An Analysis of NFS Protocol Version 4

Tom Spuhler
Solutions Specialist
Hewlett Packard
26 Mall Road
Burlington, MA 01803
tom_spuhler@hp.com
781-505-7683   FAX: 781-270-2444

Interworks 2001 #095

NFS Protocol Version 4 - PV4

- 3rd revision of the well known NFS remote file access method originally from Sun
- PV4 (and future) now part of IETF standards process
- Addresses:
  - Internet Accessibility
  - WAN and Internet performance
  - Security
  - Locking
  - Cross-platform interoperability
  - Internationalization
  - Protocol extension
- “Encouraged” by popularity of other protocols (CIFS)

Quick View: How NFS works (or most RFA methods)

You got your various RPC calls: read, write, create, lookup, etc.

1) Figure out the mix of the above needed to perform the indicated operation.

example: write a block=>(mount, lookup, access, write, commit)

2) For each operation in #2, request that the server perform the operation and wait for completion. If success, go to next until done.

Quick View: What’s wrong with this picture?

- One can spend a LOT of time waiting on network transitions and server processing
  - – And how about those high-latency networks like the Internet?
- And what if I’ve previously accessed the information in question?
  - – And how do I know it's still good?
- And do I really need to tell the server about every piddly thing I do on the file?
  - – Locks and temporary files, especially locks on temporary files

Quick View: PV4 to the Rescue!

- High latency network (anywhere latency > 0) ?
  - COMPOUND RPC and more complex procedures such as LOOKUP, Open Delegations
- Hey! I’ve used this data before!
  - Open delegations assist Client Caching
- Look, I’m just gonna use this file all by myself, why do I gotta tell you everything?
  - Open Delegations allow clients to “tell” servers to mind their own business.

What do we look for in a well performing protocol?

- Avoids re-transmission of information using local caching
  - Local caching can save 75% network throughput
- Minimum dependency on previous transaction completion before the next one can begin
- Minimum of required transactions to perform common operations (includes large data size)
- Maximum parallelism (Files, sessions, threads)
- Minimum protocol overhead
- Minimum negative impact on lower layers
- Simple, efficient implementation

NFS V4 - General

- Not dependent upon previous versions of NFS
- No longer stateless
- A congestion-management transport is required
  - TCP/IP required if available
  - UDP has been faster with PV3
- A well-known port number (2049) is used
- No mount protocol required
- Only two RPC calls
  - Null and COMPOUND
- Locking is part of the protocol and can be mandatory

NFS PV4 - General (cont)

- Leases are used to avoid “abandonment” problems The server MAY “delegate” control of a file to clients(s)
- Client Callback, if available, for best performance
- All names are encoded using UTS-8
- New security Protocol RPCSEC_GSS (RFC2203)
- New OPEN and CLOSE calls
- READDIRPLUS subsumed into READDIR
- New attributes to support FS migration and redundancy
- Protocol is extensible
- Various other goodies

NFS PV4 - Targeted Areas for Improvement

- Internet Accessibility
- WAN and Internet performance
- Security
- Locking
- Cross-platform interoperability
- Internationalization
- Protocol extension

Internet Accessibility

- Strongly encourages TCP/IP
  - requires a flow control protocol
  - Requires ability to use TCP/IP, if available
- Access through firewalls
  - Eliminated mount protocol
    - No longer uses Portmapper
  - Uses well-known port 2049
  - Public filehandle
- See Also
  - Wan and Internet Performance
  - Security

WAN and Internet Performance

- Avoid the penalty of latency
  - generally by reducing number of required commands
- Avoid re-requesting information
  - client caching
- Avoid bothering the server with things it doesn’t need to know about
- *Many of these also reduce server loading*

WAN and Internet Performance cont - LOOKUP, OPEN, CLOSE

- Avoids the penalty of latency by
  - More powerful commands
    - LOOKUP processes a path, not just a single filename
    - READDIR subsumes READDIRPLUS from PV3
    - OPEN, CLOSE

WAN and Internet Performance continued - COMPOUND

- Avoids the penalty of latency by
  - Promote execution of multiple commands in a single network transaction
    - COMPOUND RPC
      - multiple procedures serially executed until failure/ completion
      - CURRENTFILEHANDLE, SAVEDFILHANDLE, ROOTFILEHANDLE
      - VERIFY
      - GETPH, SAVEPH, PUTROOTPH

WAN and Internet Performance continued - Caching

- Avoid re-requesting information by promoting client caching
  - PV4 still NOT a strong caching protocol
  - PV3 Weak Cache Consistency information (pre and post operation attributes) has been removed
  - Change_INFO data structure returned by CREATE, LINK, OPEN, REMOVE and RENAME
  - See Open Delegation (next)

