NetApp®

Go further, faster®

NFS Server-side copy Implementation

Manjunath Shankararao (rudra@netapp.com)

Theresa Raj (traj@netapp.com)

Agenda

- Overview
- Implementation
- Protocol details
- Implementation details
- Testing tools
- Future enhancements

Overview

- Server side Copy
  - Leverages the host from performing Copy
  - Freedom to implement any File Copy Protocol (FTP, HTTP, etc)
- Implications
  - More work for Server – Auth checks, Copy
- Complexities
  - Deduplication
  - Inter-cluster
  - Information distribution
  - Resilience

Implementation

- Based on Server-side Copy Internet-Draft [https://tools.ietf.org/html/draft-lentini-nfsv4-server-side-copy-04]
- Only Intra-cluster Copy supported
  - Copy across heterogeneous servers not supported
- Operation Numbers used are proprietary

Protocol details

- RPC
  - NFSv4-like
  - Non-NFS IANA Registered Program Number
    - Until Copy I-D is approved by IETF
  - Procedures – NULL & Compound
  - Security
    - AUTH_SYS support initially
    - Later support to RPCSEC_GSSv3

Protocol details

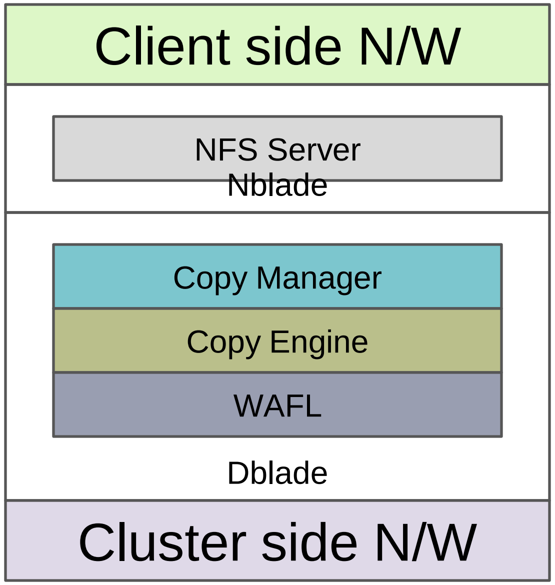
- NFSv4 like Protocol
  - Port number – 2049
  - Provides extensions for NFSv4.1 Protocol
    - Operation numbers are based from NFSv4.1
  - Server-side Copy Operation numbers after NFSv4.1 Operation numbers

Protocol details

- Selective Operations only
  - No Locking Ops
  - No Delegation Ops
  - Basic File Ops only
- New Ops
  - COPY_NOTIFY, COPY_REVOKE
  - COPY, COPY_ABORT, COPY_STATUS
  - CB_STATUS

Terminology

- Nblade – Network handling part of a Node
- Dblade – Storage handling part of a Node
- Copy Manager – Module handling Copy Operations
- Copy Engine – Module performing the actual Copy
- Intra Volume – Copy within the same data store
- Inter Volume – Copy between different data stores – Inter server/Intra server



