Reasons to Migrate from NFS v3 to v4/ v4.1

Manisha Saini
QE Engineer, Red Hat
IRC #msaini

Agenda

- What V3 lacks
- What’s improved – V4?
- Walk through on NFSv4 enhancements
- Sum up NFSv3 vs NFSv4
- Quick Overview -NFS Ganesha
- How we are moving forward?

NFSV3 -What is the Problem?

- Stateless -downside to performance and lock management issues
- Firewall -Separate port for portmapper,mountd,statd and nfsd servers
- Handling of locks- addition of NLM
- Supports only Posix ACL
- Handling of single operation per RPC

NFSV4 -What’s improved?

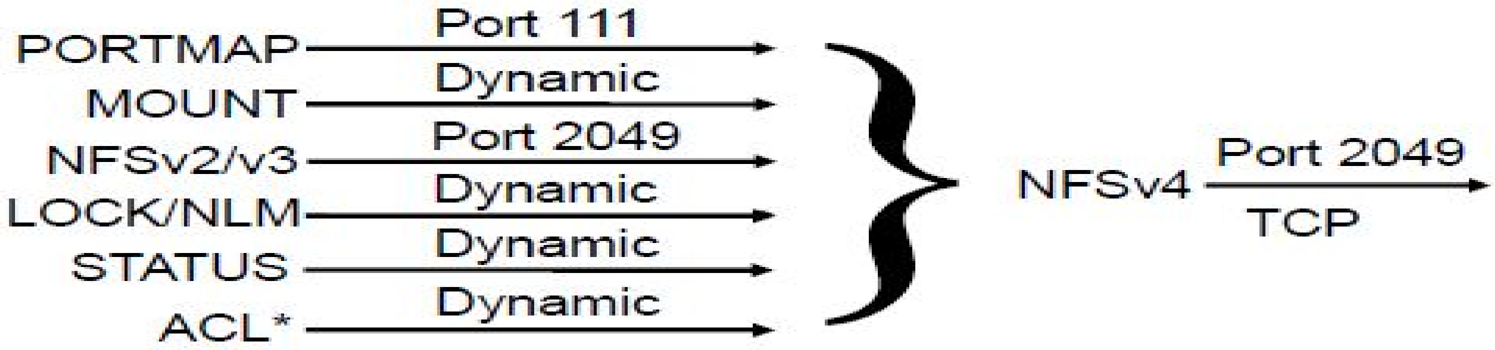
- Stateful
- Firewall
- Pseudo filesystem
- Delegation
- Locking operations(open/read/write/lock/close) are part of the protocol proper.
- NFSv4’s built-in lock leasing, lock timeouts, and client-server negotiation on recovery
- Integrated support for ACLs
- Support for NFSV4.1/pNFS
- Compound RPC

NFSV4-Stateful

- Maintain state of open files: OPEN and CLOSE
- Guaranteed consistency
- Call back /Recall functions
- Keep track of past request
- Eliminates useless write through
- Improves File locking

NFSV4-Firewall Friendly

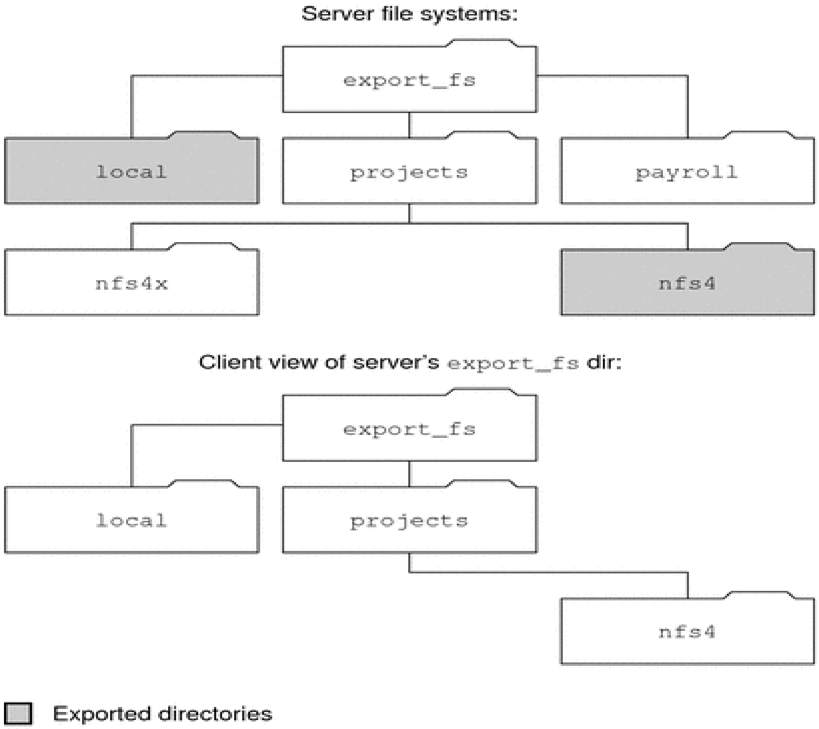
- Uses Single TCP connection
- Eliminates the need of port mapper interaction
- Well defined destination TCP port -2049



