Storage Systems

NPTEL Course

Jan 2012

(Lecture 10)

K. Gopinath

Indian Institute of Science

NFSv2 problems

- Maintaining UNIX semantics
  - Open file access permissions
    - Posix checks on 1st access; NFSv2 on every access
  - Atomic I/O operations
  - Deletion of open files: what if server deletes file?
- Cache consistency guarantees
  - NFS 2 checks if mod time of client cached data diff from server mod time. Works if server only making changes
- Security
  - Access control: User credentials
  - Securing data traffic
- Performance: UDP storms, Synch writes: ad-hoc opts.
- Needs Portmapper, mountd, lockd, statd
- Functionality: 4GB file size limit

NFSv3 enhancements

- Weak cache consistency
  - Whenever a request to NFS3 modifies a file, both pre-attr and post-attr of file sent back to client
  - If pre-attr matches that of client, valid data in client cache and attrs of cached data has to be updated with post-attr
  - Client verifies only on open or when cached attrs timeout.
- Asynchronous writes
  - COMMIT request
  - Blocks on close to sync data with server
- Crash recovery procedures
  - Server will not discard uncommitted data without changing write verifier (8 bytes), usually due to reboot
- Protocol extensions: eg. READDIRPLUS request
    - NFSv2 READDIR only lists files: iterate to get attrs of each
  - Exclusive mode CREATE request
- 64-bit operations: Mismatched clients & servers

NFSv4 overview

- Design goals
  - Better consistency mgmt
  - Performance improvements
  - Security improvements
  - Interoperability & extensibility
- Stateful protocol
  - OPEN: provides a single point where file lookup, creation, and share semantics combined
  - CLOSE: release of state accumulated by OPEN
  - Locking, shares and delegation
- Single-view of all exported FS
  - Pseudo FS
- Integration of mount & locking protocols
  - no interaction with portmapper, mountd, lockd, statd
  - fixes TCP port 2049 for NFSv4

NFSv4 changes

- COMPOUND procedure
  - Non-atomic operations
  - Error handling
- LOOKUP semantics
  - Multicomponent; traversal across multiple fs
- READDIR semantics
  - Can specify how much data can be sent per transaction
- Named attributes
  - OPENATTR procedure
- FS migration & replication
  - *fs_locations* file attribute
- Better cache consistency models but complex

NFSv4 data structures

- FSIDs & file handles
  - Volatile (vs. persistent) file handles
  - Root file handle, Current file handle
  - PUTROOTFH, PUTFH, GETFH, SAVEFH, RESTOREFH procedures
- Client IDs
  - Server & client crash recovery
- Lock state IDs
  - Locking state of a file

NFSv4 leases

- Lease semantics
  - Crash recovery operations
- Locking
  - CIFS-like share reservations
  - Lock request sequence IDs
- Delegation
  - Attribute & data caches
    - Open, close, data access, locking
  - Revocation operations via callbacks
  - Fallback to close-to-open consistency

NFSv4 security

- GSS-API-based RPCSEC_GSS
  - Kerberos
  - LIPKEY (Lower Infrastructure Public Key Mechanism)
    - Similar to SSL (with user name and passwd)
- SECINFO operation
  - Negotiate authentication, integrity & privacy parameters
- Access control
  - Windows NT ACL model
  - ALLOW, DENY, AUDIT, ALARM
- String UIDs/GIDs
  - user@domain

CIFS protocol

- Based on SMB protocol
  - 11+ dialects(!), most recent is NT LM 0.12
- Asynchronous operations
- ANDX command batching
- Services: File sharing, printing
  - Discovery, authentication, server management
- Operations: File access, locking, caching
  - Extended attributes, file change notification, protocol negotiation
- Transport independent
  - NetBIOS/TCP, TCP/IP (port 445)

CIFS features

- Performance
  - Read-ahead/write-behind
  - Caching
  - ANDX command batching
- Mounting/accessing FS
  - Tree connection
- Sharing files
  - Op-locks: Exclusive, batch, level-II
    - Requested/granted at file open
  - Op-lock breaks (revocation)
- Server referrals
  - FS replication & load-balancing

CIFS security

- Security model
  - NT/2k domains
  - Server-side session authentication
  - Share-level security
  - Client/user-level security
- Authentication schemes
  - Passwords: Plaintext/encrypted
  - Challenge/response: DES encryption
  - Pass-through authentication
- NT LM++
  - Mutual authentication
  - Message signing: MD5 hash

Unix-Windows integration

- Issues
  - Protocol mismatch
  - Access control mechanisms
  - Authentication mechanisms
  - File locking semantics
  - Management & administration
  - Infrastructure issues
- Approaches
  - Protocol emulation
  - Native implementations

Summary

- NFS a simpler model than a cluster filesystem
  - – NFSv4 most detailed wrt consistency mgmt
- Consistency issues critical