NFSv4.2 IETF89

Tom Haynes, NetApp

The Elephant in the Room

- 18 month release cycle
  - At just over 3 years
- < 100 pages
  - Hanging in at 94

Why is it taking so long?

- We needed some running code.
- Implementors were busy finishing NFSv4.1
  - Hardening
  - Enterprise ready
  - Buzzwords
    - But NFSv4.1 implementations 3 years ago were not ready for prime time!

Why do we need running code?

- Because we can find issues before we are locked into a spec
  - NFSv4.0 migration is a great example!
- Locks and Server Side Copy
  - Client opens source file and takes an exclusive lock
  - How can the destination access the file?

Is there some running code?

- Server-side Copy in Data ONTAP
  - Implementation is before the first draft
- Linux client has a prototype
- Labeled NFS is shipping in Linux
  - seVirt use case
  - No subject labels
- Other prototypes?

Current Issues (minor)

- Can the client indicate in SSC that it does not want an asynchronous copy?
  - Large files are all or nothing
    - Client may want to chunk it up
- Can the source indicate in the cnr_source_server that the source will copy bytes in consecutive order?
  - Allows the restarting of large copies which were interrupted

Current Issues (major)

- ADH as a hole does not sit well with all
  - Remove it being a hole and it looks a lot like WRITE_SAME
- Christoph views hole punching as being meta-data intensive
  - No need to be asynchronous
  - Some servers will need to write the 0s

Normative reference on RPCSEC_GSSv3

- End-to-end security requirement
  - Mandatory to implement
  - Deployment is optional
- IETF88 we decided to go down this route

- Note: RPCSEC_GSSv3 does not have a normative dependency on NFSv4.2

Next steps

- Tackle changes to SSC for the client
- Get ADH to be less intrusive on implementations that will not be doing it
  - Keep holes for 4.3?
- Encourage more prototypes