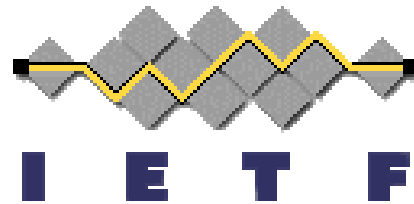

NFS Version 4 Locking

David Robinson

robinson@Eng.Sun.Com

**43rd IETF
Orlando
December 7-11, 1998**



Changes In Last Version

OPEN Procedure

- Performs share file lock
- Authenticates access based on server permissions and share lock
- Creates file if requested
- Only operates on regular files (use CREATE for everything else)
- Returns file handle and stateid

CLOSE Procedure

- Removes share file lock
- Does not remove record locks
- May fail if server does not support close with outstanding record locks



Changes In Last Version

Stateid

- 64-bit cookie returned by server to indicate file state
- OPEN, CLOSE, LOCK, UNLOCK affect stateid
- Maintained on a per owner per file basis
- Client sends to server on I/O or state changing operation to ensure both have a consistent view
- Stateid invalidated by
 - server reboot
 - server unilaterally revoking lock
 - server granting conflicting lock or I/O after lease expiration
- stateid == 0, no state has been established, server treats as least privileged for lock access
- stateid == all bits 1, reserved



Changes In Last Version

Crash recovery of state after expired lease (Shepler Scenario)

- Server may not allow a client to reclaim lock after crash if lease was expired and conflicting lock granted at time of crash
- Server must maintain on stable storage all clients with valid locks
- Server must remove client from stable storage list when conflicting lock granted
- Server only allows reclaim from clients on stable storage list
- Increasing state stored on stable storage decreases number of reclaim requests that need to be rejected

Server revocation of leases

- Within lease period - client loses all locks associated with that stateid
- Past lease period - client must revalidate all locks



Outstanding Issues

Cleanup of text

Zero length READ/WRITE versus separate RENEW procedure

Clientid and verifier semantics

Caching locks

Performance of polling on blocking locks