Linux Support of NFS v4.1 and v4.2

Steve Dickson
steved@redhat.com
Mar Thu 23, 2017

Agenda

- NFS v4.0 issues
- NFS v4.1 Supported Features
- NFS v4.2 Supported Features
- V4.2 Not Suported Features
- V4.x Other Features

NFS v4.0 – Issues

- Performance issues
- Callback issues
- Delegation issues
- Locking Issues
- Mount negotiations start with v4.2

NFSv4.1 - Features

- Sessions
  - Exactly Once Semantics.
  - Callback problem fixed.
- pNFS
  - File layout
    - Netapp
  - SCSI Layout
    - Linux
  - Flex File layouts
    - Primary Data

NFS V4.2 – Feature

- **Sparse File Support** – Sparse files are ones which have allocated or uninitialized blocks in the file. (1.4.3)
  - SEEK - Find the Next Data or Hole
    - NFS: Implement SEEK (Sep 2014)
  - READ_PLUS - READ Data or Holes from a File
    - No published upstream patches yet
    - Prototype patches improve performance with reading holes (See Anna's keynote graph).

NFS V4.2 – Feature

- **Space Reservation** - Creating holes in files by transferring just the metadata over the network
  - ALLOCATE - Reserve Space in A Region of a File
    - nfs: Add ALLOCATE support (Jul 2015)
  - DELLOCATE - Unreserve Space in a Region of a File
    - nfs: Add DEALLOCATE support (Jul 2015)

NFS V4.2 - Feature

- **Server-side copy** – Provides a mechanism for the NFS client to perform a file copy on the server without the data being transmitted back and forth over the network.
  - COPY - Initiate a server-side copy
    - NFS: Add COPY nfs operation (Feb 2017)
  - Clone - nfs42: add CLONE proc functions (Sep 26 2015)
  - Huge performance gain!

NFS v4.2 - Feature

- **Security labels** - A file object attribute allows the server to store labels on files, which the client retrieves and uses to enforce data access. (1.4.6)
  - SECURITY_LABEL
    - NFS: Client implementation of Labeled-NFS (May 2013)

NFS v4.2 – Not Supported

- **IO Advise** - Applications and clients want to advise the server as to expected I/O behavior. (1.4.2)
  - IO_ADVISE - Application I/O access pattern hints
    - not implemented

- **Application Data Block (ADB) Support** - Some applications treat a file as if it were a disk and as such want to initialize (or format) the file image. (1.4.5)
  - WRITE_SAME - WRITE an ADB Multiple Times to a File
    - not implemented

NFSv4.x – Other features

- LAYOUTSTATS
  - Performance statistics from DS to MDS
  - FlexFile use to load balance between DS(s)
- Session Trunking
  - The use of multiple connections between a client and server in order to increase the speed of data transfer.
  - pNFS File and Flexfile layout
- NFSoRDMA
  - NFS4.1 support
  - Kerberos Support

Questions

References

- RFC 5661 - https://tools.ietf.org/html/rfc5661
- RFC 7862 - https://tools.ietf.org/html/rfc7862
- Vault 2015 Talk - http://events.linuxfoundation.org/sites/events/files/slides/vault2015.pdf

NFSv4 Beyond v4.2

Part 2 of Road Map of the features in NFS v4.1, v4.2, and beyond

Dave Noveck
Netapp
Vault Conference
March 23, 2017

Contents

- A tiny bit about the NFSv4 Working group and the IETF process
- NFSv4 beyond v4.2 as approved
  - GSSRPCv3 (used by some v4.2 features but separate from them)
  - New Extension Model
  - Currently Pending Extensions
- Other working group work (mainly focused on NFS performance)
  - Revival of NFS/RDMA
  - Higher-performance pNFS options (allowing use of NVMe, RDMA)
  - Miscellaneous trunking issues

Working Group and IETF Process

- Front end (NFSv4 Working Group)
  - Cycles of drafting, review, update
  - No time limits. Process continues until everyone is ready to have Working Group Last Call for final working group review
  - Despite a seemingly unworkable process, things do get done.
- Back end (IETF superstructure)
  - Review by Area Director, IESG; RFC Editing process
  - Back end process can take a year or more
- Good news is that substantial change rarely happens in the back end
  - It is pretty safe to continue prototyping and do preliminary implementations based on final WG draft

GSSRPCv3

- Published as RFC7861 in Nov. 2016 (same day as NFSv4.2)
- Supports Mandatory Access Control for Labeled NFS
  - GSSv3 provides support for subject labels
  - Labeled NFS provides support for object labels
- Another motivation was inter-server case of server-side copy.
  - Allows target server to read file on behalf of user requesting copy.
  - No trust relationship required between source and target servers.