Implementation details – NFS Server

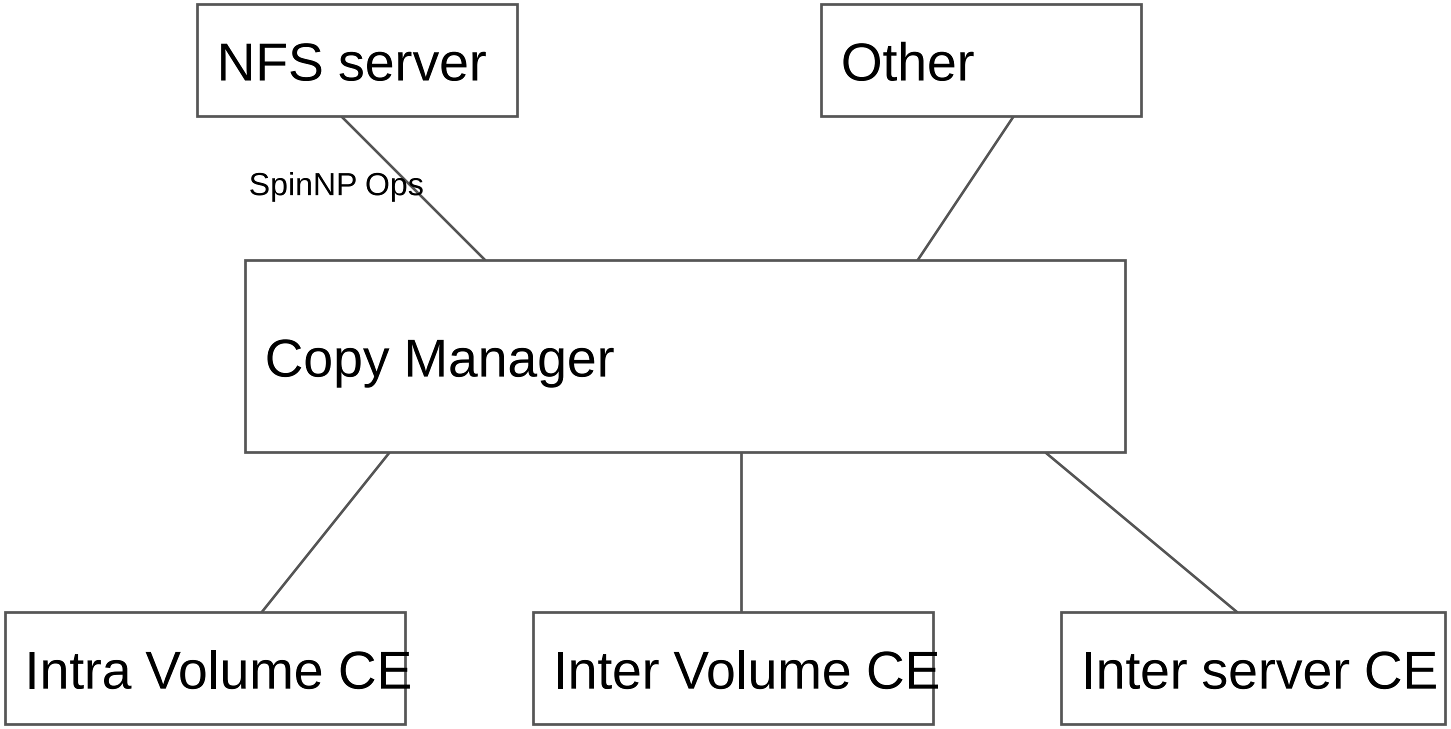
- Protocol validation
- Parameter Processing
  - Only IP support for source and dest
  - File handle => Internal file handle
  - Informs Copy Manager on Dblade owning the file handle of the Copy Op
  - Supports only COPY_GUARDED & COPY_SPACE_RESERVED Copy flags
- Maintains State information
- Deciphers Copy type based on file handles
- Copy Ops => Cluster Protocol Ops

Implementation details – Copy Manager

- Manages requests from Protocols & Copy Engines
- Source Node
  - Cluster Ops for Copy_notify & Copy_revoke
- Destination Node
  - Copy, Copy_status, Copy_abort
- Receives callback from Copy Engine after copy completion
- Passes Cluster Ops for Copy, Copy_status, Copy_abort to the appropriate Copy Engine
- Maintains Auth info
  - Internal file handles, Destination IP, User Credentials

Implementation details – Copy Manager (contd.)

- Fences Copy from invalid destinations
- No resilience for Volume migration, Source or destination reboots
- No disk level information

