State of NFsv4.1 specification
NFSv4 WG Meeting
IETF 68 – Prague
2007-03-19

Mike Eisler

email2mre-ietf@yahoo.com

Outline

- What changed between draft-...-08 and ...-09
- What changed between draft-09 and -10
- What will change in -11
- Formal review process

What changed between draft-...-08 and ...-09

- Terminology clean ups
  - clientid → client ID, deviceid → device ID, filesystem → file system, client → requester, server → replier, byte → octet
- Added a DESTROY_CLIENTID operation
  - Fails if there are sessions
- Added more explanatory text around Server Owners and trunking
- Cleaned up SECINFO/SECINFO_NO_NAME section to deal with RESTORE_FH
- Use the terms requester and replier instead
- Traded a “max slot” and “slot count” concepts for a single “highest slotid” concept
- Client IDs can now be created for specific pNFS and non-pNFS roles

What changed between draft-...-08 and ...-09

- LAYOUT_NFSV4_FILES renamed to LAYOUT4_NFSV4_1_FILES (to make it clear we are not describing NFSv4.0 storage devices)
- pNFS data types, operations arguments, operations results that have layout type specific contents now use explicit data types that consist of a layouttype4 followed by an opaque blob (with "body" in its name paying homage to RPC creds and verifiers)
- All new data types have "4" in their name
- Added prose around all the layout attributes to the file attributes chapter
- Added dacl, sacl, and mode_set_masked attributes
- Added automatic inheritance support
- Cleaned up stateid definition, defined special stateids more clearly
- Cleaned up state loss detection to reflect the session model and the status flags in the SEQUENCE result
- Change callback path testing to CB_COMPOUND/CB_SEQUENCE instead of CB_NULL

What changed between draft-...-08 and ...-09

- Introduced pNFS as an OPTIONAL feature, versus a proposal
- Discourage EXCLUSIVE4 OPEN/creates if persistent sessions are used.
  - Ban EXCLUSIVE4 if the layout_hint attribute is supported.
- Explicitly specified device ID to device address mappings as leased (and subject to revocation without a server reboot)
- MDS recovery clarifications:
  - Client has to keep a copy of modified data in memory even *after* a COMMIT but *before* a LAYOUTCOMMIT; or
  - Server cannot fail a LAYOUTCOMMIT in reclaim mode
    - Should not be an issue for NFSv4.1 storage devices
- Storage device recovery clarifications
  - draft-09 (and -10) now say that the best strategy for recovery is to write the data that has not been LAYOUTCOMMITted to the metadata server
  - Some feedback from reviewers that multipathing might permit client to write to secondary path for storage device

What changed between draft-...-08 and ...-09

- Explicitly defined pNFS terms: Unit, Pattern, Stripe, Stripe Width
- Renamed NFSv4.1 layout-type specific types to reflect their meaning.
  - E.g. nfsv4_file_layout_device4 → nfsv4_1_file_layout_ds_addr4
- Provided a more detailed example of a NFSv4.1 device (data server) list.
- Clarified STRIPE4_SPARSE versus STRIPE4_DENSE.
- Added EXCHANGE_ID (and DESTROY_CLIENTID) to list of ops an NFSv4.1 data server must support

What changed between draft-...-08 and ...-09

- DESTROY_SESSION can fail if there outstanding requests on the callback channel
- Added error code for STRIPE4_SPARSE: NFS4ERR_PNFS_IO_HOLE
- Change GET_DIR_DELEGATION results so that the operation can fail without stopping compound processing.
- Many fields of new data types changed to include a suffix that abbreviates data type name.
- Added optional “previous entry” to notifications of an addition to the directory

What changed between draft-09 and 10

- Re-clarified that delegation stateid from metadata server is appropriate for I/O to data server
- More field naming consistency issues
- Added directory filehandle to CB_NOTIFY arguments
- Put NFS filehandle in consistent places in all the NFSv4.1-only callback operations

What will change in -11

- Error codes
- Some pNFS feedback from Garth G and Rahul I.
- Locking and Delegations sections to be re-worked by Dave
- Set of issues in Issues Tracker
- What ever formal review reveals

Formal review process

- Editors believe that we need to ensure review on certain sections of specifications
- We have invited (primarily based on contributions to the NFSv4 WG mailing list) groups of reviewers for three sections/chapters
  - pNFS
  - Sessions
  - ACLs
- Process and Steps
  - Kickoff for each meeting: Editors act as moderators)
  - Reviewers get explicit sections to review and advise
    - Reviewers give feedback on whether the sections make sense
  - Volunteers sought for
    - Reader
    - Scribe
    - Reviewers
    - Moderators
  - Scribe records “defects” reported by Reviewers,
- Initially trying 3 sections/chapters
  - We will analyze effective of process in terms of defect finding and fine tune or re-visit as needed

Formal review process

- What if someone is really interested in reviewing a section but has not been invited?
  - Volunteer to Audrey VanBelleghem (Audrey.VanBelleghem_XX AT netapp.com)
    - delete _XX in the above
  - Formal reviews don't work well if there are a dozen reviewers
    - Editors are trying to ensure stuff gets reviewed
    - If more people want to review and we've no work for them,
    - Let Audrey know, and if there are multiple interested people who were not invited for a section, she'll put them in touch with each other.