Pseudo Filesystem

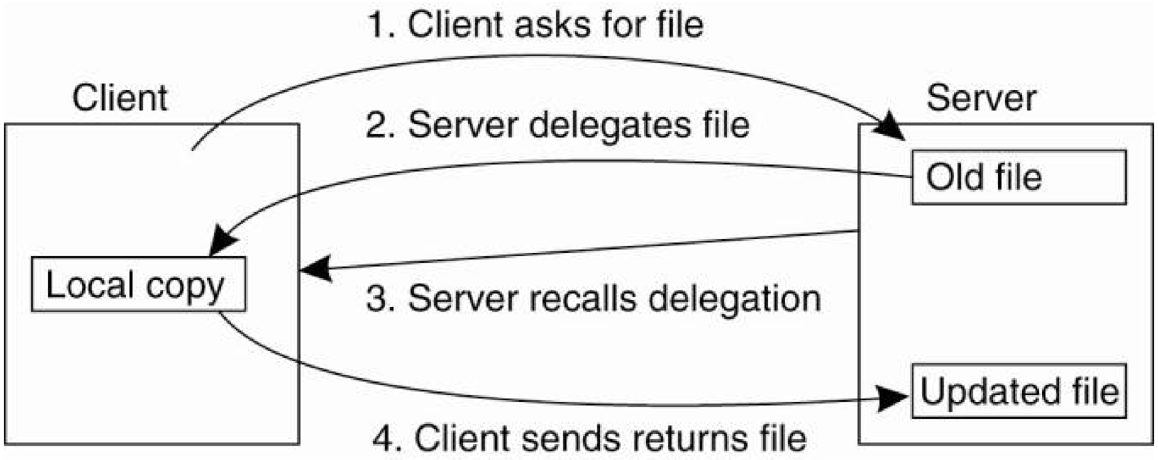
- servers create and maintain a pseudo-file system
- Generates the pseudo fsid

Eg-
Server Export:
    /export_fs/local
    /export_fs/project/nfsv4



Delegation in NFS Version 4

NFS version 4 provides both client support and server support for delegation. Uses NFSV4 callback mechanism to recall file delegation.



Delegation contd...

Advantages -

- Cut down the scope of revalidation requirement each time
- Reduces network traffic and thereby improves performance on the client and the server
- Have the access pattern of the file before providing delegation
- Server may recall the delegation at any moment of time when other client opens a file

Challenges -

- Callback daemon uses a dynamic port number. Therefore, the server might not be able to traverse a firewall, even if that firewall enables normal NFS traffic on port 2049.
- Conflicts caused by clients which are running lower version of NFSV4 in parallel.In that case an NFS server can only initiate recalls to the client that is running NFS version 4.

Leasing over HA

Issue with v3 locking-

- If client Fails,Server doesn’t get to know. Can cause complication and ambiguity issues
- If server Fails,Client may get stale file handle and problems related to locking

How it got resolved in V4-
Leasing

- Client and server are aware of each other state.
- If Client fails,Server reclaims the lock after the end of grace period to serve the other client locking request
- If Server fails,Sever are put into grace period and client to reclaim their existing locks
- During the grace period, the server reject READ and WRITE operations and non-reclaim locking requests (i.e., other LOCK and OPEN operations) with an error of NFS4ERR_GRACE.