WAN and Internet Performance continued - Delegation

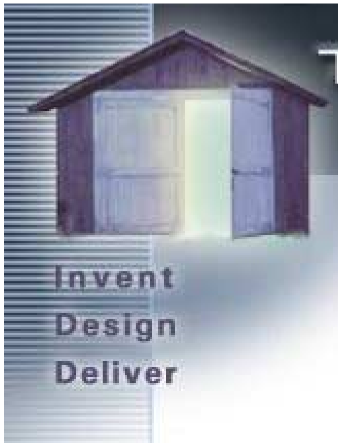
- OPEN Delegation
  - Issued and controlled by server
  - Permits client to control file
  - includes opens and closes
  - Read delegation
  - Write delegation
    - may also lock
  - if you check access time at open, then get a read delegation, the file won’t change without the delegation being revoked

WAN and Internet Performance continued - Delegation P2

- Delegations
  - delegations may be revoked
    - Callback protocol.
    - CB_NULL, CB_COMPOUND, CB_GETATTR, CB_RECALL
  - Delegations are Leased
    - A broken lease is a failure!
  - Delegation recovery is possible after server failure
    - Delegation need to be in stable store on the server

WAN and Internet Performance continued - Delegation P3

- Avoid bothering the server by
  - Write delegations
    - if a client has a write delegation for a file, it may perform most operations on that file without contacting the sever
    - Includes locking
    - Usual revocation and leases apply



Security

- Mandated strong RPC security flavors that depend on cryptography
- Negotiate security secure and in-band
- Character strings used for user and group Ids
- Window and UNIX compatible access control
  - ACLs
- Removed MOUNT protocol

- RPCSEC_GSS mandated

Locking

- Controlled by leases that need to be RENEWED
- lease recovery after server failure
- Mandatory locking available
- Share Reservations
  - full file lock between OPEN and CLOSE
- Sequence ID’s avoid duplicate request problems

- if the client has a write Delegation for a file, the client may lock it at will without contacting the server.

Cross Platform Interoperability

- Common set of features that do not favor any operating system
- Broader attribute types
- Persistent and volatile file handles
- Uniform name space with pseudo-paths and pseudo root (if necessary)

Internationalization

- All strings used for file, directory and symbolic link contents are encoded using UTS-8
  - UTS-8 is a Universal Character SET (UCS)
    - Supports mapping of 8 and 16 bit characters.
    - 8 bit encoding: 11000xx 10xxxxxx
    - Supports direct mapping of previously stored objects - US ASCII

Protocol Extension

- PV4 has provisions to support minor versioning, which should allow orderly and more regular extensions of the protocol

Other Goodies

- Protocol support for file system migration and replication

Protocol Comparison Example

- Illustrates the use of the COMPOUND procedure, elimination of the Mount protocol and *portmapper*
  - from *The NFS Version 4 Protocol* by Pawlowski, et al. www.nfsv4.org

```
mount bayonne:/export/vol0 /mnt
dd if=/mnt/home/data bs=32k count=1 of=/dev/null
```

e.g. mount remote file system. Read the first 32KB of the file.

Example from Solaris. Simplified output of network trace.

Example continued - PV3

- NFS Version 3 Network traffic

® PORTMAP C GETPORT (MOUNT)

¬ PORTMAP R GETPORT

® MOUNT C Null

¬ MOUNT R Null

® MOUNT C Mount /export/vol0

¬ MOUNT R Mount OK

® PORTMAP C GETPORT (NFS)

¬ PORTMAP R GETPORT port=2049

® NULL

¬ NULL

Example continued - PV3, p2

® FSINFO FH=0222

¬ FSINFO OK

® GETATTR FH=0222

¬ GETATTR OK

® LOOKUP FH=0222 home

¬ LOOKUP OK FH=ED4B

® LOOKUP FH=ED4B data

¬ LOOKUP OK FH=0223

® ACCESS FH=0223(read)

¬ ACCESS OK (read)

® READ FH=0223 at 0 for 32768

¬ READ OK (32768 bytes) ;

DONE!!!!

Example Continued - PV4

- **NFS Version 4 Traffic**

@ PUTROOTFH; LOOKUP “export/vol0”; GETFH; GETATTR

-- PUTROOTFH OK ¯CURFH;LOOKUP OK ¯CURFH; GETFH OK; GETATTR OK

@ PUTFH; OPEN “home/data”; READ at 0 for 32768

-- PUTFH OK ¯CURFH; OPEN OK ¯CURFH; READ OK (32768 bytes)

Done!!

- **11 round trips reduced to 2 round trips**

Implementation

- The actual implementation will have a significant impact on how a protocol performs
  - Especially true on the client!
    - Example: CIFS server implementation can have dramatic impact (eg refuse oblocks)

- Still very early in the PV4 implementation life

Implementation

- No feature, no matter how powerful, is of any use if not implemented!
- Completeness of implementation is often a reflection of the implementers resources
- Protocol complexity can drive up the cost of implementation

Learning about Remote File Access Protocols

- CIFS
  - Where’s the protocol?
  - Variety of 3rd party discussions = mostly in agreement
  - Consortium protocol definition underway
  - www.samba.org
- NFS (earlier versions)
  - Protocol available from Sun and as RFCs
  - Several very good books
- NFS V4
  - IETF now owns NFS protocol - Many RFS (RFC3010)
  - You too can implement it
  - www.nfsv4.org