessary
- striping patterns can indicate if a some or all of a pattern has mirrors
  - clients are not required to construct mirrors
  - Thus pNFS offers RAID 0 and RAID 1+0

# NFSv4.1 New Features

## pNFS: Types of Storage Devices

- pNFS supports multiple Storage Device types (aka layout types)
- A layout can stripe a file over just one type of device
- The NFSv4 working group currently specifies three types:
  - files
    - The storage “device” is an NFSv4.1 server
  - blocks
    - The storage device is an iSCSI or FC target
  - objects
    - The storage device is an Object Storage Device (OSD) target
- Additional types require a standards-track specification
- If a client does not support a given device type it can issue I/O directly to MDS

# NFSv4.1 New Features

## pNFS: File Layout

- The file layout can work with or without a backing clustered file system
  - pNFS over a clustered file system is more optimal than conventional NFS access to clustered file system because the client is directed to the optimal node
- The file layout uses the same security model (authentication, authorization, access control) as NFSv4.1

# NFSv4.1 New Features

## Data Retention

- Intended to be compatible with SNIA XAM API
- retention: if set, forces a file to be retained for a specified period of time
- retention\_event: like retention for event-based retention
- retention\_hold: bit mask of up to 64 administrative holds
  - If any bit set, file is retained regardless of state of retention, retention\_event



## NFSv4.1 New Features

# Directory Delegations and Notifications

- Enabled by new GET\_DIR\_DELEGATION operation
- Directory Delegations are a read-only in NFSv4.1
  - optimal for workloads where directories are rarely updated
  - write delegations for directories are difficult because creating entries requires a directory entry offset and a file inode number which only the server can produce
- Directory notifications allow client to be notified of updates to directories
  - Similar to CIFS change/notify
  - Optimal for workloads where directories are updated more frequently
  - Allows server to push changes to name caches versus existing poll model
  - Notifications are asynchronous
    - Designers did not believe synchronous notification would scale
    - Asynchronous model no worse than existing NFS directory caching implementations

# NFSv4.1 New Features

## Non-regular file delegations

- The new WANT\_DELEGATION operation works on all types of files except directories
- Allows one to cache contents of symbolic links
  - Symbolic links are read-only content so delegations are very apropos

# NFSv4.1 New Features

## Global Namespace

- NFSv4.0 has a “referral” feature that allows a server to re-direct a client to another NFSv4.0 server
- NFSv4.1 builds on referrals to produce a more complete definition of multi-server global namespace
  - Defines lock and session state transitions
  - Indicates whether key attributes like inode number survive migration events

# Status

- Specification is nearing end of formal inspections
  - An aggressive schedule to be ready for working group  
Last Call by December, 2007
- Prototypes/Implementations from EMC, IBM, Linux Community, NetApp, Panasas, Sun
  - Several testing events per year, next two scheduled are at U. of Michigan (October, 2007), and San Jose (May, 2008)
- EMC, U. of Michigan, NetApp, Panasas, Sun contributing to open source (Linux, OpenSolaris)

## Links

- NFSv4 working group:  
[www.ietf.org/html.charters/nfsv4-charter.html](http://www.ietf.org/html.charters/nfsv4-charter.html)
- Current draft:  
<http://nfsv4-editor.org/drafts/drafts.html>
- Blogs
  - [nfsworld.blogspot.com](http://nfsworld.blogspot.com)
  - [blogs.sun.com/shepler](http://blogs.sun.com/shepler)