

# Stage0 Issues (52, 116)

Spencer Shepler

# Issue 52: Is client migration capable?

- For servers that provide multiple network paths to data, there may be a choice between notification of migration event and just “forwarding” requests for processing
- To allow for this, server needs to know if NFSv4.1 client is capable of migrating to another server or will just return an error to application

# Issue 52: proposal

- Using EXCHANGE\_ID, add an ‘eai\_flags’ field to indicate referral and migration support
  - EXCHANGE\_ID\_FLAG\_SUPPORT\_REFERER
  - EXCHANGE\_ID\_FLAG\_SUPPORT\_MIGR
- Reason for both: client may have differing support for migration: one within referral context and one in strict migration context

# Issue 116: parallel opens

- For NFSv4.0, granularity of `open_owner` is in direct proportion of the parallelism of OPEN requests
- OPENs are sequenced with `seqid` and therefore can limit or block a client's ability to service an application's `open()` requests (e.g. multi-threaded application)
- NFSv4.1's sessions removes need for `seqid` use for OPENs. Should be able to provide for higher degree of parallel OPENs for the client.

# Issue 116

- Two solutions proposed
  - 1) keep upgrade/downgrade logic in NFSv4.1 and expose the server's stateid.seqid
  - 2) remove upgrade on OPEN and make explicit the need for the client to manage state
- 2) seems to be the best choice
- Need to clarify if Posix file locking would be broken
- Is there a need for an UNLOCK\_ALL(lockowner) as Noveck proposes.