ORACLE®

Linux NFSv4 Client-side Migration

Chuck Lever, chuck.lever@oracle.com
Principal Member of Staff

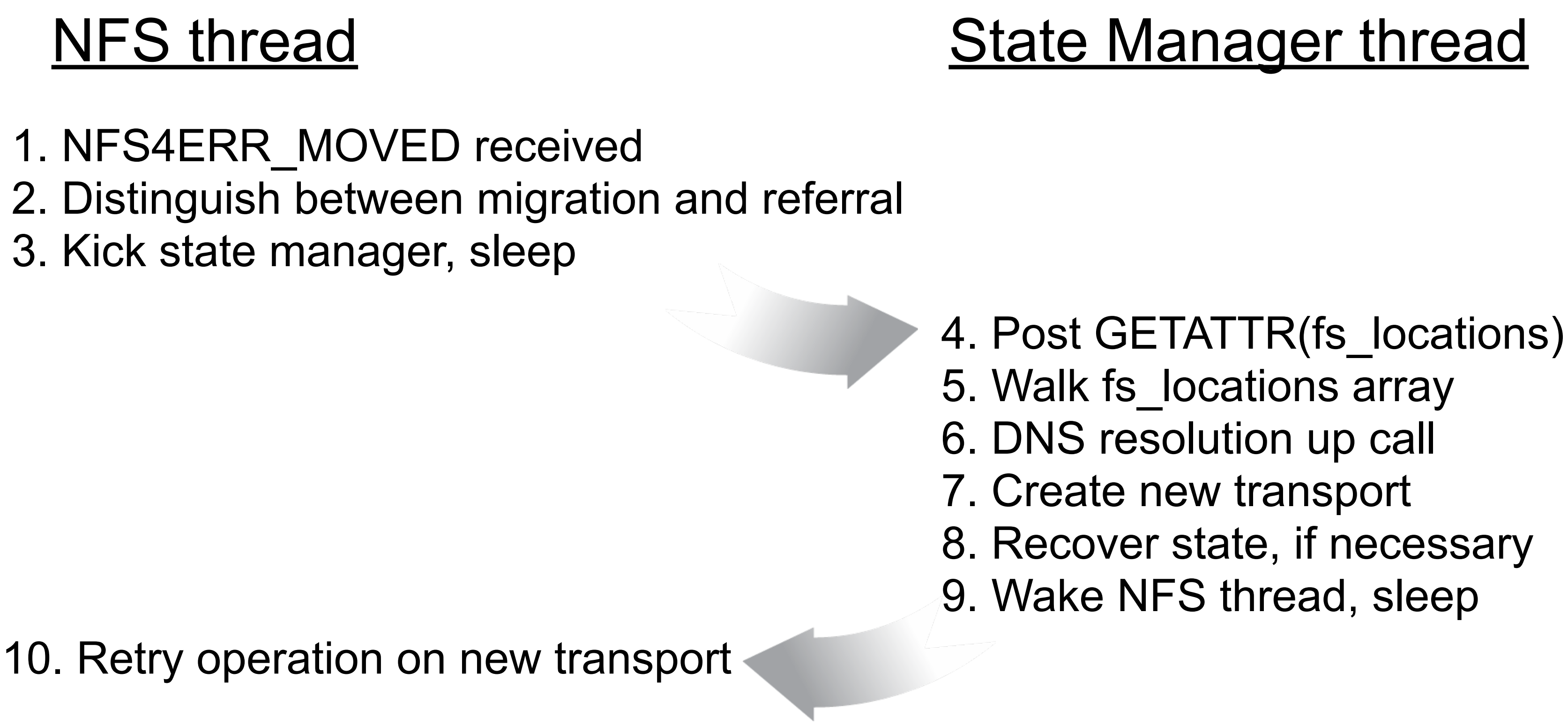
Rough, simplified flow of events

- 1. Server returns NFS4ERR_MOVED on object for which client already has state
- 2. Client responds with GETATTR(fs_locations)
- 3. Server returns array of [server, export] pairs
- 4. Client picks an fs_locations entry and connects a transport to that server
- 5. Server indicates if previous state is available; if not, client recovers state

Distinguishing Migration from Referral

- Boils down to whether client already has data and state for object
- NFS referral
  - No state
  - NFS4ERR_MOVED probably occurred during a LOOKUP
  - GETATTR(fs_locations) then under-the-cover mount
- NFS Migration
  - Have some state and/or data
  - NFS4ERR_MOVED occurred on other than LOOKUP
  - GETATTR(fs_locations) then replace RPC transport

Migration Recovery Control Flow



NFS thread

1. NFS4ERR_MOVED received
2. Distinguish between migration and referral
3. Kick state manager, sleep

State Manager thread

4. Post GETATTR(fs_locations)
5. Walk fs_locations array
6. DNS resolution up call
7. Create new transport
8. Recover state, if necessary
9. Wake NFS thread, sleep

10. Retry operation on new transport

Data Structure Organization

- Open and lock state moved from per-client into per-mount-point data structure
- More care taken with pointer references to RPC transport
- File handle of root dir saved at mount time
  - Needed to deal with NFS4ERR_LEASE_MOVED
  - Attempt to verify migration on server

What's Left To Implement

- Updating /proc/mounts
- LEASE_MOVED recovery
- SECINFO (Bryan S.)
- State recovery
  - Not needed with Solaris servers
- Volatile file handles
  - Not needed with Solaris servers
- Handling multiple concurrent migration events
- Managing client and transport state with multiple mount points on source and destination servers
- Testing, testing, testing

ORACLE IS THE INFORMATION COMPANY