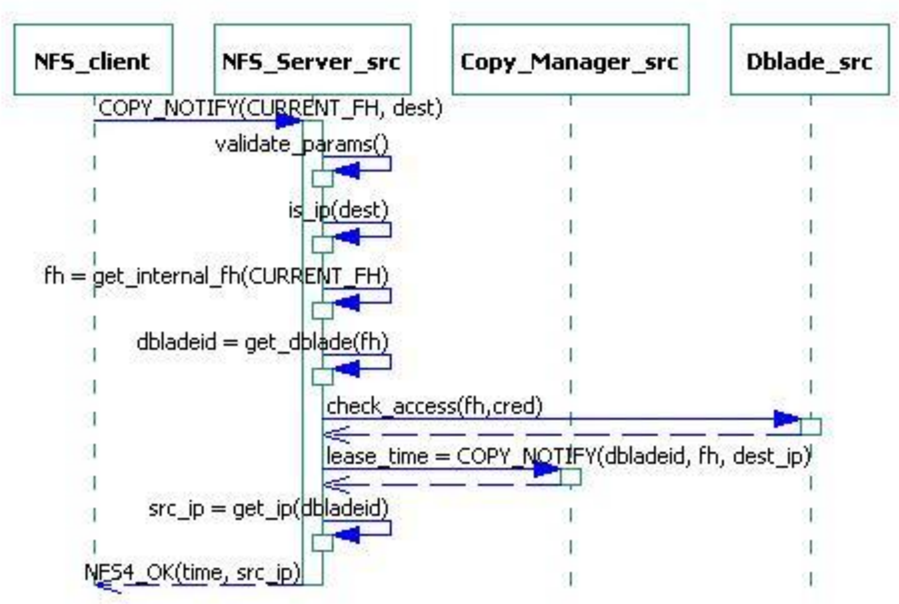


Implementation details – Copy Engines

- Dedupe aware intra volume copy engine
- Inter Volume Copy Engine
- Inter Server (Intra Cluster) Copy Engine
- Simple read, write loops
- Read performs auth verification at Copy Manager at the source

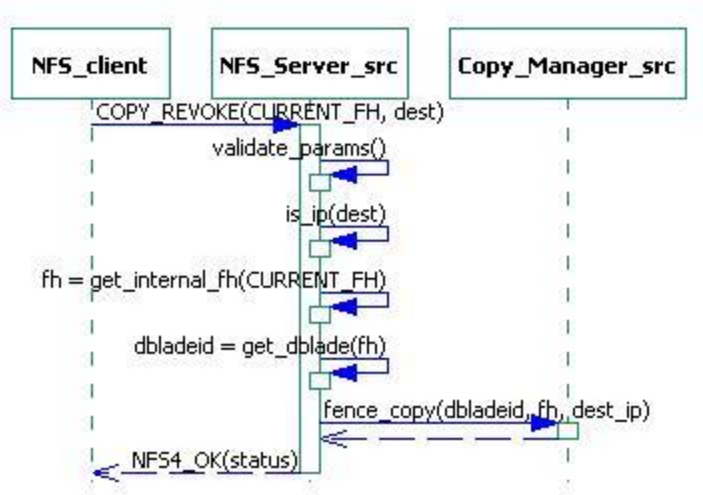
COPY_NOTIFY

- Inter-server copy
- For Intra-server copy also ?
- Populate auth @ src, to fence invalid reads


NFS_client
NFS_Server_src
Copy_Manager_src
Dblade_src
COPY_NOTIFY(CURRENT_FH, dest)
validate_params()
is_ip(dest)
fh = get_internal_fh(CURRENT_FH)
dbladeid = get_dblade(fh)
check_access(fh,cred)
lease_time = COPY_NOTIFY(dbladeid, fh, dest_ip)
src_ip = get_ip(dbladeid)
NFS4_OK(time, src_ip)

