NFSv4

March 3, 2000

Spencer Shepler
shepler@eng.sun.com
http://www.nfsv4.org/

Sun microsystems

IETF NFSv4 WG
Where are we now?

- NFSv4 WG formed in Summer 1998
- RFC 2624 "NFS Version 4 Design Considerations" June 1999
- IETF Last call to complete on March 6th
- Draft under review is:
  - http://www.ietf.org/internet-drafts/draft-ietf-nfsv4-06.txt

Sun microsystems

NFSv4 WG Charter

- Improved access and good performance on the Internet
- Strong security with negotiation built into the protocol
- Better cross-platform interoperability
- Designed for protocol extensions

Sun microsystems

WG Goals and Milestones

- Oct 99 - Begin interop testing of prototypes
- Dec 99 - Submit NFSv4 to IESG for consideration as Proposed Standard
- Jun 00 - Conduct final interop testing
- Aug 00 - Submit NFSv4 to IESG for consideration as Draft Standard

Sun microsystems

Umich/CITI and Linux

- University of Michigan/CITI working on NFSv4 for Linux - under the Linux Scalability project
- CITI has received a coordinated research grant from Sun for the work
- CITI's work will be Open Source when complete
- www.citi.umich.edu

Sun microsystems

NFSv4 - Highlights

- Stateful (OPEN/CLOSE)
- Strong security required
  - Kerberos, Public Key (via LIPKEY)
- COMPOUNDed operations to allow flexibility
- Extensible file attribute model
- Integrated file locking
- Lease based recovery
- Delegation for aggressive caching
- UTF8 for names and protocol strings

Sun microsystems

NFSv4 - RPC/Security

- RPC, XDR, RPCSEC_GSS as foundation
  - RFC1831, RFC1832, RFC2203
- Security (IETF requirement)
  - Kerberos V5
  - Public key
    - LIPKEY I-D
    - www.ietf.org/internet-drafts/draft-ietf-cat-lipkey-03.txt
  - Negotiation will be handled by new OP - SECINFO

Sun microsystems

NFSv4 - Filehandles

- ROOTFH / PUBLICFH
  - no MOUNT protocol
- Persistent and Volatile
- Client must be able to recover from volatile filehandle expiration (FHEXPIRED)
- FHEXPIRED should be on server restart only

NFSv4 - Attributes

- Mandatory, Recommended, Named
- Mandatory examples
  - object_type
  - persistent_fh
  - change
  - object_size
  - fsid

Sun microsystems

NFSv4 - Attributes

- Recommended - examples
  - ACL, mode, mime_type, owner, version
- Named Attributes
  - sub-files that contain arbitrary application data

NFSv4 - Migration/Replication

- fs_locations attribute
- Server name and root path of other locations
- Client should obtain fs_locations for each fsid
- For migration, NFS4ERR_MOVED returned
- Server must accept GETATTR on fs_locations once NFS4ERR_MOVED is generated

Sun microsystems

NFSv4 - Server Namespace

- No mount protocol
- Root filehandle for the top of the file tree
- For all shared/exported filesystems, server constructs pseudo filesystems to span name space
  - i.e. Share / and /export/home (shares are spanned)
- Client uses regular LOOKUP, READDIR operations to inspect this name space

NFSv4 - File Locking

- File locking integrated into protocol
- Supports byte range, mandatory file locking
- OPEN provides atomicity for excl-create/share
- Windows/DOS SHAREs may deny access
  - OPEN for READ/WRITE - DENY WRITE
- Lease based recovery for locking/shares

NFSv4 - Client Caching

- Mechanism referred to as delegation
- At OPEN, client will for delegation
- Requires callback mechanism so server can recall the delegation given to a client
- Client is able to cache operations when it has the delegation
  - LOCK, READ, WRITE can stay at client

NFSv4 - Compound Ops

- String together a set of operations to accomplish client specific task
- Reduce number of round-trip requests
- One RPC read
  - PUTFH/LOOKUP/GETFH/GETATTR/ACCESS/READ
- NFSv3 - 2 RPCs if not more

Sun microsystems


Sun
microsystems