More on Locking

- In NFSv4, locking operations are part of the protocol rather than separated out as it is in NFSv3 (NLM)

- Client must maintain contact with an NFS Version 4 server to continue extending its open and lock leases

- NFSV4 server and client can run on single server whereas in NLM cannot be run on same machines as it uses different ports

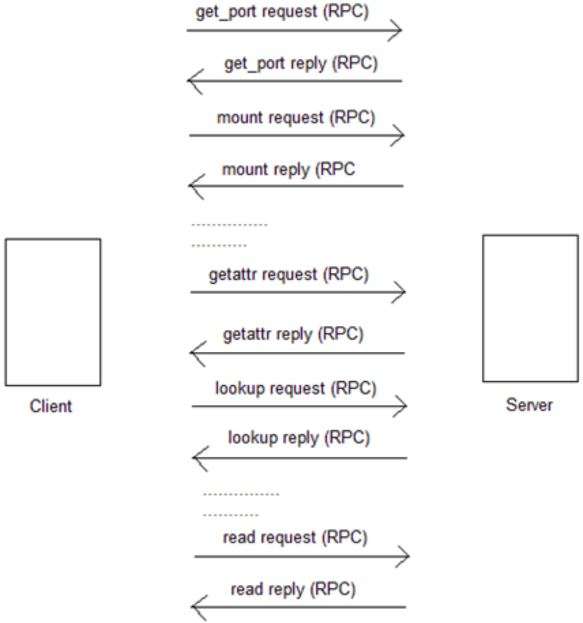
Access Control Lists

- The NFSv4 protocol includes integrated support for ACLs.
- Access Control Lists – similar to Windows ACLs
- Mapping of NFSv4 ACLs to POSIX ACLs is done in order to support with linux filesystem and store POSIX ACLs in the filesystem
- NFSv4 ACLs are richer than POSIX draft ACLs
- User and group information is stored in the form of strings, not as numeric values

Compound RPC

Problem with V3-

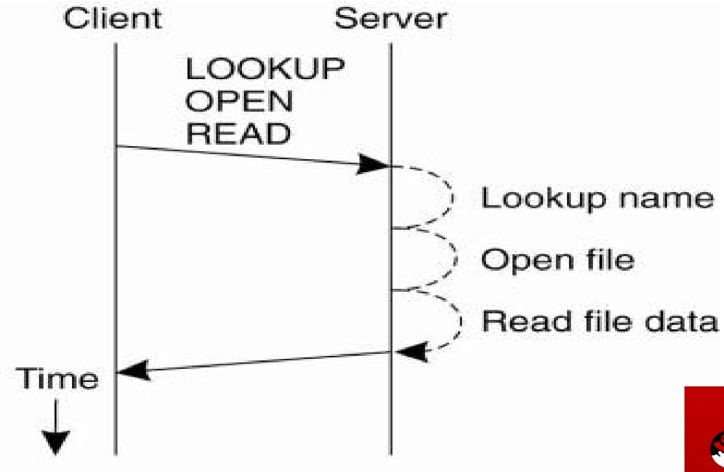
- Multiple short sharp RPC request between client and server. In order to mount and read a portion of a file , could take 20+ exchanges



Compound RPC Contd..

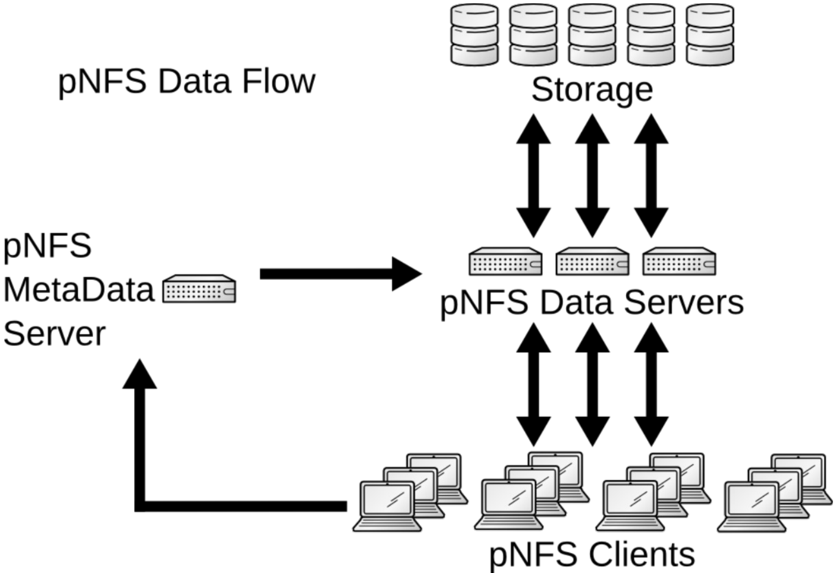
Improvement in NFSV4-

- Support Compound procedures by which several operations can be grouped into single RPC
- Better performance in wide area network