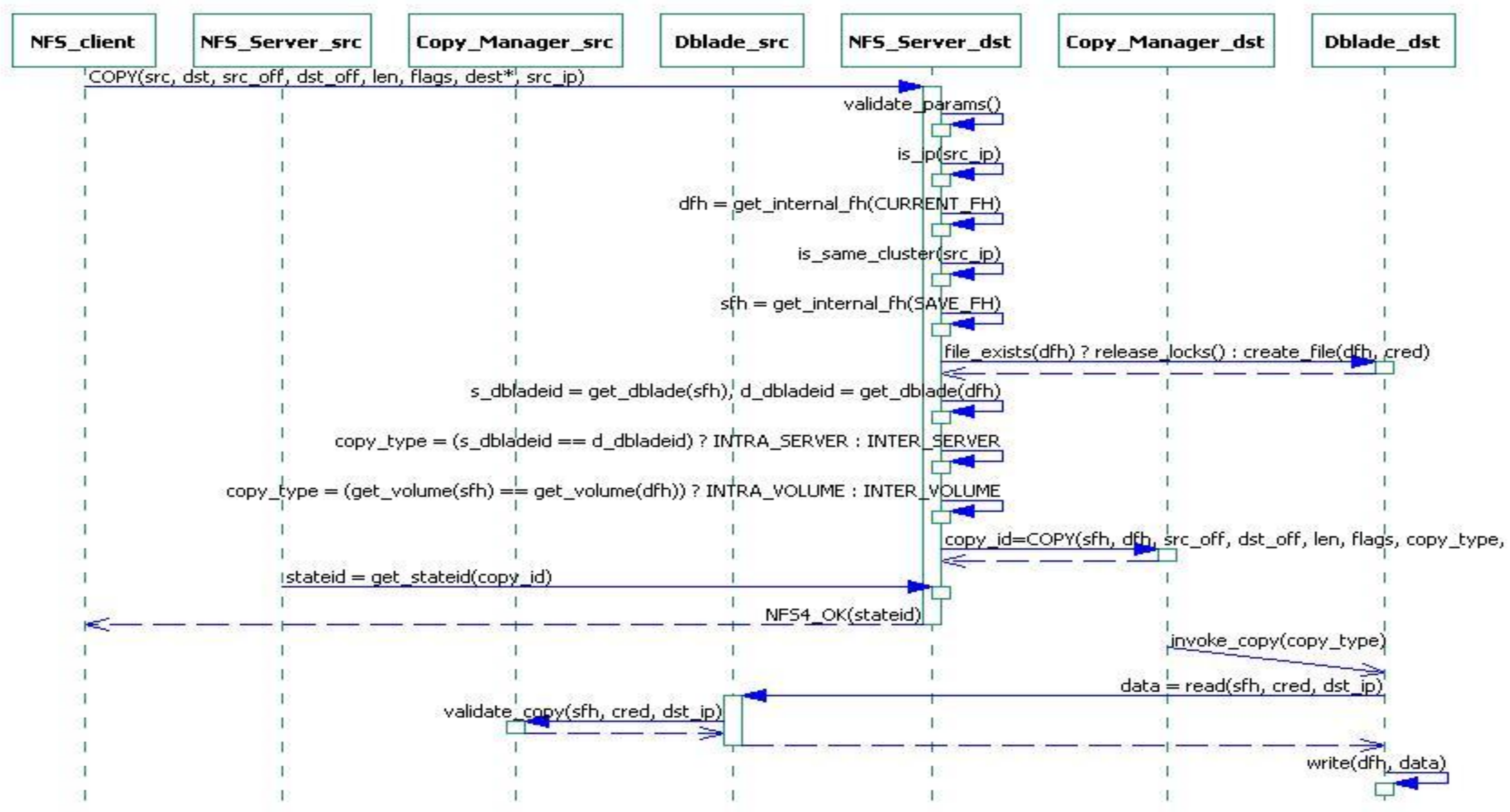
COPY_REVOKE

- Inter-server copy
- For Intra-server copy also ?
- Fence reads in case of long running copies, another way of abort


NFS_client
NFS_Server_src
Copy_Manager_src
COPY_REVOKE(CURRENT_FH, dest)
validate_params()
is_ip(dest)
fh = get_internal_fh(CURRENT_FH)
dbladeid = get_dblade(fh)
fence_copy(dbladeid, fh, dest_ip)
NFS4_OK(status)

