NFSv4 Interim WG Meeting Sessions Changes 2006-09

Mike Eisler

email2mre-ietf AT yahoo.com

Scope and Outline

- Scope of presentation: summarize sessions changes since Eisler became an editor for NFSv4.1
- I-D reorganization
- Summary of issues from Issues tracker
- Describe what has changed
- Remaining issues

I-D Reorganization

- Sessions was in its own section
- Sessions is a core aspect of NFSv4.1 with many inter relationships with other core aspects
- Produced a new “Core Infrastructure” section with these major topics:
  - RPC and XDR *(includes RPC and RPCSEC_GSS security)*
  - COMPOUND and CB_COMPOUND
  - Client Identifiers
  - Security Service Negotiation
  - Minor Versioning
  - Non-RPC-based Security Services
  - Transport Layers
  - Session

Summary of Resolved Session Issues from Issues Tracker

- 3 Sessions -- Optional or mandatory
- 10 Cleanup needed for referencing to previous versions of sessions
- 26 NFS4ERR_RESOURCE
- 28 denial of service in NFSv4.x state management
- 31 Sessions text, NFS4ERR_RESOURCE, and reply limits
- 62 Sessions: Reserved slotid values needed
- 72 Sessions chapter needs work to integrate it into spec
- 73 Interrupting RPCs and sessions
- 74 Sessions-related errors need to be added to error lists
- 98 can callback traffic show up on any connection?

Changes

3 **Sessions -- Optional or mandatory**

- **Sessions are mandatory in specification**
- **Most operations (except CREATE_SESSION, DESTROY_SESSION, and BIND_CONN_TO_SESSION) must be prefixed with [CB_]SEQUENCE**

Changes

10 Cleanup needed for referencing to previous versions of sessions

72 Sessions chapter needs work to integrate it into spec

- A lot of RDMA material was covered in the RPCRDMA
  - NFSv4.1 just needs to describe the additional RDMA features
  - Clarifies relationship between slots and RDMA credits
  - Renames CB_RECALL_CREDIT to CB_RECALL_SLOT
    - Changes argument to be a target slot count to reach versus a target slot count to return (obviates races)
- Sessions section in Core Infrastructure chapter has these subsections:
  - Motivation and Overview
  - NFSv4 Integration
  - Channels
  - Exactly Once Semantics
  - RDMA Considerations
  - Sessions Security
  - Session Mechanics - Steady State
  - Session Mechanics – Recovery

More on Issue 72 Sessions chapter needs work to integrate it into spec

- Specification mandates that the first sequenceid after a session is created MUST be 1
  - Any other sequenceid produces a new error: NFS4ERR_SEQ_MISORDERED
  - Specification suggests caching (in reply cache) for each slot a sequenceid of zero with a reply consisting of NFS4ERR_SEQ_MISORDERED
- Any sequenceid in a [CB_]SEQUENCE that is less than the slot's sequenceid or two or more greater than the slot's sequenceid results in NFS4ERR_SEQ_MISORDERED

Session section is ~1500 lines versus ~2300 lines in July, 2006

Changes

- 26 NFS4ERR_RESOURCE
- 31 Sessions text, NFS4ERR_RESOURCE, and reply limits
  - One sub-issue is that NFS4ERR_RESOURCE has been misinterpreted has a variant of EAGAIN
  - Another is that NFSv4.0 servers are free to return NFS4ERR_RESOURCE whenever they want
    - client recovery can be harder
  - Issue has been resolved by:
    - obsolescing NFS4ERR_RESOURCE
    - replacing with new specific error codes for conditions the client has violated
      - NFS4ERR_REQ_TOO_BIG
      - NFS4ERR_REP_TOO_BIG
      - NFS4ERR_REP_TOO_BIG_TO_CACHE
      - NFS4ERR_TOO_MANY_OPS
      - NFS4ERR_RETRY_UNCACHED_REP
      - NFS4ERR_UNSAFE_COMPOUND
    - allowing client to and server to negotiate maximum size of a cached reply
    - allowing client to indicate if it wants reply cached

More on Issue 26 NFS4ERR_RESOURCE

- Non-idempotent requests and reply cache
  - The flaw in previous versions of NFS was relying on the server to determine what was idempotent and what wasn't (so much for the smart client/stupid server concept)
  - With COMPOUND this became a messy
  - Current v4.1 does not try to define what a non-idempotent operation is
  - Requester decides what it wants cached (via a Boolean in [CB_]SEQUENCE)
  - Replier caches as much of the reply as it can
  - If a requester retries a [CB_]SEQUENCE that it didn't previously ask to be cached, it gets NFS4ERR_RETRY_UNCACHED_REP
- NFS4ERR_UNSAFE_COMPOUND exists for situation where the server cannot be sure whether it will be able to return a filehandle via GETFH after OPEN
  - safe: PUTFH, OPEN, GETFH
  - unsafe: PUTFH, OPEN, GETATTR, long string of ops with variable length returns, GETFH
  - The moral for client:
    - When issuing OPEN by name (you knew that NFSv4.1 lets you open by filehandle right?) , keep it really simple

Changes

28 denial of service in NFSv4.x state management

- The issue is that attackers can trivially corrupt slot table simply by connecting to the server, and sending SEQUENCE operations (NFSv4.0 had a similar issue)
- NFSv4.1 adds:
  - SET_SSV: establishes a shared secret key
  - BIND_CONN_TO_SESSION
    - used for formally binding a connection to the session
    - uses shared secret key (SSV) for verifying that BIND_ comes from session leader
- One complaint is that these steps add complexity and overhead
  - CREATE_SESSION will be changed so that clients can optionally request enforcement of connection binding

Changes

62 Sessions: Reserved slotid values needed

- This was about detecting races between the reply to a client request and a server callback involving the affected object (e.g. delegation, layout)
- Resolution is that each CB_SEQUENCE carries a variable length array consisting of:
  - sessionid
  - variable length array of slotid/sequenceid pairs
- The sessionid is necessary in case two or more sessionds are bound to a clientid

Changes

73 Interrupting RPCs and sessions

- This is about what happens when a client process' system call is signal interrupted or a soft mount times out?
- Given the slot/sequenceid architecture in sessions, a requester MUST not give up on a [CB_]COMPOUND request.

Changes

74 Sessions-related errors need to be added to error lists

- Added sessions related errors like
  - NFS4ERR_SEQUENCE_POS (the [CB_]SEQUENCE operation is not first)
  - NFS4ERR_OP_NOT_IN_SESSION (the operation has been used before [CB_]SEQUENCE)
- The actual enumeration of each operation's permissible errors is a general task that is TBD

Changes

98 can callback traffic show up on any connection?

- The resolution is that only connections designated for the backchannel (via CREATE_SESSION and BIND_CONN_TO_SESSION) can carry callbacks
  - BIND_BACKCHANNEL no longer needed
  - BACKCHANNEL_CTL added to allow client to set/add RPC authentication parameters for callbacks
- A connection is allowed to be used for the operation and/or and back channels
- A connection can be bound to multiple sessions and by extension, multiple clientids

Remaining Issues

27 SESSIONS: Provide additional sessions discussion

- `` "serving suggestion" for implementing the callback dependent operation lookup ``

30 SESSIONS: Trunking issues with regard to sessions

- Via SSV client and server can verify that a connection refers to the same session. More needed; topic of Tom's presentation

44 SESSIONS: streamchannelattrs4 is not defined

45 SESSIONS: rdmachannelattrs4 is not defined

119 SESSIONS: make BIND_CONN_TO_SESSION and SET_SSV optional for clients