Parallel NFS

- Support for pNFS came in as part of the NFS v4.1
- pNFS works by separating the duties of handling metadata and file data. The Metadata Server receives requests for file data and directs the clients to communicate with the correct Data Server.



pNFS Contd...

- Provide 3 types of storage-access protocols
  - Files (NFSv4.1)
  - Block (iSCSI, FCP)
  - Object (OSD 2)
- Removes the performance bottleneck
- Red Hat Enterprise Linux 7 supports the files layout type being included as a technology preview.

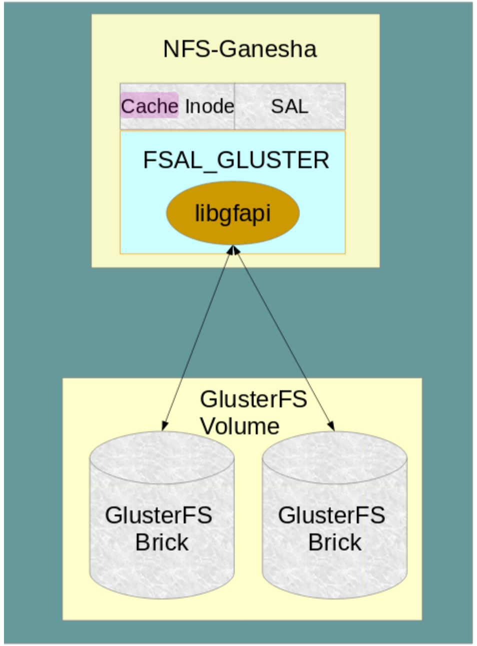
Difference between NFS V3 and NFS V4

NFSV3	NFSV4
● Stateless connection between client and server	● Statefull connection between client and server
● Exports:All exports are mounted separately	● Export:All exports can be mounted together in a directory tree structure as part of a pseudo-filesystem.
● Lock:Uses auxiliary protocol for file locking I.e NLM.NLM is stateful in which server LOCKD keeps the track of locks	● Lock:In NFSv4, locking operations are part of the protocol
● Lock:Permanent locking	● Lock:Lease based locking
● Communication:One operation per RPC	● Communication:Multiple related operations on a file are grouped into a single RPC call/response
● Firewall to know all the ports on which portmapper, mountd and nfsd servers are listening on. (can create problem when client or server are outside network)	● Mandates that all traffic (now exclusively TCP) uses the single well-known port 2049.
● Have Separate protocol for NLM,Mount,ACL,Stat,NFS	● No sidebrand protocol

Quick Overview-What is NFS-Ganesha?

- Runs on User address space,protocol-complaint NFS file server
- Uses FSAL and Libgfapi support to run on glusterfs server
- Supports v3, v4.0 , v4.1, pNFS
- Integrated HA solution using pcs,pacemaker and corosync(as of now) for gluster volumes
- Dynamically export/unexport entries using D-Bus mechanism.
- Manages Huge meta-data and data cache (Cache inode)
- Supports a lot of other fIlesystem like Gluster ,CEPH(cephFS / RGW), GPFS, Lustre

NFS Ganesha Architecture



Where are we heading with Glusterfs+NFS-Ganesha?

- Readdir performance improvement
  - introduced readdir chunking in nfs-ganesha-2.5
  - xreaddir plus support in gfapi (from glusterfs 3.13 onwards)
- Delegation support in nfs-ganesha 2.7 and glusterfs-4.0
- HA using ctdb -- storhaug 2.0
- Addition of AIO nfs-ganesha 2.7 and glusterfs-4.x

References

Mailing lists:
gluster-users@gluster.org
gluster-devel@gluster.org

IRC:
#ganesha #gluster and #gluster-dev on freenode

Links (Home Page):
https://github.com/nfs-ganesha/nfs-ganesha/wiki
http://www.gluster.org

Question?

Thank You