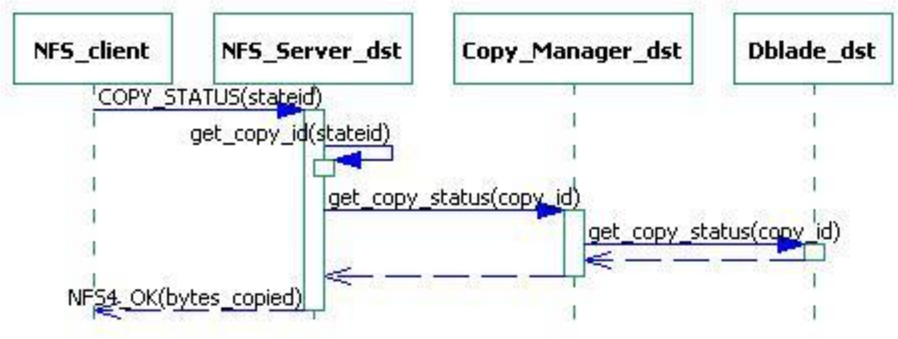
COPY

- No way to validate file handle in case of Inter cluster


NFS_client
NFS_Server_src
Copy_Manager_src
Dblade_src
NFS_Server_dst
Copy_Manager_dst
Dblade_dst
COPY(src, dst, src_off, dst_off, len, flags, dest*, src_ip)
validate_params()
is_ip(src_ip)
dfh = get_internal_fh(CURRENT_FH)
is_same_cluster(src_ip)
sfh = get_internal_fh(SAVE_FH)
file_exists(dfh) ? release_locks() : create_file(dfh, cred)
s_dbladeid = get_dblade(sfh), d_dbladeid = get_dblade(dfh)
copy_type = (s_dbladeid == d_dbladeid) ? INTRA_SERVER : INTER_SERVER
copy_type = (get_volume(sfh) == get_volume(dfh)) ? INTRA_VOLUME : INTER_VOLUME
copy_id=COPY(sfh, dfh, src_off, dst_off, len, flags, copy_type,
stateid = get_stateid(copy_id)
NFS4_OK(stateid)
invoke_copy(copy_type)
data = read(sfh, cred, dst_ip)
validate_copy(sfh, cred, dst_ip)
write(dfh, data)

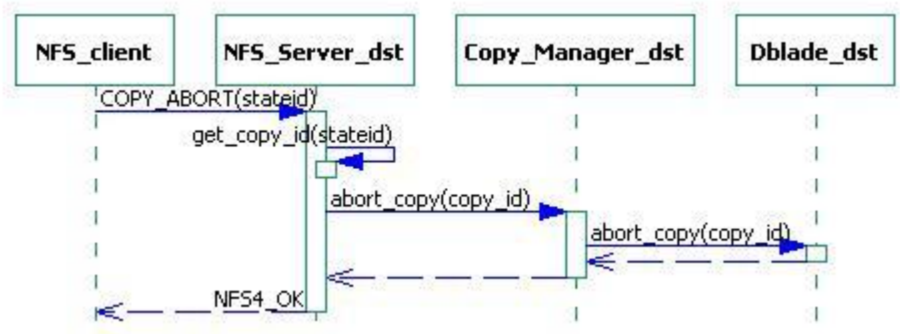
COPY_STATUS

- Query status of Copy Operations



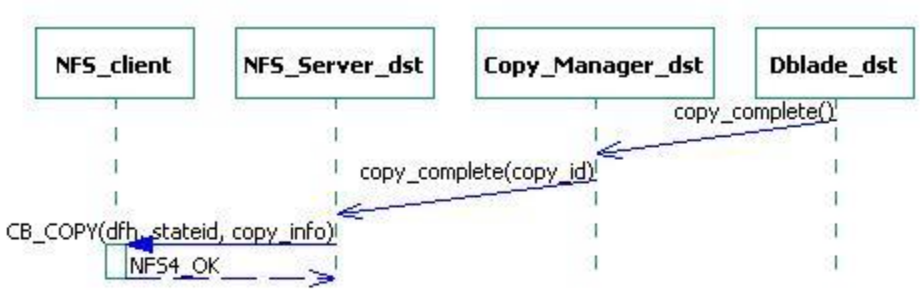
COPY_ABORT

- Stop long running Copy Operations



CB_COPY

- Callback to notify client of Copy Operation



Testing tools

- PyNFS changes for testing the Protocol
- Wireshark changes for debugging the Protocol

Future enhancements

- Support pNFS & Sessions in the new Protocol extension
- Support URL & hostnames for source & destination
- Support Inter-cluster Copy
- Copy Engine optimizations

Questions ?