New Extension Model

- No V4.3, for a while at least
- However, optional extensions to V4.2 will be possible.
- Such extensions can define:
  - New attributes
  - New operations
  - New flags or switch cases in existing operations
- New extension model described in *draft-ietf-nfsv4-versioning-09*
  - Document ready for IETF superstructure to deal with
- Two extensions are ready for approval. (see Next Slide)
  - More can be developed since v4.2 will be extensible.

Pending Extensions
Slide One of Two

- Extended Attributes
  - OTW support for size-limited extended attributes (such as Linux xattrs)
  - Without this, copying a fIle with xattrs using NFS loses data ☹
  - Separate from named attributes:
    - Those are based on multi-stream files in Windows and Solaris
  - Document ready to be considered by IETF superstructure
  - Upstream client-side patches exist for this
  - No upstream server-side patches for kernel-based NFS server
  - There are Ganesha patches for server

Pending Extensions
Slide Two of Two

- Umask attribute
  - Allowing inheritable NFSv4 ACLs to override the umask.
  - Passes umask separately from mask attribute on file creation
    - Without this, permission inheritance over NFSv4 is broken,
  - Document ready to be considered by IETF superstructure
  - There are upstream patches for both client and server parts of this.
- These two extensions and versioning document will go forward into the back-end process together.

Revival of NFS/RDMA
Background

- NFS got an early start on RDMA
  - Working group finished with docs in 2007; published in 2010
- Unfortunately,
  - Netapp changed its priorities and lost interest in RDMA
  - Tom Talpey, the driving force behind NFS/RDMA, was laid off
  - Documents were finished off in a rush and implementation lagged
  - Tom went to Microsoft and created SMB Direct
- As a result,
  - Documents were not clear enough to base new implementations on.
  - The protocol had performance problems that SMB Direct did not have
- Working group decided to revive NFS/RDMA

Revival of NFS/RDMA
Getting a Working Transport (Slide One of Two)

- Goal was to revive existing (Version One) transport.
  - Existing XDR was to be used
  - Performance issues were to be left for later
  - Also, error reporting could not be fixed due to ban on XDR changes
  - Two existing documents needed to revived/cleaned-up and one new one written.
- Rfc5666bis
  - Extensive cleanup of RFC5666
  - Clarify requirement for Upper Layer Bindings for individual protocols
  - Got rid of obsolete, never-implemented features
  - Document has just been approved by IESG.
    - After that, RFC editing

Revival of NFS/RDMA
Getting a Working Transport (Slide Two of Two)

- Draft-ietf-nfsv4-rpcrdma-bidirection
  - Needed new feature
  - Allows callbacks over RDMA, to support NFSv4.1
  - Document just approved by IESG.
  - Being worked on by RFC Editor
- Rfc5667bis
  - Also needed a major cleanup
  - Needed to be updated to meet requirements for Upper Layer Bindings
  - Document finishing up working group process

Revival of NFS/RDMA
Addressing Performance Gap vs. SMB Direct

- Performance gaps of concern
  - Need for better trunking support (see Trunking Slides)
  - Remote Invalidation (supported in Version Two)
  - Message Continuation (supported in an extension to Version Two)
- Near-term approach for performance gaps
  - Experimental draft in process of becoming working group document
  - Characteristic negotiation using CM private data
  - Upstream patches for client and server
  - Allows a simple form of remote invalidation
  - No message continuation but need for it is lessened by ability to negotiate larger receive buffers

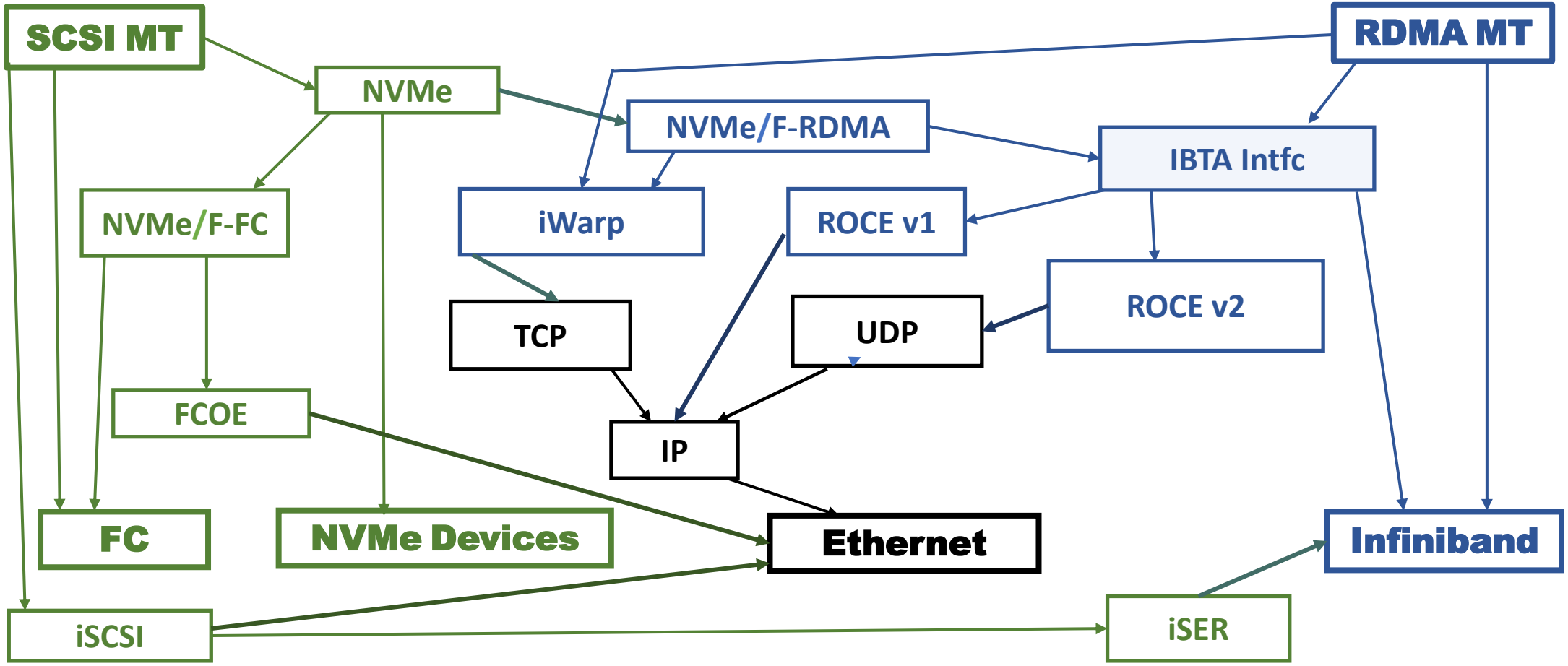
Revival of NFS/RDMA
Advancing beyond Version One

- Everything on this slide not yet an official working group document
- Base Version Two
  - Provides support for remote invalidation
  - Larger default buffer size (1K → 4K)
    - Ability to negotiate a larger value.
  - Version designed to be extensible
  - Defined in an individual submission; should be ready for promotion soon.
- Version Two Extensions
  - Message Continuation
  - Send-based Data Placement
    - Eliminates one inter-node round trip on an NFS WRITE.
    - Also a big help where remote invalidation not available (e.g. User-mode server)
  - Defined in an individual submission; discussion not far along

pNFS Mapping Types for Higher Performance

- SCSI mapping type **(Green MT in Diagram Slide)**
  - Basically, a restatement of existing block mapping type, but …
    - It has a new code and so is distinct
    - Scsi-to-NVMe mapping can be use to enable use with NVMe and NVMe/f ☺
  - Document has been with IETF superstructure for over a year
  - Should be published any month now.
  - Can be realized by **FC, NVMe Devices,** or **Ethernet** (via **FCOE**)
  - Can also be realized as RDMA fabric by **Ethernet** or **Infiniband** using **NVMe/F**
- RDMA-based mapping type **(Blue MT in Diagram Slide)**
  - Layouts could designate area in a remote memory.
    - Could access /modify data using RDMA Read and Write
  - Right now it is just a notion
  - Will take work to make it into an idea and then a submittable draft.
  - Can be realized by **Ethernet** (via **iWarp** or **ROCE**) or **Infiniband**

High-performance pNFS Possibilities



Trunking to Enable Higher Performance
Slide One of Two

- Types of trunking in NFSv4.1
  - Session Trunking
    - Multiple connections (potentially to different addresses) as part of same session.
  - Clientid Trunking
    - Multiple sessions supporting a single client; intended for clustered servers
- Reasons for Trunking
  - To get benefit of multiple wires/adaptors
  - With clustered servers, get benefit of multiple server nodes working
    - This is more suitable to client-id trunking than to session trunking used in Linux client
    - For data access, pNFS can fill the gap
    - High-intensity metadata access might need future work.
  - Get hw parallelism within adapter by using multiple queue-pairs/connections.

Trunking to Enable Higher Performance
Slide Two of Two

- Current Linux client issues with trunking
  - No trunking in the non-DS case (MDS and no PNFS use)
    - Lack of address list to drive trunking decisions
  - No support for clientid trunking
  - Limited trunking of multiple connection to same address
    - Depends on duplicates within an address-list
- Path discovery for trunking
  - Could substitute for the missing multipath_list4 in the non-DS case
  - Unclear whether relying on DNS is adequate
  - There is an individual submission under discussion
  - Not clear how this will be resolved